Bioprocess-Engineering Education with Web Technology

Olivier D.T. Sessink

Promotor
Prof. dr. ir. J. Tramper
Hooleraar in de Bioprocestechnologie, Wageningen Universiteit

Co-promotoren
Dr. H. H. Beeftink
Universitair docent bij de sectie Proceskunde, Wageningen Universiteit

Drs. R. J. M. Hartog
Programmaleider FBT bij het Wageningen Multimedia Research Centre, Wageningen Universiteit

Promotiecomissie
Prof. dr. U. von Stockar
Ecole Polytechnique Fédérale, Lausanne

Prof. dr. P. M. E. de Bra
Technische Universiteit Eindhoven

Prof. dr. ir. L. A. M. van der Wielen
Technische Universiteit Delft

Prof. dr. ir. H. Gruppen
Wageningen Universiteit

Bioprocess-Engineering Education with Web Technology

Olivier D.T. Sessink

Proefschrift

ter verkrijging van de graad van doctor
op gezag van de rector magnificus
van Wageningen Universiteit,
Prof. dr. M. J. Kropff,
in het openbaar te verdedigen
op maandag 19 juni 2006
des namiddags te vier uur in de Aula.

Olivier D.T. Sessink (2006)
Bioprocess Engineering Education with Web Technology
PhD. Thesis, Wageningen University, The Netherlands - with summary in Dutch and English
ISBN: 90-8504-430-8

Table of Contents

Chapter 1

Introduction

Introduction

This thesis covers choices that have been made, and material that has been designed during the application and evaluation of web technology in education in food and biotechnology in general, and in bioprocess engineering in particular. Special attention has been paid to the design of learning material, and the designed material itself. Design of learning material in other food and biotechnology domains of Wageningen University is described by Wilmsen for molecular biology [1] and by Diederen for food chemistry [2].

Because both bioprocess engineering education and web technology play a prominent role in this thesis, this chapter will shortly introduce some essentials in bioprocess engineering education, and in addition give a short introduction to web technology. Finally the aim of this research and the outline of the chapters of this thesis is given.

Bioprocess engineering education

Like for instance chemical engineering, bioprocess engineering is a specialized form of engineering in general. Bioprocess engineering puts a biotechnological process to work [3]. Biology, chemistry, physics and mathematics are applied to change the composition of a process stream or to convert a raw material into a product. In bioprocess engineering, the process and/or the product have biological origins. The product, for example, may be a human hormone, ethanol, or fructose, and the process may use enzymes, antibodies, or cells. Biological components are relatively unstable. Bioprocess engineering processes therefore require mild conditions, such as a pH close to neutral, temperature close to room temperature, little shear stress, and low osmotic values (and thus very diluted mixtures). Bioprocess engineering mainly deals with the design, optimization, and maintenance of processes for large-scale production of bioproducts.

Similar to chemical engineering, bioprocess engineering production processes use a chain built from individual unit operations, which can be synthesis or separation processes. Bioprocess engineering specializes in those unit operations that have specific requirements because of the instable biological components. This unit operation may be a conversion process in which a biocatalyst is present, but may also be a separation process in which the product itself requires mild conditions.

Bioprocess engineering employs a similar systematic approach to modeling as chemical engineering, based on the conservation of mass, energy and momentum. Balance equations, containing transfer and reaction terms, are used to describe some overall change in a process. The transfer and reaction terms can be described based on principles of transport phenomena, reaction kinetics and thermodynamics. The balance equations form a set of differential equations that can be solved analytically or numerically.

Essential bioprocess engineering learning objectives

Besides learning objectives that are not specific to bioprocess engineering, such as a basic knowledge of mathematics, biology, chemistry, and physics, we have identified nine learning objectives that are most essential for bioprocess engineering.

Know transport and reaction phenomena that occur frequently in bioprocess engineering operations and their mathematical representations. Transport and reaction phenomena are the foundations of bioprocess engineering. Students should, therefore, have a sound knowledge of phenomena such as convection, adsorption, diffusion, reaction kinetics and residence time distribution.

Know how common bioprocess engineering operations work, i.e. which phenomena are involved, how devices are structured. Students should know, for example, how an adsorption column is schematically constructed, what its dimensions can be, and which phenomenon its working is based on.

Know orders of magnitude of parameters and variables for common bioprocess engineering processes. Engineers often have to work with many unknowns. Assumptions then have to be made. Knowledge of orders of magnitude may help to choose assumptions as well as to validate the results. Students should, for example, know realistic cell growth rates, realistic temperatures for enzyme activity, realistic membrane surface areas, and realistic reactor volumes.

Know which phenomena and quantities have significant impact in bioprocess engineering processes. Students should, for example, know that they should take care of pH and temperature during cell cultivation, and consider if substrate inhibition or limitation will occur. Students implicitly learn that only a small number of quantities and phenomena are usually relevant for bioprocess engineering.

Know common assumptions and common knowledge used in bioprocess engineering models. Students should for example know that in batch and fed batch reactors specific rates are usually high, that CSTR reactors are assumed to be ideally mixed, and that counter-current flow enables higher transfer per surface area than co-current flow.

Reason qualitative system behavior in a few common bioprocess engineering systems. Students should for example be able to deduce if a reaction rate will decrease or increase on change of substrate concentration, or if transport rates will decrease or increase on pressure change, based on qualitative relations.

Use existing models to solve quantitative problems. Students should be able to select a model for a specific assignment, translate the given information to parameter values, and use the values and the model to complete the assignment.

Design new models in terms of balance equations. If the model is not yet known, students should be able to design a model in terms of a set of differential equations using common assumptions and phenomena.

Design bioprocess engineering unit operations, or chains of unit operations for bioprocess engineering production. Ultimately students should be able to use all of the above described knowledge and skills to design a unit operation or a chain of unit operations for a specific problem.

Summarizing, bioprocess engineering education aims to teach students knowledge about bioprocess engineering operations, and skills to design and use models of bioprocess engineering operations, to ultimately design new operations.

Web technology

Web technology is about the World Wide Web (WWW or “the Web”). The WWW, often referred to as “the Internet” (see Terminology), is a worldwide collection of interconnected documents. These documents can be viewed in a web-browser application. The web browser may request a document from the WWW. This document usually resides on a different computer, called the server. The server sends the document to the computer with the web-browser, called the client (Figure 1). The WWW was developed at CERN, Geneva, in 1991. The WWW really took of with the release of the Mosaic web browser application in 1993.

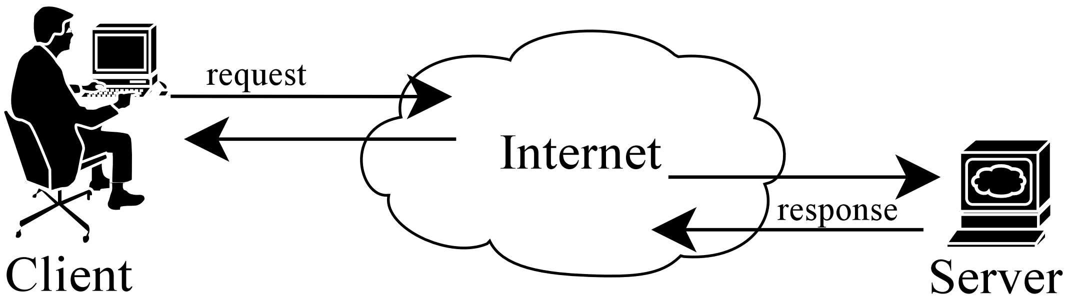

Figure 1: *The client computer with the web browser software, and the server computer with the web server software*

Web-based material

Only material that can be displayed by the web browser is considered web-based material. A web browser can download any digital material, but can not display all digital material. Files in, for example, OpenOffice.org, MS Word, or OpenDocument format, MS Windows exe or Unix elf format, or files from Mathcad, Matlab, Octave or Aspen Plus, are all digital material, but not web-based material. Material available from a web-based learning management system, such as Moodle, Blackboard, or WebCT, is thus not necessarily web-based material.

Originally, only formatted text and two image formats could be displayed by web browsers; nowadays, many extensions allow web browsers to display various other types of content. A short overview of web-browser extensions is given in Table 1. As can be seen in the table, there is a group of core extensions that are specifically designed for the web, and widely available on many platforms. Next to these core extensions, a number of other extensions are listed as example of extensions that are not web specific, or have limited availability. Whether or not these extensions are considered to be web technology is, therefore, a matter of debate. It is important, furthermore, to realize that web browsers for the disabled have limited support for most extensions. The web-based material described in this thesis has been designed to use only the core extensions, and can thus be viewed on almost all computers with a web browser and Internet connection.

Terminology

Technically speaking, the words Internet and World Wide Web are not synonymous. The Internet is a worldwide network of computers that all use the Internet Protocol (IP) to communicate. The WWW uses this network to deliver information. The information is presented to the user by the web-browser application. This web-browser application on the client computer may request content from a web-server application on a server computer using the HyperText Transfer Protocol (http). Which content exactly is requested, is specified using a Uniform Resource Identifier (URI, formerly called “Uniform Resource Locator” or URL). Each URI points to a specific piece of web content anywhere on the WWW. Pieces of web content (i.e. a page of hypertext) may, in their turn, refer to other content (i.e. an image), using an URI. The other content may be embedded, or a hyperlink, a ‘clickable’ reference to the other content, can be made. Hypertext content (i.e. content that contains hyperlinks) is formatted using the Hypertext Markup Language (HTML).

Table 1: *A list of standards and formats supported by common web-browser extensions*

Technology	***Availability***	***Web specific***	***Remarks***
Javascript / ECMAScript	all	yes	Offered internally by most browsers since 1995. Used in this thesis.
Java applet	Windows Linux Mac	yes	Since 1995 distributed by default by most browsers. Internet Explorer, however, distributes only an old version by default. Used in this thesis.
Macromedia Flash	Windows Linux Mac	mostly	Since 2000 distributed with some browsers on one platform. Nowadays available for most browsers and platforms. Used in this thesis.
Scalable Vector Graphics (SVG)	Windows Linux Mac	initially	Initially developed for the web by W3C. However, still not available by default in Internet Explorer.
Portable Document Format (PDF)	Variable	no	Several extensions exists, but browser integration and hyperlink support are not always available.
Macromedia Shockwave	Windows	yes	Supported by 54% of the browsers according to Macromedia.
Real media format	Windows Linux Mac	not Web specific, but Internet specific	Browser integration is not available with all browsers.
MPEG movie format	Variable	no	Several extensions exists, but browser integration is not always available.
Virtual Reality Markup Language (VRML)	Variable	initially	Several extensions exists, their quality differs.

Together, the web browser and its extensions make up what is called the “client-side” web technology. Common characteristics of client-side technology are the low interaction-latency (the time between a user action and a reaction), availability of dedicated computational resources, and the absence of data storage resources.

The opposite of client-side web technology is server-side web technology, built around the web-server application, often called the web server. Extensions of the web-server mostly enable the server-side generation of data that are sent to the web browser. The extensions can execute a script or program that may use user input or use data stored on the server computer to generate the data. Several well known extensions are the PHP framework, well known for its integration in the Apache web server, but also available for most other web servers on all platforms, Java-based technologies that can be integrated in various web servers on various platforms, and the Dotnet framework, integrated in the Internet Information Server web server which is only available on the Windows platform. In contrast to the client computers, an organization usually has full control over the server computer. Common availability of an extension is thus not critical, as long as the extension is available on the server computer. The choice of extension may, however, limit the choice of web server and platform. The learning material described in this thesis is developed using the PHP framework, because of its availability for most web servers and platforms.

The scripts and programs that generate the data often rely on other services for functionality. Common services are an authentication service, such as an LDAP directory, and data storage, such as a file system or a relational database. Because access to these services is commonly available and required, we consider this part of what server-side web technology may offer. Common characteristics of server-side technology are the high interaction-latency, the computational resources that are shared among all users, and the abundant availability of data storage resources.

Summarizing, web technology is the combination of client-side and server-side technologies that aim to exchange information between a client and a server via a web browser. As such, the WWW is a widely available infrastructure for information delivery.

Aim and outline of this thesis

The possibilities of ICT in general, and web technology in particular, raise high expectations for use in education. The WWW forms a ready-made, widely available infrastructure for information delivery. Web-based learning material, furthermore, is expected to enable more motivating, more activating, and more efficient learning activities. At the same time there is much confusion and uncertainty how to exactly fulfill these expectations. Especially more advanced applications of web technology, such as adaptive learning material, have not yet appeared in mainstream education. The aim of the research described in chapters 2 through 6 is, therefore, to use and evaluate the possibilities of web technology for academic education in food and biotechnology in general, and for the design of learning material for bioprocess engineering in particular. Each of these chapters is published or submitted for publication, and therefore has an independent structure.

Chapter 2 addresses security issues in the use of web technology for examination. Examination is an essential part of education, and security aspects are important when facilitating exams with ICT.

Chapter 3 covers issues in the functionality of learning management systems that have been identified during the development of state of the art learning material for food- and biotechnology. It covers both the functionality that is offered by current generation learning management systems, as well as functionality described by upcoming standards for digital learning material.

Chapter 4 describes the development of a lecturer-friendly adaptive system called Proteus, and its use and evaluation in adaptive learning material for cell growth kinetics and reactor concepts. Adaptive systems require meta data from the author. Proteus requires few meta data, and the meta data are defined using concepts that are intuitive for lecturers.

Chapter 5 describes the design, use and evaluation of a virtual experiment environment for experimenting and model related learning objectives. In this experiment environment students have to design and execute experiments, and interpret experimental results, in order to estimate model parameters.

Chapter 6, finally, is a general discussion in which all designed and developed learning material is discussed in terms of characteristics of web based learning material, guidelines from theories of learning and instruction, and guidelines from the bioprocess engineering domain.

References

1. Aegerten-Wilmsen, T. (2005). *Digital Learning Material for Experimental Design and Model Building in Molecular Biology.* ISBN 90-8504-315-8
2. Diederen, J. (2005). *Design and Evaluation of Digital Activating Learning Materials for Food Chemistry Education.* ISBN 90-8504-271-2
3. NRC (National Research Council) 1992. *Putting Biotechnology to Work: Bioprocess Engineering.* Washington, D.C.: National Academy Press.

Chapter 2

Securing web-based exams

Abstract

A number of learning management systems offer web-based exam facilities. Using these facilities increases the risks for fraud during exams in comparison to traditional paper-based exams. This article discusses a list of weaknesses in web-based exams, and proposes precautionary measures to reduce the risks. A security model that distinguishes supervision support, software restrictions and network restrictions is presented. The solutions for the security problems describe measures for exam computers with Windows and Linux respectively, secure network setups in common network architectures and tools to help supervising and monitoring of web-based exams. The article intends to raise the awareness among faculty in higher education of the risks of web-based exams, and to help the technical staff to implement precautionary measures.

Published as Sessink, O.D.T., Beeftink, H.H., Tramper, J., Hartog, R.J.M.. (2004) *Securing Web-Based Exams.* Journal of Universal Computer Science, 10(2), 145-157

Introduction

Since a few years almost every institute in higher education deploys one or more Learning Management Systems (LMSs) to facilitate students and the staff. Many of these systems use the Internet for communication and often they have a web-interface. This means essentially that the system can be accessed using a web-browser.

A number of these systems feature a test and exam facility; the Blackboard learning management system for example has its *assessment* facility [1]. Also several stand-alone test and exam functionality modules exist, for example Questionmark Perception [2] and Securexam Browser [3]. The test and exam facility is useful for self-testing, training, formative testing and final examining.

In this article the term web-based exam refers to a situation where the student uses a web-browser to access questions and submit answers and where the results of this exam (partially or completely) determine the final grade for the subject. When the test is used for the final grade it is important to assure that the student did the test in a satisfactory setting. Exam settings may have different requirements. The requirement that the student doesn't have any help from other people is very common. Also access to answers from other students is not allowed. For many settings it is required that the student has no access to the Internet (apart from access to the LMS itself), a book or personal notes. But sometimes an exam is open-book, or even open-web, where students are allowed access to a book or even the web. With the shift in learning goals towards comprehension, application, analysis, synthesis and evaluation more and more exams become open book exams.

Of course most of these requirements can, to some extent, be met by supervision, but faculty members should be aware of the fact that students have many more possibilities for fraud in a computer-room than in the traditional classroom. The aim of this article therefore is to raise awareness of faculty in higher education of the possibilities that students have with web-based exams (sections 2, 3 and 8) and to help technical staff with several solutions to support digital supervision (section 4) and to secure the computer facilities (sections 5 and 6). The solutions to secure the computer facilities are presented as a four level security model. The solutions involve the setup of the exam computer, both for Windows and Unix based computers, and the setup of the network.

Table 1: *Security model levels*

Levels	*Description*
I software accessibility	Configuration that limits the access to available software for the user
II software availability	Configuration that makes software unavailable for the user
III network accessibility	Configuration that limits the access to the available network resources for the user
IV network availability	Configuration that makes network resources unavailable to the client

Common weaknesses

For faculty members it is first of all important to realize that the new generations of students have a very high level of computer skills. Students have ways of exchanging or acquiring answers to exam questions most faculty members aren't even aware of. For example the usage of instant messenger applications such as ICQ, Trillian, AOL, MSN etc. by students at Wageningen University is estimated around 80% while the usage by faculty members is negligible.

By nature a web-based exam makes it hard to stop illegal communication. Since it is web-based, the exam computer needs to have a network connection. A student might make illegal use of this connection for communication with other students.

Figure 1: *Screenshot of a skinned chat program, very hard to detect*

Many chat programs and instant messaging programs are available for network communication. Users can customize the look of some of these programs, which is called *skinning*. It can be used to conceal the program on the computer desktop or to disguise the program. An example screen-shot is shown in Figure 1 where a chat program is hidden in the windows task-bar. This makes it very difficult for a supervisor in the computer room where the exam is being taken to detect a student who is using such a program. For example the applications ICQ, Trillian, MSN and mIRC are available for free on the Internet.

It is not even necessary to install software on the exam computer; many web sites also offer chat facilities. Also, students can install chat facilities on their personal home pages. Even the chat facility from the LMS itself might be used to communicate with other students during an exam.

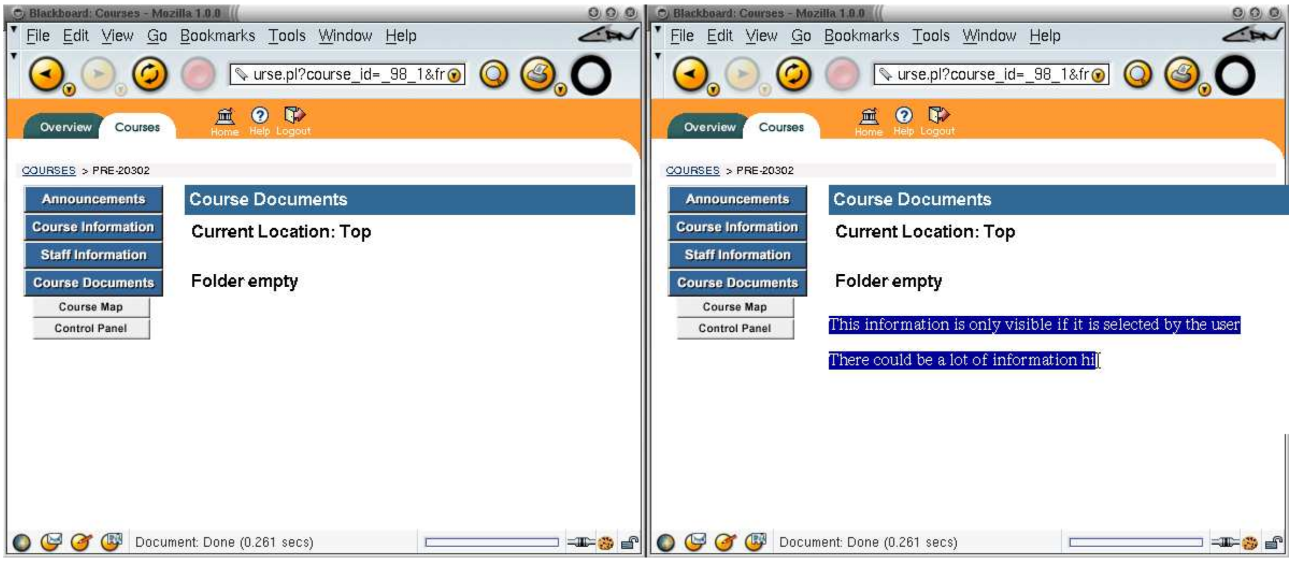

Figure 2: *Two screenshots of the same page in Blackboard. On the left the normal view. On the right the apparently white area is selected, showing the hidden information.*

While being the most well-known option, a chat facility is not necessarily required to exchange answers. Almost any communication program can be used to exchange answers. Students can upload the answers for an exam to their homepage using an ftp program, making the answers available to their fellow students. The telnet program might be used by a number of students to login to a common account on a Unix server. The email program might be used to send answers to another student. Some LMSs even offer the possibility for users to setup their own page. This page can be used to add a chat facility to the LMS or to publish answers to other students. A skilled student could even make a web-site that resembles the look of the exam facility but in reality contains the answers to the exam. An example is shown in Figure 2 where on the left a page looks like an empty web-page, but when the text is selected on the right the empty page shows answers to an exam. This is just a short list of possibilities, many more can be found.

Electronic communication is not the only security risk. The setup in computer rooms is often not very well suited for exams. Many computer rooms are setup for tutorials, where all displays face the same direction. The rows of screens are usually in short distance from each other so it is quite easy for a student to see the screen from another student a row ahead.

Many examiners use solely closed questions in web-based exams. The fundamental reason why an exam should consist of closed questions is that closed questions can be marked objectively. Another reason is efficiency. It turns out that for most web-based exams, the cost benefit ratio is such that a multiple-choice exam is preferred. But for a student, peeking during multiple-choice exams is relatively easy.

Another important issue is the identity of the student submitting the exam. Students can for example easily exchange passwords and submit each other’s exams. A student could also leave the exam after an hour, and log on to the LMS from a different location and continue the exam with help from others. A student could also pretend to do the exam, while in reality another student from a different location is submitting the exam.

On the Internet, a computer's Internet Protocol (IP) address is often used to restrict access to a service (for example access to a fileserver), or to register from which computer a service has been accessed. However in many networks it is easy to change the IP address of a computer to the address of another computer (known as *IP spoofing*). A user could change the IP address from a computer outside of the exam room to the IP address from a computer inside the exam room, and then submit the exam from the computer outside the exam room. Especially with wireless networks it is easy to do this because no physical access to the network (e.g. a network cable) is required.

A number of features are available in the different LMSs to guarantee the identity of the student. Most LMSs offer the possibility to allow access to the exam only during a certain time period. Another common feature is to stop the students from logging into the exam multiple times. Some LMSs offer the possibility to restrict the IP-address of the client computer to a specific IP address range. As said before this measure is vulnerable to IP spoofing. Some systems offer password protection of the exam. The password can be made available to the students a moment before the actual start of the exam. This can stop students at different locations to log in into the exam.

Finally a very important security issue is the security of the server. The server stores all questions, all answers, and after the exam all grades. Because the server security is not specifically related to web based exams this topic is beyond the scope of this article.

Workarounds

Next to the technical solutions, described in section 4, 5 and 6, there are of course several workarounds. Although beyond the topic of this article, this section gives a brief description of some of the most obvious workarounds.

Randomizing questions and answers

If the questions and answers are presented to the students in random order, there is less chance that they can see the answer on a nearby computer screen. The amount of communication needed to pass the right answer will also increase notably. Instead of passing "question 1 has answer b" it is needed to pass the actual text of the question and the answer, like "the question about the reaction-kinetics in a two step reaction has answer Michaelis-Menten kinetics".

Large question database

If the number of possible exam questions is much larger than the number of students, and all students get different questions, there is not much chance for any fraud. The larger the number of possible questions, the smaller the chance that two students have to answer the same question. It is however a real challenge to assure that every student will get an equivalent exam if they get different questions.

In subjects where exams involve mostly calculations, it is relatively easy to generate identical questions with numerical variations. These exams are quite common in fields like

Economy, Chemistry and Engineering.

Dedicated test application

If the questions can be answered and submitted in a dedicated application there is in most settings no need for a web-browser anymore. Many security risks can be avoided this way. Questionmark has such a dedicated application available [4].

Digital supervision

As explained above, most weaknesses involve electronic communication. In this section a set of options to support supervisors in detecting illegal electronic communication will be discussed. Faculty members should, however, be prepared that some students while accepting the traditional need for supervision during exams will claim that being "electronically watched" influences their performance during the exam.

Software supervision support

There are several possibilities to detect which software is being used by a student. Firstly, the supervisor can request a process listing in order to check all programs a user is running. Secondly, there are several software suites available to view the contents of a user's display without disturbing the user. For example Visual Network Computing (VNC) [5] and PCAnywhere [6] can be used to remotely monitor any users screen. If the students are aware that the supervisor monitors their displays, this awareness will probably prevent them from using the computer for illegal communication.

Network supervision support

As with supervising which software is being used by the student on the exam computer, the network also allows measures for supervision. A packet analyzer can be used for network monitoring. For open-web exams in particular, network analysis will make sense. Students are allowed to use the web to find information, but users should not use the web to communicate with each other. The packet analyzer can be used to analyze and log all the network traffic on a certain network. This can be used for prevention and to identify students who have communicated with each other. If students know all network traffic is logged they will not be very tempted to use any chat facility or any other way of communication that uses the network. And in case a student is suspected of cheating, the network log can be used to ascertain that the student really did.

Most networks in PC-rooms nowadays are Ethernet (IEEE 802.3). An Ethernet segment might be shared or switched. In a shared network, all network traffic is visible to all hosts in the segment. In the shared situation, the network analyzer should be connected to the same segment as the client computer. If the segment is switched, it is often not feasible to apply a packet analyzer; at least some more work needs to be done. In a switched network, the network traffic is only visible to the source and destination host. Either the switch should be configured to send all traffic to the analyzer, or the analyzer should be connected upstream from the switch. Unfortunately most switches do not support these required configuration options. Furthermore many upstream networks cannot be connected to (e.g. fiber backbones). Also many situations exist where the network is neither switched nor shared, for example a bridged network where some hosts might see each others traffic but others won't. A closer look at the network architecture is then required to find whether using a packet analyzer is feasible or not.

A powerful solution for packet analyzing is the open-source application Ethereal [7]. This application can log the network traffic itself, or use the logs from the TCPdump utility. Any computer running Linux and Ethereal can be used to log the network traffic. If the intermediate TCPdump is used, any computer can be turned into a network-logging device using just a bootable diskette or CDROM; a previous Linux installation is not required for this situation. The log can then later be analyzed by Ethereal. This setup can easily be expanded to a professional setup where dedicated remote controlled network loggers send their logs to a central logging facility for later analysis.

If the web browsers use a proxy server, the proxy log might be used as well for analysis. Especially in the case of an open-web exam, where students are allowed to browse the web, the proxy logs can be used to analyze the web sites visited by the students. A student using a chat facility from some web site during the exam can easily be detected.

Software restrictions

To address the issues related to electronic communication a four level security model structures the measures for securing the exam computers. Unix systems are quite secure by default, and offer a wide range of tools to restrict the user. Since Linux is the most popular Unix like operating system it will be used as an example. Not everybody will require the extensive security features of a Unix system, therefore measures for both Windows and Linux will be described.

For many exam settings it is desirable that the user can only use the web browser and any software required during the examination. Other software could potentially be used to communicate with other students, or to help the student answering the exam questions. We will call this software the *superfluous software*. Solutions presented for levels I and II will help to prevent students from using superfluous software.

A first thing to do when securing the accessibility of an exam computer is to make the computer only bootable from its hard disk drive and to set the BIOS password. If this measure is skipped any user can start the exam-computer from a bootable diskette or CDROM. In that case the contents of the hard disk drive would be completely accessible without any limitations. The user could for example install a chat program or remove network restrictions.

Level I - Software accessibility

Windows has a security architecture built around so-called policies. The policy settings are stored on a server, and are retrieved after a successful domain logon. The policies allow settings like <disable registry edit tools>, <run only allowed applications> and <disable control panel>. Unfortunately most of these policies are easy to circumvent. For the <run only allowed applications> option, the system administrator needs to specify strings with names of executables. If a user copies the executable to another location and renames it, the executable can be started again. The option <disable registry edit tools> only disables the Microsoft registry edit tools. Many third-party registry tools are still functional.

Windows2000 together with a Windows2000 domain features so-called group policies.

Group policies feature more policies for control over the desktop environment. An interesting feature is to install the “local security policy”. With a template, a machine can be secured against installing software by users, or against a change of configuration. But again a skilled user can circumvent these security policies. So these measures will give some barrier for a user, but are not even close to securing software accessibility.
To overcome the lack of security options, the Department of Animal Sciences at Wageningen University developed an application to run a browser inside a screensaver. This makes the underlying system inaccessible to the users from the moment the screensaver starts. The browser does not have any menu or toolbar, so the user cannot change any browser setting. This screensaver application is developed for Windows 98; some minor changes might be required to run this on top of NT or 2000.

```
menu,menubar,menuitem,menupopup,
popup,toolbar,toolbox {
   display: none;
}
```

Figure 3: *Access to the menu's and toolbars of the Mozilla web browser can easily be restricted by adding the following lines to userChrome.css*

Linux allows several different measures. If only one browser window is needed during the examination, a very powerful measure is to disable the window manager. The window manager is the application that manages the placement, resizing and starting of all application windows. Without a window manager, a user will only see the web-browser, without title bar, close button or minimize button. Therefore a user cannot hide a chat window under the web-browser or iconify it, nor can a user start any new application; it is not even possible to run multiple web-browser windows simultaneously. Only if the web-browser itself starts another program, the students have access to that program. If, for example, the browser configuration is not restricted and no measures against superfluous software are taken, a student could change the configuration from the web-browser in such a

way that an xterminal is spawned for a certain mime type. The xterminal can be used to start any other available software. It is therefore important to secure the browser configuration and to take measures against the available superfluous software.
If multiple windows are desired, there are also several very minimal window managers. Also for the current Mozilla browser a window manager is desired because it has some problems running without a window manager. Using for example the Lightweight Window Manager (LWM), the user can have multiple windows, but there is no menu and there are no icons. There are several more window managers that can be used for this type of restricted setups, like sawfish, scwm and wm2.
The shortcut keys to browser functionality (e.g. “File/Open”) can be removed by blocking modifier (control and alt) and function keys. This also blocks access to other functionality triggered by shortcut key combinations like e.g. change virtual console (control-alt-F1) or terminate the X server (control-alt-backspace).

```
xmodmap -e 'clear Control' -e 'clear Mod1'-e 'clear Mod2' \
        -e 'clear Mod3' -e 'clear Mod4' -e 'clear Mod5'
```

Figure 4: *This example shows how to disable all control and alt keys on a Xfree86 Xserver*

Also the Xservers from the Xfree86 project, the default Xserver on most Linux distributions, can easily be configured to disable all but the left mouse button, thereby disabling for example any right-click pop-up menu, see Figure 4.

Level II - Software availability level

To make superfluous software completely unavailable to the user the system needs to be stripped from all that software. A search on a typical windows 98 system resulted in more then 200 executables in the c:\windows\ directory and a typical windows 2000 system has more then 400 executables in c:\winnt\. There is no easy uninstall option for many of these executables (e.g. telnet.exe, ftp.exe, winpopup.exe), so manual selection of files is required. A problem that arises with such a stripped system is that service packs cannot be installed anymore. If a service pack is critical for the exam system (such as the service packs that were necessary against the *smbdie* exploit or the older *winnuke* exploit) the exam system needs a rebuild from scratch.

On systems using the NTFS file system it is possible to restrict access to specific executables for a group of users. As with stripping the system from executables this would require quite some effort since it is not very clear which executables are required for a normal functioning desktop. The advantage compared to a stripped system is that the same computer is still usable as normal desktop for users in a different group, and service packs will still install correctly.

To restrict access to superfluous software, a Linux system can be stripped as well. A typical Linux install has over a thousand executables installed. With Linux there is also the possibility that security updates are not installable if the system is stripped from a number of executables.

All common Linux file systems have the possibilities to deny access to executables. Groups of users can be allowed or denied access to executables. Since it is very clear which executables are required for normal system usage, it is very well possible to deny access to all executables by default and allow only the executables required during the exam for the exam users.

A different measure solving the same problem with very little effort is to set the exam user in a *chrooted* environment. Chroot is the irreversible change root utility. It sets a certain directory as root directory and therefore makes anything not in that directory or it's subdirectories inaccessible to the user. Users in a chrooted environment cannot see any files or executables from the system, except anything specifically set in their chrooted environment. Using a chrooted environment is therefore a very powerful option to make any superfluous software inaccessible. The advantage of chroot above stripping the system is that security updates and such might fail on a stripped system and they will install correctly on a normal system with a changed root for some user.

Network restrictions

A secure network is often desirable. The network provides a means for communication, and all communication can potentially be used to exchange answers or for finding answers in for example the on-line Encyclopedia Britannica. In many exam settings only communication with the LMS is required. The less other communication possible the fewer possibilities for misuse.

Networks can be secured on different levels. There are two commonly used models to describe networks, the Department of Defence (DOD) model and the Open Systems Interconnection (OSI) model. To describe the effect of a certain measure we will refer to the DOD model because it is the base for the design of the Internet protocol [8].

Table 2*: The DOD model*

DOD Layer	***Examples***
Application/Process	HTTP, FTP, TELNET, ICQ, IRC
Transport	TCP, UDP
Network	ICMP, ARP, RARP
	IP
Physical	Ethernet, Token ring

Instead of asking which communication should be disabled, it is better to ask which communication is allowed. In many situations only traffic to the LMS is desired, but a client computer might require more in order to function correctly.

There are some basic requirements for client computers. In order to access the LMS, the client computers should be capable of resolving the hostname of the LMS. The client computers need access to a Domain Name Service (DNS) server to resolve hostnames. If access to the DNS server is disabled, the client computer needs different means to resolve the LMS hostname. Also the web-browser on the client computer might be configured to use a proxy server, so access to the proxy server might be necessary.

Many clients are dependent on some Network Operating System (NOS), for example Novell, Windows NT or NIS. If the clients cannot function without access to the NOS, the communication with the NOS should be allowed. But this implies that a lot of services provided by the NOS will be available too. Many network operating systems feature chat, email and file sharing facilities that can be used for communication. If you allow access to the NOS, a close look at the capabilities of those services is important. Other restrictive measures might be necessary in that case.

There are several ways to configure a client system to restrict access to some network resource. These solutions are as safe as the client system. On a default Linux system the student will not be able to change network settings without root access, but on a default Windows 98 system a skilled student can easily change the network settings. Restrictive measures on level I or II are needed in that situation.

Level III - Network accessibility level

A simple tool is to cripple the DNS configuration of the client computer. Many communication applications need to resolve host-names in order to function. All these programs will fail to function without a proper DNS configuration. The browser should still know how to access the exam server. The name and IP address of the exam server can be set in the hosts file (/etc/hosts on Unix, c:\windows\hosts on windows) to make that server accessible. Some applications can still work without DNS configuration. Therefore restrictive measures on level I and on level II are important if you rely on this measure for level III.

The proxy option of the browser can be used to restrict the communication of the browser to the exam-server only. If the exam server is set to be the proxy server, the browser can only show pages on the exam server. There are several ways to secure the browser in such a way that changing the proxy option is impossible (see section 5.1). Not all browsers will function this way, since modern proxy servers communicate different then web-servers. Modern browser like Mozilla or Opera that try to make full use of proxy capabilities will not function with a web-server as proxy server.

192.168.10.20	client computer, with network card eth0
192.168.10.1	gateway for client computer
192.168.100.30	LMS server running the exam

the routing setup for that client would be:

route add -host 192.168.10.1 eth0

route add -host 192.168.100.30 gw 192.168.10.1

Figure 5: *Installing a crippled routing table*

A firewall-like solution on the client computer is to install a crippled routing table. This can be done quite easily by running a short script on every client just before the exam, see Figure 5. By removing the default gateway and installing a static route to the exam server the client cannot reach anything besides all routers and the exam server. Also peer to peer communication (communication from one computer to another computer within the same computer room) can be disabled with a static routing table. This is comparable to filtering in the DOD Network layer.

On Linux clients with the *dhclient* DHCP client software the /etc/dhclient-exit-hook script can be used to automatically setup this secured routing table after the DHCP information is received.

Level IV - Network availability level

By securing the communication in the network instead of the client it is much harder for the user to hack the secured configuration. The room where the network hardware is installed is usually locked, so there is no physical access possible. The most common way to secure the network is to install a firewall directly upstream from the computer room (filtering on the DOD Network and Transport layers) and to disable any peer-to-peer communication (filtering on the DOD Physical layer).

Most universities already employ routers and switches that can be used to realize firewall functionality. To stop peer-to-peer communication, the switch can be configured to put all exam computers into a private Virtual Local Area Network (VLAN) with isolated ports. To restrict upstream traffic, the router can be configured as firewall for this VLAN.

If the available network hardware has no firewall capabilities, there are several options. For example Linux and OpenBSD offer secure firewall functionality for situations where the available network hardware does not have firewall capabilities. Another option is to use a network interface card (NIC) with embedded firewall on every exam computer. For example the 3Com *embedded firewall solution* is suitable for this setup [9]. This firewall solution offers central administration, and can be used to update the firewalls for all exam computers simultaneously.

An extra extension to a firewall setup would be to use a proxy server. A proxy server allows filtering and logging on the DOD Application/Process layer. The filtering can be used to disable for example the chat facility in the LMS itself. In the closed-web situation the firewall should then only allow traffic to the proxy server, and the proxy server should allow only access to the LMS. In an open-web exam the firewall should still allow only access to the proxy server (so all other protocols besides http will be stopped) and the proxy server should allow access to any web site. The log files can be used to identify students who have illegally exchanged answers. The Squid proxy server can be used for these purposes [10]. It has a high performance and supports very flexible filtering.

Also IP spoofing can be disabled using VLAN technology [11] if the network between the computer room and the exam server is trusted. The ports for the computer room on the switch should be configured to put all the exam computers into a separate VLAN. The router should be configured to allow only traffic on that specific VLAN to use the IP range of the computer room.

If the network between the computer room and the exam server is not trusted a Virtual Private Network (VPN) can be used to stop IP spoofing. The exam server should only allow access to the exam from IP addresses from the private VPN range. The client computers should have a VPN gateway installed, or a VPN gateway should be available in the same network segment. If a VPN gateway is used it should only allow access to the VPN from the exam computers. The most dominant framework for VPNs is Internet Protocol Security (IPsec) developed by the Internet Engineering Task Force (IETF) [12]. An advantage to this setup is the possibility to link the VPN to the users identity. A number of VPN implementations can interface with smart cards, fingerprinting and iris-scan technology.

Satisfying security requirements for a specific exam

Depending on the exam requirements, some security levels need more attention than others. Apart from the exam requirements also several factors from the organization will have effect on the decisions. For example the available hardware, the knowledge and experience of the technical staff, organizational issues like how access to network hardware is organized, and also the number of exam computers will have effect on how the four level security model can be applied best.

The network architecture in many computer rooms is for example not designed to restrict communication. Only some high-end switches can be configured to disable peer-to-peer communication. Also the upstream network might be a fiber backbone, so a firewall or filtering proxy server cannot be put into place easily. If, because of this, securing level IV is not a viable option, then level III needs some extra attention. However, if the number of exam computers is very high, it is probably better to invest in hardware. For example the 3Com embedded firewall solution so large numbers of exam computers can be secured on level IV with little effort.

In section 7.1 and 7.2 the four level security model will be applied to two very common situations. The application of the four level security model in the examples also makes clear that the situation where an open-web exam needs extra software during the examination is the most difficult situation to secure. In section 7.3 the four level security model is applied to the situation at Wageningen University.

Open-web exams

For open web exams it is not an option to make all network resources unavailable to the user. In this situation the user is allowed to use the World Wide Web to gather information. The levels III and IV should allow access to the Internet for http (port 80) or to a proxy server allowing unrestricted http access. Since the World Wide Web offers a lot of options to exchange data, and the Internet access is unrestricted, logging is needed on levels III and

IV. In this situation with open Internet access it is important to pay extra attention to levels I and II.

There are also exams where only a limited part of the Internet is needed. In such a situation levels III and IV can be configured to allow access to these specific locations. If these locations do not have facilities to exchange data this situation is not much different from a closed-web exam.

Extra software needed at the exam

If some specific software is needed during the exam, for example Matlab, it will be very hard to secure levels I and II. There is a considerable effort needed to find the capabilities of a program. Matlab for example, can start other software and it can open network connections. Because Matlab can start other programs it will be extremely difficult to secure level I. Because it can open network connections it is important to pay extra attention to level III and IV.

Implementation example

Most web-based exams only require a web browser on the client. At Wageningen University both a Windows based and a very secure Linux based exam client are developed. The Windows based client is secure on level I and II, and as far as the browser secured on level III. The Linux client is completely secured on levels I, II and III. The decision to not secure level IV is made because of both the organizational structure at Wageningen University and because of the network architecture in the computer rooms. While a central ICT department administers the network, local ICT departments administer the exam computers in the computer rooms. The central department configures the network for general use and is not involved in temporary settings like exam settings. But to secure level IV support from the central department would be required. Also some computer rooms have shared Ethernet segments and some have switched Ethernet segments. A lot of effort is required to configure the network for each computer room separately. Because the exam computers are sufficient secured on levels I, II and III it is not worth the effort to secure level IV.

Conclusion

Web-based exams are more vulnerable to fraud than regular exams. This article has described the most important weaknesses for web-based exams. This article also presents a comprehensive set of measures organized in a four level security model. The levels present software accessibility measures on the client, software availability measures on the client, network accessibility on the client, and network availability on the network. The security model supports the selection of a specific combination of measures for a specific exam setting. When applying the model the organizational structure and already available facilities should be taken into account.

Acknowledgements

The authors would like to thank Gert Klein and Gerard Folkerts for their information and help on windows security.

References

1. Blackboard (2002) *Blackboard Learning System.* http://products.blackboard.com/cp/release6/LSR6WP.pdf
2. Questionmark (2002). *Perception for Web product information.* http://www.questionmark.com/uk/infosheets/qmperception.pdf
3. Software Secure, Inc. (2002). *Securexam Browser.* http://www.softwaresecure.com/browser.htm
4. Questionmark (2002). *Perception Secure Browser product info.* http://www.questionmark.com/uk/infosheets/perception_secure_browser.pdf
5. Richardson T., Stafford-Fraser Q., Wood K., Hopper A. (1998). *Virtual Network Computing* IEEE Internet Computing, 2(1): 33-38
6. Symantec (2002). *PcAnywhere fact sheet.* http://enterprisesecurity.symantec.com/content/displaypdf.cfm?pdfid=35&EID=0
7. Sharpe R., Warnicke E. (2002). *Ethereal user guide.* http://www.ethereal.com/distribution/docs/user-guide.pdf
8. Siyan K., Hawkins N., Wettern, J. (1997). *Inside Tcp/Ip: A Comprehensive Introduction to Protocols and Concepts.* New Riders Publishing

9. 3Com (2002). *Embedded firewall solution datasheet.* http://www.3com.com/other/pdfs/products/en_US/400741.pdf
10. Pearson, O. (2002). *Squid, a user's guide.* http://squid-docs.sourceforge.net/latest/html/
11. IEEE-SA Standards Board (1998). *IEEE 802.1Q: Virtual bridged local area networks.* http://standards.ieee.org/getieee802/download/802.1Q-1998.pdf
12. Kent, S., Atkinson, R. (1998). *RFC2401: Security Architecture for the Internet Protocol* http://www.ietf.org/rfc/rfc2401.txt

Chapter 3

Database functionality for learning objects

Abstract

The development of student-activating digital learning material in six research projects revealed several shortcomings in the current learning management systems. Once the SCORM 2004 and the IMS Sharable State Persistence specifications are implemented in learning management systems, some of these shortcomings will be obsolete. We have, however, identified six functionality clusters with significant added value for learning materials that are neither addressed by SCORM 2004 nor by IMS Sharable State Persistence. These clusters are: 1) support for adaptivity, 2) support for retrieval of current and previous states, 3) support for comparison of results, 4) support for student tracking for pedagogical research, 5) shared reference database functionality, and 6) problem scenario database functionality. The significance of each of these functionality clusters is discussed, and illustrated with examples from digital learning material. This article, furthermore, describes the workarounds needed to deploy these examples in the current learning management systems. A general solution that will support all six functionality clusters is presented: database functionality for learning objects. This is essentially an interface to database functionality with support for complex data models, data integrity, and concurrent access, in which authors can define the data model, and both learning objects and authors can query the database.

Published as Sessink, O.D.T., Beeftink, H.H., Hartog, R.J.M. (2005) *Database functionality for learning objects.* Journal of Technology, Instruction, Cognition and Learning 2(4): 337 - 357

Introduction

One of the means to support a constructivist approach in E-learning is student activating learning material [1]. The Food and Biotechnology (FBT) program at Wageningen University has been developing a large amount of digital learning material, targeting courses at the BSc and MSc level. The learning material has been used for several years and is appreciated by both students and lecturers (van der Schaaf, 2003, Wilmsen 2002, Wilmsen 2003, Diederen, 2003).

The FBT program comprises many smaller projects as well as several larger research projects. These larger research projects aim to exploit all pedagogical possibilities of digital learning material. While the smaller projects use mainly static learning objects, the larger research projects use more advanced learning objects. Many of these advanced learning objects process data that are often related to a user action (e.g. a mouse click in a specific region, some text submitted in a form, or a drag-and-drop action). The data are used to adapt the content or the presentation of such an advanced learning object to the user. We will call these learning objects *dynamic learning objects*. Our experience with the implementation of these dynamic learning objects in the FBT program revealed certain shortcomings of the current learning management systems (LMSs). It seems likely that these shortcomings still will exist in the next generation of LMSs.

The next generation LMSs are likely to support the Sharable Content Object Reference Model (SCORM) 2004 specification (Advanced Distributed Learning, 2004) with the Simple Sequencing specification from IMS (IMS Global Learning Consortium, 2003). Once LMSs support SCORM 2004, some shortcomings to employ dynamic student activating learning material will be resolved. But serious shortcomings remain for six functionality clusters used in the advanced learning material. These functionality clusters are: 1) support for adaptivity, 2) support for retrieval of current and previous states, 3) support for comparison of results, 4) support for student tracking for pedagogical research, 5) shared reference database functionality, and 6) problem scenario database functionality. An LMS with IMS Sharable State Persistence (IMS Global Learning Consortium, 2004) allows more functionality, but not enough to satisfy the requirements for the six functionality clusters.

This article will give an overview of these six functionality clusters, and their significance for learning material. It will, furthermore, discuss the requirements for each of these clusters, and state why these requirements are neither met by the current generation LMSs nor by LMSs supporting SCORM 2004 and IMS Sharable State Persistence 1.0. The functionality clusters will be illustrated with examples from learning material developed in the research projects of the FBT program. In addition an overview is given of workarounds that were developed to deploy this learning material in the current generation LMSs. Finally, database functionality for learning objects is described, a single solution that will allow learning objects to implement all functionality clusters. This is essentially an interface to database functionality with support for complex data models, data integrity, and concurrent access, in which authors can define the data model, and both learning objects and authors can query the database.

Functionality clusters

Support for Adaptivity

Adaptivity is a broad concept, defined as "Adaptive systems cater information to the user and may guide the user in the information space to present the most relevant material, taking into account a model of the users goals, interests and preferences" (Brusilovsky, 1998a). In an educational context, the user is a student, and the model often is a model of the users competences, extended with other user characteristics. Such a model, in which the competences are the main source of information, is called a student model. An important application of adaptivity in the educational context is student guidance (Brusilovsky, 1996).

For demo sites a, b, c, and f, a student model for the Process Engineering domain has been developed. The dynamic learning objects can set or get levels for competences defined in this student model. The data in the student model are used for *adaptive annotation* and *adaptive trail generation*. In adaptive annotation hypertext links are augmented with user specific information about the learning material behind the links (Eklund, 1998), such as information whether the prerequisites of the learning object match the competences of the student. Adaptive annotation can be used to minimize the user's floundering and make the learning with hypermedia more goal-oriented (Brusilovsky, 1995, Brusilovsky, 1998b). In Figure 1, an example of adaptive annotation in the Process Engineering domain is shown in which different users will get a different navigation view depending on their competences.

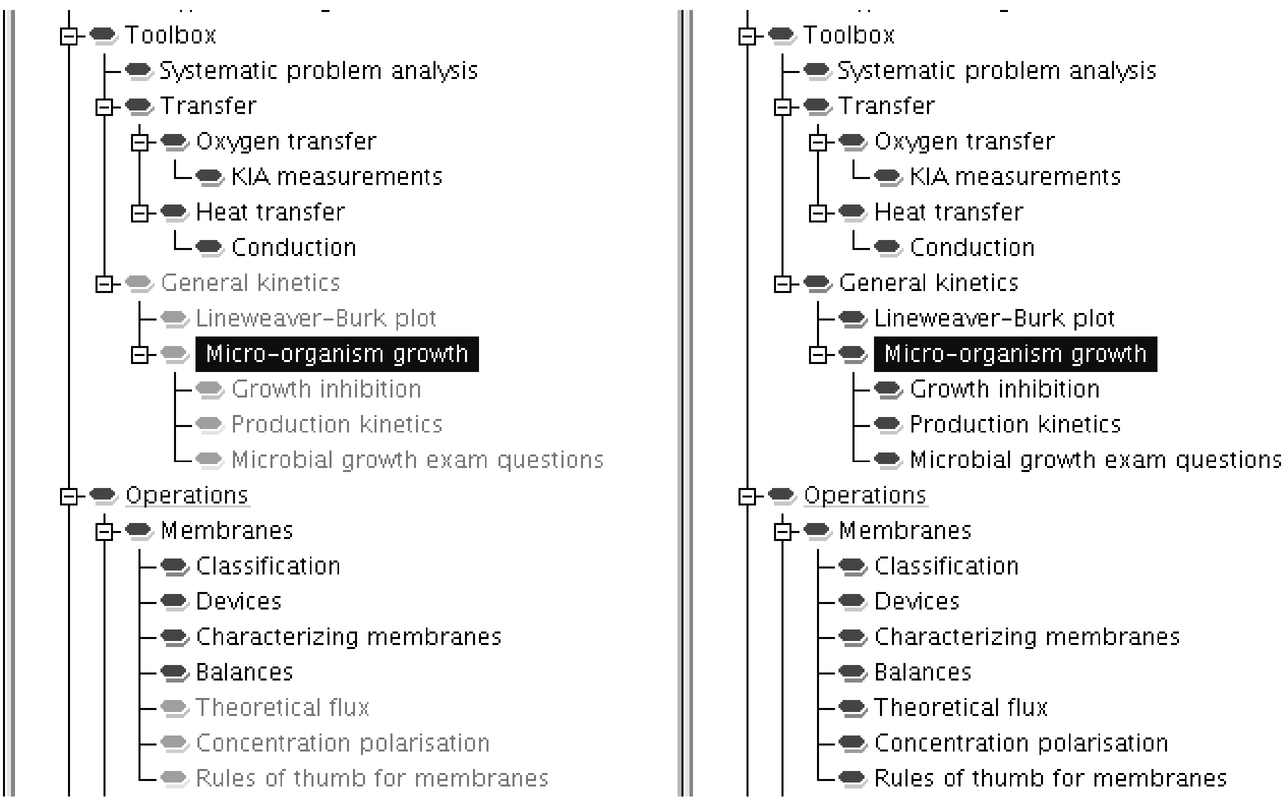

Figure 1: *Two screenshots of the navigation in the "library" for two users with different competences.*

In adaptive trail generation, a reduced selection of links is presented that forms an optimal match with the needs of the user. In demo site b adaptive trail generation is used to select assignments for the students. Both the answers to previous assignments and the choice of the user for an “easy” or a “difficult” assignment will affect which assignment the user will get next. The trail will eventually lead all students to the same level of competences, regardless of their prior knowledge.

Any adaptive application needs a user model, and therefore needs to store and retrieve data. The combination of Simple Sequencing and the CMI model in SCORM 2004 does facilitate some degree of adaptivity. Learning objects can set learner performance levels in the CMI model, such as *cmi.completion_status* and *cmi.objectives*. Sequencing rules can determine the logic how to proceed along a series of learning objects based on these learner performance levels. The only properties that can be set by the dynamic learning objects, however, are the fixed properties available in the CMI model. Adaptive systems might, for example, need a knowledge decay value, or an expiration time, which cannot be stored in the CMI model. Alternative implementations of adaptivity, therefore, need a different, more flexible way to store and retrieve user model data.

Support for retrieving current and previous states

Many dynamic learning objects can be in different states. A very simple example is shown in Figure 2, in which a Java applet graphs a simulation of bacterial growth kinetics based on a parameter set by the user.

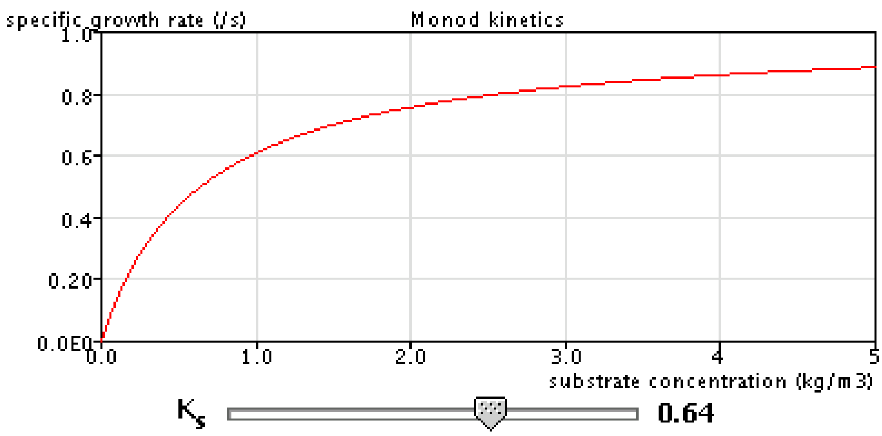

Figure 2: *Screenshot of an interactive graph, showing the Monod kinetics model.*

This parameter value is the state of the dynamic learning object in this specific example. The state of the learning object can be stored when the user quits the learning object. If the user would return to this learning object at any later time, the learning object could retrieve and restore this state. In contrast with adaptivity, where the data are stored in a well-structured student model used by many learning objects, state data are not structured for the outside world, and the data are only used by a single learning object, or a small number of closely related learning objects.

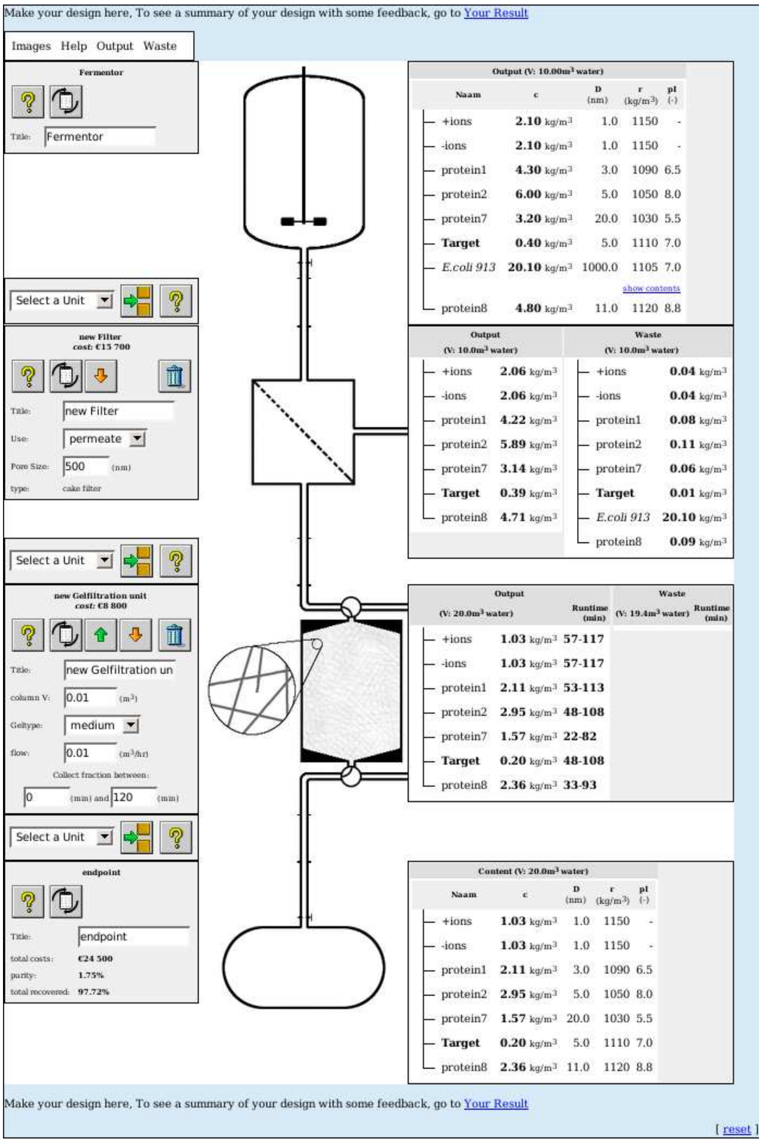

Figure 3: *The downstream process design environment with two unit operations. Typical designs need 5-8 unit operations.*

In the current one-parameter example, the need for storage and retrieval is not really pressing. In demo site c, however, an environment is developed with which Biotechnology students should design a downstream processing chain (Figure 3). This design environment is used in several learning objects. A downstream processing chain is a sequence of so called unit operations that together purify a product stream. Each of these unit operations has several properties that students have to set; in addition, the number and the order of the unit operations are also set by the students. Students use this design environment in several assignments during several days. In a first assignment, students design a downstream processing chain in order to obey given specifications. In a subsequent assignment, the specifications are changed and students need to adapt the design from their first assignment to the new situation. Given the number of settings in such a design it is essential that these learning objects can retrieve a previous state. To allow these learning objects to retrieve the design from another assignment, it is essential that learning objects can retrieve the state from each other.

The same demo site features also multiple-choice questions with feedback. The feedback is based on the particular answer of the student, but also on his history of previous attempts to the same question. A student choosing option D, after having tried two other options, will get feedback that differs from the feedback given to a student answering option D without any other tries. In demo site d, the states from several learning objects together have consequences on what is presented to the student or what feedback is given. To enable such functionality the learning objects need to retrieve a full history of state changes.

The CMI model in SCORM 2004 has limited support to store the state of a learning object. The *cmi.suspend_data* property can be used to store data. This storage is not persistent. The data are available as long as the student has not finalized the learning object. The *cmi.suspend_data* property is only accessible by the learning object itself, not by other learning objects. The history of this state, furthermore, cannot be retrieved. The size of the property is also very limited. A compound state (e.g. a state that consists of several attributes and their values) can easily grow beyond the size limit of this property.

The IMS Sharable State Persistence specification allows much more functionality. Other learning objects can retrieve the data stored by a particular learning object, and the storage size is not limited. If the learning objects themselves implement the functionality to keep previous states when adding a new state, the retrieval of current and previous states functionality cluster is fully supported in an LMS that features IMS Sharable State Persistence.

Support for comparison of results

An important aspect of student-activating learning material is the way the learning material motivates the student to become active. Being able to view the work from other students can motivate in several ways (Lunetta, 1990). This ‘work’ will be very different in different learning materials. The work might, for example, be the final result of an optimization, a solution strategy, a report, or a design.

In an area where there is not just one correct answer, such as in design or in social sciences, an overview of the work of other people might increase the awareness of the students of the different design strategies or different opinions. The differences in approach and differences in decisions and consequences by themselves stimulate many students to reconsider their own decisions and their own arguments. For design assignments specific, the awareness that design is not the same as optimization might be very important.

When the learning material is used in a distance learning setting, viewing the work of other students can furthermore improve the motivation, because it makes the student aware of other students working on the same learning material.

If it is possible to score the work of students on one or more aspects, the overview can include those scores. Showing these scores is not identical to “having a competition”, but it does awake some competitive feelings in some of the students. For these students these feelings act as a drive to perform better. For some other students, however, a competitive atmosphere reduces their confidence and satisfaction. These students are likely to become less motivated. When the competition element is emphasized, both effects are likely to become more pronounced.

We believe, therefore, that competition should be based on a score that is related to the assignment and not related to the students' knowledge and skills. In a design assignment, for example, the distance between the design and the design criteria can be a good competitive element. The reason for this is twofold. Firstly the measurement of the students' knowledge and skills is also determined by prior knowledge and skills. Secondly any negative feelings with respect to competition will be less strong if competition is directed to the assignment (e.g. the design requirements) and not to the knowledge and skills of the person or the group.

First Design: score 8.0						
		Yield (%)	Purity (%)	Waste (m^3)	Costs (€)	Steps
	You	**97.72** (#1)	**1.75** (#43)	**29.43** (#13)	**24500** (#1)	**2** (#1) , reactor, filter, gelfiltration, endpoint
best yield	Willem	**92.21**	98.89	31.45	66940	4, reactor, centrifuge, ion-exchange, gelfiltration, ion-exchange, endpoint
best purity	Ivan	85.46	**100.00**	35.32	65215	8, reactor, filter, ion-exchange, gelfiltration, ion-exchange, ion-exchange, ion-exchange, ion-exchange, endpoint
least waste	Mannis	85.02	95.42	**20.05**	37217	3, reactor, filter, ion-exchange, ion-exchange, endpoint
best price	Mannis	85.02	95.42	20.05	**37217**	3, reactor, filter, ion-exchange, ion-exchange, endpoint
least # steps	Guest	85.97	95.39	20.05	43966	**3**, reactor, filter, ion-exchange, ion-exchange, endpoint

Second Design: score 8.5						
		Yield (%)	Purity (%)	Waste (m^3)	Costs (€)	Steps
	You	**100.00** (#1)	**0.91** (#35)	**0.00** (#1)	**0** (#1)	**0** (#1) , reactor, endpoint
best yield	Sebastien	**97.10**	98.68	60.75	81117	4, reactor, filter, disruptor, gelfiltration, ion-exchange, endpoint
best purity	Florentina	93.01	**99.99**	53.20	75433	6, reactor, filter, filter, disruptor, gelfiltration, ion-exchange, ion-exchange, endpoint
least waste	Olivier	91.18	98.83	**14.28**	28064	5, reactor, filter, disruptor, filter, ion-exchange, gelfiltration, endpoint
best price	Olivier	91.18	98.83	14.28	**28064**	5, reactor, filter, disruptor, filter, ion-exchange, gelfiltration, endpoint
least # steps	Sebastien	97.10	98.68	60.75	81117	**4**, reactor, filter, disruptor, gelfiltration, ion-exchange, endpoint

Figure 4: *Scores for several criteria in the downstream process design environment.*

In demo site c, such game-like competition is stimulated (Figure 4). In the downstream process design environment, introduced in the previous section, students can compare their design with the designs from other students on several criteria, such as purity, yield, costs and waste volume. The scores are only visible to students that have finished an assignment. Many students, after seeing the scores, return to the assignment to improve their design and try to get a place in the high scores in one of these criteria.

In the cell growth experiment design in demo site a, the students can compare the costs of their experiments with the costs other students have made. In this site the scores serve another purpose as well, they will give the students an idea of the costs involved in these experiments, and also what the impact of the different experiment designs is on the costs.

Neither the SCORM 2004 specification nor the IMS Sharable State Persistence specification enable learning objects to store and retrieve information related to multiple users.

Support for student tracking for pedagogical research

When researching specific issues with learning material, for example the development of a student model for adaptivity purposes, data about the student's behavior are very valuable. One way to get these data is from observations. The amount of data from observations is, however, limited to the time of the observer. Other data can be collected using questionnaires. The data from questionnaires are, however, subject to the interpretation of the questions by the student and subject to the subjectivity of the student. Tracking data cannot replace these types of data, but can be a valuable addition. Tracking data are very abundant, every detail of every user's activity can be logged, and tracking data are objective.

These tracking data can, for example, be used to find common behavior in large groups of students, or to find learning objects that are different from others in the way students use them, or to find students that have a different behavior from other students (Mazza, 2004). This can give information about the understanding of different parts of the learning material, the difficulties students have with the learning material, and the effect of the feedback presented by the learning material to the students.

Because of the number of data, it is generally not feasible to analyze tracking data manually. There are, however, various tools available to support analysis. Text analysis tools can be used to analyze tracking data in raw text format. Tracking data in relational database management systems can be analyzed using SQL queries.

In demo site d, learning material has been developed that tracks the user's interaction in high detail. Both the interaction and navigation within a learning object, as well as the navigation between learning objects are stored. It is possible to follow all the interaction between the students and the learning material.

After interpretation of the raw logs, a typical tracking log from a student looks like:

1. student starts learning object I: *a question is presented*
2. student chooses possibility B in this learning object: *feedback on possibility B is presented*
3. student starts learning object II in the library: *information is presented*
4. students continues with learning object I and chooses possibility A: *feedback on possibility A is presented including a link to learning object III*
5. student starts learning object III, etc. etc.

In demo site b, tracking data are used to analyze the assignments used in an adaptive trail generation system. A typical tracking overview is presented in table 1.

Table 1: *Typical tracking results from an adaptive system.*

question	*total visits*	*total answers*	*correct answers*	*average tries*
transfer_ld1	24	19	18	1.7
transfer_ld2	42	23	15	4.5
transfer_ld3	23	18	18	1.2

The tracking data in this demo site are mostly used to find questions that are too difficult, which learning objectives need more assignments, or which questions appear to be too difficult.

Neither SCORM 2004 nor IMS Sharable State Persistence support storage and retrieval of this kind of tracking data.

Shared reference database functionality

Reusability of digital learning objects has always been an important quality of digital learning material. Not all reusable objects, however, can be deployed in the right context in LMSs. LMSs are designed to deploy learning objects such as a chapter in a PDF document. A user might access such an object from a listing of course documents provided by the LMS. LMSs are currently not designed to deploy tiny learning objects, such as a single definition of a specific term within a certain field of science. Presenting such small learning objects in a listing of course documents will cause a lot of overhead for the student, negatively affecting the total learning experience from the learning material.

Reference data are often such small learning objects. Reference data are usually not to be studied by the student, but add value to other learning material. Reference data are typical used in multiple learning objects. Often the reference data are identical in multiple learning objects, or at least that would be the preferred situation. It is, therefore, useful to have a single shared database for reference data. Many learning objects then can use the same reference data. For example, all learning objects about Process Engineering can use the same definition for a term like *partition coefficient*.

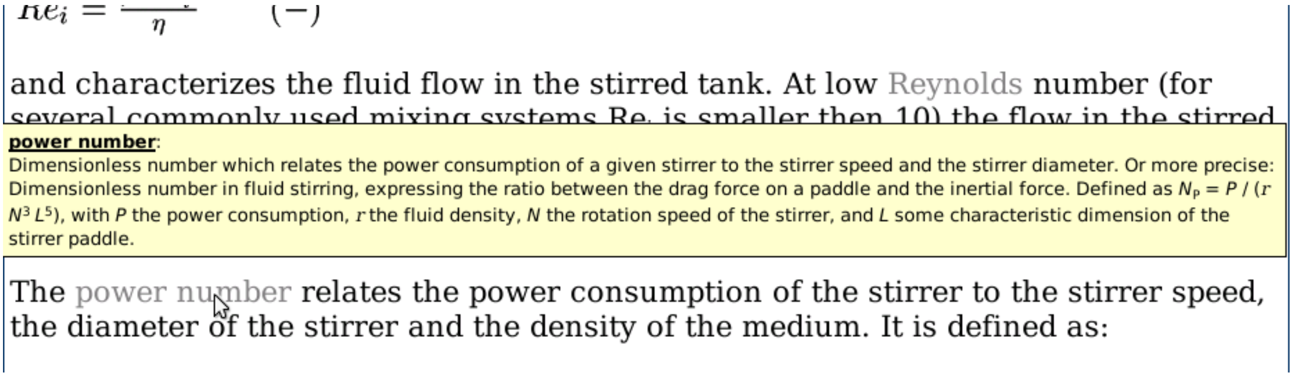

Figure 5: *Example of the shared glossary functionality.*

In demo site f, several examples of shared reference database functionality are used throughout all learning objects. The site features a shared glossary functionality, in which jargon terms are explained and translated into Dutch when required. It is a simple list with keywords and their meaning. The explanation and translation are server-side added to keywords in the learning objects. If a student moves the mouse over a keyword, a layer with the meaning and translation of that keyword is shown.

This site also features shared equation functionality. Equations play a key role in engineering education. The equation functionality needs a more complex data model. Every equation has a list of associated equations describing the symbols in the pertinent equation. If a student moves the mouse over an equation, a layer is shown explaining all the symbols used in the equation, and their meaning. Both the glossary functionality as well as the equation functionality are highly appreciated by the students.

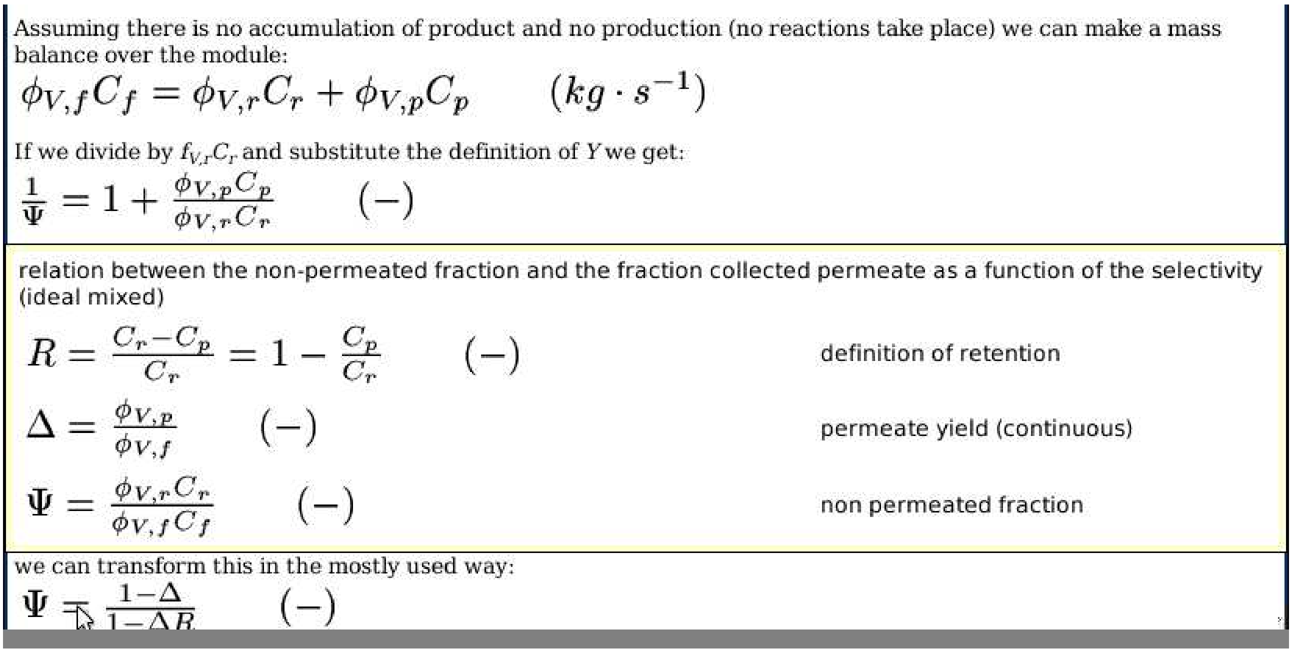

Figure 6: *Example of the shared equation functionality.*

Also in demo site e, about food safety management, six reference databases are used. One of these databases has additional features and is described in the next section.

Every definition in the glossary and every equation should be a separate object in terms of sharable content objects. Adding these objects to other objects, however, is not possible in the current LMSs. There is no interface to have learning objects exchange content. Having relations between these objects (e.g. the symbols used in the equation and their definition) would also be impossible.

An important aspect of reference data in a LMS would be the reusability of the database itself. Just like a learning object created by one author, also a reference database created by one author might be very interesting to another author. It should, therefore, be possible to copy a reference database, or to use another author's reference database.

Large data collections, especially if they are also used outside the educational context, should not be in the LMS database. This would add additional load on the LMS, and management of a large data collection would probably require a different approach. A link to an external database server would then be more appropriate. It would be very useful though, if access to an external database server could be provided using the same interface as access to a database within the LMS itself. This would certainly improve the accessibility of the external database.

The SCORM 2004 specification does not allow sharable data stores. The IMS Sharable State Persistence specification allows shared access to a persistent data store. Every learning object, however, needs low-level code to select the correct data from the data store. There is no high level query functionality specified in the IMS Sharable State Persistence specification. There is, furthermore, no interface for the author to add or update data in such a data store. For reference database functionality the IMS Sharable State Persistence specification is clearly not sufficient.

Problem scenario database functionality

A common issue with activating learning objects, especially with exercises, is that using the same object over and over again is not preferred. After using the same exercise a couple of times, many students have heard the answers from other students, and the motivation to solve the exercise by themselves might decrease. Also a single student might want to practice an exercise multiple times. Some variation within the exercises is then preferred. Digital learning material is particularly suited to offer such variation, because a single digital learning object can contain several slightly different exercises. Especially in numerical problems, often found in science courses, it is relatively easy to construct variants of the same exercise. When working with non-numerical or real-life examples in the exercises, it is often not so easy to construct multiple variants, because realistic data needs to be collected and introduced into the exercise. Often, however, different exercises can use the same realistic data. A database for these problem scenarios is, therefore, a very useful functionality. Creating equivalent but different exercises is, furthermore, often a difficult task. Such a database with problem scenarios can facilitate the creation of equivalent problems by giving a clear overview of the differences and similarities between the problems.

Demo site e, dealing with food safety management, contains a problem scenario database with microbiological hazard problems. The database contains hazard issues (e.g. *contamination with E. coli*) as well as information how to detect such a hazard and how to manage such a contamination. A learning object with an assignment can select a problem from this database and introduce it to the student. If required, even a random problem can be selected. Another learning object in the "library" section from this site allows the students to search in the same database for possible hazards, ways to detect them and how to manage them. Because these learning objects use the same database, the solution to any problem presented to the student is guaranteed to be available in the "library" section. Another advantage is that problems can be added to the database after the learning objects have been completed, or even during the course.

Also in demo site a, about cell growth experiment design, a problem scenario database is used. Students have to determine model parameters using the virtual experiments. A database with model parameters from different micro-organisms is used to give each student a unique and realistic assignment. This database with microbiological model parameters is accessed by several learning objects that feature virtual experiments.

With respect to support of problem scenario database functionality, the shortcomings of the SCORM 2004 specification and the IMS Sharable State Persistence specification are the same as described in the reference database functionality section.

Workarounds for current generation LMSs

The dynamic learning objects from the demo sites that were referenced above cannot be deployed in most available LMSs yet. Nonetheless, we have developed these dynamic learning objects and we are using them in combination with a well-known LMS. Several workarounds, therefore, have been developed. At Wageningen University we have used these workarounds for several years, with full appreciation from students and authors. Three workarounds will be described here, two have been used for the last years, the third is currently in testing stage.

In the first workaround that we developed, the functionality of the LMS is extended. Some LMSs include a database management system (DBMS) and a web server. Other LMSs need a third party web server or DBMS. In such a situation the configuration of the web server can be modified. In our situation a very common web server is providing the web interface, and a common relational DBMS provides the internal LMS storage. The web server configuration is modified to enable execution of server-side applications. A server-side application provides the communication between learning objects and the LMS. Because there is no public application programming interface (API) to communicate with the LMS, the server-side application connects directly to the DBMS.

Because of the previous experience we had, and its wide availability, PHP was chosen as server-side language. PHP has native database interfaces to all common DBMSs, including the type of DBMS our LMS used. We have developed a PHP library for communication between the LMS and our server-side learning objects. The library enables retrieval of administrative data from the DBMS. A PHP application on the web server enables client-side dynamic learning objects (e.g. JAVA or FLASH objects) to communicate with the LMS, and to store and retrieve data. The communication between the client-side dynamic learning objects and the server-side PHP application is based on the http protocol, which is supported by standard components in both FLASH and JAVA.

This workaround has two important drawbacks. Firstly it can introduce security problems. Most LMSs that use a third party web server are designed with a specific configuration for that web server in mind. These LMSs are not designed for a configuration in which the web server allows server-side execution of dynamic learning objects. The server-side dynamic learning objects are not controlled by the LMS; they can, for example, alter information in the database, or remove learning objects. Secondly, both the learning objects that use the workaround, and the workaround itself are not portable to other LMSs. The server-side learning objects depend on the availability of the PHP library for communication. And, although http communication between client-side and server-side applications is very common, the client-side learning objects depend on the PHP application on the web server for communication. The PHP library itself, furthermore, is specific to the LMS/DBMS combination we have.

The second workaround avoids the security issues with server-side learning objects on the web server by running all dynamic learning objects on a separate host. This separate host uses widely available web server and DBMS software. To synchronize sessions between the external host and the LMS host, the LMS host runs a server-side application that exports the

LMS session identifier in a cookie. Learning objects on the external host can select information from the DBMS on the LMS host using this session identifier.

Again, because of previous experience and its wide availability, the Apache web server was chosen for this external host in conjunction with the PHP server-side scripting language, and the MySQL DBMS. A PHP library is developed to select administrative data from the DBMS on the LMS host. The server-side dynamic learning objects use this library to store and retrieve user specific data. A PHP application on the external host allows client-side dynamic learning objects to store and retrieve data. Also in this workaround, the http protocol is used for the communication between client-side learning objects and the PHP application.

This workaround has also three important drawbacks. First, the learning object management functionality from the LMS cannot be used for the learning objects. The learning objects on the external host are not managed by the LMS. Also the courses in the LMS contain links to the external server, and these links are not managed either. If files are moved on the external server, the links in the LMS will be incorrect. Secondly, both the learning objects that use this workaround, and the workaround itself are again not portable to other LMSs. Although the configuration of the external host is very common, the learning objects on the external host are specific to this configuration, and cannot be deployed in regular LMSs. The PHP library on the external host and the application on the LMS host are specific for the LMS/DBMS combination. Last, the DBMS on the LMS host has to allow access to the LMS database from the external host. From a security point of view this is undesirable.

The third workaround, currently in testing stage, enables the functionality from both previous workarounds, while avoiding some of their drawbacks. This workaround is implemented as a Blackboard building block. The building block uses the API provided by Blackboard to communicate with the Blackboard LMS and to access administrative data. Client-side dynamic learning objects, managed by the LMS, use the http protocol to communicate with this building block. The building block can connect to a DBMS on an external host to store data. The building block can also redirect browsers to a web server on an external host. The building block will add a hash to the URL and store the same hash in the DBMS on the external host, together with user information from the LMS. The dynamic learning material on the external server can select the current session from the DBMS using the hash provided in the URL.

This workaround does not need a special configuration on the LMS host because it uses a public API to communicate with the LMS, and there is no need to allow access to the DBMS from an external host. The other drawbacks, however, do still apply.

Specification requirements

The previous sections have made clear that there is need for specification for LMSs such that advanced learning material as described in this article can be deployed in any regular LMS. This specification should define an interface between dynamic learning objects and the LMS. The interface should allow learning objects to select specific data from a database, and to add or update specific data in this database. The specification should allow the use of a high-level data manipulation language. The data should be in a well-defined data model. The database should enforce data security and integrity, also during concurrent access. An obvious choice would be to use a relational database and use SQL in the API. Another option would be to use an XML database and allow the use of the Xquery language in the API. Because many data in the database will be related to users and learning objects, the database functionality should allow queries to select data related to the current user, or the current learning object.

Another requirement for this specification is that the data model and the data should be portable to other LMSs. It should be possible to copy the data model and the data to another LMS just like a learning object. The specification should, therefore, define a format for the exchange of the data model and the data.

The interface between the dynamic learning objects and the database within the LMS is the most important interface for the specification. New answers from students have to be inserted or updated in the database, or definitions have to be selected from the database. SCORM 2004 includes an interface for client-side dynamic learning objects (e.g. JAVA applets or FLASH movies). This SCORM 2004 runtime API defines a javascript interface for communications initiated by a client-side learning object to the LMS. The communication is currently restricted to data that is part of the CMI model. The interface between the learning object and the database could be an extension of this API, for example a method named LMSRunSql().

An example of the code in a dynamic learning object then looks like:

```
var RunSqlResult = LMSRunSql("
     select img from equation_symbols
     inner join equation
          on equation_symbols.eq_id = equation.eq_id
     where equation.name = 'Monod'
     ");
if (RunSqlResult == 0) {
     // Succeeded, process the results.
} else {
     // Error condition, handle appropriately.
}
```

Support for server-side dynamic learning objects in LMSs is a completely new area. Much of the learning material described in this article is developed using server-side learning objects. As described in the workarounds section, introduction of server-side learning objects in current LMSs may lead to security issues. The discussion whether or not to include support for server-side dynamic learning objects in LMSs is, however, beyond the scope of this article.

There is currently no standard for learning objects describing a communication interface between server-side learning objects and LMSs. If the LMS would support server-side dynamic learning objects, however, the interface would simply be a function library or object class in the language of the server-side object. The learning object could then simply call a method from the LMS for communication.

An example of the code in a server-side learning object in PHP would look like:

```
$RunSqlResult = $LMS->RunSql("
     select img from equation_symbols
     inner join equation
          on equation_symbols.eq_id = equation.eq_id
     where equation.name = 'Monod'
     ");
if ($RunSqlResult == 0) {
     // Succeeded, process the results.
} else {
     // Error condition, handle appropriately.
}
```

Implementation requirements

An implementation of the database functionality for learning objects in an LMS essentially adds a DBMS to the LMS, and the interface for dynamic learning objects to access the DBMS. The data model for the data in the database is to be defined by the author, not by the LMS manufacturer. A user interface to implement a data model and edit data in the database should, therefore, be part of this implementation. Assuming the specification would propose to use a relational database and SQL as query language, a facility to upload an SQL data definition to the LMS would satisfy this requirement. However, the user interface would certainly be improved if a database upload system such as for instance found in MS Frontpage would be added to the LMS. The user interface should also feature import and export functionality to transfer the data model and the data from one LMS to another.

An author might want to view and analyze data in the database. Other data in the database might be provided by the author. In both cases, the LMS should have an interface for the author to work with the data in the database. Such an interface could be a web interface like phpMyAdmin (http://www.phpmyadmin.net/). Alternatively upload and download of data in well-known formats (e.g. a tab delimited text file, or MS Access file format) could be facilitated. An additional or alternative interface would be to enable JDBC or ODBC connectivity to the database in the LMS.

An important issue for the LMS is the authorization of access to the database. Both authors and learning objects may need access to specific parts of the database. The LMS should control which learning objects and authors are allowed to retrieve data from which part of the database, and which learning objects and authors are allowed to edit or store data in which part of the database. Access to a reference database, for example, should probably be allowed for every learning object. Access to a problem scenario database, however, should probably be restricted to certain specific learning objects. Rules for access control should include the possibility to grant access to a single learning object, or multiple learning objects selected by author, course, or department. The LMS should also include a user interface to configure the authorization rules for the database.

Conclusion

In four research projects exploiting the possibilities of digital learning material we have identified six functionality clusters for learning materials that have a significant added value. The six functionality clusters are support for adaptivity, support for retrieval of current and previous states, support for comparison of results, support for student tracking for pedagogical research, shared reference database functionality, and problem scenario database functionality. The learning materials that use these functionality clusters have been used and evaluated in the research projects, and are highly valued by both lecturers and students. The six functionality clusters have requirements that are not met by the current and possible future generation of LMSs. Some of the requirements are touched upon by the SCORM 2004 specification and the IMS Sharable State Persistence specification, but a list of shortcomings remains even after these specifications have been implemented in regular LMSs.

It is clear that there is a need for database functionality for learning objects. A standard is needed to guarantee interoperability of learning materials using the database functionality for learning objects. This standard should specify an interface between dynamic learning objects and the database on the LMS. The data model for the data in the database is to be defined by the author, not by the LMS manufacturer. The standard should, therefore, also specify how the data and data models in the database can be transferred from one LMS to another.

Once database functionality for learning objects is implemented in LMSs the functionality of learning objects can be extended with all six functionality clusters.

References

1. Anderson, J.R. (1995). *Learning and memory. An integrated approach.*, John Wiley & Sons, Inc.
2. Schaaf, H. van der, Vermuë, M.H., Tramper, J. & Hartog, R.J.M. (2003). *A design environment for downstream processes for Bioprocess-Engineering students.* European Journal of Engineering Education, 28(4), 507-521.
3. Wilmsen, H.M., Bisseling, A.H.J. & Hartog, R.J.M.(2002). *Web Based Learning Support for Experimental Design in Molecular Biology.* In Philip Barker, & Samuel Rebelsky, (Ed.), Proceedings ED-MEDIA 2002. (pp. 2063-2068).
4. Wilmsen, H.M., Hartog, R.J.M. & Bisseling, A.H.J. (2003). *Web based learning support for experimental design in molecular biology: a top-down approach.* Journal of interactive learning research, 14(3), 301-314.
5. Diederen, J., Gruppen, H., Hartog, R.J.M., Moerland, G. & Voragen, A.G.J. (2003). *Design of activating digital learning material for food chemistry education.* CERP - Chemistry Education : Research and Practice, 4(3), 353-371.
6. Advanced Distributed Learning (2004). *Sharable Content Object Reference Model (SCORM) 2004 2nd Edition.*, Retrieved Jan. 5 2005 from http://www.adlnet.org/
7. IMS Global Learning Consortium (2004). *Sharable State Persistence version 1.0*. Retrieved Jan 5. 2005 from http://www.imsglobal.org/
8. IMS Global Learning Consortium (2003). *Simple Sequencing and Behavior Model version 1.0*. Retrieved Jan 5. 2005 from http://www.imsglobal.org/
9. Brusilovsky, P., Kobsa, A., Vassileva, J., editors (1998). *Adaptive Hypertext and Hypermedia.* Kluwer, Dordrecht.
10. Brusilovsky, P. (1996). *Methods and techniques of adaptive hypermedia.* User Modeling and User Adapted Interaction, 6(2-3):87-129
11. Eklund, J., & Brusilovsky, P. (1998). *The Value of Adaptivity in Hypermedia Learning Environments: A Short Review of Empirical Evidence*. Proceedings of the the Ninth ACM Conference on Hypertext and Hypermedia (pp. 11-17). Pittsburgh.
12. Brusilovsky P. & Pesin, L. (1995). *Visual annotation of links in adaptive hypermedia.* CHI'95 Conference Companion (Denver CO, May 1995), ACM Press, 222-223.

13.Brusilovsky P. & Eklund J. (1998). *A Study of User Model Based Link Annotation in Educational Hypermedia.* Journal of Universal Computer Science. 4(4), 429-448. Springer Science.

14.Lunetta, V. N. (1990). *Cooperative Learning in Science, Mathematics and Computer Problem Solving.* Toward a Scientific Practice of Science Education. M. G. et. al. Hillsdal, NJ, L. Erlbaum Associates**:** 235 – 249.

15.Mazza, R., Dimitrova, V. (2004) *Visualising Student Tracking Data to Support Instructors in Web-Based Distance Education.* 13th International World Wide Web Conference (WWW 2004) - Alternate Educational Track. pp.154-161

Demo a, *Cell growth experiment design*, http://www.fbt.wur.nl/ follow content showcase, Cell Growth Experiment design.

Demo b, *Introduction to cell growth*, http://www.fbt.wur.nl/ follow content showcase, Introduction to cell growth.

Demo c, *Downstream process designer*, http://www.fbt.wur.nl/ follow content showcase, Downstream Processing Design.

Demo d, *Molecular biology cases*, http://www.fbt.wur.nl/ follow content showcase, Molecular Biology site.

Demo e, *Food safety management case*, contact Gerard.Moerland@wur.nl for access information.

Demo f, *Mixing & Membrane cases*, http://www.fbt.wur.nl/ follow content showcase, Mixing and Membranes theory assignments.

Chapter 4

Proteus: A lecturer-friendly adaptive tutoring system

Abstract

Effectively targeting a heterogeneous student population is a common challenge in academic courses. Most traditional learning material targets the "average student", and is sub optimal for students who lack certain prior knowledge, or students who have already attained some of the course objectives. Student-activating learning material supports effective training. Adaptive systems may help to support learning for a heterogeneous group of students. Development of adaptive learning material, however, is usually a complex task not easily done by the average lecturer. An adaptive tutoring system is therefore designed that requires little knowledge and skills from lecturers: Proteus. Proteus provides adaptive navigation on a micro scale based on a set of closed questions. The questions are used both to measure, as well as to stimulate student progress. Entering questions in Proteus requires little effort from lecturers. Proteus is tested and evaluated in a BSc course with ninety-one students and is received very well.

Accepted for publication as Sessink, O.D.T., Beeftink, H.H., Tramper, J., Hartog, R.J.M. *Proteus: A Lecturer-Friendly Adaptive Tutoring System*. Journal of Interactive Learning Research.

Introduction

Many academic courses are attended by a heterogeneous group of students. Students who attend a course might follow different studies, have attended different courses in the past, have attended different versions of the same course, at different universities, or even in different countries in different languages. Students, furthermore, may have poor skills, or may have exceptional skills. A common problem is to set up the education in such a way that all students are effectively and individually addressed, including students who lack some prerequisite knowledge and students who already have attained some of the course objectives.

The usual approach to target such a heterogeneous student population is to start at the "average" student level. For students who have already attained the learning objectives of the initial lectures this will be boring. For students who lack some of the course's prerequisites this will be difficult: they need to identify the gap between their knowledge and the lecture content themselves, and bridge the gap by means of self-study. Finding the right information without guidance, however, is not straightforward, and time consuming for learners [1]. The prior knowledge and skills of the students should be determined both with respect to prerequisites as well as with respect to the learning objectives of the course. Students should then be individually trained in those areas of the prerequisites and learning objectives where training is found to be required.

At Wageningen University, the BSc course 'Introduction to Process Engineering' is attended by a heterogeneous student population. The course is taken by students in Biotechnology and students in Food Science and Technology, who have an otherwise rather different curriculum. About half of the students, furthermore, have their previous education in a different country, with a different educational system, in a different language, and with a very different learning culture.

Adaptive systems may provide the means to address such a heterogeneous student population: "Adaptive systems cater information to the user and may guide the user in the information space to present the most relevant material, taking into account a model of the users goals, interests and preferences" [2]. Development of adaptive learning material, however, is often time consuming and requires specific knowledge and high-level skills from the author [2, 3, 4, 5]. This makes adaptive systems poorly accessible for lecturers outside computer science.

Requirements for a lecturer-friendly adaptive system

Several authors have stressed the deep conceptual gap between authoring systems and authors [6, 7]. A lecturer-friendly learning system should use concepts that match the intuition of the lecturers, and require little effort for effective use [8]. For adaptive systems, usability is mostly determined by the type of adaptive rules, the perceived complexity of the system and the type of information required by the adaptive rules. Perceived complexity is the complexity of the system as far as the lecturer needs to understand it in order to effectively use the system. However, there is a design trade-off: increasing the usability of a tutoring system comes at a cost. Murray [8] listed a design trade-off space for authoring of learning systems, based on breadth, depth, learnability, productivity, fidelity and cost. In lecturer-friendly adaptive system, authoring learnability and productivity are the most important requirements.

An adaptive system in education has a model of the student called a *student model*, and a model of the taught domain called the *domain model*. An adaptive system can, furthermore, assemble content (adaptive presentation), or can select links between content (adaptive navigation) [9]. Adaptive presentation requires specially crafted content. This content is usually built from small blocks, called fragments, that have meta data defining their relations. The adaptive system uses the student model and the meta data to assemble the presented content from fragments. Adaptive presentation systems are for example C-Book [10] and 2L6770 [11]. Adaptive navigation, however, does not require content built from fragments or meta data of fragments. Although many adaptive navigation systems use the electronic book metaphor, adaptive navigation has little requirements for the content. Adaptive navigation systems are for example Interbook [12], and DCG [13].

The type of adaptive rules may contribute to the perceived complexity. In some adaptive systems the lecturer should provide specific instructions for each update of the student model, using a special purpose specification language [14, 15]. This allows very detailed and customizable updates, but requires understanding of this specification language by the lecturer. Other adaptive systems use rules based on statistical theory, such as rules based on Bayesian statistics (for example [16, 17]) that require the lecturer to build a Bayesian network.

Requirements for training

In order to achieve training, there are two requirements. First, during the time spent in the system, the student's mental model should improve for each learning objective that the student did not yet fully attain. Second, after a student has finished a module (a collection of learning objectives and corresponding learning material), the student's mental model should be on a satisfactory level for each learning objective.

To support the learning process effectively, and improve retention and retrieval of knowledge, student-activating learning material should be used [18]. Student activation is for example achieved if students have to give answers in order to proceed through the learning material. If closed questions are used, a computer may effectively interact with the student to support the learning further.

In competency oriented education students will almost always touch upon several learning objectives at the same time. The closed questions are thus likely to activate students along more than one learning-objective dimension.

Design outline

In this article we describe the design of Proteus, an adaptive system that requires little effort and knowledge from the lecturers. Proteus provides adaptive navigation on a micro scale through a set of closed questions. Answers to these closed questions provide explicit knowledge to Proteus. The knowledge is explicit, because it is obtained by direct questioning of the student, as opposed to implicit knowledge that could be obtained while observing the actions of a student while learning [19]. Usage of explicit knowledge to update the student model reduces the uncertainty in the student model [20]. The questions in Proteus activate the student and stimulate the development of the student's mental model. Proteus selects questions such that students will be trained individually in specific areas where training is found to be necessary. Proteus requires only little meta data for each question. The required meta data are defined using concepts that are intuitive for lecturers. Proteus is used and evaluated in Bio-Process Engineering education.

Typical student session

In this section, a typical session of a student interacting with Proteus is described. Proteus is used to implement the module "Introduction to cell growth kinetics", which introduces theory to describe cell growth in bioreactors (see http://www.fbt.wur.nl/, follow "Content Showcase", and choose "Introduction to cell growth"). This module:

1. Introduces each student to all learning objectives and their context
2. Tests each student on the learning objectives
3. Raises awareness of insufficiencies wherever the student's knowledge is lacking
4. Trains each student individually in those areas in which they lack knowledge:
   - The student gets feedback on his/her answer, with an explanation why the answer is correct or incorrect
   - The students will get more questions in these specific areas
   - The student is guided through the relevant parts of the book, the handout or the online library
   - The student is stimulated to study this material

After starting the module "Introduction to cell growth kinetics", the student is informed that he/she has to acquire a certain number of points in order to finish the module. The student is not informed about the adaptive behavior of the module. An overview of the theory is then presented with two short movies about a bioreactor with cells and substrate. The movies show the student what the relevance of the theory is.

When the student has viewed these movies, he is confronted with a choice between a "small step", "medium step" or a "big step". These choices lead to questions in which students may gain a few, medium or many points. The student makes a selection according to his level of confidence, and a page is presented with a question, for example a multiple choice question, its options and a submit button (Figure 1). On the top of the page students see how many points they may maximally obtain from this question, and how many points are still needed to finish the module. On the right hand side there is a navigation area. The navigation area allows the students to switch to a different question, or to watch the introduction movies again. On the bottom of the page there is a link to the theory in the lecture notes or the book that is relevant for this question.

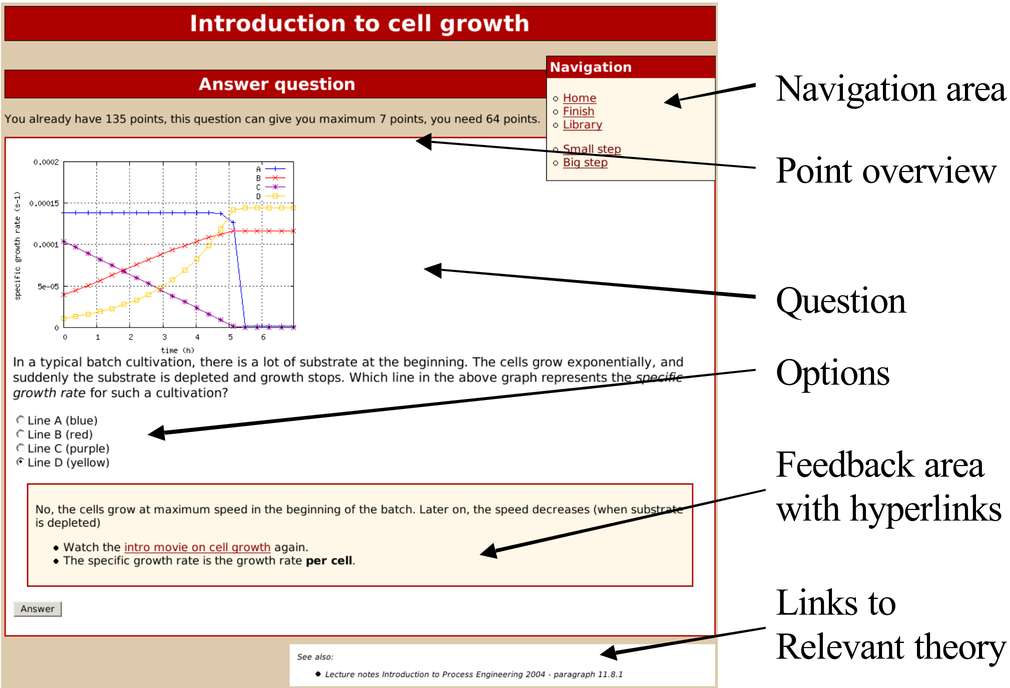

Figure 1: *A typical question in the "Introduction to cell growth" module. The student's answer is wrong, and the figure shows the feedback on this answer.*

After submitting an answer, the student receives feedback whether his answer was correct or not (Figure 1). If not, the student will get feedback why this answer was not correct and one or more appropriate hints. The feedback and hints contain hyperlinks to online available theory. The number of hints is increased if the student needs more tries to answer a question. If the answer is correct, the student receives feedback explaining why this answer was indeed the correct answer, and a link that leads to the choice for the next question: small step, medium step or big step.

After following one of these links, the adaptive system selects the most appropriate question, and this question is presented. The number of points the student still needs to acquire is changed now. If the students answered the previous question immediately correct, they acquired the maximum number of points. If students needed more tries, they acquired fewer points, or even lost points.

Typical lecturer session

When adding questions to Proteus, information about these questions has to be provided by the lecturer. The information has to be provided in terms of "levels" for "learning objectives", further described under 'Student and Domain modeling'. To create an adaptive module, the lecturer first enters the learning objectives for this module. The term "Learning objective" is used in a fine-grained definition: each concept or relation that is to be learned by the students is considered a separate learning objective. The granularity can be compared to a book; each paragraph in a book would then correspond to a separate learning objective. It should be possible, furthermore, to design questions that test and train each learning objective individually. Eventually, after questions have been entered in the system, the precise definition of each learning objective is given by the questions that address that learning objective.

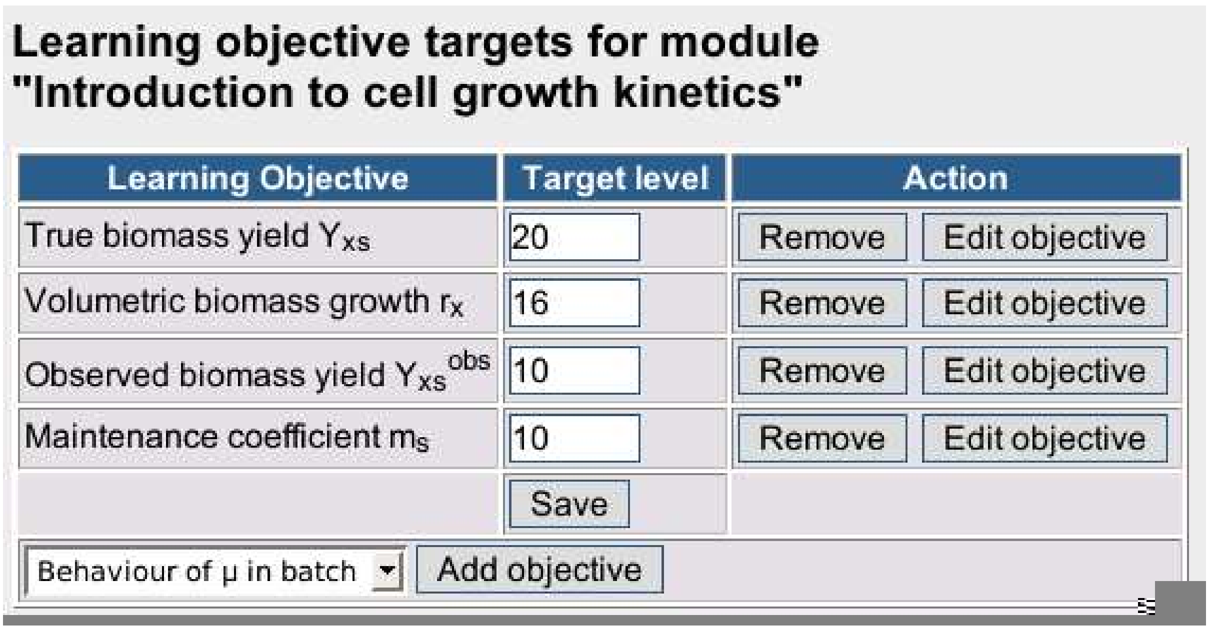

***Figure 2:** Choosing learning objectives and defining "Target levels".*

The lecturer has to assign a "target level" for each of these learning objectives (Figure 2). Students have finished the module when their student level is at least equal to this target level. The target level should correspond with the expected student effort necessary to attain this learning objective. Learning objectives that require more student effort should get a higher target level, and vice versa.

After the learning objectives have been defined in Proteus, questions are added. To add a question, the lecturer selects the learning objectives that are relevant for this question. For each of those learning objectives, a "prerequisite level" may be set for this question (Figure 3).

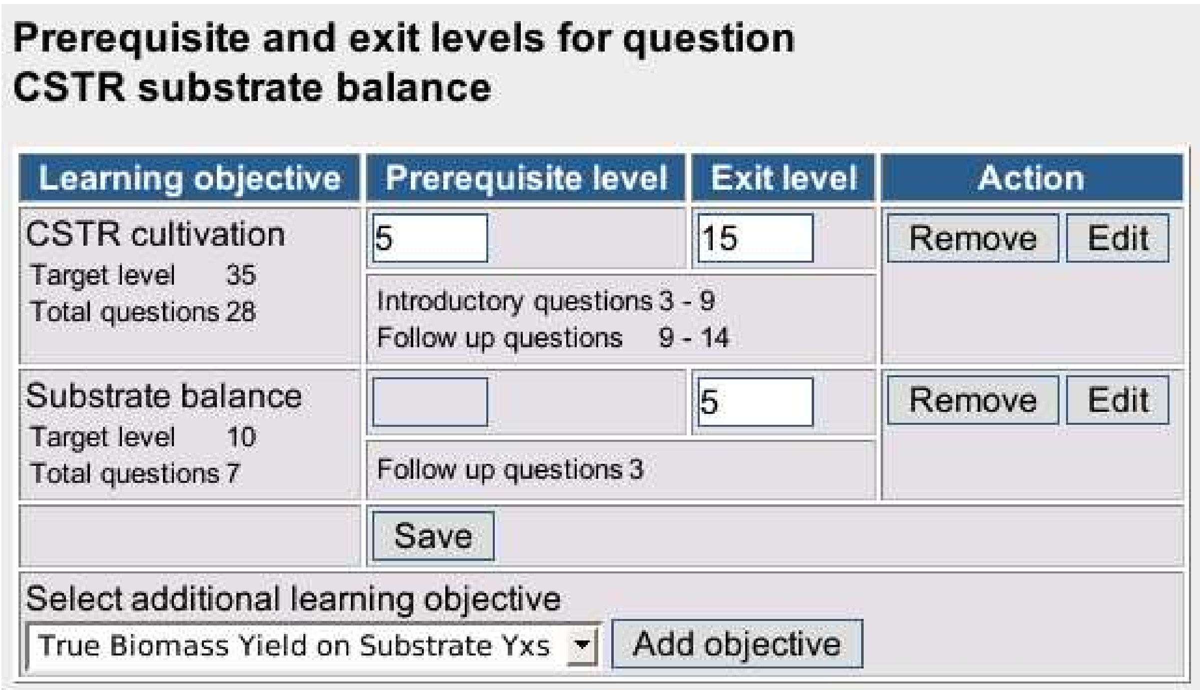

Figure 3: *The prerequisite and exit levels for a question in the lecturer interface. One prerequisite, and two exit levels have been defined.*

The prerequisite level is the minimal level the student should have for this learning objective before this question may be presented to this student. The question, furthermore, should have "exit levels" for at least one learning objective. The exit level defines which level students may reach by answering this question correctly. An exit level higher than the target level of a learning objective thus means that a student may finish this learning objective by answering the question correctly.

After the lecturer has entered prerequisite levels and exit levels, Proteus shows information how this question is placed relative to other questions. The interface shows the number of "introductory" questions and the number of "follow up" questions (Figure 3). The introductory questions are introductory relative to the present question because they have an exit level that at least equals the prerequisite level of this question. At least one of the introductory questions has to be answered by a student before the present question can be answered. Follow up questions are questions that the student may continue with after the present question is answered, because they have a prerequisite level between the prerequisite and the exit level of the present question. The number of introductory and follow up questions give an indication how much choice Proteus has to select the optimal question for a student. Numbers below three are shown in red to warn the lecturer there is little choice.

Student and domain modeling

This section describes the design of the models and rules that form the adaptive system. There is no need for lecturers who want to use Proteus to know the details described in this section. As described in the previous section, Proteus uses learning objectives for both the student model and the domain model. A learning objective is an everyday concept for lecturers. Each learning objective o in a module has a target level T_o defined (Figure 4a). The target level is the level the student should obtain to finish the module, and should correspond with the expected student effort necessary to attain this learning objective.

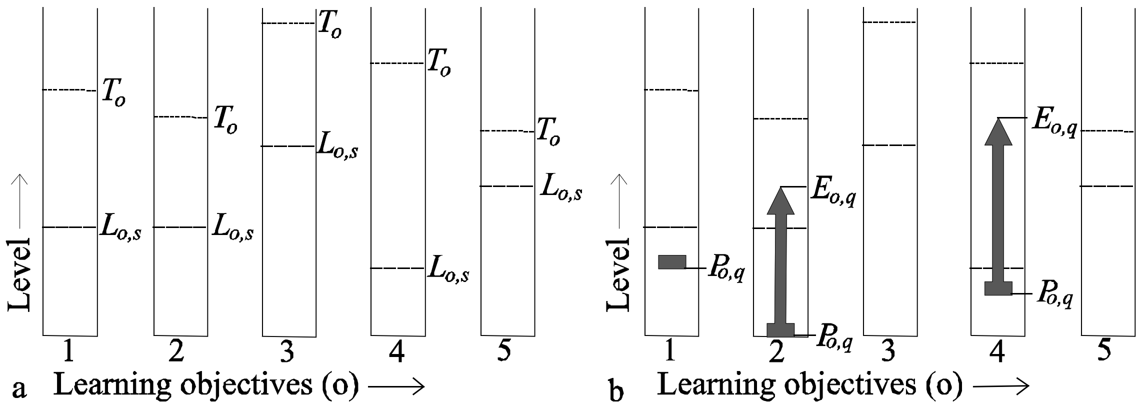

Figure 4: *A five dimensional knowledge space. In figure a each learning objective has a target level T_o defined. The current student level $L_{o,s}$ for one student is shown for each objective. In figure b a question is shown. This question has a prerequisite level $P_{o,q}$ defined for learning objective 1, 2 and 4, and has an exit level $E_{o,q}$ defined for learning objective 2 and 4.*

Together, the learning objectives for the module form an *n*-dimensional knowledge space, where *n* is the number of learning objectives. Each student *s* has a level for each of these learning objectives *o*: $L_{o,s}$ (Figure 4a). In order to finish the module, a student has to cover the complete knowledge space, and thus attain the target level T_o for each learning objective. A model in which the student model evolves to the domain model is called an “overlay model” [19]. A student raises his level for a learning objective by answering questions about that learning objective.

Each prerequisite level $P_{o,q}$ that is defined for a question defines which level the student minimally needs before this question can be answered (Figure 4b). The student should meet all prerequisite levels defined for the question. For each learning objective that is addressed by a question, the exit level $E_{o,q}$ defines to which level the student level may maximally be increased if the question is answered correctly (Figure 4b). A question can thus be depicted as a vector in the knowledge space, with a minimal starting point defined by the prerequisite levels, and a maximal end point defined by the exit levels.

When a student answers a question correctly, he may increase one or more of his student levels, and thus extend his covered knowledge space (Figure 5). When one or more of his student levels are increased, the student in general will have access to questions that previously had a too high prerequisite level. These questions may be answered in their turn to further extend the covered knowledge space. Students keep answering questions until they have covered the complete knowledge space (i.e. have attained all target levels).

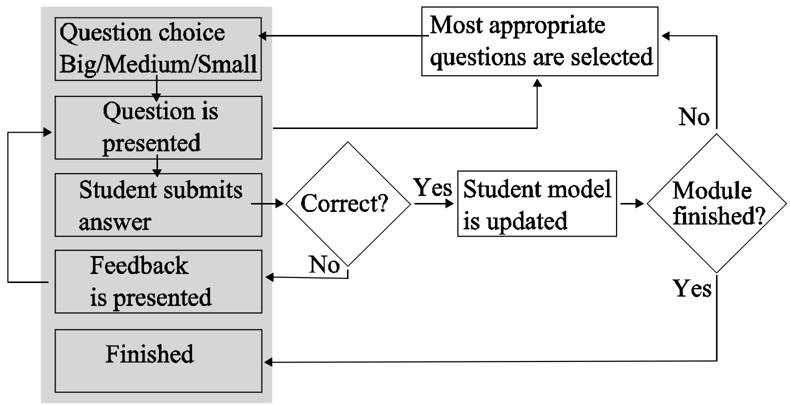

Figure 5: *The flow of actions in Proteus The gray area involves student interaction.*

Adaptive rules: Updating the student model, and selecting questions

To describe the adaptive rules, some definitions are used as outlined in Table 1.

Table 1: *Symbols and their identifiers, o for learning objective, s for student and q for question.*

Name	*Symbol*	*Indices*			*Comment*
		o	s	q	
Student level	L	#	#		updated by system after correct answer
Number of tries	N		#	#	set after student *s* answered question *q* correctly
Prerequisite level	P	#		#	defined by lecturer, default zero
Exit level	E	#		#	defined by lecturer, default zero
Target level	T	#			defined by lecturer

Updating the student model

After student *s* answered question *q* correctly in $N_{s,q}$ tries, the student model is updated. The student level $L_{o,s}$ for student s is updated for those learning objective o where question q has an exit level $E_{o,q}$ defined. The new student level $L_{o,s,post}$ is calculated with the *update function*:

$$L_{o,s,post} = \max\left(L_{o,s,prior}, E_{o,q}\right) - f1_{o,q,s} \cdot r_{o,q}$$

in which $L_{o,s,prior}$ is the student level prior to the update, and $r_{o,q}$ is the range defined by:

$$r_{o,q} = E_{o,q} - P_{o,q}$$

and in which $f1_{o,q,s}$ is a factor between 0 (1 try) and 1 (∞ tries) defined by:

$$f1_{o,q,s} = \frac{N_{s,q} - 1}{N_{s,q} + f2_{o,q,s}}$$

in which $f2_{o,q,s}$ is the confidence support term between 0 and $r_{o,q}$, defined by:

$$f2_{o,q,s} = \frac{\min\left(E_{o,q}, L_{o,s,prior}\right) - P_{o,q}}{\max\left(1, E_{o,q} - L_{o,s,prior}\right)}$$

One additional definition is used to describe the update function. For a given student s and question q, the training potential δ is defined as:

$$\delta_{s,q} = \sum_{o} \max\left(0, E_{o,q} - L_{o,s}\right)$$

which translates to “how much student s may maximally gain by answering question q”.

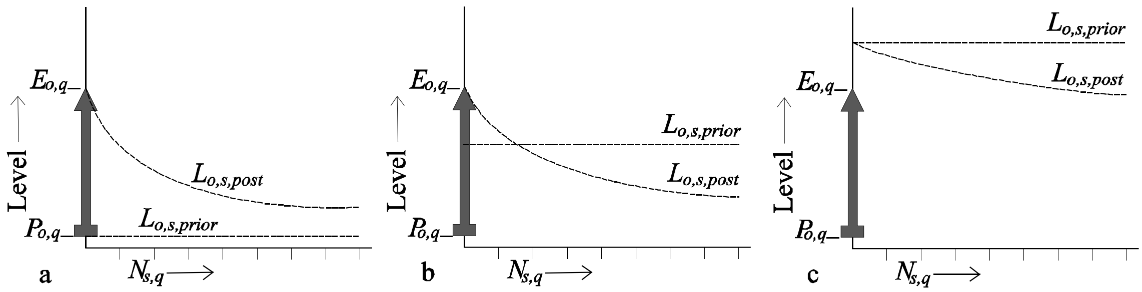

Figure 6: *The behavior of the update function that calculates the new student level $L_{o,s,post}$ as function of the number of tries $N_{s,q}$ for three situations. In a, the student level $L_{o,s,pre}$ equals the question prerequisite level $P_{o,q}$. In b, the student level $L_{o,s,pre}$ is close to the question exit level $E_{o,q}$. In c, the student level $L_{o,s,pre}$ is higher then the question exit level $E_{o,q}$. The left part of each graph is most relevant because questions are usually answered in 1 to 3 tries.*

The behavior of the update function can be seen in Figure 6. The function has five important characteristics.

First, the difference between the minimum and the maximum level that the update function may return equals $r_{o,q}$. If $r_{o,q}$ is small, a student can thus neither gain nor loose many points. This is relevant if a question has exit levels for more than one learning objective, and one of these the exit levels is only slightly higher then the corresponding prerequisite level. For that learning objective, $r_{o,q}$ is small, and thus the new student level will be close to the current student level.

Second, because factor *f1* is a reciprocal function in the number of tries, the new student level $L_{o,s,post}$ is mostly affected if the question was answered immediately correct or in multiple tries (Figure 6). Whether the question was, for example, answered in two, three or more tries has a much smaller effect on the new student level.

Third, the scale on which all levels are defined does not affect the adaptive behavior. The function is proportional to the defined levels. The relative effect of the functions is the same if, for example, all levels are defined on a scale 0-10 or on a scale 0-1000. The lecturer is thus free to choose the scale, as long as all levels are chosen on the same scale.

Fourth, the decision whether or not to split up a learning objective in two separate learning objectives does not affect the adaptive behavior. For questions with an equal training potential δ, the sum of the changes in student levels will be equal. For example, two questions with a training potential δ of 15, one that addresses two learning objectives with exit levels 10 and 5 and no prerequisites, and one that addresses a single learning objective with exit level 15 and no prerequisites, will have the same increase in the sum of the student levels for the same number of tries. The lecturer is thus free to split up learning objectives with multiple components into separate learning objectives.

Last, to stimulate student confidence, the effect of the number of tries on the new student level is decreased if the student's prior level is closer to the exit level, because the student confidence term *f2* is higher for a higher prior level. When answering the same question in the same number of tries a student with a higher prior level will thus get a higher student level than a student with a lower prior level (Figure 6a and 6b). To get their student level up to the exit level, however, they need to answer the question immediately correctly. If the student level was already higher then the exit level, furthermore, the student level will also not drop too far down if the question is answered in multiple tries (Figure 6c). If, for example, a student would answer four subsequent questions that each have an exit level of 10 for the same learning objective and no prerequisites, and if he would finish each question in two tries, his student level for that learning objective will first increase from 0 to 5, at the second question from 5 to 7, at the third question from 7 to 8, and at the fourth question his level will remain at 8. To get to the exit level of 10 the student needs to answer one of these questions immediately correct.

Selecting questions

Some additional definitions are used to describe the question selection rules. For a given student *s* and learning objective *o*, the relative training gap φ is defined as:

$$\varphi_{o,s}=\frac{T_o-L_{o,s}}{T_o} \text{ if } L_{o,s} \leq T_o$$

which translates to “relatively how much of learning objective *o* still has to be obtained by student *s*“. If $\varphi_{o,s}$=1, student *s* has no points on learning objective *o*, if $\varphi_{o,s}$=0, student *s* has finished learning objective *o*.

A question *q* is furthermore said to address learning objective *o* if $E_{o,q}$ is defined. A question may address multiple learning objectives. The main learning objective *mo* for question *q* is therefore defined as the learning objective *o* with the largest range $r_{o,q}$.

A student uses the questions to extend his covered knowledge space until he has attained the target level for all learning objectives. This defines which questions are appropriate at any given moment. For a student *s*, all questions *q* are eligible for which:

1. for all learning objectives *o:* $L_{o,s} \geq P_{o,q}$
2. for at least one learning objective *o:* $L_{o,s} < E_{o,q}$

The first requirement means that the student should meet all prerequisite levels defined for that question. The second requirement means that at least one exit level should be higher than the student level, it should thus be possible for the student to gain a few points. It is possible, furthermore, that a second exit level for a different learning objective exists that is lower than the corresponding student level.

The set of questions that match the above-described criteria for each student is often quite large. A number of additional selection filters is therefore defined, and applied in the listed order. Each filter returns a subset of the previous set. If a filter returns an empty subset, the filter is not executed.

1. The last question q that was answered by student s is removed from the set.
2. All questions that have already been answered by student s are removed from the set. Only new questions remain.
3. All questions that address the same learning objectives as the most recently answered question of student s are removed from the set.
4. For each question q the relative training gap $\varphi_{o,s}$ for its main learning objective mo is calculated. All question for which $\varphi_{mo,s}$ is smaller then 50% of the maximum found $\varphi_{mo,q}$ are removed from the set. Only questions that have a main learning objective that is relatively far from finished remain.
5. All questions that are the last question of any student are removed from the set. Only questions that are not the last question of any student remain.
6. If multiple questions remain in the selection, multiple choices may be presented to the student. The question with the largest training potential δ is presented as "big step". The question with the lowest training potential δ is presented as "small step". If a third question is available that has a training potential δ within 25% of the middle between the largest and the lowest training potential δ it is presented as "medium step".

Filters 1 – 3 provide variation in subsequent questions for the student. Filters 1 and 2 address questions that have been answered by the student, and filter 3 addresses learning objectives that have been dealt with recently by the student. Filter 2 seems to obsolete filter 1, but filter 2 might return an empty set, which means it is not executed. Filter 1 is thus relevant if filter 2 returns an empty set.

Filter 4 provides variation in the long term. Filter 4 avoids the situation that a student ends up with a number of small learning objectives that are finished, and a single large learning objective that is far from finished, causing many subsequent questions that address a single learning objective.

Filter 5 provides variation in questions among multiple students. If a group of students is working on the module in the same room, and they all get the same question, there is a chance that only few students answer the question themselves, while other students copy that answer. If all students get different questions, students cannot copy the answer. Also filter 5 seems to obsolete filter 1. However, filter 1 is relevant if filter 5 returns an empty set.

Filter 6 stimulates student confidence. Students are given some control over the question selection. This choice gives students who have little confidence the possibility to take small steps, and gives students who have more confidence the possibility to proceed with large steps. Initially the labels "easy", "medium" and "difficult" were used, after the evaluation these have been changed.

The result of these filters is that different students proceed through the total set of questions with very different paths (Figure 7).

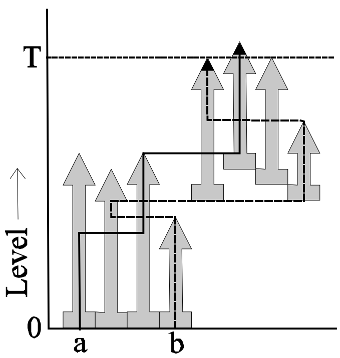

Figure 7: *Prerequisite and exit levels of eight questions, projected on a single learning objective. Students a and b extend their knowledge space with different questions.*

Question development

Questions are the means for both measuring the progress of the students to update the student model as well as stimulating the progress of the students. The student model is only a satisfactory representation of the students' mental model if the questions have measured the progress for all learning objectives well. Careful design of these questions is thus important.

Measuring student progress

The ideal question to measure student progress would be answered immediately correct by all students who have attained the exit levels defined for this question. In reality questions are never ideal, but there should be a good match between the question content and the levels defined for the question. If the prerequisites are met, the number of tries should only depend on the learning objectives that have exit levels defined. The number of tries should not be affected by other learning objective. The question should, furthermore, not have any prerequisite learning objectives other than defined by the prerequisite levels. Lack of modularity in the design of questions is thus a risk.

Although these requirements may sound very obvious, in reality it occurs that questions have implicit learning objectives that have been overlooked by the author. For example, students could *en masse* fail a question that targets the learning objective “exponential growth”, because the author assumed students know how to solve a differential equation. If it is known that many students have problems solving differential equations, either there should be a learning objective “solving differential equations” in Proteus, which should be the prerequisite for the question about “exponential growth”, or the question should be redesigned such that it can be answered without knowledge of differential equations.

Stimulating student progress

There are several reasons why the system stimulates student progress, ranging from the adaptive behavior of Proteus to the content of individual questions.

Closed questions can be activating learning material; in order to proceed through the learning material, students actively have to answer the questions. Elaboration or practice is found to improve retention of knowledge [18].

An often mentioned drawback of closed questions is that they may tempt students to guess the answer. However, Proteus discourages wild guesses. Initially students may gain a few points when answering a question with multiple tries. If the current student level is close to the exit level of a question, however, the student level will not change, or even decrease when answering that question with multiple tries. Students will quickly find that they will not acquire the required points if they need multiple tries for subsequent questions. This will stimulate students to invest time in learning the theory and in careful reasoning instead of answering the question by trial and error.

The questions themselves are annotated with links to relevant paragraphs in the book and the lecture notes, as well as hyperlinks to online theory. This allows students to access the theory with very little effort. The theory is thus directly accessible for the student. The application of just-in-time information presentation in the adaptive system reduces extraneous cognitive load [21].

All questions, furthermore, provide feedback on the students' answer, in order to assist the student in restructuring his knowledge [22]. Lack of modularity in the question design may reduce the effectiveness of the feedback. The feedback is designed following the recommendations of Narciss and Huth [23]. If students answer the question correctly, feedback is provided why this answer is indeed correct, providing both verification and elaboration. If students answer the question incorrectly, specific feedback is provided that explains why this particular answer is incorrect, again providing verification and elaboration, and hints are provided how to approach the question, providing elaboration. The hints are adjusted with the number of tries a student needs to answer the question. The more wrong answers, the more hints are given, and the more specific and concrete the hints will be. To avoid students losing confidence, most questions will, in the end, show the correct answer. Some lecturers might worry this can be abused by students, however, if a student needs a large number of tries, Proteus will not increase his student levels up to the exit levels defined for the question.

Usage and evaluation

A module "Introduction to cell growth kinetics" was developed in Proteus. This module corresponds to eight hours of study load for students, which includes for example reading the lecture notes, and exam preparation. Thirteen learning objectives were defined in Proteus. The learning objectives had target levels ranging from six for the smallest and easiest learning objective until thirty-five for the largest and most difficult learning objective. Proteus was filled with ninety-six closed questions with feedback. Once created, Proteus needs prerequisite and exit levels for each question. Adding all ninety-six questions to Proteus was completed within four hours.

The module was used twice in the course “Introduction to Process Engineering”. The first time this course was taken by 91 students, of which 63% in Biotechnology and 37% in Food Science and Technology in the second year of the BSc program. Of these students, 40% had their prior education in The Netherlands, and 60% in other countries; predominantly in China. The second time the course was taken by 52 students with a comparable distribution.

The module was introduced in a computer lab, where students could start the module. Two assistants were available to solve technical or organizational problems. Apart from creating a few accounts, no problems were reported. Students were asked to finish the module without assistance within five days, after which the lectures about cell growth kinetics would start. Students were told the module was finished after they acquired all points, and that this would cost them three to six hours. Students were not told that the number of tries affected the acquisition of points. Students were not introduced to the adaptive behavior either. Students were asked to complete a questionnaire immediately after finishing the module. Proteus was, furthermore, configured to track all student actions, student answers, and changes to the student model.

Analysis and discussion

Observations and tracking data

The assistants observed that most students started the module without using the lecture notes. After the first or second question where students needed multiple tries, students realized that this precluded a maximum score, and started studying the lecture notes if they were unsure whether their answer was correct.

Analysis of tracking data shows that different students proceeded in very different ways through the module. Students needed on average 51±14 questions to finish the module, with a standard deviation of 14 questions. The quickest student needed 2h for 17 questions, the slowest 10h for 82 questions. It took students on average 3.6±1.6 hours to finish the module. Assistants observed in 2004 that a group of 20 students in Biotechnology with a Dutch prior education proceeded very quickly through the module. The mean number of questions this group of students needed was 46. T-test analysis of the data, however, shows that this mean is not significantly different from the mean number of questions the rest of the 2004 students needed ($P=0.22$). The expected heterogeneity between different groups is thus smaller then the individual variation between students.

During working lectures following this module, the lecturer observed that students had fewer problems with introductory theory as compared to previous years without the module. All students now had a good understanding of the basic theory.

Questionnaire

The questionnaire was answered by 82 students in 2004 and 50 students in 2005 and started with an open question that asked for the students' general opinion. Students were very positive about the module. Some examples are shown:

"I think it is a very good way to digest the matter. Even I am on schedule now!"

"I think the program is useful, this way you work actively on the subject. I also think it is effective to have negative scores, because if you are not so eager anymore you would otherwise speculate (which is now more difficult). A good approach thus!"

"We are addressed in a negative way. It is unpleasant that we only see how many points we still need. It is probably useful, but not pleasant."

"I reckon it to be quite a good method to get to understand the theory, mainly because it's an active learning process, opposed to just reading lecture notes, which I find quite passive."

"I think it's forcing to read the reader and to understand it. That's very positive."

Further analysis of the answers on the open question can be found in Table 2.

Table 2: *Analysis of the open question answers, what percentage of the answers mentioned a certain category, and if it was a positive (+) or negative (-) remark.*

Category	***2004 (n=82)***			***2005 (n=50)***		
	mentioned	+	-	***mentioned***	+	-
Useful	51	51	0	54	52	2
Understanding of learning objectives	31	31	0	13	13	0
Total module, nothing further specified	25	23	2	10	10	0
Active learning	20	20	0	8	8	0
Question difficulty	15	3	12	4	2	2
Duration of the module	14	2	12	15	0	15
Concept of points	10	5	5	6	6	0
Points gaining mechanism	7	0	7	6	0	6
Accessibility of the learning material	3	3	0	4	4	0
Guidance and structuring	3	3	0	10	10	0
Order of questions	3	0	3	2	0	2

Overall the open question answers give a very positive evaluation of the adaptive module from the perspective of the students in this heterogeneous group. This is confirmed by closed questions, as is described below. The answers, however, show various possibilities for improvement.

In 2004 fifteen percent of the students referred to the difficulty of the questions, of which twelve percent was a negative comment. Several students complained the questions were too difficult. Questions should only be presented when students have acquired the prerequisites. So either the adaptive behavior did not work for these students, or the prerequisite levels for these questions were not set well, or the prior knowledge of these students did not even meet the initial requirements for the module. Students mostly complained that the "easy", or "medium" option in the module sometimes presented a question that was very difficult in their perception. Because these labels correspond to the training potential β, the same question that is initially presented as "difficult" might at a later stage become "medium" or even "easy". The labels have therefore been changed to "big step", "medium step" and “small step” after the evaluation in 2004.

About twelve percent of the students in 2004 and fifteen percent of the students in 2005 had problems with the duration of the module. Analysis of the tracking data, however, revealed that many of these students tried to finish the module in one session without a break. Most of these students needed four to five hours to finish the module, which is not very long for the module, but is very long for a single session without breaks.

Ten percent of the students in 2004 referred to the concept of gaining points, of which half had a positive remark, and the other half a negative remark. Several students with a negative remark reported it was demotivating them to see only the points they lacked, and not the points they already attained. After the evaluation in 2004 the presentation of the points has been changed to show both the number of points students already have attained as well as the number of points students still need.

Seven percent of the students in 2004 and six percent of the students in 2005 have a negative remark on the point gain mechanism. Students refer it is unclear to them how the rules are working. An error was later found in some of the questions that probably contributed to some of the 2004 remarks. And, although it is not the intention that students should know how the rules are working, introducing the students to the adaptive behavior and the update rules might increase their confidence with the system.

The open question in the questionnaire was followed by some questions on a 1-5 scale, shown in Table 3.

Table 3: *Average answers on the multiple-choice questions and their standard deviation. Questions about understanding were rated 1=poor ... 5=good, questions about statements were rated 1=disagree ... 5=agree, exceptions are shown.*

Question	***2004 (n=82)***	***2005 (n=50)***
The questions were (1=too easy ... 5=too hard)	3.3±0.7	3.1±0.6
My understanding of specific growth rate is	3.6±0.8	3.6±0.7
My understanding of Monod kinetics is	3.4±0.7	3.5±0.8
My understanding of biomass yield on substrate is	3.3±0.8	3.4±1.0
My understanding of maintenance is	3.4±0.8	3.5±0.9
My understanding of batch cultivation is	3.6±0.8	3.9±0.8
My understanding of CSTR cultivation is	3.5±0.8	3.8±0.8
My understanding of substrate/biomass balances is	3.6±0.7	3.6±1.0
This module is fun	3.3±1.0	2.8±1.2
This module is challenging	3.8±0.9	3.9±0.8
This module is motivating	3.6±0.9	3.8±1.0
This module is useful	4.1±0.8	4.4±0.8

The answers on the multiple-choice questions show that students have a positive view on the module. Especially the “useful” rating is high; in 2004 thirty-nine percent of the students gave the maximum possible score, in 2005 fifty-eight percent gave the maximum score. At Wageningen University all courses are evaluated with a standard questionnaire, including the question “This course is useful”. In the last year, 802 courses were evaluated, of which 22% received a rating of 4.1 or higher, and only 5% received a rating of 4.4 or higher.

The answers regarding the difficulty of the questions comply with the supposed adaptive working. Because Proteus should present only questions that are relevant for a student, students should not feel that questions are too easy or too difficult.

The answers on the students’ perception of their own knowledge are all within expectations. None of the topics mentioned has an answer below 3.0 nor an exceptionally high standard deviation.

Future Developments

The closed questions in Proteus currently use a non-standard XML based format, which was developed for quick development of case-based digital learning material. With the release of the IMS Question & Test Interoperability (QTI) Specification version 2.0 [24], the QTI specification can replace our non-standard format without the loss of functionality. This would enable the use of existing sets of QTI questions in Proteus.

Deploying the adaptive rules in a standard learning management system would improve interoperability further. A first approach for adaptive rules using learning standards is described by Cheniti-Belcadhi [25]. At this moment we have not found a way to implement the adaptive rules described in this article conform SCORM 2004. Database functionality for learning objects is probably required to deploy these rules in a learning management system [26].

Proteus needs to measure the student progress. Only learning objectives that allow questions to be designed such that the number of tries is a good indicator of the student progress can thus be used in the adaptive system. It is less effective for “design”, or “collaborative” learning objectives, or “whole task” assignments [27].

Conclusion

Design and development of Proteus, a lecturer-friendly adaptive system for tutoring, was successful. Proteus requires little knowledge and skills from the lecturers, because the adaptive rules are designed to use lecturers’ everyday concepts. The amount of information required by the adaptive rules, furthermore, is little. Where possible, the adaptive rules try to apply some theories of learning and instruction. Proteus provides adaptive navigation through a set of closed questions. Adding questions to Proteus is very straightforward and requires little effort. The questions are used to measure student progress, as well as to stimulate student progress. The system discourages wild guesses, and provides feedback to stimulate students to study the theory.

Addressing a heterogeneous student population effectively with Proteus is found to be very successful. A module developed in Proteus is used twice by in total 143 students in a BSc course, and is received very well. Different students followed different tracks through the questions. The quickest students needed 17 questions and two hours to finish the module. Students who required much training needed up to five times more time and up to four times more questions to finish the module. After finishing the module, students answered a questionnaire, in which students gave very positive remarks on the module, and rated the module usefulness on average 4.4 on a 1-5 scale.

References

1. Pivec, M., Baumann, K. (2004). *The Role of Adaptation and Personalisation in Classroom-Based Learning and in e-Learning.* Journal of Universal Computer Science, 10(1), 73-89.
2. Brusilovsky, P., Eklund, J., Schwarz, E.W. (1998). *Web-based Education for All: A Tool for Development Adaptive Courseware.* Computer Networks and ISDN Systems 30 (1-7), 291-300.
3. De Bra, P., Ruiter, J.P. (2001) *AHA! Adaptive Hypermedia for All.* in *Proceedings of the AACE WebNet Conference*, Orlando, FL., AACE 262-268.
4. Dagger, D., Wade, V., & Conlan, O. (2005). *Personalisation for All: Making Adaptive Course Composition Easy.* Educational Technology & Society, 8 (3), 9-25.
5. Armani, J. (2005). *VIDET: a Visual Authoring Tool for Adaptive Websites Tailored to Non-Programmer Teachers.* Educational Technology & Society, 8(3), 36-52.
6. Mizoguchi, R. and Bourdeau, J. (2000). *Using ontological engineering to overcome common AI-ED problems.* International Journal on Artificial Intelligence in Education, 11, 1-12.
7. Aroyo L., Mizoguchi, R. (2004). *Towards Evolutional Authoring Support Systems.* Journal of Interactive Learning Research, 15(4), 365-387
8. Murray, T. (1999). *Authoring Intelligent Tutoring Systems: An Analysis of the State of the Art.* International Journal of Artificial Intelligence in Education, 10, 98-129.
9. Brusilovsky, P. (2001). *Adaptive Hypermedia.* User Modelling and User-Adapted Interaction. 11 (11), 87-110.
10. Kay, J., Kummerfeld, R.J. (1994). *An individualised course for the C programming language.* In: Proceedings of Second International WWW Conference, Chicago, IL, 17-20 October
11. Calvi, L., De Bra, P. (1997) *Using dynamic hypertext to create multi-purpose textbooks.* In: T. Müldner and T. C. Reeves (Eds.) *Proceedings of ED-MEDIA - World Conference on Educational Multimedia*, Calgary, Canada, AACE, 130-135.

12.Brusilovsky, P., Eklund, J., and Schwarz, E. (1997). *Adaptive Navigation Support in Educational Hypermedia on the World Wide Web.* In *Proceedings of INTERACT97, The 6th IFIP World Conference on Human-Computer Interaction*, Sydney, Australia, New York: Chapman & Hall, 278-285.

13.Vassileva, J. (1997) *Dynamic Course Generation on the WWW.* In: B. d. Boulay and R. Mizoguchi (eds.) *Proceedings of AI-ED'97, 8th World Conference on Artificial Intelligence in Education.* Kobe, Japan, August 1997. Amsterdam: IOS, pp. 498-505.

14.De Bra, P., Aerts, A., Smits, D., Stash, N. (2002). *AHA! Version 2.0, More Adaptation Flexibility for Authors.* In M. Driscoll, & T. C. Reeves (Eds.), *Proceedings of World Conference on E-Learning*, E-Learn 2002, Montreal, Canada, AACE. 240-246.

15.Esposito, F., Licchelli, O., Semeraro, G. (2004). *Discovering Student Models in e-learning Systems.* Journal of Universal Computer Science, 10(1), 47-57

16.Henze, N., Nejdl, W. (1999). *Bayesian Modeling for Adaptive Hypermedia Systems.* In ABIS99, 7. GI-Workshop Adaptivität und Benutzermodellierung in interaktiven Softwaresystemen, Magdeburg, September 1999.

17.Conati, C., Gertner, A. and VanLehn, K. (2002). *Using Bayesian Networks to Manage Uncertainty in Student Modeling.* User Modeling and User-Adaptive Interaction 12(4), 371–417.

18.Anderson, J.R. (1995). *Learning and memory. An integrated approach.*, New York: John Wiley & Sons, Inc.

19.Brusilovsky, P. (1994). *The construction and application of student models in intelligent tutoring systems.* Computer and System Sciences International, 32(1), 70-89.

20.Beaumont I., Brusilovsky, P. (1995). *Adaptive hypermedia: From ideas to real systems.* In Maurer H. (Eds.) *Educational Multimedia and Hypermedia.* 1995. Proceedings of ED-MEDIA95, 93-98.

21.Kester, L., Kirschner, P.A., van Merriënboer, J.J.G., Baumer, A. (2001). *Just-in-time information presentation and the acquistion of comples cognitive skills.* Computers in Human Behaviour, 17, 373-391

22.Kulhavy, R., Stock, W. (1989). *Feedback in written Instruction: The place of response certitude.* Educational Psychology Review 1(4), 279-308

23.Narciss, S., Huth, K. (2002). *How to design informative tutoring feedback for multimedia learning.* In H. M. Niegemann, R. Brünken & D. Leutner (Eds.) *Instructional design for* *multimedia learning.* Münster, Germany: Waxmann 181-195

24. IMS Global Learning Consortium (2004). *Question and Test Interoperability 2.0*. Retrieved Sep 23. 2005 from http://www.imsglobal.org/

25. Cheniti-Belcadhi, L., Braham, R., Henze, N., Nejdl, W. (2004). *A Generic Framework for Assessment in Adaptive Educational Hypermedia.* Proceedings of the IADIS WWW / Internet 2004 Conference, October 2004, Madrid, Spain.

26. Sessink, O.D.T., Beeftink, H.H., Hartog, R.J.M. (2005). *Database functionality for learning objects.* Technology, Instruction, Cognition and Learning 2, 337 - 357.

27. Van Merriënboer, J. J. G. (1997). *Training complex cognitive skills: A four-component instructional design model for technical training.* Englewood Cliffs, NJ: Educational Technology Publications.

Chapter 5

Virtual parameter-estimation experiments in bioprocess-engineering education

Abstract

Cell growth kinetics and reactor concepts constitute essential knowledge for Bioprocess-Engineering students. Traditional learning of these concepts is supported by lectures, tutorials and practicals; ICT offers opportunities for improvement. A virtual-experiment environment was developed that supports both model-related and experimenting-related learning objectives. Students have to design experiments to estimate model parameters: students choose initial conditions and 'measure' output variables. The results contain experimental error, which is an important constraint for experiment design. Students learn from these results and use the new knowledge to re-design their experiment. Within a couple of hours students design and run many experiments that would take weeks in reality. Usage was evaluated in two courses, with questionnaires and in the final exam. These evaluations show that the learning objectives are well supported. The faculty involved in the two courses are convinced that the experiment environment supports essential learning objectives well.

Extended version of Sessink, O.D.T., Beeftink, H.H., Hartog, R.J.M., Tramper, J (2006). *Virtual Parameter-Estimation Experiments in Bioprocess-Engineering Education.* Bioprocess Biosystem Engineering, doi 10.1007/s00449-005-0042-z

Introduction

Cell growth kinetics and reactor concepts constitute essential knowledge for Bioprocess-Engineering students. In two courses at Wageningen University, cell growth kinetics are described with Monod kinetics, while the consumption of substrate is described with the Pirt linear growth law.

Traditionally, students learn about theoretical models in lectures and tutorials, and about practical issues in a wet-lab practical. In the lectures and tutorials, the students spend a considerable amount of time on the calculation of solutions to exercises. By solving exercises, students are supposed to discover various characteristics of the models. In wet lab practicals, students learn many practical aspects of cell culturing and apply the theory to a practical situation. Because cell culturing includes many new procedures for students (e.g. reactor building, medium preparation, sterile sampling, cell counting, etc.), application of theory receives often little attention.

Among staff there was general agreement on two opportunities for improvement of our traditional learning support. First, the Bsc-graduates' understanding of the integration of kinetics and reactor concepts and how theoretical models relate to reality could be improved. This opportunity is outlined under "Model-related learning objectives". Second, BSc-graduates' understanding how experiments relate to modeling could be improved as well. This opportunity is outlined under "Experimenting-related learning objectives".

This article discusses a virtual-experiment environment that was developed to exploit these opportunities. The virtual-experiment environment was developed for two courses: "Introduction to Process Engineering" early in the second year of the BSc of both Food Science and Technology students and Biotechnology students, and "Bioprocess-Engineering" late in the second year of the BSc of students in Biotechnology.

The virtual-experiment environment should comply with general accepted learning and instruction principles. First, retention and retrieval of knowledge improve when students actively elaborate on the knowledge [1]. The learning process is best supported if the elaboration with the experiment environment is well structured [2]. Second the cognitive processing capacity of the student is a limiting factor in learning. By minimizing the cognitive load there is more capacity left for learning [3]. In computer simulations, an important factor to minimize extraneous cognitive load is the user interface [4]. Usability guidelines (for example [5]) can help to minimize the cognitive load in the user interface. Last, student motivation is a key requirement for learning. The ARCS model on motivation [6] provides four components that should be included: Attention (stimulate curiosity and interest), Relevance (connect to the students goals and interests), Confidence (stimulate sense of success), and Satisfaction (stimulate sense of achievement).

In the virtual-experiment environment, students have to estimate some model parameters. The virtual-experiment environment allows the student to design and run multiple experiments. In each experiment the student chooses the reactor setup and the initial conditions. The initial conditions are all variables, such as substrate concentration, flow rate and initial biomass concentration. Students then run the experiment and 'take' samples. The experimental results, biomass concentration and substrate concentration, contain experimental error. From these experimental results, students estimate the parameters. Students learn from the experimental results, supported by feedback, and use the new knowledge to design a new experiment. The cycle of designing and running many experiments executed to estimate the requested model parameters now takes a couple of hours instead of a number of days or even weeks.

Model-related learning objectives

The theory of both courses describes the specific cell growth rate with Monod kinetics:

$$\mu = \mu_{\max} \cdot \frac{C_s}{C_s + K_s} \quad \left(s^{-1}\right) \quad (1)$$

where μ is the specific growth rate (s^{-1}) as function of the substrate concentration C_s (kg m^{-3}), μ_{max} the maximum specific growth rate (s^{-1}) and K_s the Monod constant (kg m^{-3}). Specific substrate consumption rate is described with the Pirt linear growth law:

$$q_s = \frac{\mu}{Y_{xs}} + m_s \quad \left(\mathrm{kg(S)\cdot kg(X)^{-1}\cdot s^{-1}}\right) \quad (2)$$

where q_s is the specific substrate consumption rate (kg(S) kg(X)$^{-1}$ s^{-1}), Y_{xs} the yield of biomass on substrate (kg(X) kg(S)$^{-1}$) and m_s the maintenance coefficient (kg(S) kg(X)$^{-1}$ s^{-1}).

Students apply these models in various reactor configurations, most notably in ideally mixed batch and CSTR configurations. Mass balances for batch (φ=0) and CSTR ($\varphi_{in}=\varphi_{out}$, $C_x=C_{x,out}$, and $C_s=C_{s,out}$) are then:

$$\frac{d\left(V \cdot C_{x}\right)}{dt} = \varphi\left(C_{x,in} - C_{x}\right) + \mu \cdot V \cdot C_{x} \quad \left(\mathrm{kg(X)} \cdot \mathrm{s}^{-1}\right) \quad (3)$$

$$\frac{d\left(V \cdot C_{s}\right)}{dt} = \varphi\left(C_{s,in} - C_{s}\right) - q_{s} \cdot V \cdot C_{x} \quad \left(\mathrm{kg(S)} \cdot \mathrm{s}^{-1}\right) \quad (4)$$

in which C_x is the biomass concentration (kg m^{-3}), V the volume (m^3) and φ the flow rate (m^3 s^{-1}).

The learning objectives supported by the virtual-experiment environment related to this model are:

1. Qualitative understanding of the behavior of biomass and substrate concentrations in batch and CSTR reactors
2. Understanding under which conditions this model is valid
3. Intuitive knowledge of the order of magnitude of parameters and variables for a few common cell types and substrates

With qualitative understanding we mean that students should be able to reason about the system's behavior without knowledge of the actual values. If some parameter or variable is increased or decreased, students should be able to infer if this will lead to an increase, decrease, maximum, minimum, or asymptote of the system's output variables.

Students will spend most of their time constructing and reconstructing their knowledge of the model. To design an experiment to estimate a certain model parameter, the student needs to understand the relation between the parameter and the system behavior, and should know under which conditions the parametric sensitivity is the highest. The students should, furthermore, be capable of reasoning that certain initial conditions will allow measurement of all unknown variables in this region of high parametric sensitivity. There are too many degrees of freedom to successfully design experiments without correct understanding of the behavior and the assumptions underlying the model.

Students will have some idea about an experiment design, which results it will return, and how these results can lead to useful information about the task. Every sample the students take will result in some new information: an increase, a decrease, a constant, a local maximum or a local minimum. Every time the students get this new information, reflective processes will be stimulated [7]. Students will consider if this result was to be expected, and, if not, they will reconsider their idea about the experiment and reconstruct their knowledge. Because the virtual-experiment environment enables a high frequency of these reflective loops, students spend most of the time reconstructing their knowledge of the model. The virtual-experiment environment is thus appropriate to support learning objectives one and two.

In traditional exercises students often get a parameter value and use it in their calculations. In such exercises there is no stimulus to acquire knowledge for the order of magnitude of these parameter values. In the virtual-experiment environment students have to estimate parameters from data with experimental error. When students know the realistic range of the parameter, they can use this information to get an idea if their experiment design and calculation could be correct. This is a stimulus for students to acquire knowledge about the order of magnitude. The virtual-experiment environment is thus also appropriate to support learning objective three.

Experimenting-related learning objectives

The learning objectives supported by the virtual-experiment environment related to experimenting are:

1. Qualitative understanding of the uncertainty in parameter estimates
2. Understanding that there may be large differences between the uncertainty in estimates of different parameters
3. Understanding that uncertainty is affected by the experiment design
4. Knowledge that constraints such as operational limits of analysis equipment, parametric sensitivity, and costs have impact on experiment design
5. Skills to design an experiment that is appropriate to answer a specific research question

Students often assume that models and parameter values are perfect. In BSc courses students are seldom confronted with uncertainty in parameter values. Too often, literature and textbooks do not show the uncertainty of parameter values. Awareness of uncertainty in parameter values is essential in modeling. In the experiment environment, students will experience experimental error as a problem during the estimation of parameters. They will find that different experiments give numerically different results. These results show various fundamentals of parameter estimation, such as the large experimental error when analyzing outside the operational limits of the equipment, and the information richness of the results. Because this will teach students to be critical towards parameter values, the virtual-experiment environment is appropriate to support learning objectives one, two, and three.

Students have, furthermore, little experience with experiment design. In a wet-lab practical, students mostly learn practical skills. Hofstein and Lunetta [8] wrote "to many students, 'a lab' means manipulating equipment but not manipulating ideas". In many practical experiments, lecturers spend most of their time managing the laboratory, and many students follow recipes without a clear understanding of the purpose of each step and without a clear goal [9]. The recipe, furthermore, is seldom questioned (for example [10, 11]). Students expect it to be correct and efficient, and expect the data gathered to be useful for their calculations. In real research, however, experimenting is tightly linked with uncertainty. Researchers should question if their experiment design is correct, if it is efficient, and if the gathered data will be useful for their research question. In the virtual-experiment environment, students are faced with the same uncertainty. Students are confronted with a design problem, and can practice experiment design. Students get feedback on their designs, and accordingly design new, better experiments. Because students design and test many experiments in a short period of time, and get much feedback, the virtual-experiment environment is appropriate to support learning objectives four and five. Proper experiment design is very important. Students should realize that poor design yields poor data. Poor data often means one has to start all over again, wasting much time and money.

Description of the virtual-experiment environment

For an overview of the virtual-experiment environment, visit the online available demo at http://www.fbt.wur.nl/, choose “Content showcase” and choose “Cell growth experiment design”.

Student task

Typically, the experiment environment is used after lectures and tutorials about the model; although online theory is available, the experiment environment itself does not introduce the model theory. Students get the task to estimate one or more parameter(s) for some model. For the model described in this article these are μ_{max}, K_s, Y_{xs}, and m_s. In order to do so, they should design and run a number of experiments. The design choices are:

1. Reactor configuration, batch or CSTR
2. Value of operational variables $C_s(0)$, $C_x(0)$, and, for a CSTR, also $C_{s,in}$, and φ
3. Time and number of samples, as well as sample analysis method

Students will measure C_s(t) and C_x(t), which include experimental error. These experimental data can be used to estimate the parameter values. To allow the environment to be used in multiple courses, the environment does not require a specific method for parameter estimation.

Because in reality, costs are often a constraint in experiment design, students are shown the costs of their experiments. The costs are, however, no constraint in the experiment environment; students have an unlimited budget. The well-defined goal and limited number of choices of this task structures the way students use the experiment environment. A structured task is required for educational simulations to support the learning process in an effective way [2].

The experiment environment may present different tasks to different students. This allows the lecturer to give unique tasks to each student within a course, but also to use the experiment environment in different courses that have different models.

In the course “Introduction to Process Engineering” the goals are to estimate the maximum specific growth rate μ_{max} and Monod constant K_s. Students in this course use the experiment environment in a computer room. To avoid students copying answers from other students, each student will get a different organism, with different target parameter values; but their model will be identical. In the course “Bioprocess-Engineering” the goals are to estimate the biomass yield Y_{xs} and maintenance coefficient m_s. Again each student will get different parameter target values and use the same model. Both tasks are considered highly authentic by staff and students. Authentic tasks might stimulate the transfer of skills to other problem solving situations [12].

Technical implementation

The experiment environment is implemented as Java applet. In order to use it, only a WWW browser with recent Java version is required, which is commonly available. The experiment environment can thus be used from any location with internet connection, at any time.

The Java applet can be controlled from the server, such that the single Java applet can accommodate different assignments. Different assignments might contain different parameter target values, different tasks, and different kinetic models.

The models can be selected from a list of available models; only a limited number of kinetic models is currently available. Extending the experiment environment with more models requires modification of the Java applet.

Experiment designs and results from each student are regularly sent to the server. They can be restored upon restart of the experiment environment, which allows students to stop and continue later on.

To allow students to do their calculations in a familiar environment, the experimental data can be copied and pasted into spreadsheet programs from all common office suites, such as Microsoft Office and Openoffice.org.

User interface

The user interface of the experiment environment consists of three windows. The first window contains the applet running in the internet browser, called the “Main window”. The second window is the “Experiment window” in which students spend most of their time. The last window is the “Test your answer window” in which students can test if their answer to the task is correct.

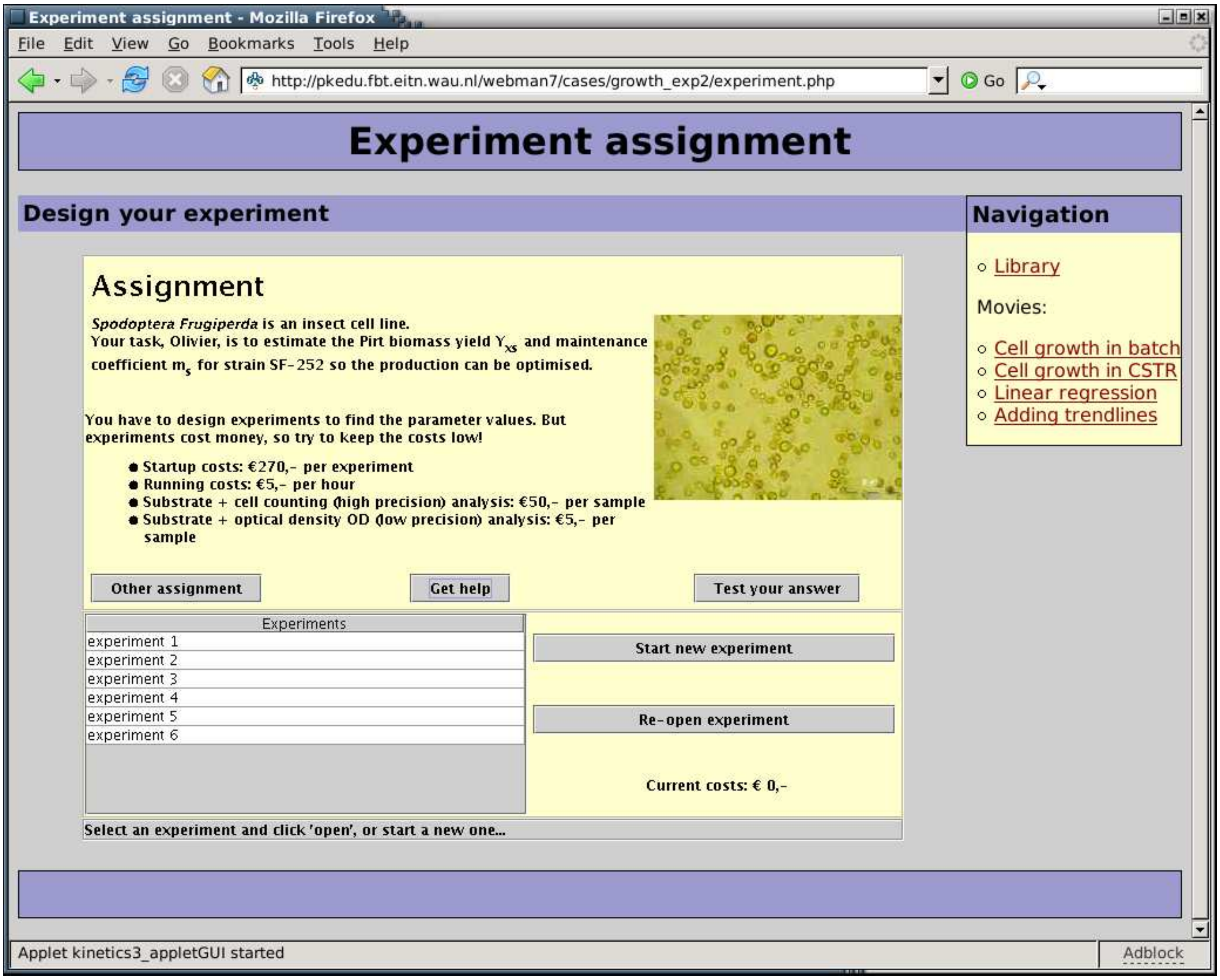

Figure 1: *The "Main window" in which the assignment is presented to the students.*

Students start in the Main window (Figure 1). This window introduces the task to the students, and it allows them to request a new task and to reset the experiment environment. Second it gives access to the experiments and it allows the students to start new experiments. Third it gives the students access to the Test window. Last it gives access to help pages that open in a new browser window.

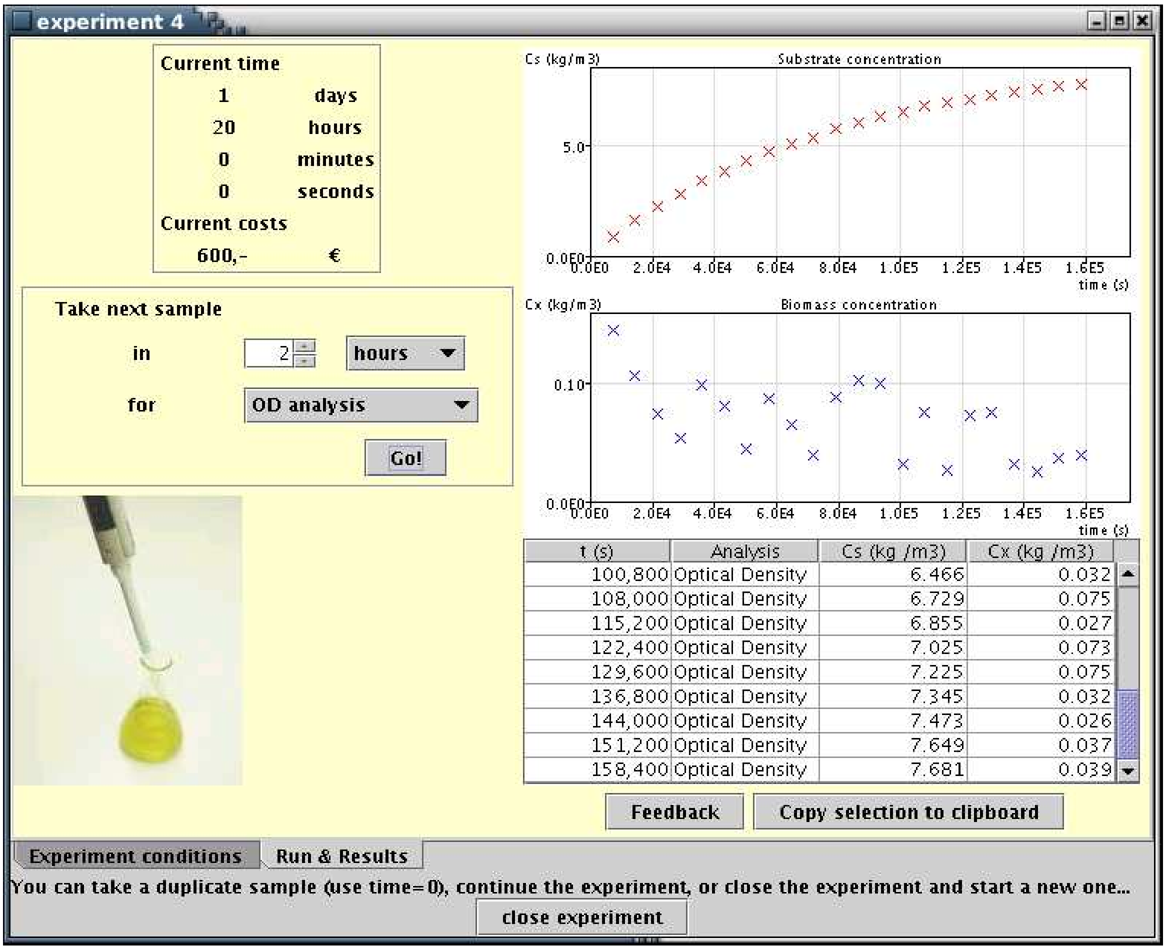

Figure 2: *The "Run & Results" tab, in which students choose when to take samples, and see the results of their experiment. These results are from a CSTR which is not yet in steady state.*

Clicking "new experiment" results in a new experiment window. Each experiment will open a separate window. The user interface for experiment setup and execution is designed to continuously adjust and limit the choices presented to the students to those that are relevant to the pertinent situation. There are, for example, two stages in each experiment: initial setup of the experiment and running the experiment. The tabbed layout matches these stages (Figure 2).

In the "Experiment conditions" tab, students can modify the reactor setup, initial biomass concentration, initial substrate concentration, feed flow rate, and feed substrate concentration. If students enter impossible values, such as concentrations exceeding the solubility, they are notified on the status bar and starting the experiment is disabled.

The “Run & Results” tab shows the buttons to run the experiment and to take samples, a button to get feedback on the results, and the results themselves, presented both graphically as well as in a table.

For sample analysis, students can choose between a cheap and rough optical density analysis and a more expensive and more precise cell-counter analysis. The reason to introduce this choice in the experiment environment is to force students to think about experimental error and costs when taking a sample.

The experiment starts with the first sample and is continued by selecting when to take the next sample. An ‘incremental time’ of zero can be used for a duplicate sample. The results, substrate and biomass concentrations, are plotted against time in graphs that automatically scale to show all results. The results are also added to the table. At any moment a student may decide to copy and paste the results to a spreadsheet program to do calculations. The “Feedback” button will show feedback on the current results in relation to the task (see below). If, for example, a student who has to find the maximum specific growth rate μ_{max} and Monod constant K_s would click “Feedback” after running a batch experiment for five minutes, the experiment environment will inform this student that these results are not very useful to find the maximum specific growth rate μ_{max}, because there is experimental error and only little difference between the lowest and the highest biomass concentration. It will also inform this student that a batch experiment is not useful to find the Monod constant K_s because estimates of specific growth rate at a low substrate concentration in batch are inaccurate.

The experiment window is not attractive to “click without reasoning”. It stimulates a more rational analysis [13], because changing initial conditions does not immediately result in new output. Students first need to run the experiment and take samples. Badly chosen initial conditions or sample times result in useless data, while at the same time increase costs.

If manipulation of conditions is unlimited, easy accessible and immediately resulting in new output, students tend to change conditions without reasoning and work very inefficiently [14]. The delay between the experiment set up and the results is furthermore a form of asynchronous feedback, which could induce the deeper thought processes demanded by the task [4].

In a couple of hours, students may use the experiment environment to design and run many experiments with different results. Whenever students have found an answer to the task, i.e. a parameter estimate, the "Test your answer" button in the main window is clicked to open a new window, in which the estimate is submitted and judged if it is within the satisfactory range defined by the lecturer. If the value is within range, the cost score will be sent to the server and shown to other students.

To stop students from guessing the parameter values to get the cheapest design without serious experiments, the experiment environment will notify the students if their experiments are inadequate. For example, if the Monod maximum specific growth rate μ_{max} is to be estimated, students should have at least an experiment that has at least two samples at a substrate concentration ten times higher than K_s and a biomass concentration within the virtual detection range, and the highest and lowest biomass samples should have a two times difference. If not, the notification message is for example: "This batch experiment is not very useful to find the maximum specific growth rate, because the biomass concentrations are all very low and hard to measure accurately".

Experimental-data generation

The experimental data that are obtained by the student is generated from a set of differential equations. The experiment environment solves these equations using a Runge Kutta integrator. Experimental error is added to the exact output. The experimental error of student measurements in a real practical were evaluated and copied into the experiment environment. The experimental error contains absolute and relative parts in such a way that students are confronted with a relatively small error above a certain threshold and a relatively large error below this threshold.

The courses "Introduction to Process Engineering" and "Bioprocess Engineering" both use the same set of differential equations. All data generated by these equations should follow Monod kinetics and the Pirt linear growth law when substrate is available. Because of the zero-order nature of maintenance, the Pirt linear growth law predicts substrate consumption in the absence of substrate. To prevent negative substrate concentrations, a different equation is needed. The Runge Kutta integrator cannot handle a model with non-continuous equations. Beeftink *et al.* [15] describe a continuous model that follows Pirt if substrate is available and that has realistic results if no substrate is available:

$$\mu_{\mathrm{obs}} = \mu - m_{\mathrm{s}} \cdot Y_{\mathrm{xs}} \cdot \left(1 - \frac{C_{\mathrm{s}}}{C_{\mathrm{s}} + K_{\mathrm{shift}}}\right) \quad \left(\mathrm{s}^{-1}\right) \quad (5)$$

$$q_{\mathrm{s,obs}} = \frac{\mu}{Y_{\mathrm{xs}}} + m_{\mathrm{s}} \cdot \frac{C_{\mathrm{s}}}{C_{\mathrm{s}} + K_{\mathrm{shift}}} \quad \left(\mathrm{kg(S)} \cdot \mathrm{kg(X)}^{-1} \cdot \mathrm{s}^{-1}\right) \quad (6)$$

These equations are used to generate experimental data for both courses. Because students use the original Pirt equation, K_{shift} is chosen very small ($K_{\mathrm{shift}} = K_{\mathrm{s}}/1000$), such that the deviation from Monod kinetics and the Pirt linear growth law is negligible in the substrate range in which students do their experiments.

Feedback

Because students should learn how to design experiments themselves, there is no recipe provided with the experiment environment. There is, however, feedback provided to help students with the most common problems, and induce verification, elaboration, and concept development.

The “Get help” function from the main window does give feedback how to approach experiment design and how to measure in cell-culture systems. This feedback focuses on concept development. The experiment design feedback is for example to split up the problem into smaller sub-problems and to measure under conditions that minimize experimental error. Feedback on measuring in cell-culture systems is for example which concentration ranges can be analyzed well, that CSTR experiments can be used to measure under constant conditions, and that batch experiments can be used to measure maximum specific rates.

In some situations, however, feedback is given that is much more focused on verification and elaboration. In situations where the experiment design is clearly flawed, the student is immediately notified. For example, if the initial biomass concentration is zero, or the feed flow rate is extremely high, a message on the status bar will notify students. To allow students to acquire intuitive knowledge about substrate inhibition, without the need to include it in the model, feedback is shown if they enter high substrate concentrations. A message on the status bar will notify them that their concentration might in reality lead to substrate inhibition.

Students can use the "feedback" button in the "Run & Results" tab to get feedback on their experimental results. Because it is unknown to the experiment environment what the intentions of the student are (e.g. to get a rough idea, or to do precise measurements), it is often impossible to give detailed feedback on the results. The feedback therefore shows how the experimental results compare to the task (e.g. are the results useful to estimate the maximum specific growth rate), and how the results compare to the measuring system (e.g. is the CSTR in steady state).

Usage & evaluation

In the course "Introduction to Process Engineering" in 2004, eighty-four students used the experiment environment in groups of two. This course is taken by students in Biotechnology and students in Food Science and Technology at the beginning of the second year of their BSc. Students had to estimate the maximum specific growth rate μ_{max} and the Monod constant K_s. In a two-hour computer-lab session they could start the assignment. A lecturer was available during this session to answer any questions. The assignment was to be finished independently during that week. Eighty-three students finished the assignment, in an average of 4.4 hours.

The assignment was followed by a test, and the course was followed by an exam. The test results contributed to the final exam results. The total time of lectures, tutorials and computer lab sessions spent on model-related and experiment-related learning objectives equaled the amount of time spent on model-related learning objectives alone in previous years. Students scored satisfactory in the test and exam, both in the previously supported learning objectives as well as the newly supported learning objectives. In comparison with previous years students attained more learning objectives in the same amount of time.

Students filled in a questionnaire immediately after finishing the assignment. The questionnaire started with an open question as to what the students thought they learned most from the experiment environment. Forty one percent referred to the model theory, forty percent referred to experiment design, and eleven percent referred to experiment peculiarities such as "how much time experiments can take", "how expensive experiments are", and "how important good experiment design is". Eight percent did not answer the question.

The open question was followed by several multiple-choice questions with answers in the range one (disagree) to five (agree). The average answers are shown in Table 1. Students indicated they consider the module very useful, and confirmed they think experiment design is an important part of their curriculum.

Table 1: *Average questionnaire answers for the closed questions in the course "Introduction to Process Engineering".*

Question	***Answer (1 disagree – 5 agree)***
did you learn much about cell-growth theory	3.4
did you learn much about cell-growth experiments	3.7
did you learn much about experiment design	3.7
do you consider experiment design important for your curriculum	4.0
do you think the module is challenging	3.6
do you think the module is fun	3.1
do you think the module is difficult	3.1
do you think the module is motivating	3.2
do you think the module is useful	3.9

Faculty involved with the course were very enthusiastic about the virtual-experiment environment, especially the way it activates the students to work with the abstract theory. Students were very motivated and worked very hard. The faculty is convinced that the experiment environment supports essential learning objectives, and supports those well.

In the course "Bioprocess Engineering" in 2005, fifty students used the experiment environment in groups of two. This course is taken by students in Biotechnology at the end of the second year of their BSc. Students had to estimate the biomass yield on substrate and the maintenance coefficient. The module was used and evaluated similar to the course "Introduction to Process Engineering". Forty-nine students finished the assignment, in an average of 3.3 hours. The exam question on model-related and experimenting-related learning objectives was answered satisfactory.

In the open question of the questionnaire, seventy two percent of the students indicated they learned much from the experiment environment. Thirty nine percent indicated they learned much about model-related learning objectives. Another thirty nine percent indicated they learned much about experiment-related learning objectives. Twenty two percent indicated that the assignment was very difficult in the beginning, only after a while they understood how to approach the assignment. Seventeen percent indicated they had problems to process the experimental results on the computer. Six percent indicated they felt that one needs good luck to finish the assignment quickly.

The open question was again followed by several multiple-choice questions, shown in Table 2. Students gave slightly higher ratings in this course.

Table 2: *Average questionnaire answers for the closed questions in the course "Bioprocess Engineering".*

Question	***Answer (1 disagree – 5 agree)***
Did you improve your experiment design skills	3.8
did you improve your understanding of the model	3.8
did you improve your parameter estimation skills	3.7
do you think the module is challenging	3.8
do you think the module is fun	3.3
do you think the module is difficult	3.2
do you think the module is motivating	3.6
do you think the module is useful	3.9
give an overall rating for the module	3.7

An interesting observation in some of the multiple-choice answers is the trend they showed in time: from more positive for students that rapidly finished the assignment to less positive for students that finished the assignment more slowly. The questionnaire was immediately answered after students finished the assignment. Most students in the course "Bioprocess Engineering" finished the assignment in the same afternoon they started it. The time at which the questionnaire was submitted was, for those students, thus representative for the time they needed to finish the assignment. The "overall rating", for example, was rated 4.1 by the first half of the students, and rated 3.4 by the second half of the students. Other questions show similar trends in the answers: students that need more time to finish the assignment give less positive answers. Factors that may have played a role are the fit between prerequisite knowledge and prior knowledge, the group culture, and student confidence.

All faculty members involved with the course were again very enthusiastic about the virtual-experiment environment.

Discussion

Lessons learned

During the development of the virtual-experiment environment several preceding designs have been created and tested with formative evaluations. "Formative" evaluation refers to structured evaluation that takes place during the use of the learning material with the intention to improve the material [16]. The results of these evaluations have, to some degree, shaped the current virtual-experiment environment.

First, formative evaluations of a prototype virtual-experiment environment that included many practical aspects of cell culturing showed that mixing theoretical and practical learning objectives in a virtual-experiment is not effective. If experimental results were unexpected, students resorted to practicalities to solve the problem. Students did not consider if their approach was theoretically sound at all.

Second, the same formative evaluations showed that simulating practical issues such as equipment peculiarities should be done very well or should be avoided. Although we spent a considerable amount of time to simulate the equipment correctly, students indicated that the relation between the peculiarities of the simulated equipment and the peculiarities of real equipment is not clear to them. Students indicated this affected their motivation in a negative way.

Last, in several tests students indicated that it is frustrating if a computer environment makes them do work manually that is easily done by a computer. Manually calculating concentrations from computer generated HPLC results is, for example, considered more frustrating then manually calculating concentrations from real HPLC results.

Because of these three observations, all practical learning objectives were removed from the virtual-experiment environment.

Extending the learning objectives

The experimenting-related learning objectives all have a statistical background. The virtual-experiment environment may be extended with more statistical learning objectives. The model as described in this article is particularly interesting to introduce statistical theory.

The model's behavior is non-linear in time, and linearizing or non-linear fits are thus required for parameter estimation. This allows the lecturer to introduce aspects such as error propagation and the statistical impact of linearization (for example [17, 18]). There are, furthermore, many publications specific for this model. For example about optimal experiment design for parameter estimation [19, 20, 21], as well as optimal experiment design for model discrimination [22].

Depending on the statistical background of the students and the learning objectives of the course, three extensions are envisioned for the virtual-experiment environment.

First, calculation of the confidence interval could be added to the assignment. Currently the experiment environment does not include error estimates as part of the assignment. The experiment environment only tests if the difference between the model parameter values and student parameter estimates are within a certain range. In reality, students should calculate the confidence interval of their estimate, and decide if it is sufficiently small. The experiment environment should thus check if the confidence interval found by the students is correct. Calculating the confidence interval from data from multiple different experiments, however, requires advanced statistical knowledge.

Second, the experiment environment could be modified to include model discrimination, i.e. the determination of the best model for a certain set of data. Students can use the virtual-experiment environment to practice experiment design for model discrimination, and practice methods for model discrimination (for example [23, 24]).

Third, the experiment environment could be used to introduce the concept of optimal experiment design. An experiment design is optimal if it allows the most precise calculations of the parameters for a given model. Using knowledge of the parametric sensitivity, the error in the measurements, and an estimate of the parameters, the optimal design can be found. Optimal experiment design is only possible if it is known that the experiment object can be described with a certain model, and if the experimental error is known. Because the experiment is simulated, and thus the model and experimental error are known, the environment can thus be used to introduce optimal experiment design to students.

Conclusion

The virtual-experiment environment successfully exploits the opportunities in model-related learning objectives and experiment-related learning objectives. Students get an authentic task: students are asked to estimate model parameters. In order to do so, students can design and run experiments. Experimental results contain experimental error, which is an essential constraint for the design.

It is argued that the experiment environment supports qualitative understanding of the behavior of biomass concentrations and substrate concentrations in batch and CSTR reactors, understanding under which assumptions this model is valid, and intuitive knowledge of the order of magnitude of parameters and variables, for a few common cell types and substrates. It is, furthermore, argued that the experiment environment supports qualitative understanding of the uncertainty in parameter estimations, understanding that there may be large differences between the uncertainty in estimates of different parameters, understanding that uncertainty is affected by the experiment design, knowledge that constraints such as operational limits of analysis equipment, parametric sensitivity, and costs have impact on experiment design, and skills to design an experiment that is appropriate to answer a specific research question.

Design and development of the experiment environment as digital web-based learning module was successful. The experiment environment is implemented as java applet, and the user interface is designed to structure the learning process. The environment gives feedback that supports the learning objectives to the student.

The experiment environment is used by 134 students in two different courses in the second year of the BSc curriculum Biotechnology and the BSc curriculum Food Science and Technology. Usage is evaluated with questionnaires and in the final exam. Students rate the environment's usefulness 3.9 on a 1-5 scale. Students indicate that the learning objectives are well supported by the experiment environment. The faculty involved in these courses, furthermore, is convinced that the experiment environment supports essential learning objectives, and that it supports those learning objectives well.

References

1. Anderson, JR (1995) Learning and memory. An integrated approach. John Wiley & Sons, Inc, New York
2. De Jong T, Van Joolingen WR (1998) Scientific discovery learning with computer simulations of conceptual domains. Rev. Educ. Res. 68(2):179-202
3. Merriënboer van JJG (1997) Training complex cognitive skills. Educational Technology Publications, Englewood Cliffs
4. Williams V (2003) Designing Simulations for Learning. J. Instr. Sci. Tech. 6(1), http://www.usq.edu.au/electpub/e-jist/docs/Vol6_No1/contents.htm
5. Nielsen J (2000) Designing Web Usability: The Practice of Simplicity. New Riders Publishing, Indianapolis
6. Keller JM (1987) Development and use of the ARCS model of motivational design. J. Instr. Dev. 10(3):2-10
7. Norman D (1993) Things That Make Us Smart - Defending human attributes in the age of the machine. Reading, Addison-Wesley, Massachusetts
8. Hofstein A, Lunetta VN (2003) The Laboratory in Science Education: Foundations for the Twenty-First Century. Sci. Educ. 88(1):28-54
9. Fisher D, Harrison A, Henderson D, Hofstein A (1999) Laboratory learning environments and practical tasks in senior secondary science classes. Res. Sci. Educ. 28:353-363
10. Champagne AB, Gunstone RF, Klopfer LE (1985) Instructional consequences of students' knowledge about physical phenomena. In West LHT, Pines AL (eds) Cognitive structure and conceptual change. Academic Press, New York, pp 61-68
11. Eylon B, Linn MC (1988) Learning and instruction: An examination of four research perspectives in science education. Rev. Educ. Res. 58(3):251-301
12. Savery JR, Duffy TM (1995) Problem-based learning: An instructional model and its constructivist framework. Educ. Tech. 35(5):31-37
13. Anderson, JR (1993) Rules of the mind. Hillsdale. Erlbaum, New York
14. Trudel C, Payne SJ (1995) Reflection and goal management in exploratory learning. Int. J. Hum. Comput. Stud. 42:307-339
15. Beeftink HH, Heijden Van der RTJM, Heijnen JJ (1990) Maintenance requirements: energy supply from simultaneous endogenous respiration and substrate consumption. FEMS

Microbiol. Ecol. 73:203-210
16. Carroll JM, Rosson MB (1995) Managing evaluation goals for training. Comm. ACM, 38(7): 40-48
17. Sagnella GA (1985) Model fitting, parameter estimation, linear and non-linear regression. Trends Biochem. Sci. 10:100-103
18. Smith LH, McCarty PL, Kitanidis PK (1998) Spreadsheet Method for Evaluation of Biochemical Reaction Rate Coefficients and Their Uncertainties by Weighted Nonlinear Least-Squares Analysis of the Integrated Monod Equation. Appl. Environ. Microbiol. 64:2044-2050
19. Munack A (1989) Optimal feeding strategy for identification of Monod-type models by fed-batch experiments. In Fish NM, Fox RI (eds) Modelling and control of biotechnical processes. Proc. 4th Int. Congr. Comp. Appl. Ferm. Technol., Cambridge, Pergamon, Oxford
20. Baltes M, Schneider R, Sturm C, Reuss M (1994) Optimal Experimental Design for Parameter Estimation in Unstructured Growth Models. Biotechnol. Prog. 10:480-488
21. Dette H, Melas VB, Pepelyshev A, Strigul N (2003) Efficient design of experiments in the Monod model. J. Roy. Statist. Soc. Ser. B. 65:725-742
22. Takors R, Wiechert W, Weuster-Botz D (1997) Experimental Design for the Identification of Macrokinetic Models and Model Discrimination. Biotechnol. Bioeng., 56:564-576
23. Burham KP, Anderson DR (1998) Model Selection and Inference. A Practical Information and Theoretic Approach. Springer Verlag, New York
24. Stewart WE, Shon Y, Box GEP (1998) Discrimination and goodness of fit of multiresponse mechanistic models. AIChe J. 44:1404-1412

Chapter 6

Web-based education in bioprocess engineering

Abstract

The combination of web-technology with knowledge of bioprocess engineering and with theories on learning and instruction may yield innovative learning material for bioprocess engineering. In this article, an overview is given of the characteristics of web-based learning material, as well as guidelines for the design of learning material from theories of learning and instruction, and from the bioprocess engineering domain. A diverse body of learning material is presented that illustrates the application of these guidelines. The material has been developed over the last six years for different courses, mostly at the BSc level. It illustrates how web-based learning material may enable various different approaches to learning objectives that may improve overall learning. The learning material has been used for several years in education, has been evaluated with very positive results. It is now part of the regular learning material for bioprocess engineering at Wageningen University.

submitted for publication as O.D.T. Sessink, H. van der Schaaf, H.H. Beeftink, R.J.M. Hartog, J. Tramper. *Web-Based Education in Bioprocess Engineering*.

Introduction

The landscape of academic learning is changing since the mass adoption of the Internet in education. Up till now most changes have addressed communication and distribution. Web technology, however, enables new approaches for content. This may change the way future learning material is designed and used [1,2].

In this article, it is argued how the combination of web-technology with knowledge of Bioprocess Engineering and with theories for learning and instruction may yield innovative Bioprocess Engineering learning material. First an overview is given of the characteristics of web-based learning material. Second, guidelines for the design of learning material are given from theories of learning [3], as well as guidelines from the Bioprocess Engineering domain. Third, a diverse body of web-based learning material is presented to illustrate the application of these guidelines, and to show the various approaches to learning objectives that are facilitated by web-based learning material.

Web-based learning material

Web-based learning material is created with web-technology (See box 1). Web-based learning material is a subset of digital learning material. It therefore has characteristics that are unique for web-based learning material, but also characteristics of digital learning material in general.

Few requirements

Web-based learning material has few requirements for the client computer. Digital learning material in general, may require for example the availability of a modeling program, a specific platform, or a specific viewer program on the client computer. A web-browser, however, is available by default on almost all platforms. A computer with Internet Explorer on Windows satisfies the requirements. However, a free web-browser on a free platform such as Mozilla Firefox on Linux also satisfies the requirements, and does not force students to buy or use proprietary software.

Direct access

If the above requirements are met, web-based learning material can be accessed independently from time and place by many students simultaneously. In contrast to digital learning material that is distributed on physical media (e.g. a CDROM), students directly access the original source of the material. Students will thus always have the latest version of the learning material. After learning material is updated, all students have immediate access to the updated material. Authors may update their learning material during a course to incorporate feedback from students, even if the material is used in distance education.

Hyperlinking

Although not unique to web-technology, the hyperlink is perhaps the most well known characteristic of web-technology. The hyperlink allows one piece of web-based material to reference any other piece of web-based material, such as learning material from another university, literature, or an online encyclopedia.

Rich content

Digital learning material in general allows for various forms of multimedia and interaction. It may include formatted text, drawings, photos, animations, simulations (see below), video, interactive applications (see below) and communication components (see below). Web-based learning material in particular, offers various high-level components to build such rich material. Using multimedia in learning material may, if used appropriately, improve learning effectiveness [4,5].

Interaction and feedback

Digital learning material enables interaction with the student. The learning material may show a response to a student action such as mouse actions, or keyboard actions.

One of the most important applications for interaction in learning material is feedback. Feedback is information about progress towards ones goal [6]. There are several design guidelines for feedback available [7]. Feedback may, furthermore, influence the students motivation. If, for example, a student is struggling for a long time on some task, the feedback may become very concrete and even contain the correct answer to avoid the student losing confidence.

Simulation

Digital learning material has access to computational resources, which can be used to provide simulation. Using a model, simulations produce output from student input. In a flight simulator, for instance, the inputs are the handles and the control stick, and the output are changes in the cockpit window and the indicators. If learning material allows only a few combinations of inputs, all possible outputs can be included in the material and simulation is not required [8]. If, in contrast, a very wide or unlimited range of student inputs are allowed, a simulation is appropriate [9, 10]. The effect of input on output is present in many learning objectives: from the operation of equipment to the discovery of system behavior. These learning objectives may be the ability to detect anomalies, to identify signs of specific behavior, to predict behavior or to predict a state, or to find the combination of inputs required to obtain a specific state or specific behavior [11].

Simulations may be used to support a learning objective in a way that would be impossible or unacceptable in reality because of danger, costs, availability of equipment, time or irreversible damage. Students may use simulations to practice situations they may never meet during their education, or to prepare for situations for which access in education is very limited [9, 10].

Simulations may, furthermore, exaggerate reality to provide more effective learning experiences [12]. The simulation may, for example, accelerate time, and thus enable students to discover how a parameter affects long term system behavior, or the simulation may confront all students with an important but normally infrequently occurring phenomenon, or the simulation may simplify reality and assume ideal behavior.

Collaborative learning material

Although collaborative learning does not necessarily require special learning material, web-based material may offer advantages for collaborative learning. Collaborative learning needs communication, and web-based learning material may integrate a communication component.

Integrated learning experience

An integrated learning experience means that a learning environment allows students to engage in multiple tasks related to learning, without a conscious switch between these tasks. In a well integrated environment, for example, students may move their attention from a text, to a video, and then to a communication component, without the feeling that any attention was required for anything other than learning. Digital learning material can support a variety of tasks, and may thus enable an integrated learning experience [13].

Adaptivity

Adaptive learning material builds a model of the goals, preferences and knowledge of each individual user, and uses this model throughout the interaction with the user, in order to adapt to the needs of that user (adapted from [14]). Adaptive learning material may, for example, present more detailed and deep information to more qualified students, or suggest a set of most relevant links to each individual student. This may for example minimize the user's floundering, make the learning more goal-oriented and make the learning more efficient [15].

Adaptive material needs to identify individual users, track their activities, and build, retrieve and update the model of the user. Any digital learning material may provide the necessary means. Web-technology in particular has many means for authentication, tracking, and structured data storage, and is thus well suited for implementation of adaptive functionality.

Guidelines for learning material design

Six guidelines for the design of learning material have been used in this article. These guidelines are further referred to as learning material guideline (lg).

lg1: Learning material should activate students

Learning is found to be more effective if students participate actively [16]. To acquire intuitive knowledge about system characteristics, for example, learning through experimentation is found to be more effective than learning from a book [17]. Learning material that stimulates students to put effort into learning is called activating learning material. The most simple example of activating learning material is an assignment that students have to complete.

Digital assignments are very well suited to activate students. Digital assignments may interact with students, they may provide hints to resolve the most common problems, and may provide verification if students answered them correctly. Students work at their own pace, and depend little on the lecturer if they encounter problems. Slower students, furthermore, may get the satisfaction of answering the assignment by themselves before it is presented by the lecturer. Experience at Wageningen University learns that a much larger percentage of students is actively working with digital activating learning material compared to activating assignments in a tutorial.

lg2: Learning material should minimize cognitive load

The cognitive load (mental effort) imposed by learning material should be optimized for learning. Cognitive load theory assumes that learning is limited by the total cognitive capacity of a student. Extraneous cognitive load, for example caused by an ill-designed user interface, or caused by a non-native language, should thus be minimized. Intrinsic cognitive load, the load that is directly related to the learning objectives, should match the students level and the available time to work with the learning material [18].

Many guidelines for user interface design can be used to reduce extraneous cognitive load [19]. Web-based learning material should use web-site components that students are accustomed to. The material should, furthermore, use visual aids such as images, highlights and pop-up windows with extra information where appropriate.

lg3: Learning material should be modular by design.

Modularly designed learning material addresses only one or a few learning objectives simultaneously, and requires little unnecessary prerequisite knowledge. In a heterogeneous student population not all students have attained the same learning objectives. Learning material that requires unnecessary prerequisite knowledge may thus exclude some students from effective learning. Learning material about enzyme kinetics, for example, should not require knowledge about protein structures and composition.

Feedback, furthermore, is likely to be more effective if it is only triggered in a very specific situation, for example if the student has problems with a specific learning objective. If students have many different reasons to give a certain wrong answer to a question, feedback is likely to be less effective [7].

Some learning material designers tend to add many ‘catches’ to assignments that are not essential to learn the learning objectives. Students who have overlooked a minor detail will fail to complete the assignment. Unless this detail is the learning objective, the ‘catch’ is bad for confidence for the students who do know the subject, but fail because of the overlooked detail, and the feedback is probably not effective for those students either.

lg4: Learning material should provide authentic context and tasks

Authentic learning theory suggests that students learn better how and when to apply knowledge if the learning context and learning activity match the student’s world more closely. For example, biotechnology students should not learn theory about mixing in the context of paint mixing, but in the context of cell cultivation, and the task should not be to describe the color of a reactor, but to set up a mass balance over a reactor. An accepted way to provide an authentic context is to simulate it with digital learning material [20].

Real-life context and activities may cause a cognitive overload, and thus conflict with cognitive load guidelines [20]. It is the opinion of the authors that the cognitive load guideline should then precede.

lg5: Learning material should use personalized support where possible

Academic courses are typically attended by a heterogeneous student population. Students who attend a course might follow different studies, have attended different courses in the past, have attended different versions of the same course, at different universities, or even in different countries in different languages, and have a different learning style. Students, furthermore, may have poor skills, or may have exceptional skills.

Addressing a heterogeneous student population with the same learning material may be not be as effective for each individual student. Adaptable learning material allows students to choose different paths through the material, different levels of guidance, or different levels of feedback. Novice students, however, hardly use adaptable features [21]. In contrast to adaptable material, adaptive material automatically adapts to the users' needs based on the system's assumptions about student needs, and may thus be more appropriate address a heterogeneous student population.

lg6: Learning material should be motivating

ARCS, Attention, Relevance, Confidence and Satisfaction, determine a set of guidelines for motivation [22]. First, learning material should capture the student's attention: it should induce curiosity and interest. Second, it should be relevant for the student; it should thus connect to the world of the student, and students should feel that they can apply the newly learned knowledge. Third, the student should be confident that he/she can handle the assignment, and the material should stimulate sense of success. Learning material should for example make clear what the expectations are and when all expectations have been met by the student. And last, students should get satisfaction from finishing the assignment, there should be very few students who fail to finish the assignment, and the assignment should stimulate sense of achievement.

A few students at Wageningen University, furthermore, indicated that aesthetically pleasant learning material improved their motivation. Because large aesthetic improvements may cost a considerable investment, the aesthetics should only be addressed when the improvements are significant.

Guidelines for Bioprocess Engineering education

The guidelines for the Bioprocess Engineering domain are derived from its learning objectives. In Table 1 the most essential Bioprocess Engineering learning objectives can be found.

Table 1: *The most essential Bioprocess Engineering learning objectives*

#	***Objective***
1	Know transport and reaction phenomena and their mathematical representations that occur frequently in Bioprocess Engineering processes.
2	Know how common Bioprocess Engineering operations work, i.e. which phenomena are involved, how devices are structured.
3	Know orders of magnitude of parameters and variables for common Bioprocess Engineering processes.
4	Know which phenomena and quantities have significant impact in specific Bioprocess Engineering processes.
5	Know common assumptions and common knowledge used in Bioprocess Engineering models.
6	Reason qualitative system behavior in a few common Bioprocess Engineering systems.
7	Use models to solve quantitative problems.
8	Design models in terms of balance equations.
9	Design a unit operation, or a chain of unit operations for Bioprocess Engineering purposes.

These learning objectives have lead to the formulation of four guidelines for Bioprocess Engineering learning material. These guidelines are further referred to as Bioprocess Engineering guideline (bg).

bg1: Learning material should balance quantitative and qualitative practice

McDermott [23] describes a large scale experiment in which undergraduate students had to order light bulbs on relative brightness. The light bulbs were in simple circuits in parallel or series. Only 15% of the students managed to order the light bulbs correctly. Many of those students had no problems to use Ohm's law to calculate currents in much more complex setups. This experiment shows that success on quantitative exercises does not indicate qualitative understanding.

Most learning objectives from Table 1 are qualitative. Students should thus practice qualitative reasoning next to quantitative exercises. Solving quantitative assignments is, however, an important engineering skill. Students should, therefore, spent an equal amount of time on qualitative as on quantitative assignments.

An often heard argument from Bioprocess Engineering lecturers against qualitative assignments is that they are more difficult to grade than quantitative assignments. However, as the light bulb assignment illustrates, it is very well possible to create qualitative assignments that are easy to grade.

bg2: Learning material should use multiple ways to connect model theory to physical reality

The first time the learning material for experiment design and parameter estimation (see below) was used, students had to design an experiment to estimate the maximum specific growth rate of micro-organisms. Although students had already attended several tutorials in which they answered various quantitative assignments about Monod kinetics, quite some students asked their supervisors what the maximum specific growth rate actually was. Those students had no qualitative idea what they had been calculating in the earlier assignments. From their perspective, they were given a mathematical parameter μ which they used in some equation called Monod.

Students need more practice how model theory and formal symbols relate to physical reality and vice versa. Many Bioprocess Engineering textbooks feature mostly assignments in which students calculate system output given some parameter values, listed with their formal symbol. There are, however, more aspects of the connection between model theory and physical reality, as listed in Table 2.

Table 2: *Aspects of model theory and physical reality*

#	***Aspects of model theory and physical reality***
1	Mechanism / phenomenon
2	Mathematical representation
3	Qualitative or quantitative system behavior
4	Natural language representation
5	Model domain (valid input range)
6	Operational definition of symbols
7	Assumptions

Students should be confronted with all aspects in their assignments. Table 3 lists example assignments that address different aspects of the connection between model theory and physical reality.

Table 3: *Example assignments that address different aspects of the connection between model theory and physical reality. The aspects numbers refer to Table 2*

Assignment	*Aspects*
Estimate parameter values for a given model from given experimental data.	3, 6
Choose the most appropriate mathematical representation given a set of experimental data and a set of possible mechanisms.	1, 2, 3
Reason the behavior of a system after a given parameter change.	3
Adapt a mathematical representation to include some assumption.	2, 7
Calculate a steady state given a mathematical model and parameter values.	2, 3
Give the model domain for a model and its underlying assumptions.	2, 5, 7
Choose a mathematical representation for a system described in natural language.	1, 2, 4
Give a formal representation for a given mechanism described in natural language.	1, 2, 4

bg3: Learning material should allow students to practice design skills

Design assignments state objectives and specify requirements, and allow the student to combine methodical steps and personal decisions to achieve any result that meets these objectives and requirements (adapted from [24]). Design assignments do not have a single correct result; there are many results that meet the requirements possible. In the competences listed for graduates in Bioprocess Technology at Wageningen University design skills are prominently listed. Graduates should be able to design biotechnological products, production processes, biocatalysts, and scientific experiments.

At Wageningen University, students currently practice design only a few times. Because each design is unique, students find it difficult to judge their own results, and thus require much assistance. Grading design assignments is, also because of the uniqueness, time consuming. Even experiment design is rarely done by students themselves; in practical courses, students often follow a recipe [25, 26].

Design can be practiced with web-based learning material. This learning material may track all student activity, and assist the student with frequently occurring problems. The material may, for example, force a step by step structured approach, in which every step is checked for compliance with previous steps. Alternatively the design environment may allow students to simulate the operation of their design, such that they can learn from its behavior, and improve their design accordingly.

bg4: Learning material should confront students with uncertainty in parameters and models

Students, especially in BSc courses, falsely assume that models and parameter values are perfect. Students are seldom confronted with uncertainty [9], many textbooks do neither show the uncertainty in parameter values, nor mention the limited applicability of models.
When using existing models, student should for instance always be notified what the application domain of the model is, to what degree the system in reality exhibits ideal behavior, how large the uncertainty in parameters is, how large the uncertainty in model predictions is, and how sensitive the system is to changes in temperature, pressure and composition. When learning how to design new models, students should be informed that models with parameters that cannot be measured accurately, or not at all, have a limited applicability.

Approaches to Bioprocess Engineering learning objectives

A variety of web-based learning modules for Bioprocess Engineering has been developed at Wageningen University over the last six years. Each of the modules has been designed with a focus on one or a few of the guidelines, which has resulted in learning material with different approaches to the learning objectives. The material is online available, visit http://fbt.wur.nl/ choose 'Content showcase', and then choose any of the listed Process Engineering modules. The modules can be categorized into case based material, adaptive material, material for experiment design and parameter estimation, material for downstream process design, and material for systematic model design.

Case based learning material

In the developed case based learning material, students are set into a scenario in which they have to solve some problem. This material provides the most authentic context of all designed material. The context and task are chosen from real problems described in literature or by industry, and are thus authentic (lg4), and students find them interesting and relevant (lg6). Students finish the task by answering a series of activating closed questions (lg1). This approach borrows various aspects from adventure games, such as the storyline, role play, and discovery of new information. Feedback on student answers is provided in a slightly personalized 'staged' manner (lg5); the more tries students need to answer a certain question, the more feedback is given, and the more concrete hints in the feedback are. Together the sequence of questions and staged feedback guide students through the task.

Several case based modules have been designed for BSc courses, such as "Mixing and oxygen transfer", "Membranes" and "Heat transfer". Most of these cases start with qualitative questions, and slowly move to more quantitative questions (bg1). Although the scenarios initially present a design problem, students do not really practice design; all students follow the same questions and find the same result, and the 'open' character of design as described in guideline bg3 is thus absent.

These modules are used in a computer lab, where students work on the assignment in groups of two. The computer lab session is preceded by a regular lecture in which the subject is introduced. Assistants are available, but experience learns that little assistance is asked for. Assistants indicate they like the material, because they spend their time very effective with students who lack some prerequisite knowledge, or students who have in-depth questions. Yearly evaluation results indicate that students like these modules next to regular lectures and tutorials.

A framework for these modules is designed that allows for quick creation of a sequence of closed questions. The framework supports five types of closed questions: multiple choice, multiple answer, numerical value, item selection and ordering, and balance equation setup, and can be extended where required.

Adaptive learning material

One module of adaptive learning material has been designed, about "Cell growth kinetics in reactors". In this material each student follows an individual, personalized, path through a collection of questions (lg5), such that each student will get the 'most appropriate' questions to answer [27]. Depending on the answer of all previous questions, the next 'most appropriate' question is selected. For instance, a student who can immediately select the correct definition of the biomass yield may get more in-depth questions about biomass yield. A student who needs multiple tries before selecting the correct definition, however, will get more questions on the definition of biomass yield before any in-depth questions. Other criteria for question selections enable variation in topics between subsequent questions, variation in questions between different students working at the same time, and matching to the students confidence level (lg6). The questions should be mostly independent from other questions, and thus modularly designed (lg3), such that the adaptive system has much choice to choose the next question. Questions may, however, require some prerequisite knowledge. These questions are only presented after students have acquired this prerequisite knowledge. Students have to acquire the prerequisite knowledge by working through a group of other questions.

The questions in the module about "Cell growth kinetics in reactors" are similar to the questions described in the case based learning material, they are activating questions (lg1) with the same staged interactive feedback (lg5). Many questions in this module activate qualitative reasoning (bg1). The questions, furthermore, confront students with many different aspects of the connection between model theory and physical reality (bg2). The module thus not address guideline lg4 at all; it does neither present an authentic context, nor an authentic task.

The module is introduced in a short computer lab session. Students are then asked to finish the module individually before the regular lectures about this subject start. Students indicate they like the module very much, rating its usefulness 4.4 on a scale 1-5. Students indicate that this personalized material is very effective, and they indicate that they very much like the qualitative questions and the questions that connect model theory and physical reality. During the tutorials following this material, the lecturer observed that students had fewer problems with introductory theory as compared to previous years without this material.

A framework for adaptive learning material is designed that allows for quick creation of these adaptive modules. It is based on the framework for case based learning material, and adds a conceptually simple interface to specify the required information to use these questions in an adaptive sequence [27].

Experiment design and parameter estimation learning material

In the developed learning material for experiment design and parameter estimation, BSc students practice design of experiments for parameter estimation, execute these experiments virtually, and use the results for parameter estimation [9]. Students enter an abstract laboratory environment in which they can execute cell growth experiments. The results include experimental error and operational limits of analysis equipment, which are important constraints in experiment design. The abstract environment allows students to focus on model and experiment related learning objectives that often receive little attention in real laboratory experiments [25, 26].

Experiment design for parameter estimation is an activating (lg1), authentic (lg4) task for students, which they find very relevant (lg6). The learning material does not provide an authentic context such as described in guideline lg4; only a laboratory would. The material, however, avoids cognitive overload (lg2); in a real laboratory students are overwhelmed with the number of practical details, and fail to focus on design and estimation issues. The material enables students to practice the design process itself (bg3); the task is very open, and students do not receive much guidance. During design of experiments and interpretation of experiment results students continuously practice qualitative reasoning (bg1) and reconstruct their mental model of the system. Students are confronted with the uncertainty in parameters because students will find uncertainty in their own estimates (bg4). By estimation of parameters, students assign physical meaning to model theory themselves (bg2). Experiments are simulated with accelerated time, which allows students to practice and execute many designs in a short time, and learn from their choices. The material is modularly designed with respect to parameter estimation (lg3). The material does neither include nor require a specific parameter estimation method, which allows it to be used in multiple courses.

The learning material is used in two BSc courses. Although students have followed several practical courses prior to these courses, students indicate that real experiment design is mostly new to them. Assistants note that students indeed have little idea how to start. When started, however, students quickly learn from their mistakes and continuously improve their designs. Students rate the usefulness of the material 3.9 on a 1-5 scale.

The developed material includes various kinetic models, but is restricted to cell growth experiments in batch or CSTR reactors. A collection of Java components is, furthermore, developed for this material that can be used to create for example other virtual experiments.

Downstream process design material

In the developed learning material for downstream process chain design, BSc students have to design a process chain that meets certain purity, yield, and cost requirements. Students see an overview of their design, with the fixed composition of the starting material, and the variable composition of the final purified product, and the performance of each unit operation in between [28]. Students may insert and delete unit operations, or change operational settings for each unit operation.

The design task is an authentic task for students (lg4). Although several advanced design applications exist that would provide a more authentic context as described in guideline lg4 (e.g. SuperPro Designer and Aspen Plus), none of these design applications is suitable for education early in the BSc because they have a long learning curve.

To activate students to discover effects of changes in their design, the performance of each unit operation and the content of the final product immediately reflect those changes (lg1). This material is used in the first course on Bioprocess Engineering. To avoid a cognitive overload, the number of unit operations as well as the number of operational parameters for each unit operation are kept low (lg2). The low number of operational parameters allows students to discover a qualitative relation between these parameters and the performance of the unit operation (bg1).

Students practice the design process itself, because they are free to design any process chain that meets the design requirements; students are not guided to a specific answer (bg3). After students have created a first design, furthermore, an overview is given how their design relates to several designs from other students. The designs that achieved the highest yield, the highest purity, the lowest costs, and the lowest waste are shown. This shows students fundamental design characteristics: different designs may all match the requirements, and different optimizations are possible within the requirements. This comparison, furthermore, motivates students to improve their own design (lg6).

This learning material is used in an early BSc course. Students indicate this learning material is both challenging and fun, and indicate they like the possibilities for discovery how parameters affect unit operation performance, and how unit operation performance affects the total chain performance. Many students are very much motivated to optimize their design to get it shown in the high score list. Assistants, furthermore, noticed that students were working very concentrated and were often engaged in on-topic discussions.

If required, the developed material can be extended with other unit operations, or with unit operations with more operational parameters. The functionality of the unit operation has to be modeled based on a few simple properties of the stream, such as density, pI, and size.

Systematic model design material

In the developed material for systematic model design, BSc students practice design of a mathematical model in order to answer an assignment [29]. Students engage in a large number of activities grouped into four stages: general analysis, detailed analysis, model composition, and model evaluation. The material guides students very tightly through these stages, but still allows students to design their model freely.

In the designed material, students are asked to predict when oxygen becomes limiting in a certain aerobic cultivation. During this cultivation, the biomass concentration, substrate concentration and viscosity change. This makes a model the appropriate means for this prediction. Both the prediction of oxygen limitation in a cultivation as well as design of a mathematical model are considered to be activating (lg1), relevant (lg6), and authentic (lg4) tasks for students. There are many software applications that provide an authentic context to enter a model and run simulations, such as Matlab or Mathcad. These applications, however, assume the model is already available; none of them supports the model design process itself (bg3). These applications, furthermore, require a specific syntax or user interface with a long learning curve, which adds too much extra cognitive load for BSc students. The structured approach and the available interactive feedback provide guidance for students. Together, these reduce the cognitive load for the students (lg2), and make student more confident to handle the design assignment (lg6). The material is activating; the interactive questions continuously activate students to provide some answer, or to improve their answer (lg1). The learning material may help students to connect model theory to physical reality, students first design a model for physical reality, and then use this model to answer an assignment about physical reality (bg2).

Students use this material in the end of the BSc. Students indicate they like the structuring and unit checking, and they like the fact that they can design a model without getting stuck in mathematical manipulation. Students would appreciate it if such material would be used more often in Bioprocess Engineering courses.

The developed material uses the closed questions from the case based material framework, and some Java based tools for model composition and numerical simulation. These tools can be used to create learning material for any model described by differential equations. A collection of Java components is developed, furthermore, that can be used for simulation, unit checking and search for steady-states.

Discussion

Digital learning material should, in the authors opinion, complement other learning material where appropriate. Computer mediated education should not overtake all other forms of education, which is sometimes feared by students. As such, digital learning material can actually prepare students better for lectures and tutorials, enable more effective interaction with lecturers and assistants during tutorials, allow students to practice more challenging and motivation assignments, and enable students to engage in otherwise impossible learning activities.

The developed learning material requires extensive database functionality from the learning management system (LMS). Most current LMSs, such as Moodle, Blackboard, or WebCT, however, do not yet offer database functionality for learning objects. It is, furthermore, important that a standard interface for database functionality is available. Standards guarantee interoperability between different LMSs, for example when moving to a new LMS, or when sharing learning material between different lecturers or between different universities that use a different LMS. New standards for learning material that describe some of the required functionality have already been developed [30], and we expect that future standards will include adequate functionality.

In order to develop web-based learning material that exploits all possibilities of web-technology, advanced knowledge of both the specific domain and of web technology are currently required. We expect that the near future will bring high-level building blocks that are specifically designed to build learning material from. We expect, furthermore, that applications for learning material development that require less web-development skills will become available. Together this would make the development of web-based learning material more accessible to authors without a web-development background. Such high level components and user-friendly development applications, however, come at the price of reduced flexibility.

To some extent, the Bioprocess Engineering guidelines above were a starting point for learning-material development. Their final shape, however, emanated from the development, usage, and evaluation of the various presented learning material. Experience during a number of years has led us to the believe that these guidelines are essential for Bioprocess Engineering education. Evaluation results of both tutorials and exams in which guidelines bg1 – bg4 were applied indicate that students need more practice with learning material that implements these guidelines.

The presented learning material has been designed over the last six years. In the view of the guidelines from this article there may be several opportunities for improvement. A large part of the developed material addresses learning objectives that were at Wageningen University previously not supported by specific learning material. All of the learning material has been used and evaluated with very positive results and satisfactory exam results, which indicates that the material is at least on par or better compared to regular learning material.

The developed material that supports previously unsupported learning objectives shows that evaluation of new technologies for learning material in Bioprocess Engineering has lead to innovation in Bioprocess Engineering education as well. Perhaps most importantly, the development of new learning material has induced many valuable discussions about the current Bioprocess Engineering education and Bioprocess Engineering learning material at Wageningen University.

Conclusion

Web-based learning material is a useful addition for Bioprocess Engineering education. Combination of web-technology with general guidelines for learning material and guidelines from the Bioprocess Engineering domain allows for various new approaches to learning objectives, and has lead to innovation in Bioprocess Engineering education.

A variety of learning material has been designed and developed for Bioprocess Engineering. The designs may serve as inspiration for the development of new material. Next to being proof of concepts, most of the learning material has been used for several years in education, has been evaluated with very positive results, and is now part of the regular learning material for Bioprocess Engineering.

References

1. Evans, C., Gibbons, N.J., Shah, K., Griffin, D.K. (2004). *Virtual learning in the biological sciences: pitfalls of simply "putting notes on the web".* Computers & Education 43, 49-61
2. Yazon J.M.O., Mayer-Smith J.A., Redfield R.J. (2002). *Does the medium change the message? The impact of a web-based genetics course on university students' perspectives on learning and teaching.* Computers & Education 38(1-3), 267-285
3. Reichert, R., Hartmann W. (2004). *On the Learning in E-Learning.* Proceedings of EDMEDIA 2004 World Conference on Education Multimedia, Hypermedia and Telecommunications. 1590-1595.
4. Weiss, R.E., Knowlton, D.S., Morrison, G.R. (2002). *Principles for using animation in computer-based instruction: theoretical heuristics for effective design.* Computers in Human Behavior 18, 465-477
5. Mayer, R.E. (2003). *The promise of multimedia learning: using the same instructional design methods across different media.* Learning and Instruction 13, 125-39
6. Posner, G., Rudnitsky, A. (1986). *Course Design: A Guide to Curriculum Development for*

Teachers (3rd ed.). New York: Longman.

7. Narciss, S., Huth, K. (1999). *Motivational effects of the informativeness of feedback.* Annu. Meet. Am. Educ. Res. Assoc., Montreal, Quebec, Canada, American Educational Research Association, Washington DC
8. Harris, T., Leaven, T., Heidger, P., Kreiter, C., Duncan, J., Dick, F. (2000). *Comparison of a virtual microscope laboratory to a regular microscope laboratory for teaching histology.* The Anatomical Record, The New Anatomist, 265(1), 10 – 14
9. Sessink, O.D.T., Beeftink, H.H., Hartog, R.J.M., Tramper, J. (2006) *Virtual parameter-estimation experiments in Bioprocess-Engineering education.* Journal of Bioprocess and Biosystems Engineering DOI: 10.1007/s00449-005-0042-z
10. Gonzalez-Cruz, J., Rodriguez-Sotres, R., Rodriguez-Penagos, M. (2003). *On the Convenience of Using a Computer Simulation to Teach Enzyme Kinetics to Undergraduate Students with Biological Chemistry-related Curricula*. Biochemistry and Molecular Biology Education 31(2), 93-101
11. Aegerter-Wilmsen, T., Janssen, F., Kettenis, D., Sessink, O, Hartog, R., Bisseling, T. (accepted for publication). *Introducing molecular life science students to model building using computer simulations.* Journal of Computers in Mathematics and Science Teaching.
12. Darius, P., Schrevens, E., Van Der Knaap, H., Portier, K., Massonnet, G., Lievens, L., Duchateau, L., Thas, O. (2003). *Using Web-Based Tools for Teaching Statistical Concepts and Experimentation Skills.* proc. of Statistics & the Internet conf. http://www.stat.auckland.ac.nz/~iase/publications.php?show=6
13. Helic D., Maurer, H., Lennon, J., Scerbakov, N. (2001). *Aspects of a Modern WBT System.* Proc. Intl. Conf. on Advances in Infrastructure for Electronic Business, education, Science, and Medicine on the Internet, SSGRR 2001, CD-ROM publication (ISBN:88-85280-61-7), paper 38.
14. Brusilovsky, (2001). *Adaptive Hypermedia.* User Modeling and User-Adapted Interaction 11, 87-110.
15. Brusilovsky P. (2003). *Adaptive navigation support in educational hypermedia: the role of student knowledge level and the case for meta-adaptation.* British Journal of Educational Technology, 34(4), 487-497
16. Anderson, (1995). *Learning and memory, an integrated approach.* John Wiley & Sons, Inc, New York, USA.

17.De Jong, T., van Joolingen, W.R. (1998). *Scientific discovery learning with computer simulations of conceptual domains.* Review of Educational Research Summer, 68(2), 179-201
18.van Merriënboer, J. J. G. (1997). *Training complex cognitive skills.* Englewood Cliffs, New Jersey, Educational Technology Publications.
19.Mayer R.E., Moreno, R. (2003). *Nine Ways to Reduce Cognitive Load in Multimedia Learning.* Educational Psychologist 38(1), 43-52
20.Herrington, J., Oliver, R. (2000). *An instructional design framework for authentic learning environments.* Educational technology research and developments 48(3), 23-48.
21.Jorgensen A. H. & Sauer A. (1990). *The personal touch: A study of users' customization practice.* Proceedings of INTERACT '90, Elsevier Science Publishers, Amsterdam, 561-565.
22.Keller JM (1987). *Development and use of the ARCS model of motivational design.* J. Instr. Dev. 10(3), 2-10
23.Mc Dermott, L.C. (1993). *How we teach and how students learn - A mismatch?* American Journal of Physics 61(4), 295-298
24.Van Keulen, H. (1999). *Design teaching views and practices in Delft.* N.P. Juster (ed) The continuum of design education. Proceedings of the 21st SEED Annual Design Conference and 6th national conference on product design education. (Glasgow, UK), 51-58.
25.Hofstein A, Lunetta VN (2003). *The Laboratory in Science Education: Foundations for the Twenty-First Century.* Sci. Educ. 88(1), 28-54
26.Johnstone, A. H. & Al-Shuaili, A. (2001). *Learning in the laboratory; some thoughts from the literature.* University Chemistry Education 5, 42-51.
27.Sessink, O.D.T., Beeftink, H.H., Hartog, R.J.M., Tramper, J. (accepted for publication). *Proteus: A lecturer-friendly adaptive tutoring system.* Journal of Interactive Learning Research
28.Van der Schaaf, H., Vermue, M., Tramper, J., Hartog R.J.M. (2003). *A design environment for downstream processes for Bioprocess-Engineering students*. European journal of engineering education 28(4), 507-521
29.Van der Schaaf, H., Vermue, M., Tramper, J., Hartog R.J.M. (accepted for publication) *Support of modelling in process engineering education.* European journal of engineering education
30.Sessink, O.D.T., Beeftink, H.H., Hartog, R.J.M. (2005). *Database functionality for learning objects*. Tech. Cog. Instr. Learn. 2(4), 337-357

Summary

This thesis covers choices that have been made, and material that has been designed during the application and evaluation of web technology in education in food and biotechnology in general, and in bioprocess engineering in particular. Special attention has been paid to the design of learning material, and the designed material itself. This thesis comprises six chapters.

In Chapter 1 is explained that bioprocess engineering education teaches students what is needed to apply biotechnology in large scale production processes. Additionally, it is described that web technology is the technology that allows you to present information on the World Wide Web, and to retrieve this information.

Chapter 2 covers the increased risk for fraud during web-based exams as compared to traditional paper-based exams. It is important that the staff at universities is aware of this. Three types of measures are presented and illustrated to help technical staff in order to secure web-based exams: supervision support measures, software restriction measures and network restriction measures.

In Chapter 3 the capabilities of web-based software facilities for learning, called Learning Management Systems, are discussed. These systems currently limit authors to use all the possibilities of web technology. Six types of additional functionality that offer significant added value for learning material are described. The six types are illustrated with examples from learning material in food and biotechnology. They are: 1) support for adaptivity, 2) support for retrieval of current and previous states, 3) support for comparison of results, 4) support for student tracking for pedagogical research, 5) shared reference database functionality, and 6) problem scenario database functionality. All six types could be supported if Learning Management Systems would support database functionality for learning objects. Future standards should include such support. Standards are important to guarantee that learning material will work in all Learning Management Systems.

In Chapter 4 an adaptive system for learning called Proteus is described. This system adapts automatically to each student, and tries to deliver the most appropriate learning material to

each individual student. The power of adaptive systems is that they can effectively address students with different backgrounds, but they are also difficult to work with. Proteus, however, is designed such that it is usable for lecturers without ICT background. The system has been used to create learning material about cell growth in reactors. This learning material has been evaluated with very positive results.

In Chapter 5 a virtual experiment environment is described in which students have to design experiments in order to estimate parameters for a given model. These experiments are carried out in a virtual laboratory, and the results, including experimental error, are presented to the students. Students learn how to design experiments, and how to deal with experimental error. Students could do this in a real laboratory, but in a real laboratory there are too many other things to learn for students, and only little time is available. A virtual experiment does not have these disadvantages.

In Chapter 6 all designed and developed web-based learning material is discussed. An overview is given of various approaches to bioprocess engineering learning material that are enabled by web technology. The different approaches are compared in terms of six general guidelines for learning material, as well as four guidelines for bioprocess engineering learning material. The learning material has been evaluated with positive results. Next to the designed learning material itself, the evaluation of web technology for design of learning material has led to innovation in bioprocess engineering education, as well as valuable discussions about bioprocess engineering education.

Samenvatting

Dit proefschrift behandelt het toepassen van webtechnologie ten behoeve van het onderwijs in de levensmiddelentechnologie en de biotechnologie in het algemeen en ten behoeve van de bioprocestechnologie in het bijzonder. Speciale aandacht wordt geschonken aan de keuzes die zijn gemaakt bij het ontwerpen van dit onderwijsmateriaal en aan het eigenlijke leermateriaal dat is ontworpen. Het proefschrift omvat zes hoofdstukken.

In Hoofdstuk 1 wordt beschreven hoe bioprocestechnologieonderwijs studenten leert wat nodig is om biotechnologie toe te passen op praktische productieprocessen. Ook wordt er beschreven hoe webtechnologie het mogelijk maakt om informatie op het World Wide Web aan te bieden en te verkrijgen.

Hoofdstuk 2 behandelt de relatief grote risico's op fraude bij web-gebaseerde examens, vergeleken met traditionele schriftelijke examens. Het is belangrijk dat dit bekend is bij de stafmedewerkers van universiteiten. Er worden drie groepen maatregelen gepresenteerd om technisch medewerkers te helpen webgebaseerde examens veiliger te maken, te weten: surveillance-maatregelen, programmatuur-beperkende maatregelen en netwerk-beperkende maatregelen.

In Hoofdstuk 3 worden de mogelijkheden besproken die web-gebaseerde leerhulpmiddelen, ook wel leeromgevingen genoemd, bieden. In hun huidige vorm zijn de leeromgevingen beperkend, auteurs kunnen niet alle mogelijkheden van webtechnologie gebruiken. Zes groepen aanvullende functionaliteiten die waardevol zijn voor leermateriaal worden beschreven. Deze worden geïllustreerd met voorbeelden uit levensmiddelen- en biotechnologie-leermateriaal. De zes typen zijn: 1) ondersteuning van adaptiviteit, 2) ondersteuning voor het ophalen van de huidige en vorige status, 3) ondersteuning voor het vergelijken van resultaten, 4) ondersteuning voor het volgen van studenten voor pedagogisch onderzoek, 5) databankfunctionaliteit voor gedeelde referenties 6) databankfunctionaliteit voor gedeelde probleemscenario's. Alle zes typen kunnen ondersteund worden als leeromgevingen databankfunctionaliteit voor leerobjecten zouden ondersteunen. Toekomstige standaarden zouden deze ondersteuning moeten opnemen.

Standaarden zijn belangrijk om te verzekeren dat leermateriaal werkt in alle leeromgevingen.
In Hoofdstuk 4 wordt een adaptief systeem beschreven, genaamd Proteus. Dit systeem past zich automatisch aan aan elke student, en probeert, per individuele student, het geschiktste leermateriaal te leveren. De kracht van adaptieve systemen is dat zij studenten met verschillende achtergronden effectief kunnen bereiken, maar ze zijn tegelijkertijd ook ingewikkeld. Proteus is echter zo ontworpen dat het bruikbaar is voor docenten zonder ICT-achtergrond. Het systeem is gebruikt om leermateriaal over celgroei in reactoren te maken. Dit leermateriaal is geëvalueerd en is erg goed ontvangen.
In Hoofdstuk 5 wordt een virtuele experimenteeromgeving beschreven waarin studenten experimenten ontwerpen om parameters in een gegeven model te schatten. Deze experimenten worden uitgevoerd in een virtueel laboratorium en de resultaten, inclusief een experimentele fout, worden gepresenteerd aan de studenten. Studenten leren hoe ze experimenten moeten ontwerpen, en hoe ze met experimentele fout om moeten gaan. Studenten zouden dit ook in een echt laboratorium kunnen leren, maar in een echt laboratorium zijn er te veel andere dingen die studenten moeten leren, en is slechts weinig tijd beschikbaar. Een virtueel experiment kent deze nadelen niet.
In Hoofdstuk 6 worden alle ontworpen webgebaseerde leermaterialen besproken. Er wordt een overzicht gegeven van de verschillende benaderingen van bioprocestechnologieleermateriaal die mogelijk zijn met webtechnologie. Deze benaderingen worden vergeleken in termen van zes algemene richtlijnen voor leermateriaal en van vier richtlijnen voor bioprocestechnologieleermateriaal. Al het leermateriaal is zeer positief geëvalueerd. Het evalueren van webtechnologie voor het ontwerp van leermateriaal heeft, naast het opgeleverde leermateriaal, ook geleid tot innovatie van het bioprocestechnologieonderwijs en tot waardevolle discussies over bioprocestechnologieonderwijs.

Nawoord

Hier ligt dan mijn proefschrift. Na zes jaar gewerkt te hebben aan het ontwerp van digitaal leermateriaal, eerst als toegevoegd onderzoeker, even via het uitzendbureau, gevolgd door een promotietraject bij Proceskunde en tegelijkertijd ook nog ondersteunend voor andere vakgroepen, is dit het resultaat.

Het pad door deze zes jaar is zeker geen gladde rechte weg geweest. De richting van het onderzoek is een aantal keer gewijzigd, en sommige richtingen zijn na een aantal flinke stappen weer volledig verlaten. Ook de vorm van het proefschrift was lange tijd nog onzeker. Zelfs de wetenschappelijkheid van het onderzoek is onderwerp van discussie geweest. Nadat het aantal publicaties begon toe te nemen verstomde deze discussie snel en kwam ook de huidige vorm van het proefschrift boven drijven.

Gedurende de zes jaar hebben heel wat mensen direct of indirect een bijdrage geleverd waar ik ze voor wil bedanken.

Allereerst wil ik Rik bedanken, die zes jaar lang mijn resultaten, frustraties en ideeën heeft aanhoord, die mij zes jaar lang *de wet van alles* uit mijn hoofd heeft gepraat, die mij zes jaar lang heeft uitgelegd wat een *dangling modifier* is, en die vooral honderden discussies over ons onderwijs en over mijn onderzoek heeft gevoerd. Ook Rob wil ik bedanken, die vele discussies over het onderwijs en mijn onderzoek heeft gevoerd, en die in de concept artikelen van vele formuleringen de eventuele onbedoelde alternatieve betekenissen heeft blootgelegd. Hans en Marian wil ik bedanken voor de vele discussies over het Proceskunde onderwijs. Hans heeft vele artikelen vol ICT terminologie door moeten werken, en heeft daarmee bijgedragen aan de toegankelijkheid van dit proefschrift. Maar bovenal heeft hij zich ingezet voor het mogelijk maken van het promotieonderzoek.

Hylke wil ik bedanken voor de vele discussies over het onderzoek, alle *nerd-talk* (ook Jeroen bedankt!), voor het lunchgezelschap buiten in het arboretum of bij Unitas (waarvoor ook Dione, Sebastiaan, Marleen en alle anderen bedankt!), de buitenlandse reizen die we samen gemaakt hebben en natuurlijk de goede muziek op onze kamer. Tinri wil ik bedanken voor de prettige samenwerking bij het ontwerpen en bouwen van leermateriaal voor

Moleculaire Biologie, en natuurlijk de vele discussies over het wel of niet lopen van onze onderzoeken. Leandro, Bart en Koos wil ik bedanken voor hun bijdragen aan het onderzoek tijdens hun afstuderen, en Koos en Bart ook nog als student-assistent. Zonder Bart was ik Marin misschien wel niet tegengekomen!

Pieter wil ik bedanken voor het samen spelen met het clusterspeelgoed. Ook Remko bedankt dat hij het vertrouwen had dat wij het aanzienlijke budget goed zouden besteden. Gelukkig voor de vakgroep resulteerde ons spelen binnen korte tijd in een operationeel Beowulf cluster.

Verder wil ik alle medewerkers en studenten van Proceskunde bedanken, die met alle sociale activiteiten Proceskunde tot de gezelligste vakgroep van de universiteit maken: borrelen, zeskampen, labuitjes, bomberclonen, spelletjesavonden, brainstormweken, barbecuen, kerstdiners, AIO-reizen (met aansluitend vakanties), Sinterklaas en natuurlijk alle zinnige en onzinnige discussies tijdens de koffie.

Voor het onderzoek beschreven in dit proefschrift, en ook voor het schrijven en publiceren van dit proefschrift, is alleen open-source software gebruikt. Vele honderden mensen hebben, betaald of onbetaald, meegeschreven aan de vele programma's die ik kosteloos heb kunnen gebruiken. Ook deze mensen bedankt!

Naast alle mensen die een grote bijdrage hebben geleverd zijn er natuurlijk velen die een kleine bijdrage hebben geleverd. Zonder al die kleine bijdragen was dit proefschrift niet geweest wat het nu is. Dus, iedereen die een bijdrage heeft geleverd, bedankt!

En soms kom je iemand tegen waarna er wonderlijke dingen gebeuren met jezelf. Dat gebeurde toen ik jou tegenkwam Marin! Jij bent voor mij het belangrijkste resultaat van de afgelopen zes jaar bij Proceskunde, en ik hoop dat dat nog lang zo mag blijven!

Publications

1. van der Schaaf, H., Sessink, O.D.T., Vermuë, M., Tramper, J., Hartog, R.J.M. (to be submitted). *Design of activating digital learning materials to support complex learning objectives.*
2. Sessink, O.D.T., van der Schaaf, H., Beeftink, H.H., Hartog, R.J.M. Tramper, J. (submitted). *Web-Based Education in Bioprocess Engineering.*
3. Sessink, O.D.T., Beeftink, H.H., Hartog, R.J.M. Tramper, J. (accepted for publication). *Proteus: A Lecturer-Friendly Adaptive Tutoring System.* Journal of Interactive Learning Research.
4. Sessink, O.D.T., Beeftink, H.H., Hartog, R.J.M. Tramper, J. (2006). *Virtual parameter-estimation experiments in Bioprocess-Engineering education.* Journal of Bioprocess and Biosystems Engineering, doi 10.1007/s00449-005-0042-z
5. Aegerter-Wilmsen, T., Janssen, F., Kettenis, D., Sessink, O.D.T., Hartog, R.J.M., Bisseling, T. (accepted for publication). *Introducing molecular life science students to model building using computer simulations.* Journal of Computers in Mathematics and Science Teaching.
6. Sessink, O.D.T., Beeftink, H.H., Hartog, R.J.M. (2005). *Database functionality for learning objects.* Technology, Instruction, Cognition and Learning 2: 337 – 357.
7. Sessink, O.D.T., Beeftink, H.H., Tramper, J., Hartog, R.J.M. (2004). *Securing web-based exams.* Journal of Universal Computer Science 10: 145 – 157.
8. Sessink, O.D.T., Beeftink, H.H., Tramper, J., Hartog, R.J.M. (2003). *Author-defined storage in the next generation learning management systems.* In: Proc. International Conference on Advanced Learning Technologies 2003, pp. 57 - 61. Athens, Greece.
9. van der Schaaf, H., Sessink, O.D.T., Vermuë, M., Tramper, J. (2001). *Student aan het werk binnen Wageningen UR virtual consultancy.* Ontwikkeling van digitaal leermateriaal. NPT Procestechnologie 2: 13 – 15.
10. Hartog, R.J.M., de Gooijer, K., Sessink, O.D.T., van der Schaaf, H., Vonder, O. (2000). *Comparing Web based course development with and without a learning environment.* In: Proceedings Webnet 2000 (G. Davies & C. Owen, eds), pp. 240 - 247. San Antonio : AACE.

Curriculum Vitae

Olivier Sessink werd geboren op 30 december 1975 te Venlo. In 1994 behaalde hij het VWO-diploma aan het Thomascollege te Venlo. In datzelfde jaar begon hij aan de studie Bioprocestechnologie aan Wageningen Universiteit, met specialisatie Bioreactoren. Als afstudeeronderzoek voerde hij een onderzoek uit naar de beluchtingsregeling van afvalwaterzuiveringen bij de sectie Meet- Systeem- en Regeltechniek, gevolgd door een onderzoek naar het modeleren van reactoren waarin baculovirussen m.b.v. dierlijke cellen geproduceerd worden bij de sectie Proceskunde. De studie werd afgesloten met een stage aan de University of Queensland in Brisbane, Australia. In september 1999 studeerde hij *cum laude* af. Gedurende zijn studie had hij een eigen bedrijf “Visual Upload” dat freelance website-ontwikkeling leverde. Door deze activiteiten kwam hij in aanraking met Open Source-software, en begon hij ook met het schrijven van de eerste Open Source-programma’s. Sinds die tijd is hij project manager van verscheidene kleine en grote Open Source-projecten. Direct na het afstuderen begon hij voor het Food- and Biotechnology-project te werken aan onderwijsinnovatie met behulp van web-technologie. In 2000 werd dit werk voortgezet met het promotieonderzoek bij de sectie Proceskunde zoals beschreven in dit proefschrift, aangevuld met advies en ondersteuningswerk bij het Wageningen Multimedia Research Centre. In september 2005 is hij als Open Source-specialist bij Open Office in dienst gekomen. Heden werkt hij als Senior Innovatiemanager bij de Defensie Telematica Organisatie.