Design
of
Digital Learning Material
for
Bioprocess-Engineering Education

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

Co-promotoren
Dr. ir. M.H. Vermuë
Universitair docent bij de sectie Proceskunde, Wageningen Universiteit

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

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

Prof. ir. A.J.M. Beulens
Wageningen Universiteit

Dr. ir. A.M.W. Bulte
Universiteit Utrecht

Prof. dr. ir. M.H. Zwietering
Wageningen Universiteit

Dit onderzoek is uitgevoerd binnen de onderzoekschool VLAG

Hylke van der Schaaf

Design of Digital Learning Material for Bioprocess-Engineering Education

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 dinsdag 30 januari 2007
des namiddags te half twee in de Aula.

Hylke van der Schaaf

Design of Digital Learning Material for Bioprocess Engineering Education
PhD. Thesis, Wageningen University, The Netherlands, 2006 – with summary in Dutch

ISBN 90-8504-580-0

Contents

Abstract

With the advance of computers and the internet, new types of learning material can be developed: web-based digital learning material. Because many complex learning objectives in the food- and bioprocess technology domain are difficult to achieve in a traditional learning environment, a project was started to explore the possibilities of digital learning material to address those learning objectives. The material that has been developed, the choices that led to the material and the lessons learned are discussed in this thesis.

In a previous project digital learning material was developed consisting of web-based, linear cases, in which the student is placed in the role of junior consultant. In this role the student is sent to a company to solve a realistic problem. The web-based cases introduce the student to his role, and guide him through the problem by asking the student questions. If the student answers a question incorrect he will get incremental feedback. After giving the correct answer, the student will get an explanation why that answer was correct. Putting the student in a role at a virtual consultancy motivates the student to work actively with the material and to learn about the subject. The kind of adaptive feedback used also helps in activating the student and keeping him motivated. Animations and simulations are very useful in explaining the many models that are used in (bio)process technology and a modular approach makes flexible use of the material possible.

This thesis describes the continuation of this research. One of the learning objectives in the process engineering education, that where not well supported by the available learning material, nor by the previously developed linear cases, was the design of downstream processes. Linear cases with only closed questions of standard types (multiple choice, multiple answer, value, etc.) are not very suitable to teach students how to design, because design is a non-linear, open process. Therefore a design environment, called the DownStream Process Designer (DSPD) was developed. The DSPD allows students to design a downstream process. The student can chain unit-operations together, and tune the unit-operations in order to create a process that operates within the specified requirements. The DSPD is used in a case where the student has to design a process that can isolate a protein from a mixture and purify it to a minimal purity, while staying within an acceptable loss of protein and limiting the amount of waste and costs. By listing the top-scoring designs for each design requirement, a game element was created that greatly stimulated the student and enhanced the learning process. Students liked the ability to create a design and directly see the results of their actions.

Another learning objective that was difficult to reach was the design of models. Students often do not know where to start or how to proceed when they have to design a model. Lecturers, being expert designers themselves, often skip steps in their explanations, because they do those steps unconsciously. A stepwise approach to designing models was created, and in each step the student is supported by digital tools that help him in his design. Two tools of this set, that help the student when entering a model into the computer and that allow him to run simulations with his model, are described in detail. The stepwise approach and the tool set are implemented in a design-oriented case on oxygen transfer that is used in an educational setting. The stepwise approach and the supporting activities for each step

helped the student in keeping an overview over their design process. The students also liked the fact that they where not distracted by collateral problems like detailed mathematics.

Guidelines for the design of digital learning material have been inducted from the development and use of the above described material. The first thing to do when designing new learning material is making an analysis of the learning objectives that this new material should cover. Learning objectives can be classified based on the degree of freedom inherent to the topics and skills covered by the learning objectives and learning material should offer a matching degree of freedom to the student. Topics and skills that require a low degree of freedom in thinking and acting can be effectively supported by simple adaptive systems or cases containing closed questions with dynamic feedback. Skills that imply a high degree of freedom, like design, require open-ended learning materials that offer the student that degree of freedom. Learning material should help the student focus on the learning objectives. Therefore, tasks that are not directly related to the learning objectives should be automated as much as possible.

After making the didactic design of the learning material, the material can be digitised and made available for the students. Of course it is important to keep all the technical constraints in mind during the didactic design of the material. The technical aspects of web-based digital learning material and how some of these aspects contribute to the total investment needed to create digital learning materials are addressed. One of the things that can lead to a reduction in costs is the re-use of components and tools that have been previously developed by the institution itself, or by a third party. Cooperating with other universities to share development costs and gain a larger target audience for the material is a good way to improve the balance of the costs per user and the quality of the learning material.

The following web address gives access to all material that has been designed, developed and used within the scope of this thesis work: http://pkedu.fbt.wur.nl/hylke/thesis.

Introduction

This thesis describes the design and development of digital learning material for the bioprocess-technology education of the Food- and Bioprocess Engineering Group of Wageningen University. The digital learning material is developed to address complex learning objectives in the food- and bioprocess technology domain that are difficult to achieve in a traditional learning environment. This thesis describes the developed material, discusses the decisions that where made during the design and development and concludes with lessons learned from the development, implementation and use of this material.

In this chapter the required competences of (bio)process technologists and the associated learning objectives are specified. An explanation of digital learning material is given and the advantages it has over traditional learning material. Next the preceding project that led to this thesis project is described and discussed and the contents of the thesis are summarised.

Bioprocess technology

In our technology-oriented society, one of the competences that a (bio)process technologist needs to develop is designing and optimising biotechnological production processes. These processes use micro-organisms and enzymes to reach a more effective and cleaner production than possible with traditional production methods. In these processes, a micro-organism or an enzyme converts raw materials that are added to the production process into a usable component, after which the usable component is separated from the reaction mixture. The difference with most other technological fields, like chemical engineering, is that the production process contains a sensitive, unstable, often living, biological component that requires special treatment.

In the bioprocess-technology curriculum at Wageningen University, the design of bioprocesses, the design of mathematical models that describe these processes and the design of experiments to measure parameters for the models are important learning objectives. These learning objectives are complex learning objectives, because design is an open process, where a combination of systematic methodology and personal decisions leads to a unique result. Because of the many choices that have to be made during the design process and the large personal influences of the designer on the design process, there are many possible results and there is no way to say what the optimal result is. To make a design, the bioprocess technologist also needs much knowledge about the subject of the design, which means he needs to have much knowledge, understanding and skills in fields relevant for bioprocess technology, such as microbiology, biochemistry and physics [3].

The fact that design is an open process makes designing a difficult subject to teach to students. As a result lecturers involved in late-curriculum courses in bioprocess design have observed that MSc students have developed insufficient design competences in previous courses. As digital learning material offers many opportunities to support these complex learning objectives it was decided to explore the possibilities of digital learning material.

Web-based digital learning material

Learning material is everything that a lecturer uses to teach his classes, like a book, lecture notes, a collection of problems on paper or a computer program. Digital learning material is learning material that is used on a computer. Web-based digital learning material is digital learning material that can be accessed over the internet.

The use of web-based digital learning material seems to have advantages over the use of traditional learning material. Some of the advantages of digital learning material are the possibility to have the material adapt to the student and the possibility to show animations and simulations to the student. The distribution of digital learning material is very easy, especially when the internet can be used. A student with a valid user name and password can access the material from any internet-connected computer anywhere in the world. From a technical point of view, digital learning material can also very easily be made in a modular way. This means the material can be used in a flexible way across courses or for self-study.

Many people fear that the use of much digital learning material will lead to a situation where students only study behind their own computer and no longer meet lecturers or other students. Of course this scenario is possible with digital learning material, but the same scenario is also possible with traditional learning material such as books. Like a book, digital learning material can be used in a computer class scenario, where at scheduled hours a lecturer is present in a computer room and students can work there, together, under supervision. If this opportunity is offered, most students will use it, because students like to meet other students and lecturers in a learning environment.

Initial work

In 1999 the Food- and Bioprocess Engineering Group of Wageningen University started developing digital learning material within the Food and Biotechnology (FBT) programme. One of the aims of this programme was to develop digital modules in the field of food- and biotechnology in order to support students in achieving a number of learning goals that previously received insufficient attention. Two cases where developed, one about mixing in bioreactors and one about the use of membranes.

Wageningen UR virtual consultancy

The developed learning material is centralised in a website. Students entering this website are put in the role of junior consultant in the virtual company “Wageningen UR Virtual Consultancy”. After choosing a topic the student can decide to go to the on-line library, or to go to the management of the virtual company and obtain an assignment. After the student receives an assignment from the management of the virtual company, a series of questions lead the student to the solution of the problem. This virtual environment and the tasks the student receives provide a satisfactory balance between authenticity requirements [1] and the practical constraints in higher bioprocess-technology education. The virtual environment and the built-in guidance provide sufficient handles to capture the student's attention, to make the student aware of the relevance of the problems for bioprocess engineers, and to foster confidence and satisfaction [2].

This set-up of cases together with a library was chosen because it is suitable for different styles of learning [8]. A student who prefers to learn from theory can start by learning the material from the on-line library and then test his knowledge by going through the case study. A student who prefers to look at the theory from an applied point of view can start with the case study and access the library when he needs to. The learning objectives covered are the same for both approaches.

Interactive cases

One of the unique aspects of the developed learning material in the digital cases is that when a student gives a wrong answer to a question he receives specific feedback on his answer and a list of general hints to lead him in the right direction. The specific feedback catches many of the standard mistakes that most of the students make. If the student gives repeatedly incorrect answers, the list of general hints grows and the hints get more specific. This avoids the situation that the student gets stuck and looses his motivation because he can not find the correct answer to a question. This incremental feedback also assures that each student receives the feedback he needs; some students will only need a few hints, while other students can get the help they need.

If a student gives the correct answer to a question he receives feedback specifying why the answer was correct, so the student can check whether the line of reasoning that led him to the correct answer was indeed correct.

The lecturer does not need to address most standard problems that students run into, because these are now caught by the feedback in the material, so he is more available to address in-depth questions.

On-line library

All relevant process-technology theory is available in the on-line library. The basic tools for process technologists, like setting up balance equations, transport equations and dimension analysis only need to be presented once in this library and are always available. In the theory about topics that use these basic tools, a link is given to the relevant tools. By using one large, modular library and not a separate library for every case study duplication is avoided. It also makes it easier to make consistent use of symbols and definitions. One library also gives an overview over all available theory and this makes the relation between different process-technological topics more clear. Where possible the library uses animations and simulations to give better insight in the subjects.

The developed material is arranged in modules, so a module can be used in multiple courses and a lecturer can use multiple modules in a course. The on-line library is also arranged in modules and this gives the possibility to only show certain parts of the library to certain students [7]. Students working on an introductory course will only see the introductory pages of the relevant topics, while students working on advanced courses get all pages in the library.

Evaluation of the interactive linear cases

The Mixing and Membrane modules developed in the FBT project can be used as independent learning material, but are also actively used as supplement in existing computer classes. In these classes students are stimulated to work in pairs on a module, while a lecturer is present to respond to questions. Both students and lecturers where very positive about the quality of the material and the way it was used in classes. The lecturers observed that students where very actively engaged with the material. The lecturers themselves could focus on the more complex questions that students had as the basic questions where answered by the learning material.

The development of this initial digital learning material by the process engineering group, mainly focused on linear cases containing closed questions, to support learning objectives such as knowledge and understanding of particular process-technology subjects.

From this learning material we learned that putting the student in a role at a virtual company helps to motivate the student to work actively with the material and learn about the subject, though the extra step of first introducing them to a "virtual consultancy" and then sending them to another virtual company was unnecessary. Adaptive feedback also helps in activating the student and keeping him motivated. Animations and simulations are very useful in explaining the many models that are used in (bio)process technology. The modular approach made flexible use of the material possible.

Most students prefer to use their books over the on-line library to learn about the theory. The interactive parts, animations and simulations in the library however where appreciated by the students. Therefore, in further material, the library was only used to explain those things that can be better explained using digital techniques like models and animations.

Linear cases with closed questions are not very suitable to support non-linear, open learning objectives like design and modelling. The next challenge was to design and develop digital learning materials that supported the achievement of these learning objectives in process engineering contexts.

Aim and content of this thesis

This thesis describes the continuation of the initial work. The aim was to design and develop digital learning material for more complex technical learning objectives, such as design of processes and mathematical models, and to extract guidelines for the design of digital learning material from the experience gained from this development.

Chapter two describes the design and development of a design environment for downstream processes for BSc students [4]. This design environment presents the student with a fermentation broth containing a product among many other components. The goal for the student is to purify the product so it conforms to the given specifications. To do this, the student can create a chain of unit operations by adding and configuring unit operations. Students like this way of applying their knowledge into a design, and seeing how their design decisions make a difference for the final product. After completing the design, the student is presented with the top-scoring designs in the fields of costs, yield, purity and waste production. These top-scores motivate many students to re-evaluate their design in order to increase their score. This game element activates the students and stimulates the student's learning process.

Chapter three describes the development of two tools that can be used in a step-wise approach to the design of models [5]. These two tools can be used by a student to build a model, and to run simulations with the model that was created. Building the model is done in a structured way, by combining equations that have been pre-defined by a lecturer. Students liked this way of composing a model, but they missed a way to get a semantic overview over their model, and an easy way to check the units used in their model. In further developments these problems where addressed and more tools where developed.

Chapter four describes the design and use of a set of tools for the systematic design of models [6]. This set includes improved versions of the two tools described in chapter three. A stepwise approach guides the student through the analysis of the problem, the model composition and model evaluation steps, and the tools support the student in each step. The stepwise approach and the tool set are implemented in a design-oriented case on oxygen transfer and the use in an educational setting is evaluated.

In chapter five a set of guidelines, that are derived from the material are described and illustrated with examples of the material developed by the Food- and Bioprocess Engineering Group. These guidelines can be helpful when designing digital learning material. They describe how learning objectives can be classified based on the degree of freedom inherent to the topics and skills covered by the learning objectives. Topics and skills that require a low degree of freedom in thinking and acting can be effectively supported by simple adaptive systems or cases containing closed questions with incremental feedback. Skills that imply a high degree of freedom, like design, require open-ended learning materials that offer the student that degree of freedom.

The final chapter describes the technical side of the development of digital learning material. It explains the basics of the technology used in web-based digital learning material and how the technical aspect of the creation of digital learning material factors into the total investment needed for the design of the learning material.

The following web address gives access to all material that has been designed, developed and used within the scope of this thesis work: http://pkedu.fbt.wur.nl/hylke/thesis.

References

1. Honebein, P. C., Duffy, T. M., Fishman, B. J. (1991) “Constructivism and the design of learning environments: Context and authentic activities for learning”, In *Designing environments for constructive learning*, Vol. 105, pp. 87 - 108, Berlin: Springer.

2. Keller, J. M. (1987) “Strategies for stimulating the motivation to learn”, *Performance and Instruction Journal*, 26(8), 1-7.

3. Meriënboer, J. J. G. van (1997) *“Training complex cognitive skills. A Four-Component Instructional Design Model for Technical Training”* Educational Technology Publications, Englewood Cliffs, New Jersey, ISBN 0-87778-298-9.

4. Schaaf, H. van der, Vermuë, M., Tramper, J. & Hartog, R. (2003) “A design environment for downstream processes for Bioprocess-Engineering students”, *European Journal of Engineering Education*, Taylor & Francis, vol. 28, nr. 4, pp. 507-521.

5. Schaaf, H. van der, Vermuë, M., Tramper, J. & Hartog, R. (2006) “Support of Modelling in Process-Engineering Education”, *Computer Applications in Engineering Education*. Vol. 14, No. 3, pp. 161-168.

6. Schaaf, H. van der, Vermuë, M., Tramper, J. & Hartog, R. (2006) “A digital tool set for systematic model design in Process-Engineering education”. *European Journal of Engineering Education*. Vol. 31, No. 5, 619-629.

7. Valcke, M., Kirschner, P., Bos, E. S. (1999) “Enabling technologies to design, produce and exploit flexible, electronic learning materials”, in J. v. d. Akker, R. M. Branch, K. Gustafson, N. Nieveen & T. Plomp (Eds.), *Design approaches and tools in education and training* (pp. 249-263). Dordrecht/Boston/London: Kluwer.

8. Vermunt, J. D. H. M. (1992) *“Leerstijlen en sturen van leerprocessen in het hoger onderwijs”*. Amsterdam: Swets & Zeitlinger.

Downstream processing design

A bioprocess engineer should have at least a set of basic design skills. Bioprocess design is a complex cognitive skill, which should be trained in every year of an academic bioprocess-engineering curriculum. However there is little existing learning material to support the initial training of design skills early in the curriculum. For this reason a web-based DownStream Process Design environment has been developed called DSPD. This article describes the design criteria for the development of this design environment. It describes the design environment itself and it gives an impression of the use of the design environment in a course for first-year students.

Published as:
Schaaf, H. van der, Vermuë, M., Tramper, J. & Hartog, R. (2003) "A design environment for downstream processes for Bioprocess-Engineering students", *European Journal of Engineering Education*, Taylor & Francis, vol. 28, nr. 4, pp. 507-521.

Introduction

The Food and Biotechnology (FBT) research programme is a research programme on design of digital learning material. The programme was initiated at Wageningen University in September 2000 and currently counts 6 large projects and a number of smaller projects. The intention of the FBT programme is that the digital learning material will be used by students of Wageningen University, but also by students from many other institutions. It is expected that the use of the learning material outside Wageningen University will lead to constructive criticism from students and staff from other universities. The ensuing improvements will raise the quality of the learning material and thereby will be beneficial for both students of Wageningen University and the external ones. Furthermore sharing of web-based learning material will be one step on the path to internationalisation of higher education [1]. This adds a new perspective to the use of information and communication technology in engineering education [2]. Within the FBT programme a four-year research project on the design, development and use of web-based digital learning material for food- and bioprocess-engineering education is carried out. Material that has been developed in this project has been used at Wageningen University, École Polytechnique Fédérale de Lausanne (EPFL) in Lausanne (CH), the Technical University of Lodz (PL) and is accessible to any other university in the world [3].

A bioprocess engineer should have at least a set of basic design skills. Textbooks in the fields of process engineering and biotechnology however, do not offer sufficient information about design processes nor do they offer students the possibility to elaborate on design knowledge. There are a number of process-engineering textbooks having the term "Design" in their title [4, 5, 6] but these textbooks mainly present knowledge about typical process operations, conceptual tools like balance equations and typical computational procedures. In fact no learning material has been found that supports all aspects of training design skills. To comply with the need of industry for competent bioprocess designers, Wageningen University has inserted a set of instructional activities targeted at design competencies in the process-engineering curricula.

This article will start with a definition of design and a description of how ideas about learning to design have been implemented recently in the bioprocess-engineering curriculum at Wageningen University. Next it will elaborate which skills are essential in design processes in general and in process engineering in particular. It will then explain why existing design environments do not satisfy these requirements. For this reason a server-based DownStream Process Design environment has been developed called DSPD. The paper describes the design environment and evaluation results of how the DSPD is used in the early stages of the curriculum and how students respond to the new possibilities, which are offered to them.

What is design

On the one hand there are many definitions of design, on the other hand the list of publications about design and about design education that avoid to commit to one specific definition of design, is very long [7, 8, 9, 10, 11, 12]. This shows how ubiquitous the concept of design is and at the same time how difficult it is to grasp all aspects of design to everybody's satisfaction in just a few lines. The following quote implies a very broad definition of **design:**

> *'Everyone designs who devises courses of action aimed at changing existing situations into preferred ones.'* [10, p 111].

A slightly more specific but still very abstract definition of design is given by Dym and Little:

> *Engineering Design is the systematic, intelligent generation and evaluation of specifications for artefacts whose form and function achieve stated objectives and satisfy specified constraints. [12]*

Although clearly definitions of *design* are inadequate in general [9], it still may be good to set the stage for a short paper by selecting one of them. For this paper the following definition has been adopted:

> '*Design is an open process that is both object and context dependent. Within this process, a combination of methodical steps and personal decisions leads to the realisation of a material or immaterial product or process.' [11].*

Design is an open process, there is more than one way to look at a problem, there is more than one good solution and it is not possible to determine one best solution. Design is an object-dependent process. How you design depends on what you are designing. Design is a context-dependent process. The design depends on where and how the product or process is going to be used. Design is about making decisions. When facing a design problem, there are in theory an infinite number of possible answers and it is impossible to make an evaluation to say which answer is the best. There usually is much irrelevant information available and much relevant information missing.

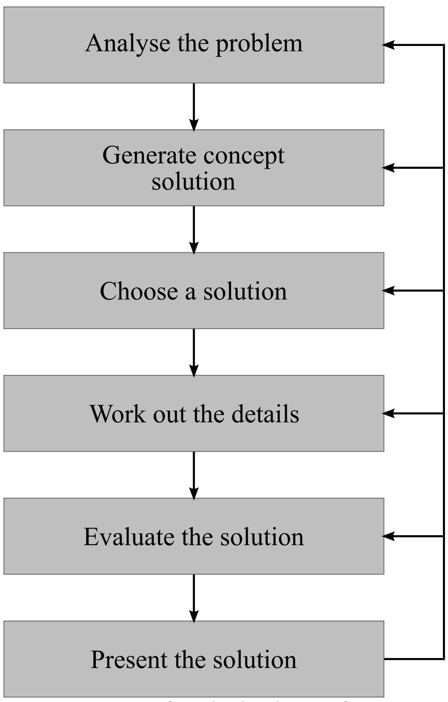


Figure 1: A set of methodical steps for designing.

For most design processes there is a standard set of steps that can be used to structure the design process. The details of these methodical steps depend on the object and context of the design and can be different for each situation. For the design of a biotechnological process the set of methodical steps given by Jones can be used [7]: Analyse the problem, generate concept solutions, choose a solution, work out the solution in detail, evaluate the solution, if necessary, change the design according the findings and present the solution (Figure 1). This is the approach we want our students to experience and to become familiar with.

Design in process-engineering education

Because designing is a major learning objective of the bioprocess-engineering curriculum of Wageningen University, designing is introduced early in the programme. During the first year of their study, students in bioprocess engineering are introduced to the design of downstream processes. The main function of a downstream process is to separate a product from a mixture of components. In this context designing a downstream process usually means choosing the unit-operations, ordering those unit-operations and choosing the operational settings for those unit-operations.

When looking at teaching the basics of the design of downstream processing to first year students, one can identify several skills and types of knowledge that the student must acquire, which are important for the design process. On the one hand the student has to build up knowledge of unit-operations (filter, ion-exchanger, etc.) commonly available for the configuration of a downstream process, on the other hand the student has to learn how to order and configure those unit-operations to get the desired results.

While specific knowledge of the different unit-operations is important when designing a downstream process, it's not necessary for a student to have all possible knowledge of unit-operations before he is able to design a functioning downstream process for a certain product. When a student discovers during the design process that he lacks some necessary knowledge he is motivated to acquire that knowledge. This motivational aspect is an important reason to offer the necessary information about unit-operations just in time during the design activities of the student.

Requirements for an environment that supports initial training in the design of downstream processes

To facilitate the learning process for first-year students, there is a need for an easy-to-use environment for designing downstream processes. We defined a set of requirements for this environment that we will first list and then explain. The main requirements for this environment are that it should:

1. Offer the possibility to insert, move and remove unit-operations.
2. Provide easy introduction for novice designers to the concept of unit-operations.
3. Offer the possibility to adjust the control parameters of unit-operations.
4. Directly show the consequences of any change in the design.
5. Limit the cognitive load for the student.
6. Enable personalised feedback.
7. Be directly accessible for any authorised student on any computer.
8. Have a modular design that can be reused in different situations.

The first six requirements are inspired by, or derived from theory and assumptions about how students learn complex cognitive skills such as design, or about the typical problems of novice designers [13, 14, 12, 15].

Because an adaptive-content framework for web-based learning had been developed already at the process-engineering department, the personalised feedback requirement for the DSPD resolves in the technical requirement that the DSPD should have an interface with the adaptive learning environment of the process-engineering department.

The last two requirements are derived from general principles on system design (modularity) and from the goals that have been set in the FBT research programme (the intention to offer world- wide access with minimal administrative load).

An easy to-use-design application is desired, so that first year students do not have to spend much time learning the application before they can start learning downstream processing. In other words, extraneous cognitive load should be minimised (requirement 5). Furthermore, the application should contain the most common unit-operations used in downstream processing. The student should be able to play with the unit-operations to get an idea of what a specific unit-operation does and how it works. Elaborating knowledge by running simulation models of, in this case, unit-operations can be an effective way to support the learning process if the right accompanying measures are taken to structure the students' use of the simulation models [16]. This means that the student has to be able to change the settings of a specific unit (requirement 3) and that he directly sees the effects those changes have on the performance of this unit and the effects those changes have on the performance of all units following this specific unit (requirement 4). A student also has to have the option to get an overview of the entire downstream process that summarises the performance of the different units and of the total design, so the student can easier identify bottlenecks in his design [14].

Finally, it is deemed important that the student gets feedback on the overall design he has made, for instance when the student orders unit-operations in a way that does not make much sense, but is not impossible either, such as creating a cascade of identical centrifuges.

Existing process design environments do not satisfy our design requirements

There are several existing design environments that are used to design process schematics, like Aspen Plus© and SuperPro Designer©. These programs are designed to allow the design of almost any possible production process, and include complete simulation, documenting and scheduling tools and more. Because of this the user already has to be familiar with process design in general, with the specific unit-operations he wants to use, and with the design environment itself before being able to create a functional design in one of these design environments.

These existing design environments are also too complex to use for a student who has only just been introduced to downstream processing and isn't even aware of the unit-operation concept. To add one unit-operation to your flow sheet in SuperPro Designer©, you first have to select the unit-type, then click on your worksheet to add a unit of that type.

After adding your units you have to manually draw the streams between the units of which there can be many for a single unit-operation. The manual of SuperPro Designer© needs seven pages to explain the process of adding and connecting a unit-operation. After adding unit-operations the user has to specify in detail what the contents of each stream are, what happens in each unit, what the separation efficiency is for each component in the product stream, etc. The total manual of SuperPro Designer© is well over a hundred pages. Like SuperPro Designer©, Aspen Plus© is a complete design environment for industrial use and not easy to learn. Learning to work with these complex programs would require an intensive course on its own.

It is possible to design a downstream process using these design environments by adding, and moving unit-operations and changing settings of those unit-operations, but they do not provide an easy introduction to downstream processing for novices by limiting cognitive load or directly showing the results of a change.

The commercial design environments are also not available on every computer a student has access to. Furthermore they cannot easily be implemented in a web-based course in a way that allows automatic feedback on the design the student has made.

Description of the DSPD

When a student opens a page with a design exercise for the first time, he is confronted with the starting situation of the process he has to modify. The most simple form of this starting situation would be a reactor with some content, with the assignment to isolate one of the components from the reactor. A more complex starting situation could be a complete process with the assignment to identify and 'fix' a bottleneck somewhere in that process.

Given the first assignment, to design the process for isolating a component from the reactor, the student can then start adding unit-operations between the reactor and the endpoint of the process chain. When a unit is added the initial settings of this unit will allow as much components as possible to pass. The student will have to tune the unit to his liking. The result of these changed settings are directly available and based on these results the student can decide what further changes to the settings should be made or what other unit-operations should be added.

There are no restrictions to the order of unit-operations. However, the results and feedback generated will warn the student if a design is illogical. For instance, if the student places an ion-exchange unit that cannot handle a flow containing solid components behind a unit that outputs a flow that contains solid components, the ion-exchange unit will be clogged and generate an empty product stream. The input stream will be redirected to the waste stream.

How to try it yourself

A link to the downstream process designer can be found on the content showcase on the FBT web site (http://www.fbt.wur.nl/). Use the link "Try the Downstream Processing Design Case".

The product stream

The DSPD has to be modular (requirement 8) and because unit-operations can be added in every possible order, it is very important to clearly define what information is passed from one unit-operation to the next. In a linear downstream-process chain, the product stream is passed from one unit-operation to the other.

The definition of the product stream must contain all parameters that are relevant for the isolation process. Some of these parameters refer to the liquid, like density, viscosity and the type of liquid. Some refer to the substances or components in the liquid and in case the components are cells, they may have substances inside them that are released when the cells are broken.

For example, for filtration, it is important to know the size of the components in the product stream as this determines if a component can go through the filter or not. For an ion-exchange unit, the iso-electric point of a component determines if the component is bound to the ion-exchange column. The size of components is also important in an ion-exchange unit, as components that are too large will block the column. The properties of components that have to be known can be different for each unit-operation.

In this list of components in the product stream, not all parameters make sense for all components. An ion does not have an iso-electric point and it is not possible to break an ion in pieces like a cell, so it has no parameter that describes how strong the ion is. The list is also extendable, if a new type of unit-operation is defined that requires more information about a component, this information can be added to the definition of the starting product. All existing unit-operations will ignore the new parameter so the new unit-operation can use it.

User interface

The user interface is what the student sees and has to work with. Figure 2 is an example of the downstream process designer used in a case. The Downstream Process Designer has to display a lot of relevant information for the student. It is important that the student isn't overwhelmed with information, but at the same time he has to be able to find the information he needs [14].

Each unit-operation in the process stream shows the following fields:

- Unit-operation properties: The name of the unit-operation, icons to move, update or delete the unit-operation. The properties of the unit-operation that the student can change.
- Unit image: A graphical representation of the unit-operation.
- Output / waste: The listing of the output and waste streams generated by the unit-operation.

The unit-operation properties are specific for each unit-operation. Some unit-operations have more properties than others. There is one property that every unit-operation has: the name the student wants to give to the unit-operation. The storage vessel unit-operation (called endpoint in Figure 2) has only this standard field, while, for example, a disruptor also has fields for setting the pressure drop over the disruptor and the number of times the stream is passed through the disruptor. If the student is allowed to make modifications to the process chain then every unit has the option to remove that unit from the process chain. Between every two units an option is available to add a unit-operation between those two units.

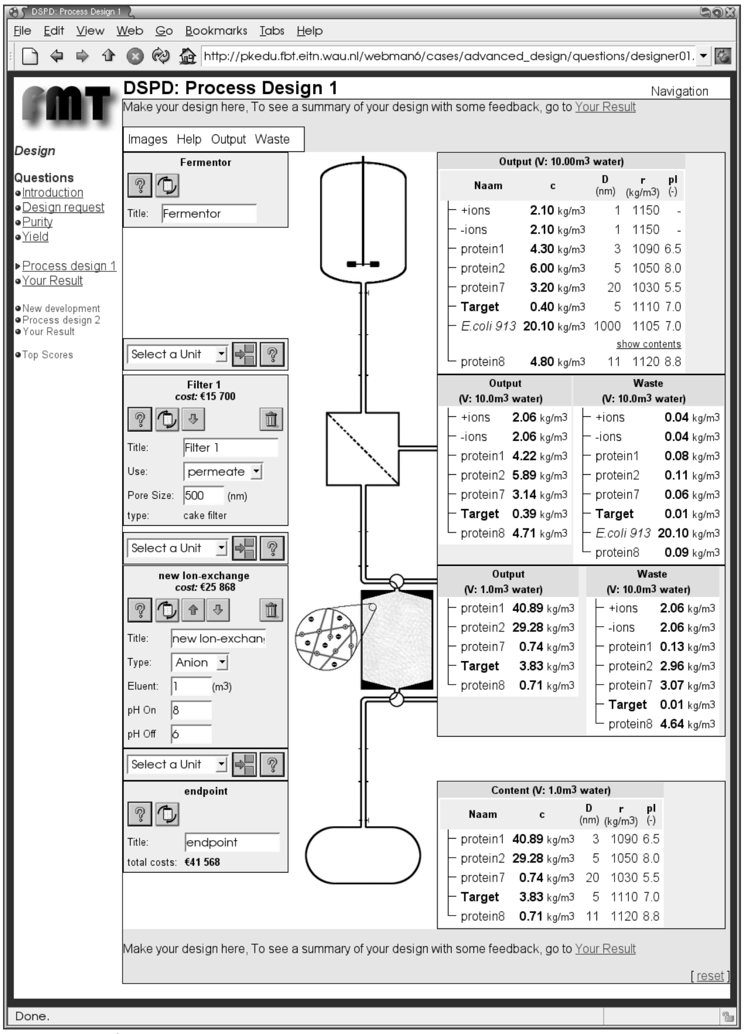


Figure 2: The DownStream Process Designer in use in a case.

The image of the unit mainly serves as a quick way to recognise the type of the unit-operation. For some units the image also gives visual feedback on a setting of the unit. For a filtration unit, it shows whether the permeate or the retentate of the filtration step is used for further processing.

The output/waste lists of each device describe the type and volume of the streams and the components in the streams. For each type of component the name and concentration is given and there is a field that can be used by a unit-operation to give specific information, e.g. the run-time of the component through a gel filtration. The reactor and endpoint units do not have a 'waste' stream. The extra screen space available as a result of this is used to list other properties of the components, like the density and iso-electric point. The contents of cells can also be shown next to the reactor and endpoint units, but the student can hide the contents of the cells to save more screen space. As a result, the student does have direct access to all information about the components, but this information is not repeated for every unit-operation.

Help function

The DSPD module has a built-in help function. This help function contains a short explanation of how the DSPD works and for each individual unit-operation it explains what the unit-operation does and what settings the user can change for that unit-operation. For each unit-operation there is also a demonstration of the unit-operation. The demonstration uses the DSPD itself, with a process consisting of a reactor with a suitable demonstration content, the unit-operation itself and a storage vessel. In this demonstration the user can play with all the settings of that unit-operation with the restriction that the user cannot add or remove any other unit-operations.

Architecture

The DSPD is a server-side program. When a student works with the DSPD, the program is executed on the web-server. The student only sees the result of the processing displayed on his local computer. This system has several advantages. First, because all data are processed on the server these data can easily be stored on the server. So if a student does some work and later logs in from a different computer, he can directly continue where he stopped. Te data can also be linked to a student model to track for instance student progress or to add a

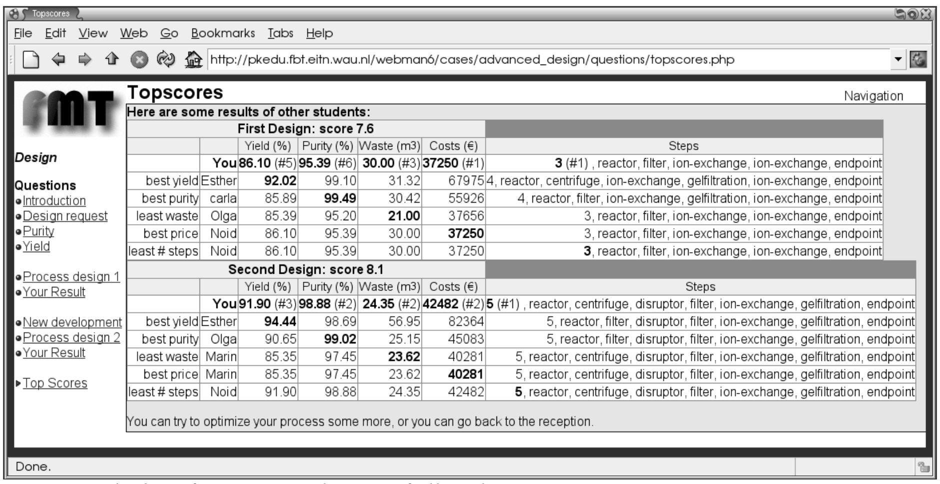


Figure 3: The list of top-scoring designs of all students.

competitive element where the student can compare his results with the results of other students, as seen in figure 3.

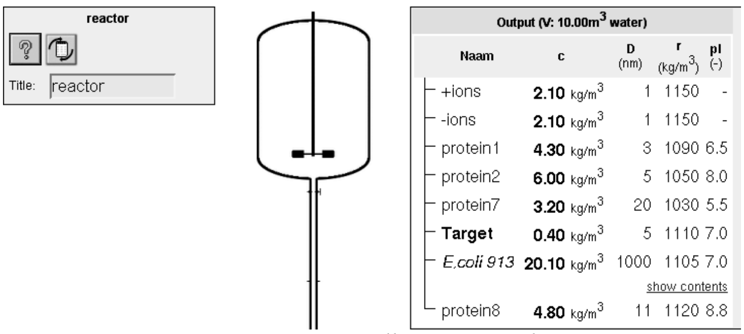
Second, because all complex processing is done on the server side, there is no need to install any additional software on the client side. In many universities the computers that are available to students are very restricted in what the student can and cannot do. Installing software is something that is often impossible for the student. Third, because the output of the DSPD is standard HTML, it can be viewed with any browser the student prefers to use on any operating system. Especially if the material is also used at other universities we do not have control over what the student has available on the client computer.

Because all a student needs to access web-based learning material like this is a user-name and password, it is also very easy to make this material available to, for instance, other universities. Several web-based applications developed in Wageningen University for process engineering are being used at the EPFL in Lausanne and the Technical University of Lodz.

Server-side processing also has some disadvantages. The user interface is limited to the possibilities of standard HTML. Also, for every action the user takes, a request has to be send to the web-server and the appropriate response has to be send back. If the user is on a slow connection this process can be slow.

Use of the DSPD in Bioprocess-engineering Education

There are several ways to use the DSPD in education. It can be used to illustrate the working of a device in a lecture about the theory of that device. In our education, the DSPD is used in a case, where the student is put in the role of junior consultant of a consultancy firm. In this role the student is given the assignment of designing the downstream process for a new product. The case starts with an introduction with some questions. After this introduction the student is given the task of designing the downstream process, based on specifications he gets from the 'research department' of the company and with set requirements for the purity of the product, the total amount of product recovered and a budget. After making a successful design, the student gets some new data from the research department and is asked to change his design for the new situation. This case can be used in a tutorial, where the students work alone or in pairs on the case, while a lecturer is present



Naam	c	D (nm)	r (kg/m³)	pI (-)
+ions	**2.10** kg/m³	1	1150	-
-ions	**2.10** kg/m³	1	1150	-
protein1	**4.30** kg/m³	3	1090	6.5
protein2	**6.00** kg/m³	5	1050	8.0
protein7	**3.20** kg/m³	20	1030	5.5
Target	**0.40** kg/m³	5	1110	7.0
E.coli 913 (show contents)	**20.10** kg/m³	1000	1105	7.0
protein8	**4.80** kg/m³	11	1120	8.8

Figure 4: A reactor containing water, cells, proteins and ions.

to answer any questions the students might have. The case could also be used as basis for a group discussion with a tutor present, or by a student with an internet connection at home, as preparation for a lecture or exam.

At Wageningen University the case is used by first-year students in Bioprocess Engineering. These students have little knowledge about downstream processing or the unit-operations used and have no design experience. The case is used in the course Process Engineering. The learning objectives of this course, which this case helps to achieve, are

- Design of a flow sheet for a typical biotechnological product.
- Recognise the most common unit-operations.
- Describe the function of the unit-operations.
- Describe how they work.
- Order unit-operations in a flow sheet.

When the user is building a new downstream process, and has to decide what unit to add next, he has to choose a unit based on the composition of the mixture offered. To make this decision the student has to check the properties of the components in the mixture and find out which unit operations will separate the components based on these properties. In figure 4 we can see that E.coli 913 has a diameter that is very different from the other components in the mixture. To remove E.coli 913 the student could use a unit operation that separates components of different size, like a filter or gelfilter.

After adding a unit operation, the settings of that unit have to be changed to achieve the desired separation. A filter for example (Figure 5) separates large components from smaller components and the student has to select if he wants to use the large components (the retentate) or the smaller components (the permeate). The student also has to choose the pore size, as that is the control parameter that determines what components can pass through the membrane and what components are blocked.

An example of a situation where the order of two units makes a difference, is when a solution with large volume is passed through both an ion-exchange unit and a gelfiltration unit. In the example in figures 6 and 7 the starting volume of the flow is $10m^3$. When the gelfiltration unit is placed first (Figure 6), a large gelfiltration unit is needed to get a good separation. The flow coming out of this gelfiltration step will still be large and the gelfiltration step will produce a lot of waste.

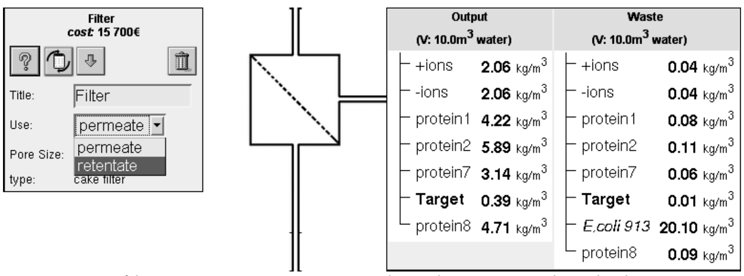


Figure 5: A filter separates components based on size. Either the larger or the smaller components can be recovered for further processing.

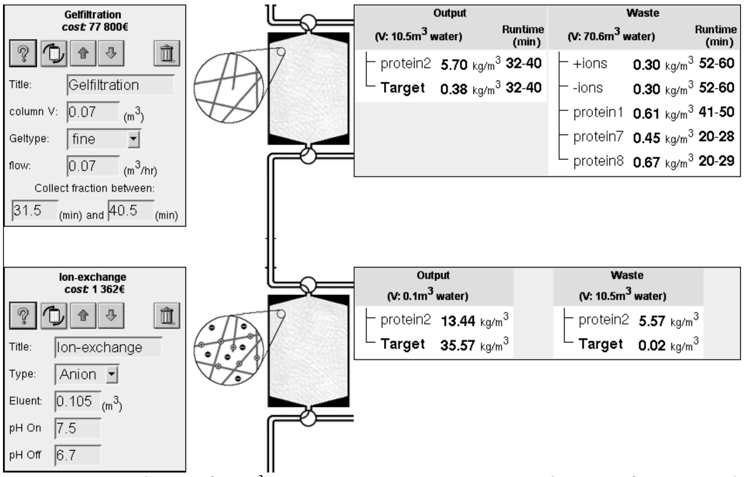


Figure 6: A volume of 10m³ water containing proteins and ions is first treated with a gelfiltration unit and subsequently with an ion-exchange unit.

When the ion-exchange unit is placed first (Figure 7), the flow from the ion-exchange unit into the gelfiltration unit will be much smaller then the original $10m^3$. The gelfiltration unit can thus be much smaller in this situation, resulting in a much smaller waste volume and a cheaper process. The student should realize that an ion-exchange unit can be used both for purification and for concentration of the product flow while gelfiltration is only suitable for separation.

The course is problem oriented, one of the assignments in the course is to solve the case that was built around the DSPD. After solving the case the students have to make a report about their solution to the problems in the case.

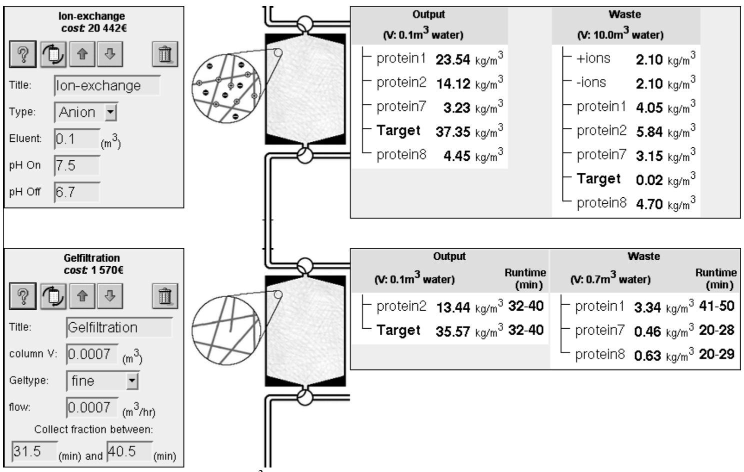


Figure 7: A volume of 10m³ water containing proteins and ions is first treated with an ion-exchange unit and subsequently with a gelfiltration unit.

The last page of the case shows how the students' design compares to that of the others in the fields of product purity, product recovery, costs, amount of waste and number of units used. It also gives an overview of who made the design with the highest product purity, the highest product recovery, the lowest costs, the least waste and who used the shortest process. This introduces a competition element and inspired some students in this first try-out to try and get the highest score in as much fields as possible.

Evaluation by students and lecturers

Currently the DSPD has been used by about 40 students in 2 groups. For both groups the DSPD was embedded in a case as described above. Both in order to improve the DSPD as well as in order to improve the way in which the use of the DSPD is embedded in the bioprocess-engineering curriculum evaluations have been carried out. First of all the students were observed carefully while they were working with the DSPD. Actually one of the most striking aspects is the intense concentration and on-topic discussion that can be observed in a classroom with students working - mostly in pairs - with the DSPD. Initially the students need about 15 minutes to find their way around in both the case environment as well as in the DSPD. Once they know how to navigate through the case and recognize the navigation logic in the DSPD they were all actively engaged with the DSPD. Furthermore it was observed that the option to compare your own results with results of other students which was improved after the first group, clearly led students to reconsider their first solutions. This resulted in activities to improve on their first solutions and evaluating discussion between different groups of students.

After the case the students in the first group had to write a report and they were asked to fill in an evaluation form and with the students in the second group an in depth group interview was carried out. The main results of these evaluations are described here.

The case for the first group was relatively "open" and so where the learning goals for this case. From the reports of the students in the first group it was concluded that the case should be more structured and less "open" at least for students in an early stage of their study. The students in the second group indicated that both the learning goals of the case as well as the assignments in the case were clear. Furthermore they felt that indeed they did achieve these learning goals due to their activities with the DSPD. This experience is in coherence with the conclusion in Jong and Joolingen 1998, that learning materials like computer simulations and activating learning materials like the DSPD should be implemented carefully within a course. If it is not sufficiently clear to the students what they are supposed to do with the material, they will not benefit optimally from their experience with the material [16].

Almost all students indicated that they found the DSPD challenging and very much fun to work with. This confirmed essentially the impression of the lecturers during their observations.

The competitive element was considered positive by the second group and these students (except for one) told that they really had a strong desire to get a better score then the others. Apart from the score, the possibility to compare their results with the other students stimulated the students to take a better look at their own designs and a desire to better understand which settings or orderings of operations could lead to an improvement of their designs.

Students also really liked the fact that they felt they where working on something real instead of some theoretical academic problem. In addition, students also liked the fact that there is no risk attached to mistakes, something that is usually not the case in real life. They could try the things they wanted to try, without getting penalties if it didn't work. Students were satisfied with the balance between the requirement that a mistake must be corrected before the student is allowed to continue (which forces the student to understand fully what he is doing) and the ease with which a mistake can be corrected once the student does indeed understand what he is doing. Indeed the adagium that "Significant learning often occurs in a setting where it is safe to try." [17] was one of the guidelines during the development of the DSPD.

The lecturers where very positive about the activating and motivating properties of the DSPD. In particular the observations during the course of the student pairs actively discussing the subject had impressed the lecturers. However, they also had to conclude that some students in the first group tried to put as little effort in the course as possible. The conclusion of the lecturers was that first-years students especially need well structured assignments to make sure they are introduced to all aspects of the design of downstream processes. These observations and the reports of the students in the first group led to a more structured case for the second group and the option to also compare results in addition to listing the top scores.

Based on these results the lecturers have decided to deploy the DSPD in more instructional situations and to increase the use of the DSPD in bioprocess-engineering education.

Conclusions

For education of first-year students in bioprocess engineering there was a demand for an easy to use, easy-to-distribute downstream-process design environment. Existing design environments are not suitable to support the first stages of learning how to design a downstream process. The DownStream Process Design application described here is web-based, runs on the web server and is therefore accessible from any internet-enabled computer with a web browser. In fact web-based applications that have been developed for process engineering within the FBT programme are being used already at EPFL in Lausanne (CH) and the technical university of Lodz (PL).

The DSPD supports the design of a single linear process-chain. A downstream process starts with a reactor-type operation and ends with a storage vessel. Unit-operations of any available type can be inserted at any point between the reactor and the final storage vessel. There is virtually no restriction to the number of operations between the reactor and storage vessel. An operation takes the output of the previous operation and generates an output and a waste stream from its input (in => out + waste). The student can also create any order of operations, but will soon find that some sequences of operations do not make sense.

The application has a graphical user interface that is easy to use for students that are not yet familiar with the subject. It takes students less then half an hour to get used to the DSPD interface, after that they are exploring the different unit-operations and searching for the best combinations. For some students it takes some time before they realise that 'just clicking around' is not going to get them a working design. Once they realise that if they really put some thought into the design it will give a better result, their motivation increases.

Two groups of students have now used the DSPD. The experience with the first group unveiled the need for a more structured case and more detailed assignments to guide the students. The first year students that used the DSPD needed more guidance than a single assignment offers. Most did not continue to search for alternative solutions after finding a working solution.

For a second group a case that provided more structure was offered. Furthermore a feature that enables the students to compare their design with the design of others was added. Evaluation with a second group showed that these measures invited the students to reconsider their first working solution and to more involvement in the task.

The teachers who implemented the DSPD were very positive about the DSPD in the sense that they are convinced that the DSPD supports the learning goals of the course and motivates the students.

Since the original publication of this paper the case containing the DSPD has been used in education for several years. A question was added to the exam to test for the learning objectives associated with downstream process design. In 2005 and 2006 a total of 65 students participated in the exam. The average score for the total exam was 6.9 out of 10, while the average score for the question about downstream processing was 7.5 out of 10. From this the lecturers concluded that the students had indeed reached the set learning objectives. In the evaluations the students gave the DSPD case an overall rating of 3.9 out of 5.

References

1. Irandoust, S., Sjöberg, J. (2001) International Dimensions: a challenge for European engineering Education, *European Journal of Engineering Education* VOL. 26, no.1, pp 69-75.
2. Brandt, D., Henning, K. (2001) Perspectives of information and communication technologies for engineering education EUR.J.ED., vol 26, NO.1,63-68.
3. http://www.fbt.wur.nl/, go to the content showcase, follow Try the Downstream Processing Design Case.
4. Riet, K. van't, Tramper, J. (1991) Basic Bioreactor Design (New York: Marcel Dekker Inc.).
5. Asenjo, J.A., Merchuk, J.C. (1994) Bioreactor system design (New York: Marcel Dekker Inc.).

6. Cabral, J.M.S., Mota, M., Tramper, J. (2001) Multiphase bioreactor design (London: Taylor & Francis).

7. Jones, J.C. (1984) A method of systematic design, In: Cross, N. (ed), Developments in Design Methodology (Chichester: Wiley).

8. Chandrasekaran, B. (1990) Design Problem Solving: a Task Analysis, *AI Magazine* , Vol 11, no 4, pp 59-71.

9. Dasgupta, S. (1991) Design Theory and Computer Science (Cambridge UK: Cambridge University Press).

10. Simon, H.A. (1996) The Sciences of the Artificial 3e edition (Cambridge MA: MIT Press).

11. Keulen, H. van (1999) Design teaching views and practices in Delft. Ln: 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, pp51-58. ISBN 1 86058 2087.

12. Dym, C.L., Little, P. (2000) Engineering Design: A Project-Based Introduction (New York: John Wiley & Sons).

13. Anderson, J.R. (2000) Cognitive Psychology and its implications (New York: Worth Publishers).

14. Merriënboer, J.J.G. van (1997) Training Complex Cognitive Skills. A four-component instructional design model for technical training. (New Jersey: Educational Technology Publications Englewood Cliffs) ISBN 0-87778-298-9.

15. Cross, N. (2000) Engineering design methods: strategies for product design 3rd ed. (Chichester England: John Wiley & Sons).

16. Jong, T. de, Joolingen, W.R. van (1998), Scientific Discovery Learning with Computer Simulations of Conceptual Domains*, Review of Educational Research Summer*, Vol 68, No. 2, pp 179-201.

17. Posner, G.J., Rudnitsky, A.N. (1997) Course Design: A Guide to Curriculum Development for Teachers. (New York: Longman).

Support of Modelling in Process-Engineering Education

An objective of the Process Technology curriculum at Wageningen University is to teach students a stepwise modelling approach in the context of process engineering. Many process engineering students have difficulty with learning to design a model. Some common problems are lack of structure in the design activity, 'getting lost' in the abundance of available mathematical equations that describe similar processes in different situations and problems with correctly typing the equations into the modelling software. To reduce these problems there is a need for a set of educational tools to support students when learning how to design models. This paper describes a set of tools that can easily be integrated within a web-based learning-management system. The tool set is based on a 5-step modelling approach. The design and use of two tools of the set are described in detail: ModelComposer and ModelRunner. ModelComposer supports the student when composing a mathematical model. ModelRunner lets the user execute experiments with the model. The tools were used in a case study to get student and teacher feedback.

Published as:
Schaaf, H. van der, Vermuë, M., Tramper, J. & Hartog, R. (2006) “Support of Modelling in Process-Engineering Education”, *Computer Applications in Engineering Education*, Vol. 14, No. 3, pp. 161-168.

Introduction

An objective of the Process Technology curriculum at Wageningen University is to teach students a stepwise modelling approach in the context of process engineering. In process-engineering education many students face difficulties with learning to design mathematical models; i.e. to translate a given process situation into a mathematical model that can be used to find answers to process-engineering problems.

In process-engineering education, when asked to design a model, novice students find it difficult to design an adequate model in a methodical way. They have problems in selecting a starting point and a suitable order for their subsequent actions. Novice students usually have all theory and equations at hand, but they miss a structured way to find the right equations for the situation they are trying to model. For example, which of the many different equations for the required stirrer diameter should the student use for the type of stirrer, type of fermentor and type of liquid in the situation he is trying to model. Students also often have difficulty entering their model into the modelling software they have to use. A small typing error can be very difficult to find, especially when a student is not even sure he used the correct equations in the first place.

Existing professional modelling and simulation software requires extensive training before it can be used effectively and does not give a student much guidance when designing a model. Some modelling and simulation software is developed for educational purposes, but this is often more geared towards learning from using or “playing with” simulations than learning to design models. Others are difficult to use in combination with a learning management system (LMS). [3, 4, 5, 6]

To support the students in process engineering when learning to design models, we need a set of flexible, easy-to-use digital tools, that can be used within a (web-based) learning management system. This paper describes the design of two tools from this proposed tool set, the ModelComposer and the ModelRunner. An evaluation with students and a teacher was also done. The models used here are continuous, dynamic models, specified by differential equations.

The work presented here is part of the FBT-2 program. The FBT-2 program aims at the creation of a rich body of digital learning materials for food- and biotechnology [1]. One of the projects in this program is aimed at the development of learning material for process engineering [2].

Supporting a 5-step modelling approach

This section describes a methodology for designing and testing models. This methodology fits the teaching approach used in process-engineering education at Wageningen University [7]. The process of designing a model can be described by a set of steps:

1. Description of the Problem.
2. Detailed Analysis.
3. Model Composition.

4. Model Validation.
5. Resolution of the described Problem.

Step one is important because when making a model, one usually has a reason for it [8]. Students sometimes forget this; they just start making calculations with the equations from their textbooks right away and end up with a model that cannot be used to resolve the problem that inspired the designing of the model in the first place.

The second step involves a detailed analysis of the system the model should describe. This includes for instance specifying exactly what belongs to the system to be modelled; what the system boundaries are and what processes take place within these boundaries.

The third step consists of the actual composing of the model. In our case this involves choosing balance equations that describe the state changes of the system, expanding these balance equations by expanding each expression in the balance equations and checking the units of all input parameters and state variables of the model. This process of expanding a tree-like structure guides the students in finding the correct equations for their model and gives much direction to the line of reasoning of a student.

The fourth step is to validate the model after it has been composed. Part of this involves checking the results the model generates against experimental data, and checking if the model displays expected behaviour in several thought-experiments.

The fifth step is to use the model to generate an explicit answer for the initial problem and to evaluate this answer. If a model made by a student predicts values that are clearly unrealistic or even impossible, the student should backtrack on his design steps to find the cause of the unexpected results.

In this 5-step approach there is a distinction between building the model and using the model [8]. This process of designing and testing models is, like all design processes, not a linear process. Sometimes one has to do more than one iteration for some of the steps.

Example: Biomass growth in a batch fermentor

The third step of the 5-step modelling approach consists of expanding the chosen balance equations in a tree-like structure. The following example gives an idea of what this could look like in a bioprocess-engineering problem.

Problem: How much biomass can be obtained in 10 hours with a micro-organism that has a maximum specific growth rate of $0.77h^{-1}$ in a fermentor of $1m^3$ starting with an initial biomass concentration of $0.01kg/m^3$. The system in this model will be the contents of the

$$\frac{dM_X}{dt} = \Sigma reaction\ rates + \Sigma transfer\ rates \quad (1)$$

$$\begin{array}{l} \vdash \Sigma reaction\ rates \\ \llcorner\ \Sigma transfer\ rates \end{array} \quad (2)$$

Figure 1: The mass balance. dMx/dt needs to be calculated, Σreaction rates and Σtransfer rates need to be specified further.

bioreactor. As we are interested in the amount of biomass in the reactor we describe how the amount of biomass in the bioreactor changes in time due to growth.

The Model Composition step starts with the balance equations we need. In this case we need a biomass balance.

The biomass balance equation, (1) in Figure 1, contains two expressions that need to be expanded further (2): the sum of the transfer rates and the sum of the reaction rates. Growth is assumed the only reaction for the biomass. The growth rate is the specific growth rate times the amount of biomass in the fermenter: $r_x = \mu * M_x$. This leads to Figure 2.

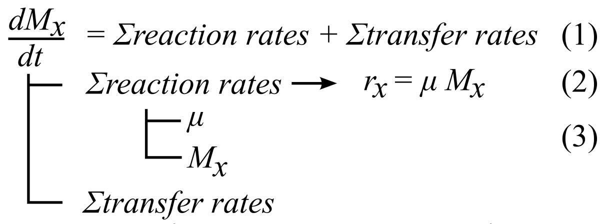


Figure 2: Expanding one expression. rx is assigned to Σreaction rates, μ and Mx need to be specified further.

Now we need to expand the specific growth rate μ, and the amount of biomass in the reactor M_x (3). The specific growth rate usually depends on the substrate concentration. If we assume there is always excess substrate for the biomass to grow, we can assume maximum growth. Therefore, we can say that the growth rate equals the maximum specific growth rate: $\mu = \mu_{max}$. The maximum specific growth rate is a constant value. The amount of biomass in the reactor, M_x, is the state variable we are calculating with this balance equation.

Because there are no flows into or out of the reactor, the sum of the transfer rates is zero, thus constant (Figure 3).

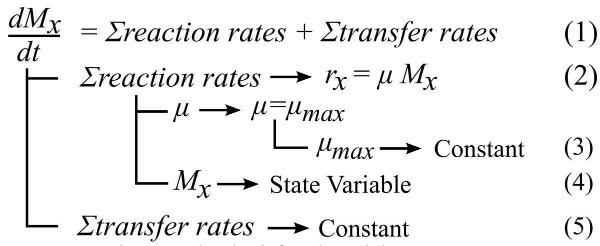


Figure 3: The completely defined model.

This model is only valid for the given situation. All substrates that the micro-organism needs for growth have to be present in non-limiting concentrations and cell concentrations can not be so high that they become limiting either. This model will predict unlimited exponential growth. In reality all substrate will eventually be consumed by the biomass, so the model will only be valid while there still is enough substrate.

To get an answer to the original problem, the constants are filled in the model and the necessary calculations are done. The model will predict that after 10 hours the biomass concentration will be approximately 22 kg/m^3.

Design of two tools of the set

As stated before, the goal of the project is to create a set of tools to support individual steps of the described 5-step modelling approach. In this paper two tools will be described, the ModelComposer and the ModelRunner. The ModelComposer is typically used in step 3 of the described modelling approach. The ModelRunner is typically used in steps 4 to run simulations with the model in order to validate it, or in step 5 to answer the question that initiated the design of the model.

The ModelComposer

When the student starts the actual composition of the model, he first selects one or more balance equations that describe the states of the system to be modelled. After that the student can one by one expand each balance equation by expanding each expression.

The mathematical representation of the model is a set of trees (Figure 3). The root of the trees are the balance equations the student selected for his model. Each expression in the balance equations is a node. If the student decides that an expression should be described by an equation, the expressions in this equation become new nodes of the tree.

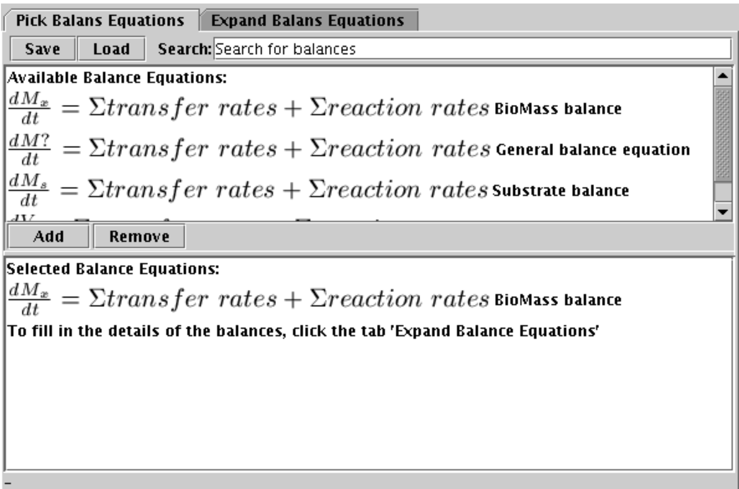


Figure 4: Choosing balance equations for the model.

In a complete model, the leafs of the resulting tree would be either constants, or state variables that are calculated from the balance equations.

When a student is building a model during a process-engineering course, he usually has books or handouts at hand that contain all the equations he could possibly need. The task for the student is to determine which equations are needed for the situation at hand.

Therefore, the ModelComposer lists all available equations for the student, so the student does not need to type them into the modelling environment. This will allow the student to focus his learning efforts on the equations used and relieve him of subtasks associated with typing in equations. These subtasks such as finding typing errors, mistakes in the placing of brackets and so forth, mainly induce cognitive load and are not essential for the learning goals.

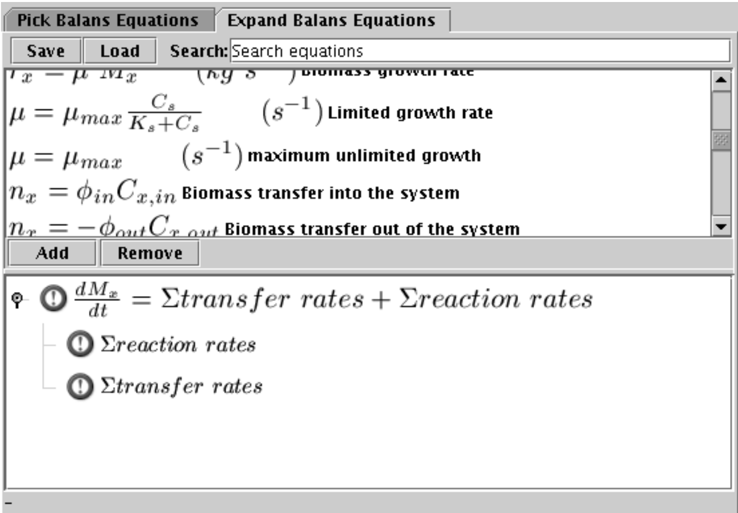


Figure 5: Expanding the balance equations.

The ModelComposer contains two tabbed views, one for selecting the balance equations that define the state variables of the model and one for expanding those balance equations into a completely defined model. The tab that lets the student select balance equations for the model consists of two parts: a list of the available balance equations and a list of

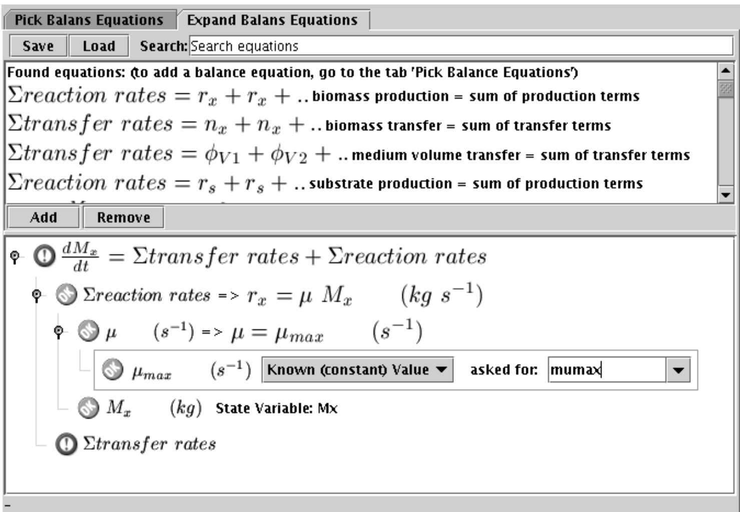


Figure 6: A more expanded tree.

balance equations the student selected for his model. (Figure 4). The tab that lets the student expand the balance equations into a completely defined model also consists of two parts: a list of the available equations the student can use in his model and a tree view. (Figure 5).

At the top of the tree are the balance equations. When the view of a balance equation is expanded, the expressions that make up the balance equation are shown as branches. In front of each item in the tree is an icon that indicates if the branch below that item is completely defined, or if there are still some expressions undefined in the branch (Figure 6). The student can then define each expression by determining if the expression is defined by a constant value, a state variable, or by another equation.

If the expression is defined by a constant value, the student is asked to name the value, so it can be referred to later. The student can choose a name of a constant he defined elsewhere, so the same constant can be used in several places [8, 3]. If the expression is defined by one of the state variables, the student is asked to select the state variable. If the expression is defined by an equation, the student is asked to select the equation he wants to use from the list of equations and “connect” it to the expression.

When the student connects an equation to an expression, the ModelComposer will check if the equation contains the expression the student connected the equation to. The other

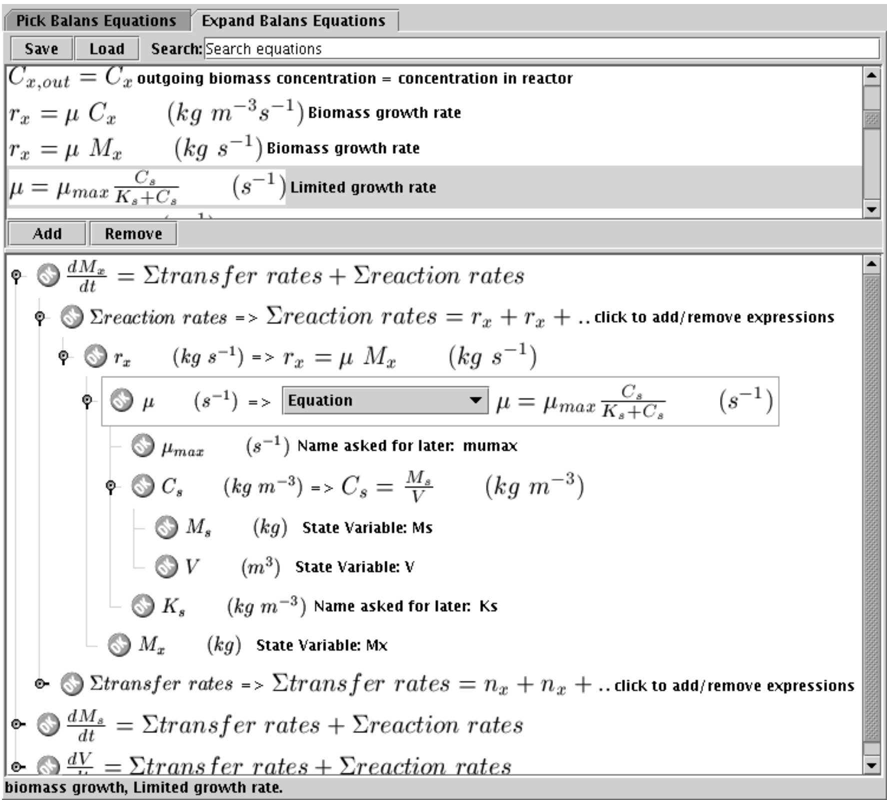


Figure 7: A completely defined model.

expressions in the equation become new nodes in the tree and the student can then in turn define how these expressions should be described in the model (Figure 6, 7).

When the student completes a sub tree of the model, the icons in front of the nodes will indicate this and the student can then focus on another part of the model that is not finished yet.

The ModelRunner

After completely defining a model, the student has to run simulations with the model to validate the model. The student also runs simulations to answer the question that initiated the designing of the model. Before running a simulation the student needs to supply values for the constants he defined in the model and he will need to give boundary conditions needed for the simulation.

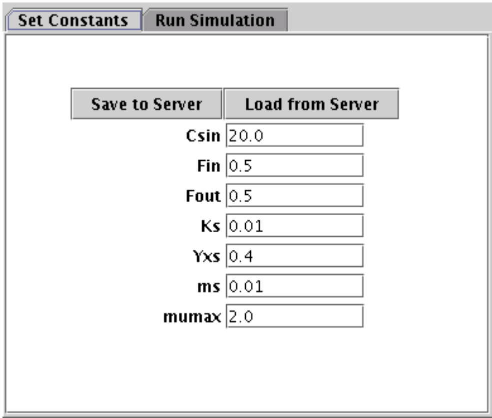


Figure 8: Assigning values to constants.

The ModelRunner displays two tabbed views. In the first tab, the student can enter values for all the constants he defined in the model (Figure 8). In the second tab, the student can enter starting values for the state variables of the model and the end time for the simulation run. A graph of the simulation run is displayed. Pressing the 'Run' button will start the simulation run. The values of the state variables in time are plotted in a graph (Figure 9). The student can select the state variables he wants to see in the graph.

Architecture and technical implementation

This chapter describes the ModelComposer and ModelRunner and their communication with the Learning Management System (LMS). To satisfy the requirement for a web-based solution, a combination of Java applets and server-side scripts is used. Java applets are used for the student interface and can store and retrieve information, like the available equations and the current model of the student, by communicating with a server-side script that will handle any database communication necessary. The data transferred between the applets and the server-side script is encoded in XML to allow easy parsing and extendibility. For the server-side scripts to be able to store the state of the model of the student, the LMS needs to implement some form of 'author-defined storage', where the author of the learning material can define what will be stored, in relation to the current logged-in student [9]. For instance on the BlackBoard LMS the server-side scripts can be implemented as a BlackBoard Building Block.

The two tools described in this paper are implemented as two Java applets. Both Java applets use the same classes to handle the composing and simulation of the model. A model consists of equations. Equations are composed of expressions. In a model an expression can be defined by a constant value, a state variable, or another equation. In the Java implementation therefore the classes *model*, *expression* and *equation* were created. To make numerical calculations with a completely defined model possible, the mathExpr package was used [10]. The *model* class implements an interface needed to communicate with a numerical integrator. Each instance of *expression* and *equation* contains the methods needed to numerically evaluate itself using the results of the numerical evaluation of the expressions and equations further down in the model tree. When displayed as a tree, the nodes of the model are expressions, therefore the *expression* class also implements the interface needed for displaying the tree, the *MutableTreeNode* interface[12]. Because the top of the tree of a model is a differential equation, a child class of *equation* was created, *differentialEquation*, which also implements the *MutableTreeNode* interface.

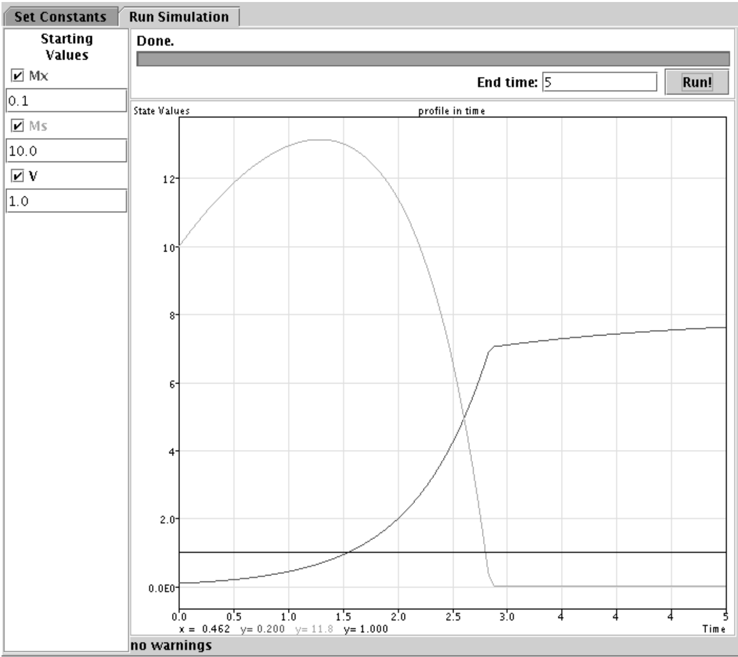


Figure 9: After running a simulation, the results are displayed in a graph.

All values of the instance variables of the available equations and expressions are stored in a relational database. In the database is also stored which expressions each equation contains. For each expression and equation there is also a TeX representation stored to allow easy generation of images for display in browsers or applets.

The list of equations and their expressions is converted into an XML document by the server-side scripts and can be loaded by the Java applets that need them.

When the Java applets is initialised, the applets will load the current state of the model. When the student just started the modelling exercise, he could start with an empty model, or with a partially finished model created by the author of the learning material [11], which the student has to finish. It can also be the partial model the student has been composing during a previous session.

First use by students

To evaluate the usability of the two modelling tools, 9 students where asked to use the ModelComposer and ModelRunner in an exercise to design a model. The students were first-year BSc students in biotechnology and did not have much experience with designing models. They were however somewhat familiar with the mathematical equations to be used in the model they had to design in this test.

The students were given an overview of the modelling approach, with the first two analysis steps already completed, and were asked to complete the other steps (composition, validation and problem resolution) using the ModelComposer and the ModelRunner.

Most students managed to find their way around in the tools within 15 minutes. They did indicate a short introduction to the tools would be useful to get a quicker understanding of the systematics of the ModelComposer.

After composing a model, students started playing with the model they created, by trying different values for the different parameters of their model and this evoked active discussion between the students about the models designed. The observed they discussed more than they usually do when modelling.

The students liked the alternative approach to compose a model and the ease with which they could play with the model they created. However, they did indicate they missed a systematic overview of their model that shows what parameters and state variables are used in each balance equation. They would like to be able to get this overview in the form of an information flow diagram of the model they composed so far.

During the test it also became clear that an easy way to check the units used in the model would be an important addition to the tool set. Some students had difficulty finding the problems in their model that where related to units; for example, some of them used concentrations of a substance (unit: kg m^{-3}) and total mass present of a substance (unit: kg) inconsistently.

An information flow diagram, units checking and other details suggested by the students are being implemented.

Conclusions

The two tools ModelComposer and ModelRunner can be used in any LMS that supports the implementation of 'author-defined storage' [9]. These two tools are part of a digital tool set that is being developed. The tool set is based on a 5-step modelling approach used in process-engineering education. The two tools presented here can be used in steps 3, 4 and 5 of this 5-step modelling approach. Typical for the methodology is the distinction between the design of the model and the use of the model to do experiments [3]. An author can use any number or combination of parts of the tool set in his digital learning material.

Using the ModelComposer the student can define a model. The ModelComposer provides the student with all the equations he needs. Because of this, the student does not have to type in equations and he can learn to work with the complete equations instead of seemingly distinct symbols. This relieves the student of subtasks that mainly induce cognitive load and are not essential for the learning objective. The student builds a tree of the model, supported by the system. This tree-structure gives much direction to the line of reasoning of a student and helps to determine which parts of the model are not completely expanded yet. Using the ModelRunner the student can execute experiments with a model in order to validate his model, or answer the question that initiated the design of the model.

Students who used the ModelComposer and ModelRunner liked the alternative way of composing a model. From the discussions provoked by the use of these tools, the teacher concluded that the students got better understanding of modelling in process engineering.

Further research efforts will be directed to the design of tools that support the remaining steps in the 5-step modelling approach.

References

1. Homepage of the FBT-2 program: http://www.fbt.wur.nl/
2. Schaaf, H. van der, Vermue, M., Tramper, J., Hartog, R. (2003) A design environment for downstream processes for Bioprocess-Engineering students, *European Journal of Engineering Education*, Taylor & Francis, vol. 28, nr. 4, pp. 507-521.
3. Kramer, M.R., Scholten, H. (2001) The SMART approach to modelling and simulation. EUROSIM 2001, June 27-30, Delft, The Netherlands.
4. Jong, T., de, et al. (1998) Self-directed learning in simulation-based discovery environments, *Journal of Computer Assisted Learning*, Blackwell Science Ltd, nr. 14, pp. 234-246.
5. Stella homepage (visited 10-2004): http://www.hps-inc.com/
6. 20Sim homepage (visited 10-2004): http://www.20sim.com/
7. Mettes, C. T. C. W., Pilot, A., Roossink, H. J., & Kramers-Pals, H. J. C. E. (1980) Teaching and learning problem solving in science. Part I A general strategy. *Journal of Chemical Education*, 57(12), 882-885.
8. B. P. Zeigler, H. Praehofer, and T. G. Kim (2000) *Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems*, 2nd ed: Academic Press.
9. Sessink, O.D.T. et al (2003) Author-defined storage in the next generation learning management systems. In -- (Ed.), *IEEE International Conference on Advanced Learning Technologies 2003*, Athens, Greece.
10. mathExpr homepage (visited 10-2004): http://www-sfb288.math.tu-berlin.de/~jtem/mathExpr/
11. J.G. van Merriënboer, *Teaching introductory computer programming - a Perspective from Instructional Technology*, PhD thesis, Enschede, ISBN 90-73028-03-5
12. Java API Specifications (visited 10-2004): http://java.sun.com/reference/api/

A digital tool set for systematic model design in Process-Engineering education

One of the objectives of the Process Technology curriculum at Wageningen University is that students learn how to design mathematical models in the context of process engineering, using a Systematic Problem Analysis approach. Students find it difficult to learn to design a model and little material exists to meet this learning objective. For these reasons, a set of digital tools has been developed to support students when learning to design mathematical models. The set of tools enables a process engineering student to do each step in the systematic approach for designing models while providing feedback on the actions of the student. The article describes both the system, the underlying design decisions and how one such case is used in a regular educational setting. Evaluation after use in a regular educational setting show that students are very positive about the fact that there is feedback on every step of the design process and that there is no need to deal with complicated mathematics.

Published as:
Schaaf, H. van der, Vermuë, M., Tramper, J. & Hartog, R. (2006) “A digital tool set for systematic model design in Process-Engineering education”, *European Journal of Engineering Education*, Vol. 31, No. 5, pp. 619-629.

Introduction

One of the objectives of the Process Technology curriculum at Wageningen University is that students learn how to design mathematical models in the context of process engineering. Modelling, in the context of process engineering, is the process of creating an abstract mathematical representation of a physical process, that can be used to find solutions to process-engineering problems. The models students design in process-engineering education are usually created by combining existing standard equations that describe sub-processes of the system to be modelled. An example would be the creation of a model that describes the growth of micro-organisms, which consume glucose and oxygen, in a bioreactor. By combining standard equations for microbial growth and glucose and oxygen consumption in mass-balance equations, a model can be designed that enables the calculation of changes in biomass, glucose and oxygen concentrations in the bioreactor given certain boundary conditions.

Students are taught that many problems in process engineering can be solved by using a systematic approach. Students have to learn to apply this systematic, stepwise approach in the design of models as well.

Students find it difficult to learn to design a model

In process-engineering education many students face difficulties when learning how to design models.

The first type of difficulties are related to design methodology. Students need to learn which modelling steps are important in the process of designing a model and be able to execute each step. At the same time, students need to keep an overview over the whole design process so they can make sure they are still working from their starting point towards a solution of the problem. Decisions made in one step set the stage for subsequent steps in the design process and students have to recognise this and learn that previous decisions may require adjustment. In practice students often have to make adjustments in steps earlier in the design process.

The second type of problems are related to the digital tools used to design models. Most of the digital tools that are standard for the design and application of models have been developed for industrial or research purposes and are in general not specifically suitable for use in an educational setting. If students have to use this kind of modelling software for the design of their model, they often have difficulty entering their model into this modelling software. A small typing error for example can be very hard to trace, but it can cause an experiment with the model to result in behaviour that is totally different from the relevant behaviour.

Some software that is used in education, like Mathlab or Mathcad requires training before a student can effectively use it, and it is difficult to learn how to use modelling software if you are not yet a proficient modeller. The modelling and simulation software that is specifically aimed at education, like Stella, 20sim or SMART [1, 2, 3, 4, 5, 6, 7, 8] is usually more geared towards learning from existing models by “playing” with them and not learning how to design new models. In addition, the correct use of units when building a model is of major importance, however not all of the available modelling software has facilities to check the units used in a model. Most modelling software is also not readily available on every computer a student uses.

Most modelling software does not give the student feedback on his design process or on the choices the student makes in the design and a lot of modelling software does not support the student in using a systematic approach to the design process.

Teachers find it difficult to teach how to design a model

Teachers usually teach systematic problem analysis and systematic model design by giving the students a lot of examples of how to design a model for a given problem and how a systematic approach is used in the design of these models and in solving the problem. Teachers who teach modelling are generally themselves expert model designers. They are trained in the design of models and finding solutions to process-engineering problems. However, they are called experts because they can go from goal a to goal c while seemingly skipping goal b. They do process goal b, but unconsciously, often they do not tend to make the goal explicit, they just "do" it [9, 1]. Because of this, it is for students sometimes hard to recognise the systematic approach that is used by an expert designer.

Systematic model design

To support students when learning to design models, a systematic model design approach has been developed, based on the Systematic Problem Analysis approach [10, 11] that was already in use in the courses taught at the Food and Bioprocess Engineering Group of Wageningen University. The process of designing a model can be described by four steps, each subdivided into several sub-steps. A more detailed explanation of the sub-steps is given in chapter 4, with the description of the learning material that has been developed.

1. General analysis.
   - Determine the function of the model, what question should be answered by the model.
   - Describe the system to be modelled in words.
   - Make a schematic drawing of the system to be modelled.
2. Detailed analysis.
   - Make assumptions explicit.
   - Define the system boundaries & define subsystems.
   - Make a drawing including all available data.
   - Determine which variables and parameters could be important.
3. Compose the model.
   - Search for and select a set of usable standard equations.
   - Check the number of equations and the number of unknowns.
   - Check the units.

4. Answer the question and evaluate.
   • Use the model to answer the question.
   • Evaluate the answer.

An important aspect of the design process is that the students will often have to backtrack through the sub-tasks. Sometimes students have to take another look at a previous step and sometimes a sub-step can be skipped altogether. Students might have to go back on the main track. For instance, when the units do not match they will have to check their set of standard equations, or when they discover that their model is not yet finished during the evaluation, they will have to go back to the composition step, or maybe even revisit the analysis steps.

Design requirements for the Systematic Model Designer

The systematic model design approach has been implemented in a web-based system. This system is called the Systematic Model Designer (SMD). Other design requirements for the SMD pertain to the relevant learning objectives, the educational model and the learning environment.

Learning objectives

Many models in process engineering are based on balance equations that describe a change in time. The Systematic Model Designer thus should support learning to design this type of model. A balance equation describes the change, in time, of a quantity in a system. This change in time is a function of the state of that system. For example, to calculate the amount of biomass in a fermenter (M_X), we can write:

$$\frac{dM_X}{dt} = \Sigma\, transfer(M_X) + \Sigma\, reaction(M_X)$$

In words: the change in the amount of biomass in the fermentor is the sum of the amounts of biomass flowing into and out of the fermentor, and the sum of the amount of biomass produced by growth and decreased by death in the fermentor. The amount of biomass produced by growth and decreased by death are a function of the state of the system, including the current amount of biomass present.

To solve this type of differential equations students usually require specific knowledge of numerical methods. Because we want the students to be able to focus on the systematic approach to the design of their model and not be distracted by detailed mathematics [1, 12], we want the system to handle these mathematics transparently for the student.

Educational model

For the design of learning materials at Wageningen University, the ARCS model [13] is used. From the ARCS model we derive that the material has to activate the students, it has to be relevant to the students, it has to give students confidence that they can successfully complete their assignments, and the students should feel satisfaction after completing (parts of) the assignment. To activate students, we want the material to be interactive. Students have to be able to perform each step in the model design approach by themselves. The feeling that they achieved the goal by themselves will give students satisfaction. To give students confidence, we require the material to give the students feedback on what they do and help them if they get stuck.

Learning environment

Like most other universities Wageningen University has installed a web-based Learning Management System. In order to provide the student an integrated learning experience we want the SMD to be web-based as well and compatible with *de facto* standards for current learning management systems.

Description of the SMD module

The stepwise approach is implemented in a set of digital tools. Each step of the model design approach is supported with a student activity. To make use of the tools in education, they are used in a bioprocess-engineering related case where the students have to design a model. By combining the SMD with a relevant case a teacher can make learning material for a student to use. If a teacher wants to focus on a specific part of the model design, he can use a subset of the tools, or let the student start with a (partly) finished model.

Step 1 General analysis

The general analysis is meant to get a clear view on the question that is to be answered using the model, in what way the model is going to help to solve the problem and what system the problem is related to.

Thus, the system asks the student to describe, in his own words, the problem to be solved by the model. After a student has given a description, we ask the student to compare his description with several descriptions provided and to pick the description that optimally fits his own suggestion. This stimulates the student to reflect on his own description and at the same time provides hints and clues as to possible better descriptions, so the student can improve his. After the student has picked a description, he receives feedback on his choice, to further improve his own description.

In a similar way the student is asked to make, in his own words, a description of the system being modelled.

In the third part of the general analysis, the student is asked to make a drawing, on paper, of the system he is trying to model. After making the drawing, the student can continue. He is asked to compare his drawing with several presented drawings and to pick the one that resembles his own drawing best. The student then receives feedback on his choice so he can improve his own drawing.

Step 2 Detailed analysis

In the detailed analysis, students have to analyse the details of the system they are trying to model, of what processes take place in this system, what sub-systems they can define, what assumptions they have to make and how they can reduce the real-life situation to a manageable degree so it can be described by a model.

Often, some details in the system being modelled are not known, while other details can be ignored because they are not important. During the design process students have to make assumptions about these details and it is important that the students are aware of the assumptions they make.

To make assumptions explicit, a multiple-answer question is used. The student is presented with a list of numerous possible assumptions. Some of those assumptions are needed to model the system, some should not be made, and some are irrelevant. The student can choose which assumptions he thinks are necessary, and gets feedback on the choices he made. The given feedback consists of thought-provoking comments and not of “correct” or “incorrect”, in order to activate the student to think carefully about his answer.

Another step is making a detailed drawing of the system to be modelled. This is done in the same way as in the General Analysis step. The student is first asked to make a drawing on paper, after that, the student can choose, from several drawings, the drawing that fits his best.

To divide the system into subsystems and to define the system boundaries, the student is given a list of possible systems he might define and is asked to select the ones he thinks are essential to make a model for the total system. After defining subsystems the student is

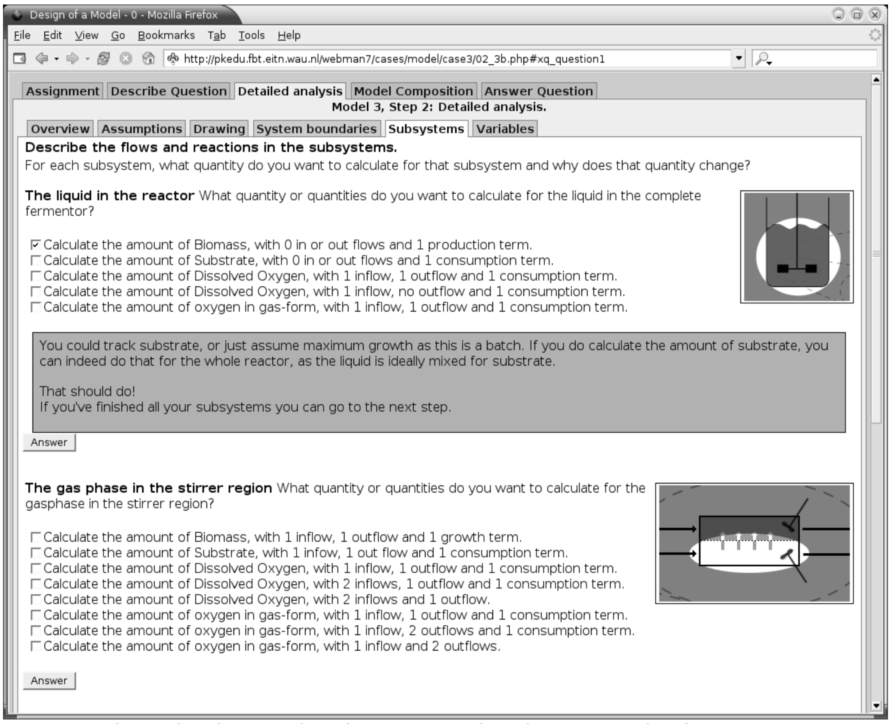


Figure 1: The student has to select the quantities that change in each subsystem.

asked for each subsystem what quantities he wants to calculate for that subsystem and how many in- and output flows, production and consumption reactions there are for that quantity in that subsystem (figure 1).

In one of the sub-steps, the student has to determine what variables he needs as the basis for his balance equations. The student is presented with a long list of possible variables to choose from, he can search in the list by giving (part of) the name of the variable he needs. The list is then shortened to only the variables that fit that name. After adding a variable to his list of essential variables, the student is given feedback on his choice (figure 2).

Step 3 Compose the model

In the composition step, the students gather standard equations, combine these into their mathematical model and check if the units used in this model are consistent. While composing a model, the student will get summaries of the various choices he made during the analysis steps of the design process.

First the student chooses the balance equations he needs. He has already determined what variables are the base for these balance equations in the detailed analysis (step 2), now he is presented with a list of standard balance equations and can select the ones he needs for his model.

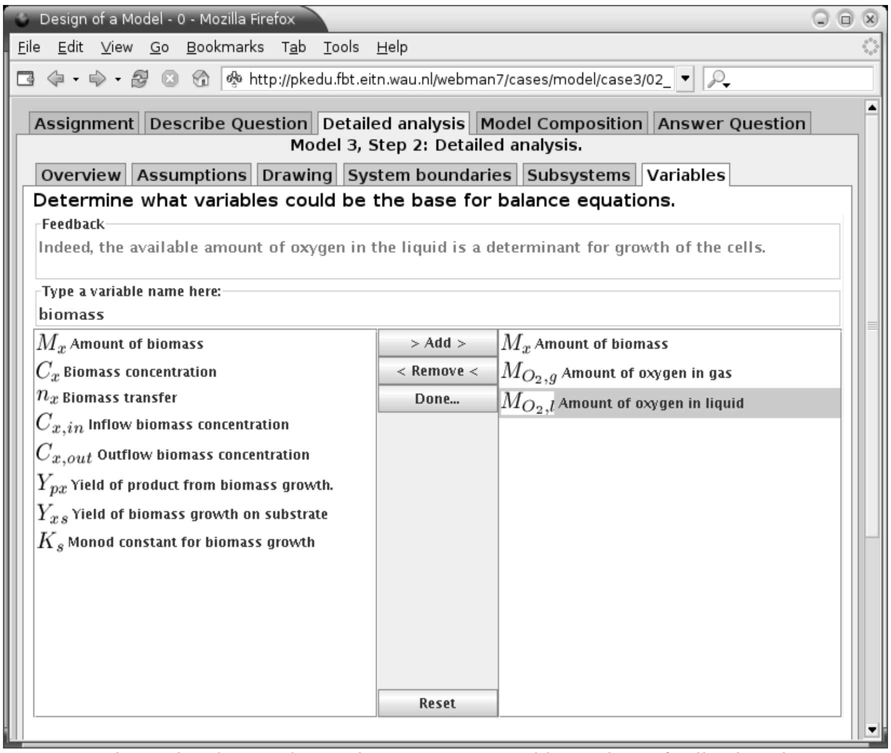


Figure 2: The student has to choose the important variables and gets feedback on his choices.

After selecting one or more balance equations, the student can expand these balance equations. The balance equations contain several expressions that the student has to fill in. An expression can either be a known parameter, a state variable or be defined by another equation. If an expressions is defined by an equation, then the student has to choose which equation he wants to use to define the expression. This new equation may in turn contain new expressions that need to be filled in. This process of setting up equations creates a tree structure, with the balance equations at the base. All branches will eventually end with model parameters or state variables and when this is the case, the number of (independent) equations should be an equal to the number of unknowns. Icons in the tree will point the student to the nodes in the tree that are not yet complete (figure 3).

The equations that the student can use to expand his balance equations are provided by the system. The student can search those equations by giving a keyword and the system will then show only the equations that match that keyword (figure 3, "search" is located near the middle).

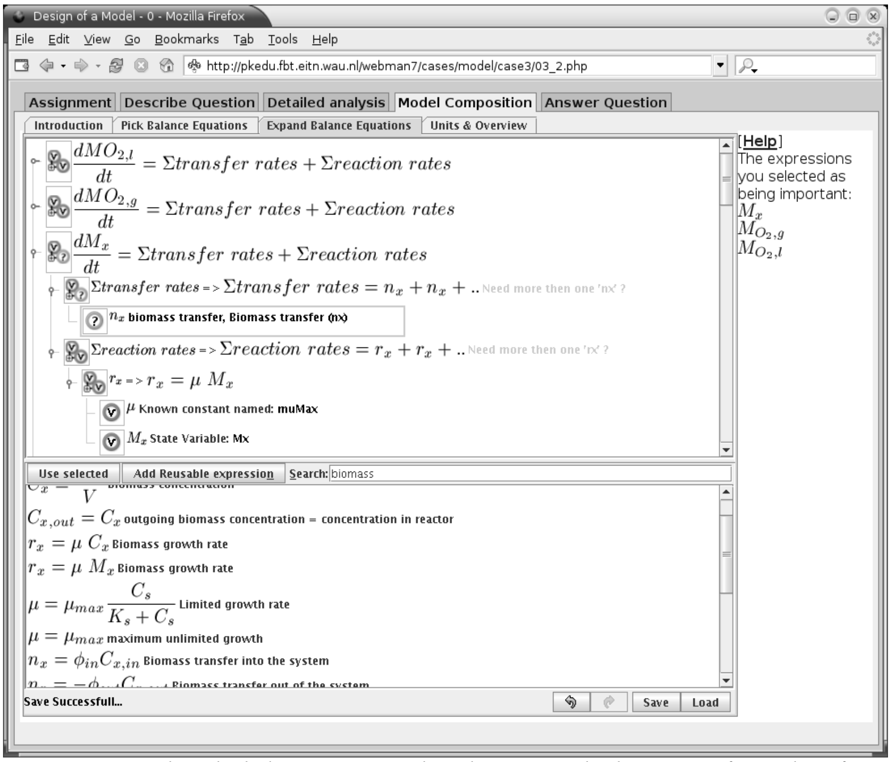


Figure 3: Expanding the balance equations by selecting standard equations from a list of supplied equations.

Once the student has developed a numerically-solvable model, he can supply units for all the parameters and balance equations in the model. The system will then check if all the units are used in a consistent way throughout the model. If the system finds any discrepancies, the student is alerted and shown the place where the inconsistency is located. This way the student can track errors in his model more easily and that gives him the confidence that he can complete the model (figure 4). This screen also shows an overview of all variables and parameters that the student has defined, and where they are used.

Step 4 Answer the question

In the last step, the student uses his model to answer the original question.

Before the student can do any calculations with his model, he has to supply values for the parameters he defined in his model. After supplying values for all parameters the student can choose to either run a simulation in time with his model, or search for a steady state of his model.

If the student runs a simulation in time, he can supply boundary values for the state variables of his model. The system then uses a Runge-Kutta solver to calculate the changes

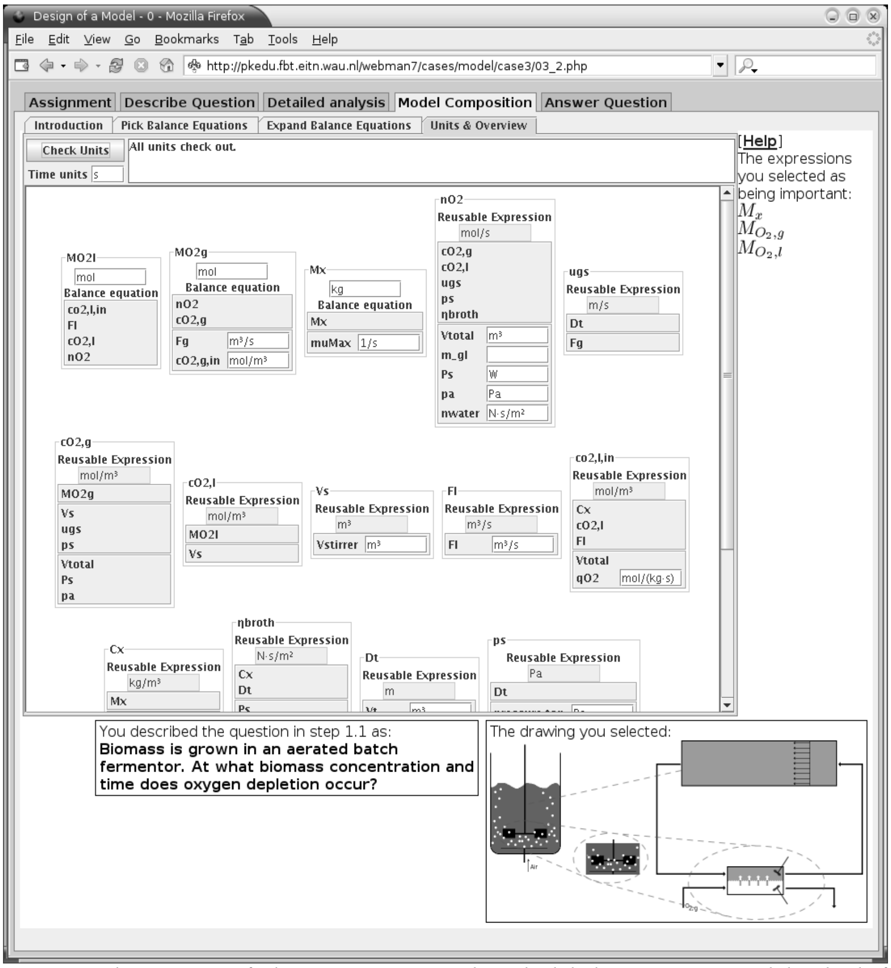


Figure 4: The overview of what parameter is used in which balance equation and the check if all units are consistent throughout the model.

of the state variables in time. Graphs are plotted of the variables of the model that the student is interested in (figure 5). The system also provides the possibility to find steady-state values of the model. Next, the student has to evaluate his answer to check if it really answers the question he defined in the basic analysis and if the answer found is realistic and in the range in which the student expects it to be.

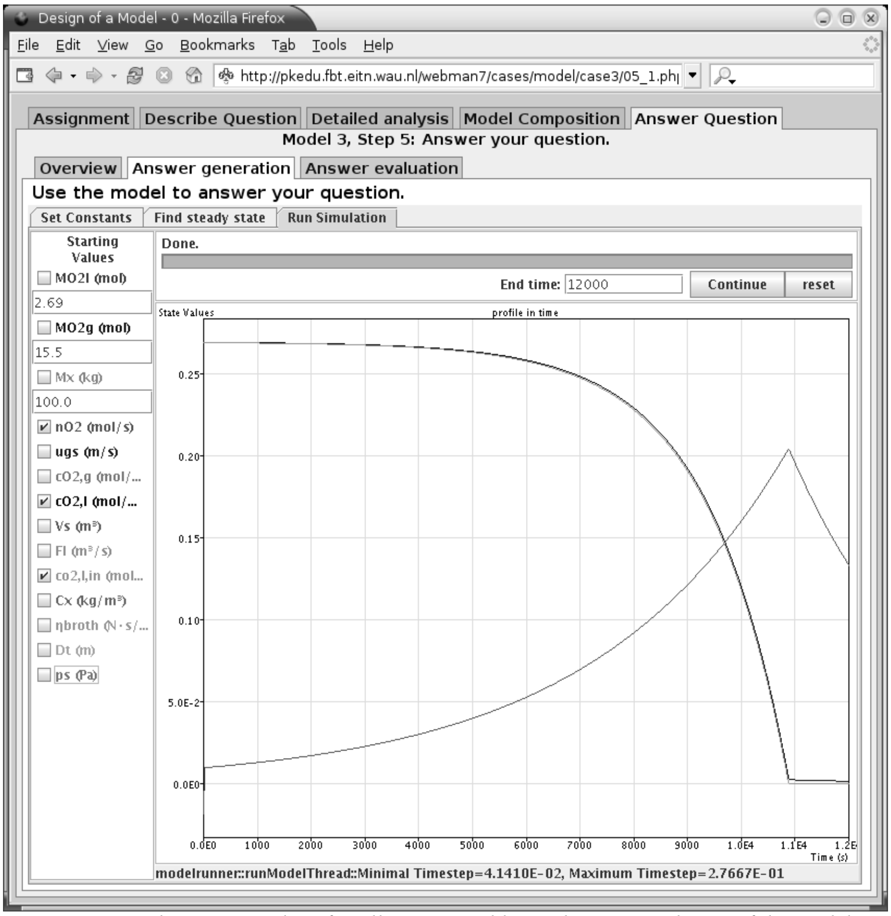


Figure 5: Provide starting values for all state variables and run a simulation of the model.

Use in education

The set of tools was used in a case on oxygen consumption and oxygen limitation in a bioreactor. This case was used in the course Bioprocess Design for Master students in Biotechnology, with a group of 20 students. For these students the subject of oxygen consumption and limitation is highly relevant. After completing the case the students had to write reports about their design process.

The students where actively working on designing models without being distracted by collateral problems. Students liked the fact that they could concentrate on the model-design process and did not have to worry about mathematics. They also appreciated the ability to easily check the units of their model. Students commented that they really liked the fact that they could follow an explicit structure and that they received feedback on every step in the model-design approach. They also noted that they would appreciate more use of this system in their process-engineering courses.

From the reports delivered by the students it was clear that all students had a very good view on the problem to be solved, the system to be modelled and what assumptions needed to be made and why those assumptions where necessary. The students also had a good grip on how the different subsystems could be defined and how those subsystems interacted with each other.

During the case all students created a model and where able to do simulation experiments with it. Seeing that their model worked inspired several students to play with the parameters of their model to see what the influence of the parameters was on the outcome of the model.

While all students created a complete model, not all of them created a correct model. The model the students had to create in this case contains four different subsystems, all with a different volume. Several students confused some of these different volumes when composing their model, and substituted the volume of the wrong subsystem in some places in their model.

Most of the errors did not result in a very large deviation in the final result, however some students did report results that where obviously not correct and did not mention this in their reports. A student should recognise obvious incorrect model output.

These observations pointed out that further development should focus on the generation of feedback in the composition and evaluation parts of the model-design process.

References

1. Merriënboer, J.J.G. (1997) Training Complex Cognitive Skills: a four-component instructional design model for technical training, Educational Technology Publications Inc., Englewood Cliffs, (ISBN 0-87778-298-9).

2. Jong, de, T. et al. (1998) Self-directed learning in simulation-based discovery environments, *Journal of Computer Assisted Learning*, **14**, 234-246.

3. Teodoro, V. D. (1998) From formulae to conceptual experiments: interactive modelling in the physical sciences and in mathematics. *Invited paper presented at the International CoLos Conference New Network-Based Media in Education.*

4. Dimitracopoulou, A., Komis, V., Apostolopoulos, P., Politis, P. (1999) Design principles of a new modelling environment for young students, supporting various types of reasoning and interdisciplinary approaches, *AI-ED 99, 9th International Conference on Artificial Intelligence in Education*, pp. 109-120.

5. Kramer, M.R., Scholten, H. (2001) The SMART approach to modelling and simulation. *In Proceedings of EUROSIM 2001.*

6. 20Sim, Controllab Products B.V., http://20sim.com/ (accessed 8-2005).

7. Berkeley Madonna, University of California at Berkeley, http://www.berkeleymadonna.com/ (accessed 8-2005).

8. Stella, isee systems, inc., http://hps-inc.com/ (accessed 8-2005).

9. Larkin, J., McDermott, J., Simon, D.P., Simon, H. A. (1980) Expert and Novice Performance in Solving Physics Problems. *Science*, **208**, 1335-1342.

10. Mettes, C. T. C. W., Pilot, A., Roossink, H. J., & Kramers-Pals (1980) H. J. C. E., Teaching and learning problem solving in science. Part I A general strategy. *Journal of Chemical Education*, **57**(12), 882-885.

11. Zeigler, B. P., Praehofer, H., Kim, T. G. (2000) *Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems, 2nd ed*, Academic Press.

12. Kirschner, P.A. (2002) Cognitive load theory: implications of cognitive load theory on the design of learning, *Learning and Instruction*, **12**(1), 1-10.

13. Keller J.M. (1983) *Development and use of the ARCS model of motivational design*, Report nr. IR 014 039. ERIC document reproduction service No. ED 313 001.

Design of activating digital learning materials to support complex learning objectives

"Tell me and I will forget. Show me and I may remember. Involve me and I will understand" - Confucius, 250BC

The use of digital learning materials can have advantages over traditional learning materials when teaching complex technical topics and skills. The first step in the design of digital learning material is to investigate what types of learning objectives need to be supported. Complex learning objectives can be classified based on the degree of freedom that is implied in the topics and skills covered by the learning objectives. Topics and skills that require a low degree of freedom in thinking and acting can be effectively supported by simple adaptive systems or cases containing closed questions with dynamic feedback. Skills that imply a high degree of freedom, like design, require open-ended learning materials that offer the student that degree of freedom. In this paper we discuss guidelines for the design of digital learning materials, together with examples of the development and use of digital learning materials.

Submitted for publication as:
Schaaf, H., van der, Sessink, O.D.T., Hartog, R.J.M., Tramper, J., Vermuë, M. (2006) "Design of activating digital learning materials to support complex learning objectives", *International Journal of Engineering Education.*

Introduction

In a technologically-oriented society, (bio-)process engineers more and more need to carry out complex cognitive tasks, ranging from fault management and trouble shooting to the design of completely new production processes. These complex tasks are complex in the sense that they require a number of highly interrelated constituent skills. Some of these skills need to be executed as well-learned procedures, while others rely on a good understanding of one of the specific process engineering subject domains and the ability to reformulate and solve problems (i.e. find procedures) in this subject domain [1]. Education in this field should thus focus on training these complex technological skills.

At Wageningen University, during courses in food- and bioprocess engineering it was observed that some important learning objectives in the process engineering domain are difficult to address, particularly the design-related learning objectives, such as process design, experiment design, and design of mathematical models. The situation is often further complicated by the often very heterogeneous student population. To reach these learning objectives in a traditional learning environment (classroom lectures combined with lecture notes and books, tutorials and practicals) would require either a lot of time, available teaching staff, or utilities.

Given the problems encountered, the FBT project [2] was started at Wageningen University to explore the possibilities for solving these problems by developing digital learning materials. In this paper we give an overview of the developed digital learning materials. The focus is on education at university level in the domain of bioprocess engineering. From these materials we developed a set of guidelines to support the design of digital learning material for training complex technical skills in process engineering. These guidelines aim to enable future designers to design effective and motivating digital learning material to address complex technical skills.

Constraints for the design of activating digital learningmaterials

At the start of any design process, one has to specify the constraints for the desired end-product [3]. It is obvious that the designed learning material should support the intended learning objectives. Besides this obvious constraint, we defined the most important constraints below. The designed digital learning material should:

- Activate students.
- Motivate students.
- Limit extraneous cognitive load.
- Be suitable for students with different backgrounds and learning styles.

At Wageningen University the education in food and bioprocess engineering aims at an active approach to learning. In all courses in the curriculum the students are stimulated to actively use the technical skills that they are supposed to learn.

To support an active attitude of the students, it is important to motivate the student. The digital learning material should therefore at all times motivate the student by treating relevant topics and helping the student in case of problems encountered during the individual learning process.

The learning process of a student is limited by the total cognitive load a student can process. All cognitive tasks have an intrinsic load that cannot be changed [1]. Tasks that are not essential to the learning objectives induce an extraneous cognitive load. All the effort a student has to invest in those tasks it not directed to achieving the learning objectives, therefore extraneous cognitive load should be minimised.

At universities the student population is often very heterogeneous; Students may have different learning styles, different prior knowledge, etc. Digital learning material should cater to all these different students.

Classification of complex technical learning objectives

A general approach in the design of learning materials for training complex technical skills starts with the analysis of the complex learning objectives involved [1]. Different types of learning objectives can benefit from different types of learning materials. Of course, this applies to digital learning material as well.

From the digital learning materials developed at Wageningen University, we derived a classification that can help to decide what kind of digital learning material to use to support different kinds of complex technical learning objectives (Figure 1).

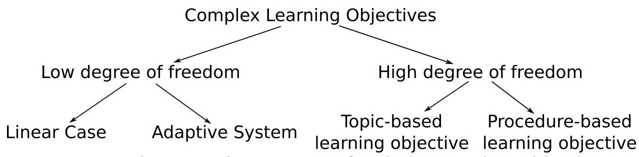


Figure 1: A schematic decision tree for designing digital learning materials.

Complex technical learning objectives can be classified on basis of the amount of freedom they involve: learning objectives with a low degree of freedom and learning objectives with a high degree of freedom.

Learning objectives with no or few degrees of freedom deal with topics in which problems usually have only one correct solution, with a clearly defined way to get to this correct solution. An example of these learning objectives is to obtain knowledge and understanding of a specific topic, such as mixing.

Learning objectives that involve topics with a high degree of freedom on the other hand, usually involve a design process in some form. Design is an open process, where a combination of a systematic methodology and personal decisions leads to a result. Designing is an activity that is done everywhere in the scientific world, like the design of processes in industry to the design of models and the design of experiments in research. A key issue within any design process is that there is no fixed way to make a design; it is a flexible, variable and iterative process. During the design process there is a large degree of freedom and many choices have to be made, while there is no way to say which choices are optimal. When you start a design you usually do not know what the end result will look like and because there is more than one correct choice, different designers will end up with different designs.

How to address learning objectives with a low degree of freedom

Learning objectives with a low degree of freedom can be supported by using closed questions as exercises about these objectives usually have one correct answer and a clear way to reach this answer. Using closed questions makes it possible to give the student feedback that may change, depending on the answer the student gives. Different types of feedback can provide verification and elaboration that can confirm and explain correct responses and specifically help the student to correct his errors. This feedback in the form of hints, references and explanations stimulates the student's learning process [4].

With incremental or staged feedback it is possible to change the type and amount of feedback depending on how many times the student gave a wrong answer to a question. This way the correct answer is not given directly after the first try, which stimulates the student to search for the correct answer himself [4]. At the same time it prevents the situation where the student gets stuck and cannot continue because there is not enough feedback, or the feedback is not specific enough. The certainty that he will be able to answer the question in the end will give the student confidence that he will be able to finish the exercise, which in turn increases the student's motivation.

Closed questions in a linear case

Some learning objectives deal with the relationship between several aspects within a topic, or even with aspects between several topics. These learning objectives can be supported by presenting the student with a series of related questions and combining these questions in a case. The questions in a case can have a strong relation with each other and this relation can be used to help the student gain understanding of the pertinent topic.

In addition, cases with a realistic setting, that students can recognise as relevant, and that provide them with an authentic task, is a good way to increase students motivation and the learning process. Associating information with a relevant and realistic context makes it easier for students to later recall the knowledge if they are confronted with a similar situation [5, 6].

Closed questions in an adaptive system

Lecturers often face a very heterogeneous group of students, where each student has different prior knowledge, has done different previous courses, maybe at different universities, in different countries, with a different native language and may even have a different learning culture. When trying to get all these different students to the same prerequisite state of knowledge using an adaptive system is helpful.

In a situation with such a diverse group of students, a linear case with a fixed set of questions is not very useful. For example, some students will already know most of the topics of the case, and will quickly loose interest if a topic they already mastered, keeps coming back in the questions. On the other hand, other students that do have a gap in their knowledge might want more questions about that same topic.

With an adaptive system, each individual student can get specific, individual learning material on the specific topics he has problems with. If the system notices that a student lacks knowledge in a specific topic, the system can give the student more learning material on that specific topic. If a student already mastered a topic, the system will quickly move to other topics and not bother the student too much with learning material about the topic the student already mastered.

In an adaptive system it is not possible to have a strong relationship between the individual questions other than that they address the same topics or skills. If two questions are strongly related, and the system would decide that the student needs a lot of in-between questions after the first one, then the questions become separated and the relation between the questions will be lost. It is therefore hard to use a system like this to address learning objectives that require a series of related questions and that cannot be addressed in one single question.

How to address learning objectives with a high degree of freedom

Most learning objectives with a high degree of freedom deal with design. During a design process choices have to be made and depending on the choices made, the result of the design process can be very different. When making a design, the student will have to deal with the uncertainty of not knowing whether the choices made where the best ones. A student cannot spend all his time looking at every detail of one aspect of the design before making a choice. He has to keep an overview over his whole design process and make a decision about the amount of time he can spend looking into the different details of the design. This switching between overview and details introduces a large cognitive load. Together with additional cognitive load induced by the subject matter, this can cause a cognitive overload and reduce the learning efficiency [1].

When designing learning material it is therefore important to reduce any extraneous cognitive load. The first step in reducing extraneous load from the material, is determining what part of the load is important for the learning objectives and what part is not. It is possible to distinguish between two types of learning objectives in relation with cognitive load: *topic-oriented* learning objectives and *procedure-oriented* learning objectives (Figure 1).

Topic-oriented learning objectives

On the one hand, a learning objective can be a knowledge and understanding of a specific topic, where the student for instance has to learn the complex interactions inherent to the topic. The process the student should follow to get his answers is not part of the learning objectives and thus should be transparent for the student. The student should be able to focus his attention on the topic-related matters. Computer simulations and discovery learning, where a student can discover how a model works by changing the parameters of the model, can be very useful in this case.

Procedure-oriented learning objectives

On the other hand, a learning objective can be more about learning a procedure, like how to design, and less about a specific topic. In this case you want the students to become aware of the procedures they are following. They should not be distracted by details that are specific to the topic.

For example, in many fields of science students have to be taught how to design an experiment so that the experiment yields data that can be used to draw conclusions. When teaching students how to design an experiment, you want the students to actually design many experiments and then execute those experiments. Often it is not practical to have students execute many experiments, especially if those experiments are expensive, or take a long time to execute. The experiments might fail or might yield useless results, because the student is still learning to design experiments. In this case, learning material in which the experiments are simulated can allow the student to practice with designing experiments.

Another example of a situation where the procedure is the important learning objective, is teaching students how to design mathematical models. When students have to design a model they often do not know where to start or what to do and they just start typing equations in their model environment without properly analysing the problem. At the same time, most electronic modelling environments impose a large amount of extraneous cognitive load on a student, especially if the student is still learning how to design models. Learning material can be made that can be used when teaching a student how to design a model, that can give the student a clear structure of the design process, and that reduces cognitive load on the mathematical part.

Examples of digital learning materials to support learning objectives with a low degree of freedom

Learning objectives with a low degree of freedom can be supported using closed questions. The examples given here are on mixing in bioreactors and cell growth kinetics.

Mixing in bioreactors

This material for 2nd year BSc students can be found on the FBT website [2] in “Content Showcase”, “Mixing theory assignment”. In this subject students have to learn the ins and outs of mixing in bioreactors, i.e. why the contents of bioreactors are mixed, how they are mixed and how to calculate the speed of the mixing device. This material is a typical example of a case-based system. In this case students are placed as a junior consultant in a virtual firm and given the assignment to design the mixing equipment for a bioreactor.

Using a series of closed questions, students are led though the case. If a student has given a wrong answer, he is first given some general hints that can help him find the answer to the problem. If the student gives more wrong answers to the same question, the hints become more specific, until finally the correct answer is revealed, with a detailed explanation why that answer is correct. If a student gives the correct answer, he is given an explanation of why that answer was correct.

Where appropriate, bug-related feedback is given [4]. For example, if a question requires the student to enter the result of a calculation, and the student gives an answer that is a factor 1000 larger or smaller then the correct answer, then the student is given the hint to check the units of their answer (because it is likely that the student is using a unit that is different from the required unit).

The ordering of the specific questions and the storyline of the case is used to illustrate the relationships that exist between different topics and show typical problems that often arise in real-life situations.

Lecturers noted that the students are very motivated and active when working with the material. Students like to explore all the feedback on the questions. Even if they directly gave the correct answer students often try the wrong answers to see the feedback with the explanation of why that answer was wrong.

Cell growth kinetics

This material can be found on the FBT website [2] in “Content Showcase”, “Cell growth experiment design”. At Wageningen University the 2nd year BSc course “Introduction to Process Engineering” is followed by students with a very diverse set of prior knowledge. This often gave problems during traditional classes, because a large group of students was missing some part of the required prior knowledge, but each individual student lacked knowledge on a different topic. A large part of the available time in the course was lost on getting all students to the same level of prior knowledge in all topics.

A large set of questions of varying level of difficulty was created, covering all topics. Each question addresses one or more topics and was rated for their difficulty on each of those topics. The questions were then inserted into an adaptive system called Proteus [7]. This system offers the student questions, and keeps track of the proficiency of the student in each topic, based on how many tries the student needs for each question. If needed, the system offers the student more questions about the topics that he does not yet fully master. The questions contain elaborative feedback to help the student understand the topic and as the level of proficiency of the student rises, the system will offer more difficult questions.

Students who already have all required knowledge can complete the module about cell growth in about 20 questions, while students that do lack a lot of pre required knowledge sometimes have to make up to 80 questions and take considerably longer to get their knowledge up to par. After the student has studied the module he has all the required knowledge on the subject and the lecturer can teach his class more effectively.

A problem of existing adaptive systems is, that they are difficult to use for the lecturer, who has to put his material into the system. The advantage of the Proteus system developed is that the lecturer, besides making the questions, only has to specify what topics a question is relevant for, and how difficult the question is. These are typical tasks that experienced lecturers can easily handle.

Examples of digital learning materials to support learning objectives with a high degree of freedom

Learning objectives with a high degree of freedom are difficult to support with closed questions and linear cases and are best supported with open questions and non-linear progression. The examples given here are on the design of downstream processes, the design of experiments and the design of mathematical models.

Downstream process design

This material can be found on the FBT website [2] in "Content Showcase", "Downstream process design". It has been created for a first-year BSc course on Process Engineering. The learning objectives that this module of learning material supports are:

- Getting to know the different unit-operations in downstream processing.
- Learning how to design a downstream process that produces a product within the desired specifications.
- Learning the effect of a single unit-operation on the outcome of the entire process.
- Learning the effect of different ordering of unit-operations.

These learning objectives include design. Any design that meets the constraints is a correct design and it is important that students realise that there is not one single correct design. In this learning material the focus is on the downstream-processing specific part of the design, and not on the design process itself. These first-year students have to learn how unit-operations work and interact first. They will learn later during their studies what design methodology is best used.

To teach students how the different unit operations in downstream processing work, and especially what different combinations of unit operations do, we wanted to give the students the ability to "play" with the unit operations. Students should be able to try out different settings for the different unit-operations and try out different orderings of the unit operations. This playing stimulates the learning process [8]. The students should not be distracted by the complex details of how each individual unit operation works, e.g. the student can use an adsorption column in his process, but should not be distracted by details like flow patterns or diffusion limitations that may occur in the column.

A design environment, called the DownStream Process Designer (DSPD) [9, Figure 2], is created in which the student can design a downstream process using different unit operations. After each change the student makes, the DSPD calculates the effect of each unit operation on the product stream and shows the student the effect on output and waste.

This design environment is used in a digital case. In this case, the student works for a company producing an enzyme. The company presents the student with an unrefined product and the student has to design a downstream process to purify the product to a given purity, minimizing the loss of active component, while keeping within a given budget. After completing this design, the company makes a change in the production process that causes a change in the unrefined product. The student then has to adapt his design to this new situation.

At the end of the case there is a list with the best scoring designs of students, in the fields of purity, product yield, price and amount of waste. This list of top-scores often highly motivates students to review and improve their designs, trying to get their own name on the list. Because there is more then one correct answer students can learn much by comparing their own design with the design of other students. By looking at other design results they can see that there are other ways to solve the same problem, they might notice things they overlooked and get ideas on how to improve their own design.

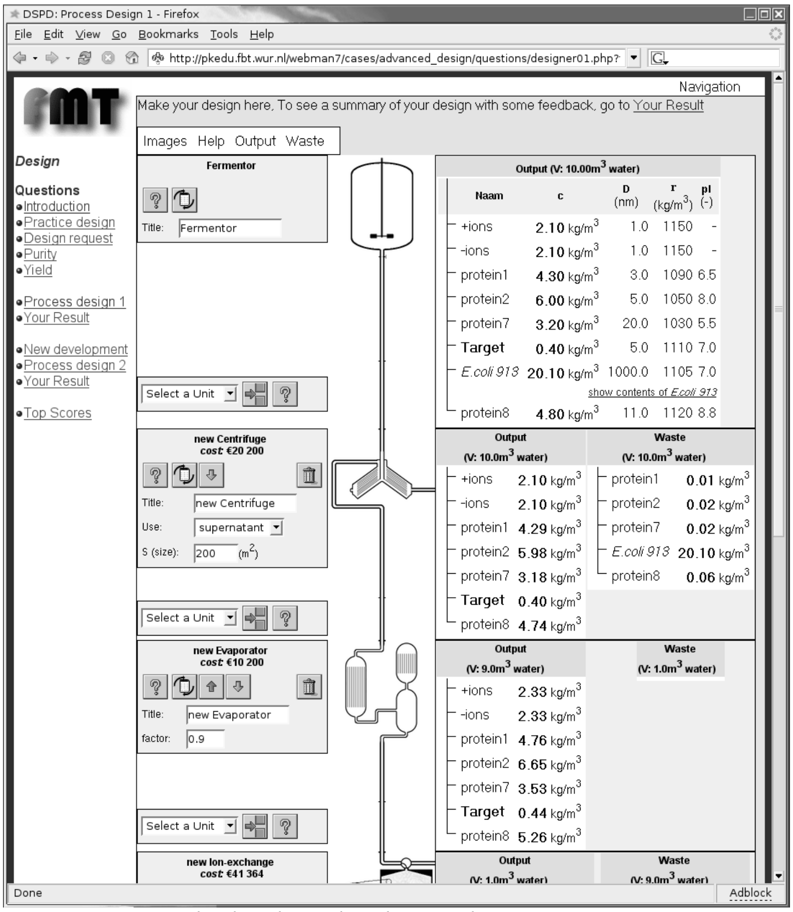


Figure 2: A system that lets the student design a downstream process.

In this learning material the assignment is clear. The process the student wants to follow is completely free; the student can try any approach. The end result of the process is also free. Within the restrictions set for the design, the student is free to optimise on any of the design parameters in order to get a top-score. The student does not have to do any calculations himself and can concentrate on “playing” with the different unit operations in his design.

Evaluation showed that almost all students liked to work with the DSPD; they found it challenging. Especially the competitive element introduced by the top-score list motivated the students to take a better look at their design.

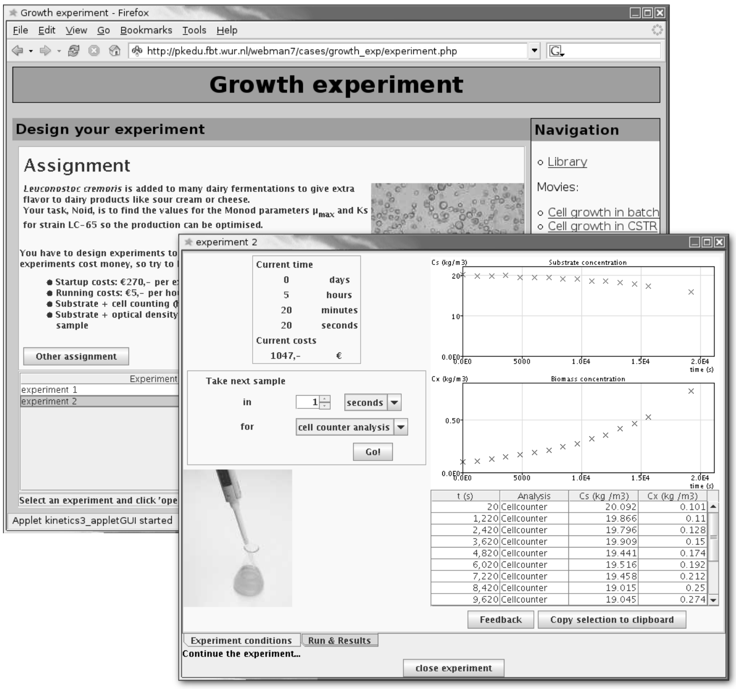


Figure 3: An environment that allows students to do cell-growth experiments.

Cell-growth experiments

This material for 2nd year BSc students can be found on the FBT website [2] in "Content Showcase", "Cell growth experiment design". In process engineering education the topic about the design of experiments was missing in the curriculum. When teaching how to design experiments you want students to practice designing their own experiments and see the results of their experiments. A problem in Bioprocess Engineering is that experiments in cell growth usually take a long time to execute, so it is not feasible to let students design and execute many experiments.

To address this problem, new learning material has been created [10, Figure 3]. The learning objectives for this material are:

- Getting a feel for the behaviour of an often used cell-growth model.
- Learning to design experiments that result in the required data.
- Learning to process data that have an experimental error.

The students already know the model on a theoretical basis; they already know what equations the model is composed of, but they do not really have a feel for what this means in practice.

An environment was created where students can carry out cell-growth experiments. The students have to determine several organism-specific parameters for a growth model of microorganisms, such as the maximum growth rate and the biomass yield. To determine these parameters, they have to do growth experiments with the microorganism. Students will have to learn how the model works and what kind of experiments are needed to determine the desired parameters. The measurements the students do during these experiments have a random error applied to them, to simulate experimental error. The students will have to take these measurement errors into account and design their experiments accordingly.

Evaluation shows that the students find that the virtual experiments support the learning objectives well and that they are useful. The lecturers involved in the courses in which the experiment environment is used, were enthusiastic about the virtual experiments for the support of essential learning objectives.

Designing models

This material for 3rd year BSc and 1st year MSc students can be found on the FBT website [2] in “Content Showcase”, “Experimental material on Systematic Model Design”. At Wageningen University the curriculum for bioprocess engineering lacked adequate material to teach students how to design a model in a structured way. In general, lecturers find it difficult to teach students how to design models and students find it difficult to learn how to design models. One of the reasons for this is that students that are just learning how to design a model do not yet have a good overview over all the steps involved in the design of a model. As a result the students often do not know where to start, or what to do next. The lecturers on the other hand tend to “skip” steps in the modelling process because they can do those steps without much conscious thought. This makes it difficult for a student to follow what the lecturer is doing.

When learning how to design a model, the focus of the student's learning has to be on the design process and not on the details of the subject being modelled or the mathematics of the model. Students have to practice their design skills on a subject. It is important that this subject offers the student enough challenge so that he has to use the full modelling process to solve the exercise. If the subject is not complex enough, the students will tend to skip steps in the process, or will find steps tedious and not useful, and will thus not practice the full modelling methodology that they have to learn. If the subject is too complex, the student will get stuck in the details of the subject and will not be able to focus his attention on the overall modelling process. In addition, when designing a mathematical model, the student should not spend a large part of his time figuring out how to execute a certain mathematical procedure in the specific modelling environment used.

To help students when learning to design models, a system was developed that helps students to design a model in a systematic way [11, 12, Figure 4]. To help the student keep an overview over his model-design process, the model design process is divided into four steps: general analysis, detailed analysis, model composition, model evaluation. Each step is divided into several sub-steps. Each step is supported by an activity for the student that generates feedback that helps the student in his design. The analysis steps contain open and closed questions. The composition step contains tools for the student to enter his model and check his model for consistency in the used mathematics and units. The analysis step contains tools that allow the student to run simulations with his model, or to search for steady state situations.

In this learning material, the procedure the student follows is not entirely free. The student is presented with a fixed set of steps, but he can freely choose his path through these steps. The end result of the design process is free; each student can design a different model, though the variation in models is not very high in practice because the models do have to match the design requirements. Every step of the design process is present in a clear navigation structure. The student gets feedback on every step in the process. Activities that cause extraneous cognitive load, like mathematical and computational operations and unit checks, are automated as much as possible.

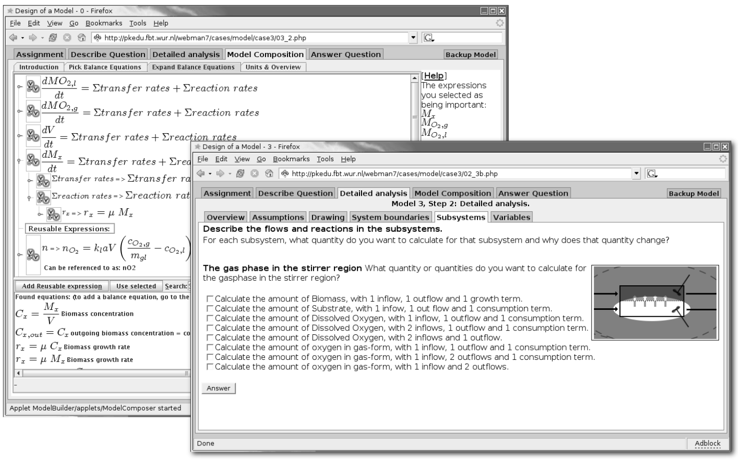


Figure 4: an environment to teach students model design.

Evaluation showed that students liked the explicit structure the system gave to their design process. They also appreciated the fact that they did not have to worry about mathematics and that they had an easy way to check the units of their model. They also noted that they would like to see more use of the system in their process-engineering courses.

Concluding remarks

The base for the design of any learning material is the learning objectives that the material has to support. Designs based on different learning objectives can thus lead to different types of learning materials. The more complex the learning objective, the more complex the design process of the learning material may be. This applies to all learning materials, thus also to the design of digital learning material.

In the design of digital learning material one should keep in mind that the learning material should help the student focus on the actual learning objectives. The student should not be distracted by elements that are not part of the learning objectives. In addition, any activity that does induce cognitive load, but that is not supporting the learning objectives should be automated in the learning material, or at least handled in such a way that this activity does not hamper the student while using the material.

When learning, students should be actively training the skills they have to develop. The degree of freedom the student gets when working with the learning material should match the degree of freedom that is normally associated with the skill that is being trained. Complex, open-ended learning objectives that include a high degree of freedom, such as design, require open-ended learning materials that offer the student that degree of freedom. On the other end of the spectrum, learning objectives with a low degree of freedom and few internal relations between different aspects do not need learning material that offers such high degree of freedom. These learning objectives can be efficiently supported with an adaptive system like Proteus.

These guidelines for the development of digital learning material lead to a very diverse set of learning materials as can be seen from the examples discussed in this paper. These materials can be used in various educational settings, from demonstration material during a lecture, to a computer lab with a lecturer present, to independently-used self study material. The amount of contact between student and lecturer is not necessarily changing when introducing digital learning material in a course. Faculty and students perceived that, depending on how the material is used, the contact is more effective.

The material presented here has been successfully used at Wageningen University as well as at other universities in Europe, namely EPFL in Switzerland and Technical University of Lodz in Poland. At Wageningen University the material is mostly used in a computer-lab setting, with students alone or in pairs behind a computer and a lecturer present to answer any additional questions. Students comment that they like the material because it allows them to work at their own speed. Most of the problems they face are remedied by the material, and only if that is not sufficient, or if the student is interested in aspects of the subject that fall outside of the scope of the learning objectives, they need extra support from the professor. Faculty comment that they can focus their attention much better on the students that have problems in understanding the subject treated, or students that have questions that go beyond the level required for the course, without holding back the other students. They also observed that students are less distracted and much more focused on the learning material used in computer classes, in comparison with traditional tutorial classes.

The development of digital learning material has led to a further increased attention to education in general at the Process-Engineering department. The new possibilities offered by web technology inspired many of the faculty to rethink all their learning material and teaching strategy and not just the parts that could be digitised.

The development of the digital learning material offers the opportunity to support learning objectives that were difficult to support with traditional learning material, like designing experiments and model building.

In the authors' experience, the development of digital learning material does not take much more time than the development of traditional material. Most of the time needed to develop learning material is used for the analysis of the learning objectives and the specification of the learning activities. The actual digitisation of the material takes only a very small part of the total development time.

References

1. Meriënboer, J. J. G. van. (1997) "Training complex cognitive skills. A Four-Component Instructional Design Model for Technical Training", Educational Technology Publications, Englewood Cliffs, New Jersey, ISBN 0-87778-298-9.
2. FBT project homepage: http://fbt.wur.nl (as of 2006).
3. Zeigler, B.P., Praehofer, H. & Kim, T.G. (2000) "Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems", 2nd ed: Academic Press, ISBN 0127784551.
4. Narciss, S. & Huth, K. (2003) "How to design informative tutoring feedback for multi-media learning", R. Brünken H. Niegemann and D. Leutner, editors, Instructional Design for Multimedia Learning. Waxmann, Münster.
5. Bennett, S., Harper, B. & Hedberg, J. (2002) "Designing real life cases to support authentic design activities", *Australian Journal of Educational Technology*, 18(1), 1-12. http://www.ascilite.org.au/ajet/ajet18/bennett.html
6. Herrington, A. & Herrington, J. (2006) "Authentic Learning Environments in Higher Education", Edited by Anthony Herrington and Jan Herrington, pp. 321, Information Science Publishing, Hersley, ISBN 1-59140-595-5
7. Sessink, O.D.T., Beeftink, H.H., Hartog, R.J.M. & Tramper, J. (2006) "Proteus: A Lecturer-Friendly Adaptive Tutoring System", *Journal of Interactive Learning Research* Accepted for Publication.
8. Jong, T. de & Joolingen, W. R. van (1998) "Scientific discovery learning with computer simulations of conceptual domains", Review of Educational Research Summer, 68(2), 179-201.
9. Schaaf, H. van der, Vermuë, M., Tramper, J. & Hartog, R. (2003) "A design environment for downstream processes for Bioprocess-Engineering students", *European Journal of Engineering Education*, Taylor & Francis, vol. 28, nr. 4, pp. 507-521.
10. Sessink, O.D.T., Beeftink R., Tramper J. & Hartog R. (2006) "Virtual Parameter-Estimation Experiments in Bioprocess-Engineering Education", *Bioprocess and Biosystems Engineering*. Accepted for Publication.
11. Schaaf, H. van der, Vermuë, M., Tramper, J. & Hartog, R. (2006) "Support of Modelling in Process-Engineering Education". *Computer Applications in Engineering Education*, Vol. 14, No. 3, pp. 161-168.
12. Schaaf, H. van der, Vermuë, M., Tramper, J. & Hartog, R. (2006) "A digital tool set for systematic model design in Process-Engineering education". *European Journal of Engineering Education*, Vol. 31, No. 5, pp. 619-629.

Technical implementation

This chapter describes the technical aspects of the development of digital learning material. It explains the basics of the technology used in web-based digital learning material and how the technical aspect of the creation of digital learning material factors into the total investment needed for the design of the learning material. Learning Management Systems can reduce the cost of creating digital learning material, but their support for complex learning material is yet too limited. The cost for the creation of digital learning material can be greatly reduced if components and tools can be re-used.

Introduction

Designing learning material starts with the didactic design. After the material has been designed from a didactic point of view it has to be made available to the student in a functional form. If the material is a book, it needs to be printed, bound in a cover and shipped to the shops. If the material is a video, the actors play out the script in front of the camera, the scenes are edited, DVD's are pressed and shipped. Likewise, if the material is digital, using web technology, the web pages are made, any required programs are programmed and the material is installed on the web server.

The creation of digital learning material that is innovative, both from a didactic and technical point of view, requires a serious investment both on the didactic side as the technical side. In this chapter the technical side of the development of web-based digital learning material is addressed.

Web technology

We use the term web technology when the material is located on a server computer and is accessed from a client computer, using a web browser. Most web content can be accessed from any computer with an internet connection, but it is possible to restrict the access based on where the client computer is located. Most intranet websites for instance can only be accessed from within the company network, and not just from any computer with an internet connection anywhere in the world.

It is possible to place any type of digital content on a web server, but not all types of content are considered to use web-technology; the content must be usable by the web browser. A MS-Word™ document placed on a website can only be viewed if the client has the MS-Word™ program and the file has to be downloaded and then opened in MS-Word to be viewed (though the browser may do this automatically). The word document can then no longer communicate with the web page and the server.

Some types of content can not be displayed by the browser itself, but need a “plug-in” to be displayed. Examples of this are Flash applications and Java applets, but there are many more. These types of content are considered web technology, because the way the plug-in is integrated into the browser allows the content to communicate with the browser and the website. The use of these types of content has to be carefully considered because some users may not have the correct plug-in installed on their computer and thus might not be able to use the content because of that. The Flash and Java plug-ins are examples of plug-ins that are installed on most computers by default, so using those is unlikely to cause problems. Java and Flash are therefore often used.

Client-side and server-side technology

A client in web-technology is a program or computer that can send a request for data to a server. A server is a program or computer that listens for requests for data from clients. A server is permanently ready for requests and the client can send a request to the server at any time. The client and server can be different computers, but sometimes the client and server are different programs that run on the same computer. The context in which the words client and server are used should make clear whether they refer to computers or computer programs or both.

Dynamic web-based applications and content can be divided into two groups: applications and content that uses server-side technology and applications and content that uses client-side technology. Of course all web material is stored on a web server and displayed in a web browser, so it all uses a bit of both. The distinction is made on the basis of where the programming code that makes the material dynamic, is running; does this code run on the web server, or does it run in the web browser.

Examples of server-side technology are the Downstream Process Designer, described in chapter 2 of this thesis, and most learning environments, like Blackboard®. From the browser's point of view, the pages generated by these systems are just html pages. These pages could have been stored on the web server just as the browser receives them. When a user follows a link to one such page, however, a program on the web server gathers all information that is needed to create the web page. This information can be the exact link that the user clicked on, it can be information of who the user is and what the user has done previously, or it can be information stored in databases. The program on the web server then uses all this information to create the page that is send to the browser. The page can be different for each user. It can also be different each time the user visits the same page.

Examples of applications that are based on client-side technology are Flash applets and Java applets, like the ModelBuilder and ModelRunner applets, described in chapters 3 and 4 of this thesis. From the web server's point of view, these applets are just files containing data, like any other file. When a user follows a link to one such applet, the web server just sends the file of the applet to the web browser, without doing anything with it. When the web browser receives the applet, it starts the plug-in associated with the applet, and the plug-in then starts to execute the code in the applet. The code in the applet, running on the user's computer, can then, for instance, run a model and display the results to the user.

Some dynamic content has parts that are client-side and parts that are server-side. Some Java applets need to store some data, like the ModelBuilder that needs to store the student's model and the ModelRunner that needs to fetch the student's model so it can run simulations with it. These data can best be stored on the web server, because on his next visit the user might be using a different desktop computer. If the user's data where stored on the desktop computer that the user used the last time, he would not be able to access it this time. To store information on the web server, there has to be a program on the web server that can communicate with the Java applet in order to store and retrieve this information. Whenever we want a user to be able to access his own data from different client machines (i.e. desktop or laptop computers) this set-up is required. This set-up requires that content is coordinated on both the server- and client-side and the communication between the two parts introduces additional complexity.

Learning management systems

Almost every university in the world uses a Learning Management System (LMS), or Virtual Learning Environment. An LMS is a website that facilitates easy communication and on-line distribution of material between students and lecturers. Most commonly deployed LMS's enable the student to log-in using their web browser and see his course schedules, download course material, communicate with lecturers and other students and do other things depending on the features that the LMS supports.

So far, most of these systems have focused on the authorised management of content, authorised access to content, student management and communication aspects of digital learning. These systems are mostly content-distribution systems that do not (yet) have much support for innovative and interactive learning materials. Most teachers use them to distribute their lecture notes, make their lecture schedule available to students, support groups of students working together and communicate assignments and grades.

There is much potential for the support of interactive, activating, personalised learning material beyond the support that current LMS's offer. Advanced material could greatly benefit from the functionality that LMS's already do offer: distribution and management. Often, new learning material needs to store data, like the student's answers, somewhere in a database [1]. Current learning environments do not yet have standard methods for learning material to do this. Some learning environments do offer facilities to extend the functionality of the learning environment. For these learning environments an extension can be made that allows a learning object to store data in a database, but these extensions are learning-environment specific and need to be developed for each separate learning environment. There are standards in development that address this problem [1, 3, 4], but these standards are not finalised yet and support for these standards is not yet fully implemented in the current learning environments.

If the learning material requires certain facilities, like the ability to store data, and the LMS that is used does not have these facilities by default, these features will have to be created either as a plug-in for the LMS, or in a separate website outside of the LMS.

It is to be expected that, as learning environments continue to be developed, better, standardised facilities will become available for learning material to communicate with the LMS and advanced learning material will become easier to create.

Developing material

There are several factors that determine how complex and time consuming the development of digital learning material is. It is the author's experience that in most cases, after the didactic design of the material has been done, the actual realisation (i.e. “digitising”) of the learning material takes relatively little time. The effort needed for the technical implementation of the learning material depends both on the type of learning objectives that need to be supported and on the tools and components that are available that can be used in the development of the material.

The type of learning objectives that need to be supported is one of the first things that determine the level of complexity of the technical design of the learning material. In general, material with a high degree of freedom is technically more complex to implement than material with a low degree of freedom. Presenting a student with a multiple choice question does not involve much programming, but giving a student the ability to design a model or a downstream process requires considerably more effort. Giving a student automated feedback based on a design also requires much more effort than giving a student automated feedback based on his choice in a multiple-choice question.

With respect to the availability of tools that can be used in the development of the learning material we can identify three situations:

- Material that can be created in an LMS.
- Interactive material that has to be created outside of an LMS and uses previously developed components or tools.
- Interactive material that has to be created outside of an LMS, that is all new.

Of these three, the technically easiest material to develop is material that can be authored in an existing LMS. The tools available in an LMS have a user-friendly user-interface and even a lecturer who has little computer experience can use them to create his learning materials. Most of the time needed to develop material of this kind will be spend on the didactic design and the digitisation takes relatively little time.

The development of material that re-uses components and tools that have been created previously, but that does exceed the standard capabilities of LMS's is more complex. These tools and components can be commercially available third-party software, like Macromedia Authorware™ or self-built software like Protheus [5] or the ModelDesigner [8]. Depending on how well developed the tools and how generic the components are, using them in learning material may require someone with experience in programming and developing websites. The lecturer knows exactly how the learning material is supposed to behave, as he has seen how the tools and components are used elsewhere, and as a result the programmer can follow the exact instructions of the lecturer and does not need to know anything about the subject that the learning material covers. The programmer can use the previously made material as an example of how to implement the new material.

Because of this, the technical implementation of material that re-uses components and tools that have been created previously takes relatively little time. To create a new topic for a system like the adaptive system Proteus [5], a teacher only needs to design a new set of questions, and enter those into the system. In this case there is no programming that needs to be done and the digitising of the questions takes less then 5% of the total time investment. To implement a digital case as described in chapter 1, some knowledge of web technology is needed to create the layout for the setting of the case, the photo's used in the case and the navigation through the case, but no advanced programming is needed as the components that handle the questions can be re-used. Most of the total development time of the case will be taken up by writing the storyline and designing the questions and feedback.

The most effort is needed for the technical implementation of new learning material that is too different for any existing tools or components to be useful. This takes more effort and often requires more programming than most lecturers are able to do themselves. For instance, the creation of a design environment that allows students to design a downstream process [6], experiments [2], or a set of tools that allow students to design a model [7, 8], involves considerable programming effort. If the lecturer who does the didactic design of the material has no experience with web technology and programming, and the programmer does not have any knowledge about the subject the learning material covers then very intensive contact between the lecturer and the programmer is necessary. In this case the lecturer does not have a good idea about what is possible with web technology and the programmer has no good idea about what is needed to teach the subject to students. If the programmer does have experience with the subject then he can actively suggest ideas to the lecturer and the lecturer will have to do a lot less explaining about how the material should work for the student.

How it is applied

The material described in this thesis uses a combination of client- and server-side technologies and is implemented in a separate website. To facilitate the students at Wageningen University that uses an LMS for a lot of educational functions, an extension for this LMS was created that allows students to use the external learning material as if it where integrated with the LMS.

Where possible, the “tools and components” approach was used. For instance, for the questions in the cases, a code library was created that can read a question and its feedback from an easy to create XML file. This way lecturers can create and edit questions themselves and digitising a new case after it has been designed by a lecturer is relatively little work. Another example is the set model design tools described in chapter 3 and 4. If a lecturer wants to show the student a model in a new case, these tools can be used to do this.

Like the development of high quality, traditional learning material, the creation of high quality, activating, digital learning material is often considered expensive, but there are ways to make this investment worthwhile. If the programming of innovative material is done in a modular way, this investment only needs to be made once and the resulting tools can be re-used in other learning material as well. The use of digital learning material should not be limited to just one university. All universities have internet connections so the material of one university has the potential to be used on any university in the world. This makes the development of innovative digital learning material very suitable for cooperative projects with other universities. If several universities develop material using the same components and tools, they can share the costs of developing those tools and by using the same tools it becomes easier to share learning material. Sharing learning material between universities greatly increases the use of the material and this can lead to an improvement of the quality of the learning material that both universities will benefit from. This potential audience for the learning material can be maximised by minimising the prerequisite knowledge that is needed to use the material.

References

1. 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), Accepted for Publication.
2. Sessink, O.D.T., Beeftink R., Tramper J. & Hartog R. (2006) “Virtual Parameter-Estimation Experiments in Bioprocess-Engineering Education”, *Bioprocess and Biosystems Engineering*. Accepted for Publication.
3. Advanced Distributed Learning (2004) Sharable Content Object Reference Model (SCORM) 2nd Edition., Retrieved Jan. 2005 from http://www.adlnet.org/
4. IMS, Question & Test Interoperability Specification, Retrieved 03-2006 from http://www.imsglobal.org/question/index.html
5. Sessink, O.D.T., Beeftink, H.H., Hartog, R.J.M. & Tramper, J. (2006) Proteus: A Lecturer-Friendly Adaptive Tutoring System, *Journal of Interactive Learning Research* Accepted for Publication.
6. Schaaf, H. van der, Vermuë, M., Tramper, J. & Hartog, R. (2003) A design environment for downstream processes for Bioprocess-Engineering students, *European Journal of Engineering Education*, Taylor & Francis, vol. 28, No. 4, pp. 507-521.
7. Schaaf, H. van der, Vermuë, M., Tramper, J. & Hartog, R. (2006) “Support of Modelling in Process-Engineering Education”. *Computer Applications in Engineering Education*, Vol. 14, No. 3, pp. 161-168.
8. Schaaf, H. van der, Vermuë, M., Tramper, J. & Hartog, R., (2006) “A digital tool set for systematic model design in Process-Engineering education”. *European Journal of Engineering Education*, Vol. 31, No. 5, pp. 619-629.

Samenvatting

Met de opkomst van computers en het Internet kan een nieuw type leermateriaal ontwikkeld worden: web-gebaseerd, digitaal leermateriaal. Omdat veel complexe leerdoelen in het levensmiddelen- en bioprocestechnologie domein moeilijk te bereiken zijn is er een project gestart om te verkennen wat de mogelijkheden zijn om deze leerdoelen te ondersteunen met digitaal leermateriaal. Het materiaal dat ontwikkeld is, de keuzes die geleid hebben tot dit materiaal en de conclusies die getrokken zijn worden besproken in dit proefschrift.

In een voorgaand project was leermateriaal ontwikkeld dat bestaat uit lineaire casussen, waarin de student in de rol van Junior consultant geplaatst wordt. In deze rol wordt de student uitgezonden naar een fictief bedrijf om daar een realistisch probleem op te lossen. De web-gebaseerde casussen introduceren de student in zijn rol, en leiden hem door het probleem, door hem vragen te stellen. Als de student een vraag incorrect beantwoordt, krijgt hij feedback. Bij nogmaals incorrect antwoorden wordt de feedback specifieker. Nadat hij het correcte antwoord gegeven heeft, krijgt de student uitleg waarom dat specifieke antwoord correct is. Door de student deze rol bij een virtueel consultancy bureau te geven wordt hij gestimuleerd om actief met het materiaal bezig te zijn en meer over het onderwerp te leren. De adaptieve feedback houdt de student gemotiveerd en actief. Animaties en simulaties zijn zeer geschikt om de vele modellen te verduidelijken, die binnen de (bio)procestechnologie gebruikt worden. De modulaire opzet van het materiaal maakt het materiaal flexibel inzetbaar.

Dit proefschrift beschrijft de voortzetting van dit onderzoeksproject. Binnen het proceskunde onderwijs werden een aantal leerdoelen nog niet goed ondersteund door het beschikbare leermateriaal en ook niet door de ontwikkelde digitale, lineaire casussen. Een van de leerdoelen was het ontwerpen van processen voor het opwerken van een biotechnologisch product. Lineaire casussen met alleen gesloten vragen van het standaard type (meerkeuze, meerkeuze met meerdere antwoorden, exacte waarde, etc.) zijn niet erg geschikt om studenten te leren ontwerpen, omdat ontwerpen een niet-lineair, open proces is. Om dit probleem te ondervangen is een digitale ontwerpomgeving ontwikkeld, genaamd de DownStream Process Designer (DSPD). De DSPD geeft studenten de mogelijkheid om een opwerkingsproces te ontwerpen. De student kan standaard procesapparatuur aan elkaar koppelen en deze apparatuur vervolgens instellen om zo een proces te creëren dat aan de gestelde specificaties voldoet. De DSPD is gebruikt in een digitale casus, waarin de student een proces moet ontwerpen waarin een eiwit wordt geïsoleerd en gezuiverd uit een mengsel van stoffen, tot een minimaal vereiste zuiverheidsgraad. Het verlies aan product moet tevens binnen gestelde grenzen blijven, de kosten van het proces moeten binnen een gegeven budget blijven en de hoeveelheid geproduceerd afval moet worden geminimaliseerd. Door de top-scores van de beste ontwerpen te presenteren aan het einde van de casus wordt een spel element in de casus gebracht dat de studenten stimuleert om nogmaals over hun ontwerp na te denken om zo zelf een top-score te behalen. Studenten waren zeer te spreken over de mogelijkheid om zelf een ontwerp te maken en de gevolgen van hun aanpassingen direct te zien.

Een ander bioprocestechnologisch leerdoel dat ook moeilijk te bereiken is, is het ontwerpen van modellen. Studenten weten vaak niet waar ze moeten beginnen, of hoe ze verder moeten als ze bezig zijn met het ontwerpen van een model. Docenten die zelf zeer ervaren zijn in het ontwerpen, slaan in hun uitleg vaak stappen over, omdat ze deze stappen onbewust maken. Er is daarom een stapsgewijze methode voor het ontwerpen van modellen ontwikkeld en geïmplementeerd in digitaal leermateriaal, waarin de student bij iedere stap ondersteuning krijgt. Twee onderdelen van dit digitale hulpmiddel worden in detail beschreven. Het ene hulpmiddel helpt de student bij het samenstellen van zijn model en het invoeren in de computer en het andere kan de student gebruiken om simulaties uit te voeren met zijn model. De stapsgewijze aanpak en de verschillende hulpmiddelen zijn geïmplementeerd in een ontwerp-gerichte casus over zuurstofoverdracht in een bioreactor en deze casus is gebruikt in het onderwijs. Uit evaluatie bleek dat de stapsgewijze aanpak en de ondersteunende activiteiten voor iedere stap de studenten hielpen bij het houden van overzicht over hun ontwerpproces. De studenten waren ook zeer te spreken over het feit dat ze bij het ontwerpen niet afgeleid werden door bijzaken, zoals wiskundige bewerkingen.

Uit het ontwikkelen en het gebruik van het beschreven materiaal zijn leidraden voor het ontwerpen van digitaal leermateriaal geïnduceerd. De eerste stap bij het ontwerpen van nieuw digitaal leermateriaal, is het analyseren van de leerdoelen die moeten worden ondersteund door het te ontwikkelen materiaal. Leerdoelen kunnen geclassificeerd worden naar de vrijheidsgraad die inherent is aan de onderwerpen en vaardigheden die in de leerdoelen beschreven zijn en leermateriaal moet een passende vrijheidsgraad bieden aan de student. Onderwerpen en vaardigheden die een lage vrijheidsgraad in denken en doen van de student vereisen, kunnen effectief ondersteund worden door eenvoudige adaptieve systemen, of door lineaire casussen met gesloten vraagstukken en adaptieve feedback. Vaardigheden die een grote vrijheidsgraad impliceren, zoals ontwerpen, hebben leermateriaal nodig, dat de student die vrijheidsgraad biedt. Leermateriaal moet de student helpen zich te concentreren op de leerdoelen. Taken die niet direct gerelateerd zijn aan de leerdoelen moeten daarom zoveel mogelijk geautomatiseerd worden.

Na het maken van het didactische ontwerp wordt leermateriaal gedigitaliseerd en beschikbaar gesteld aan de studenten. Vanzelfsprekend is het belangrijk om al bij het didactisch ontwerp de technische limitaties in het achterhoofd te houden. De technische aspecten van web-gebaseerd digitaal leermateriaal en hoe sommige van deze aspecten bijdragen aan de totale investering die nodig is voor het creëren van hoogwaardig digitaal leermateriaal worden besproken. Een van de mogelijkheden die kosten kunnen reduceren is het hergebruiken van componenten en hulpmiddelen die al eerder ontwikkeld zijn door het instituut zelf, of door een externe partij. Samenwerken met andere universiteiten, om zo de ontwikkelkosten te delen en een grotere doelgroep te creëren voor het materiaal, is een goede manier om de balans tussen de kosten per gebruiker en de kwaliteit van het leermateriaal te verbeteren.

Een pagina die toegang geeft tot al het materiaal dat besproken wordt in dit proefschrift kunt u vinden op: http://pkedu.fbt.wur.nl/hylke/thesis.

Nawoord

Het is alweer ruim zeven jaar geleden dat ik afstudeerde in September 1999 en dat ik reageerde op een email die Rob rond had gestuurd, dat hij iemand zocht “voor wat html werk”. Vier dagen later had ik een aanstelling voor vijf maanden, dat werd een jaar en toen werd het een Aio plaats... En dat terwijl ik tijdens mijn studie altijd had geroepen dat ik geen Aio zou worden. Ok, het was geen full-time Aio positie, maar slechts vier dagen in de week, aangezien mijn werkzaamheden dat eerste jaar ook veel technische ondersteuning was voor andere vakgroepen en iemand dat toch moest blijven doen. Eén dag in de week bleef ik dus gewoon medewerker.

Aio zijn, met als doel het ontwerpen van innovatief digitaal leermateriaal was af en toe best lastig, want niemand wist eigenlijk hoe dat er uit zag, innovatief digitaal leermateriaal. Gelukkig had ik begeleiding van twee heel verschillende personen, Marian en Rob, zodat ik altijd wel bij een van de twee terecht kon voor mijn problemen. Bij Marian kon ik altijd terecht voor een didactische en proceskundige kijk op de zaak, terwijl Rob altijd oog had voor de technische kant en me altijd weer kon wijzen op punten die we als proceskundigen soms al te makkelijk voor bekend aannemen. Bij Marian kon ik ook altijd terecht als ik weer een groepje studenten nodig had om mijn materiaal te testen. Van beiden heb ik veel geleerd over ontwerpen, onderzoek en artikelen schrijven, daarvoor wil ik ze beiden dan ook hartelijk bedanken.

Hans en Rik wil ik bedanken voor hun commentaar en suggesties. Hans heeft zich nooit uit het veld laten slaan door de technische kanten van de artikelen. Zonder zijn inzet was dit promotie traject nooit van de grond gekomen. Ook Tinri, Julia, Cora, Bert-Jan, Gerard en Ayalew wil ik bedanken voor de interessante discussies over onderwijs en aanverwante zaken en de goede sfeer binnen het FBT project.

Voor de echt technische discussies en nerd-talk was Olivier natuurlijk altijd beschikbaar, daarnaast hebben we vele buitenlandse reizen gemaakt naar congressen en andere universiteiten. Zes jaar lang waren Olivier en Sebastiaan goede kamergenoten in het Biotechnion, met uitstekende muziek smaak en goed voor vage discussies tijdens de lunches in het arboretum en over een pannenkoek bij Unitas. Hiervoor wil ik natuurlijk ook Dione en Marleen bedanken. Ook Jeroen blijkt een aardig woordje nerd-talk te spreken. Bedankt voor de vele discussies over camera's, monitoren en whisky.

De proceskunde-onderwijs-afstudeer-studenten Koos, Leandro, Bart en Tamara wil ik natuurlijk bedanken voor hun waardevolle input en hun goede beslissing om het ontwikkelen van digitaal leermateriaal als afstudeervak te kiezen. Dankzij alle medewerkers van Proceskunde heb ik een zeer leuke tijd gehad op de vakgroep. Er is heel wat geborreld, gebombercloned, koffie gedronken en gefotografeerd, er zijn een hoop Aio-reizen, brainstorm weken, lab-uitjes en we-days gepasseerd. Bedankt voor de goede sfeer.

Buiten het werk wil ik “De Vriendjes” bedanken. Michiel, David, Hendrik en Maurice, de verjaardagen, bond-films en oud-en-nieuw vieringen zijn altijd weer een gezellig weerzien. De D&D spelers en spellen spelende, (niet al te)hardlopende (ex)Schermers wil ik ook hartelijk bedanken voor alle gezelligheid.

Tot slot wil ik Michiel en Wietske hartelijk bedanken dat ze mijn paranimfen willen zijn, Arjen voor het ontwerpen van de kaft van dit boekje en mijn ouders voor alle steun, aanmoedigingen en advies.

Publications

1. Hartog, R., Gooijer, C.D. de, Schaaf, H. van der, Sessink, O., Vonder, O.W. (2000) Comparing Web based Course Development With and Without a Learning Environment, Webnet 2000: World conference on www and the internet, San Antonio, Gordon Davies & Charles Owen: 240-247.
2. Schaaf, H. van der, Sessink, O., Vermuë, M., & Tramper, J. (2001) Student aan het werk binnen Wageningen UR virtual consultancy. Ontwikkeling van digitaal leermateriaal. *NPT Procestechnologie* 2: 13 - 15.
3. Schaaf, H. van der, Vermuë, M., Tramper, J. & Hartog, R. (2003). A design environment for downstream processes for Bioprocess-Engineering students, *European Journal of Engineering Education*, Taylor & Francis, vol. 28, nr. 4, pp. 507-521.
4. Hartog, R., Schaaf, H. van der, et al. (2003) "eLearning" *Agro Informatica* 16(4): 9-11.
5. Schaaf, H. van der, Vermuë, M., Tramper, J. & Hartog, R. (2006) Support of Modelling in Process-Engineering Education. *Computer Applications in Engineering Education.* Vol. 14, No. 3, pp. 161-168.
6. Schaaf, H. van der, Vermuë, M., Tramper, J. & Hartog, R., (2006) A digital tool set for systematic model design in Process-Engineering education". *European Journal of Engineering Education.* Vol. 31, No. 5, 619-629.
7. Sessink, O.D.T., Schaaf, H. van der, Beeftink, H.H., Hartog, R.J.M., Tramper, J. (2007) Web-based education in bioprocess engineering. *Trends in Biotechnology*, accepted for publication.
8. Schaaf, H. van der, Sessink, O.D.T., Vermuë, M., Tramper, J., Hartog, R., (2007) "Design of activating digital learning materials to support complex learning objectives", Submitted for publication.
9. Summary page giving access to all material described in this thesis: http://pkedu.fbt.wur.nl/hylke/thesis.

Curriculum Vitae

Hylke van der Schaaf werd geboren op 29 augustus 1976 te Hellevoetsluis. In 1994 behaalde hij het VWO-diploma aan de Christelijke Scholen Gemeenschap Jacob van Liesveldt in Hellevoetsluis. In datzelfde jaar begon hij met de studie Bioprocestechnologie aan de toenmalige Landbouwuniversiteit Wageningen, met als specialisatie Bioreactoren. Na afstudeervakken bij de sectie Meet-, Regel- en Systeemtechniek en de sectie Proceskunde en een proceskundige stage bij Diosynth in Oss, kreeg hij zijn ingenieurs titel in September 1999. Direct na zijn afstuderen ging hij werken aan onderwijsinnovatie met behulp van web-technologie binnen het onderzoeksprogramma Food- and Biotechnology. In 2000 werd dit werk voortgezet in een promotietraject, dat beschreven is in dit proefschrift. Naast dit promotietraject was hij werkzaam bij bij het Wageningen Multimedia Research Centre waar hij advies en ondersteuning gaf op het gebied van ICT en web-technologie. Hier is hij tot op heden werkzaam als ontwikkelaar van educatieve software.

Hylke van der Schaaf was born on August 29, 1976 in Hellevoetsluis. In 1994 he completed the secondary school at the CSG Jacob van Liesveldt in Hellevoetsluis. In the same year he started with the MSc curriculum Bioprocesstechnology at the (former) Wageningen Agricultural University, The Netherlands. During his MSc he specialised in bioreactors, did his MSc theses in the Systems and Control Group and the Process engineering group and an internship at Diosynth in Oss. He graduated in September 1999. After his graduation he started working on the development of digital learning materials in the Food- and Biotechnology research project. In 2000 this work changed into a PhD project of which the results are described in this thesis. Besides this PhD project he worked for the Wageningen Multimedia Research Centre, giving advice and support on the field of ICT and web-technology. He is still employed here as a developer of educational software.