European Journal of Engineering Education
Vol. 28, No. 4, December 2003, 507–521

A design environment for downstream processes for Bioprocess-Engineering students

H. VAN DER SCHAAF[†]*, M. VERMUE[†],
J. TRAMPER[†] and R. HARTOG[‡]

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.

1. 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 (Irandoust and Sjöberg 2001). This adds a new perspective to the use of information and communication technology in engineering education (Brandt 2001). 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 (http://www.fbt.eitn.wau.nl/).

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

---

[†] Food and Bioprocess Engineering Group, P.O. Box 8129, 6700 EV Wageningen, The Netherlands.

[‡] Bode 143, P.O. Box 9101, 6700 HB Wageningen, The Netherlands.

* To whom correspondence should be addressed. e-mail: hylke.vanderschaaf@wur.nl

European Journal of Engineering Education
ISSN 0304-3797 print/ISSN 1469-5898 online © Société Européenne pour la Formation des Ingénieurs (SEFI)
http://www.tandf.co.uk/journals
DOI: 10.1080/03043790320001O1863

elaborate on design knowledge. There are a number of Process-Engineering textbooks having the term 'Design' in their title (Van 't Riet and Tramper 1991, Asenjo and Merchuk 1994, Cabral *et al.* 2001), 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 that are offered to them.

2. 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 (Jones 1984, Chandrasekaran 1990, Dasgupta 1991, Simon 1996, Keulen 1999, Dym and Little 2000). 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*.' (Simon 1996, p 111).

A slightly more specific but still very abstract definition of design is given by Dym and Little 2000:

> *Engineering Design is the systematic, intelligent generation and evaluation of specifications for artefacts whose form and function achieve stated objectives and satisfy specified constraints.* (Dym and Little 2000)

Although clearly definitions of *design* are inadequate in general (Dasgupta 1991), 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*.' (Keulen, 1999).

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.

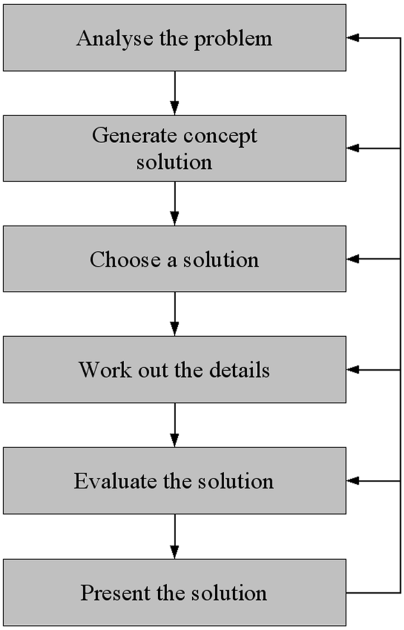


Figure 1. A set of methodical steps for designing.

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.

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 (Jones 1984): 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.

3. 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 that 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 is 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.

4. 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. (Anderson 2000, Merriënboer 1997, Dym and Little 2000, Cross 2000).

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 (Jong and Joolingen 1998). 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. (Merriënboer 1997)

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.

5. 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.

6. 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.

6.1. *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.eitn.wau.nl). Use the link 'Try the Downstream Processing Design Case'.

6.2. *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.

6.3. *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 (Merriënboer 1997).

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.

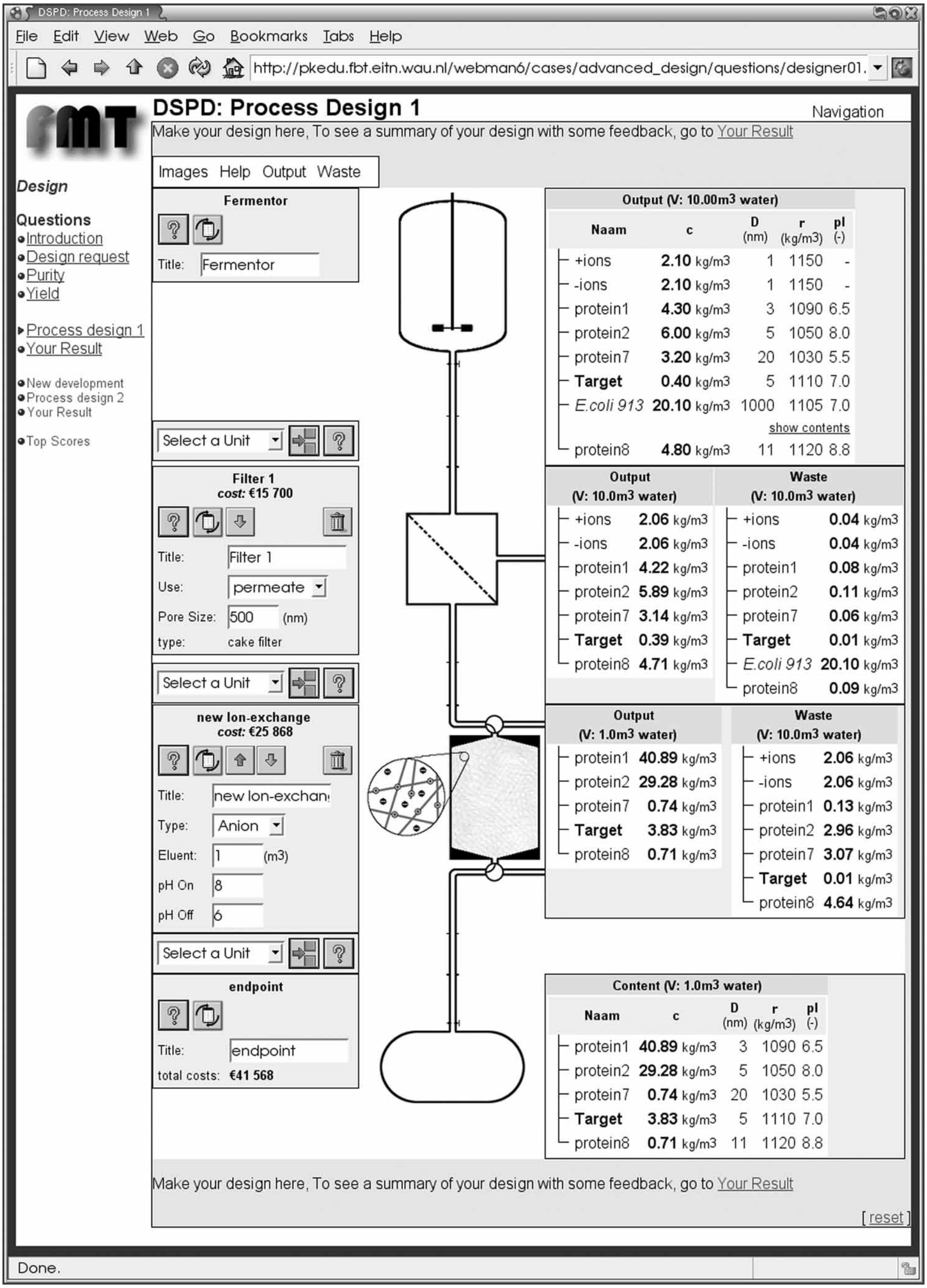
If the student is allowed to make modifications to the process chain then every unit has the option to remove that unit from the process chain. Between every two units an option is available to add a unit operation between those two units.

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.

6.4. *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.



http://pkedu.fbt.eitn.wau.nl/webman6/cases/advanced_design/questions/designer01.

DSPD: Process Design 1

Make your design here, To see a summary of your design with some feedback, go to Your Result

Fermentor

Title: Fermentor

Output (V: 10.00m3 water)

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	20.10 kg/m3	1000	1105	7.0
protein8	4.80 kg/m3	11	1120	8.8

show contents

Filter 1
cost: €15 700

Title: Filter 1
Use: permeate
Pore Size: 500 (nm)
type: cake filter

Output (V: 10.0m3 water)		Waste (V: 10.0m3 water)	
+ions	2.06 kg/m3	+ions	0.04 kg/m3
-ions	2.06 kg/m3	-ions	0.04 kg/m3
protein1	4.22 kg/m3	protein1	0.08 kg/m3
protein2	5.89 kg/m3	protein2	0.11 kg/m3
protein7	3.14 kg/m3	protein7	0.06 kg/m3
Target	0.39 kg/m3	Target	0.01 kg/m3
		E.coli 913	20.10 kg/m3
protein8	4.71 kg/m3	protein8	0.09 kg/m3

new Ion-exchange
cost: €25 868

Title: new Ion-exchan
Type: Anion
Eluent: 1 (m3)
pH On 8
pH Off 6

Output (V: 1.0m3 water)		Waste (V: 10.0m3 water)	
protein1	40.89 kg/m3	+ions	2.06 kg/m3
protein2	29.28 kg/m3	-ions	2.06 kg/m3
protein7	0.74 kg/m3	protein1	0.13 kg/m3
Target	3.83 kg/m3	protein2	2.96 kg/m3
protein8	0.71 kg/m3	protein7	3.07 kg/m3
		Target	0.01 kg/m3
		protein8	4.64 kg/m3

endpoint

Title: endpoint
total costs: €41 568

Content (V: 1.0m3 water)

Naam	c	D (nm)	r (kg/m3)	pI (-)
protein1	40.89 kg/m3	3	1090	6.5
protein2	29.28 kg/m3	5	1050	8.0
protein7	0.74 kg/m3	20	1030	5.5
Target	3.83 kg/m3	5	1110	7.0
protein8	0.71 kg/m3	11	1120	8.8

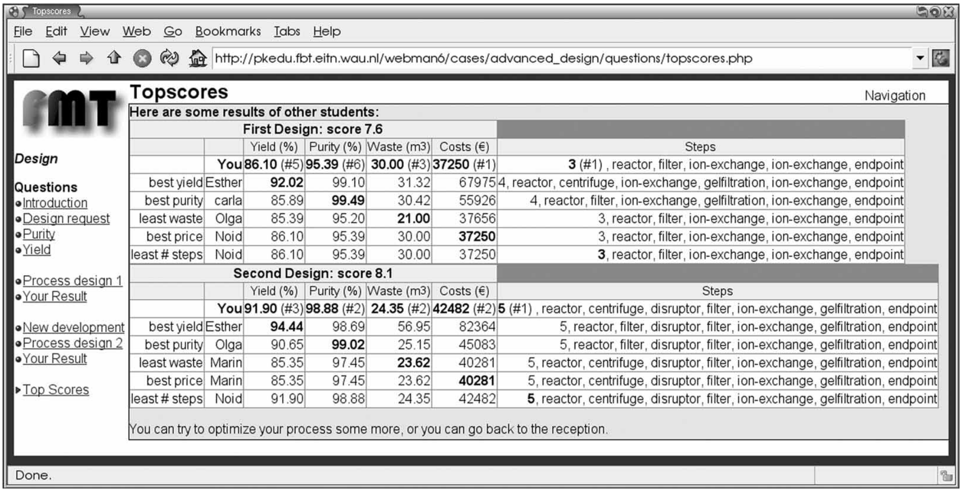
Make your design here, To see a summary of your design with some feedback, go to Your Result

[reset]

Figure 2. The downstream process designer in use in a case.

6.5. *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,



Topscores

Here are some results of other students:

	First Design: score 7.6					
		Yield (%)	Purity (%)	Waste (m3)	Costs (€)	Steps
	You	**86.10** (#5)	**95.39** (#6)	**30.00** (#3)	**37250** (#1)	**3** (#1) , reactor, filter, ion-exchange, ion-exchange, endpoint
best yield	Esther	**92.02**	99.10	31.32	67975	4, reactor, centrifuge, ion-exchange, gelfiltration, ion-exchange, endpoint
best purity	carla	85.89	**99.49**	30.42	55926	4, reactor, filter, ion-exchange, gelfiltration, ion-exchange, endpoint
least waste	Olga	85.39	95.20	**21.00**	37656	3, reactor, filter, ion-exchange, ion-exchange, endpoint
best price	Noid	86.10	95.39	30.00	**37250**	3, reactor, filter, ion-exchange, ion-exchange, endpoint
least # steps	Noid	86.10	95.39	30.00	37250	**3**, reactor, filter, ion-exchange, ion-exchange, endpoint

	Second Design: score 8.1					
		Yield (%)	Purity (%)	Waste (m3)	Costs (€)	Steps
	You	**91.90** (#3)	**98.88** (#2)	**24.35** (#2)	**42482** (#2)	**5** (#1) , reactor, centrifuge, disruptor, filter, ion-exchange, gelfiltration, endpoint
best yield	Esther	**94.44**	98.69	56.95	82364	5, reactor, filter, disruptor, filter, ion-exchange, gelfiltration, endpoint
best purity	Olga	90.65	**99.02**	25.15	45083	5, reactor, filter, disruptor, filter, ion-exchange, gelfiltration, endpoint
least waste	Marin	85.35	97.45	**23.62**	40281	5, reactor, centrifuge, disruptor, filter, ion-exchange, gelfiltration, endpoint
best price	Marin	85.35	97.45	23.62	**40281**	5, reactor, centrifuge, disruptor, filter, ion-exchange, gelfiltration, endpoint
least # steps	Noid	91.90	98.88	24.35	42482	**5**, reactor, centrifuge, disruptor, filter, ion-exchange, gelfiltration, endpoint

You can try to optimize your process some more, or you can go back to the reception.

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

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. The data can also be linked to a student model to track for instance student progress or to add a 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 at 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.

7. 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 to be 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 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.

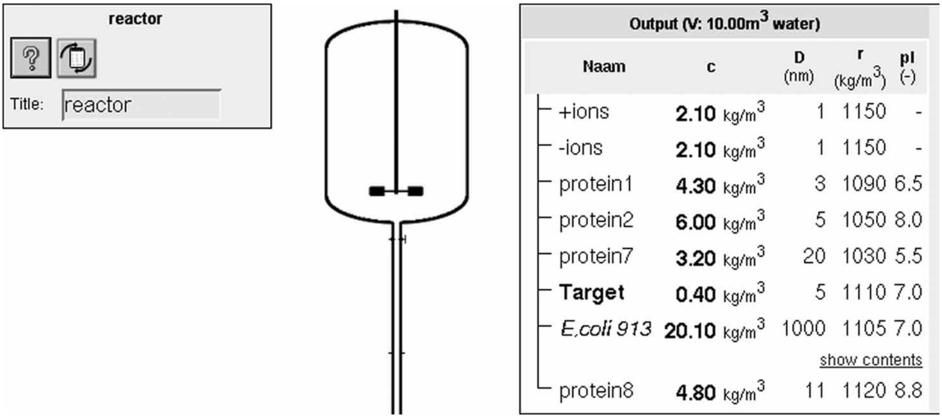
At Wageningen University, the case is used by first-year students in Bioprocess Engineering. These students have little knowledge about downstream processing or the unit-operations used and have no design experience.

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

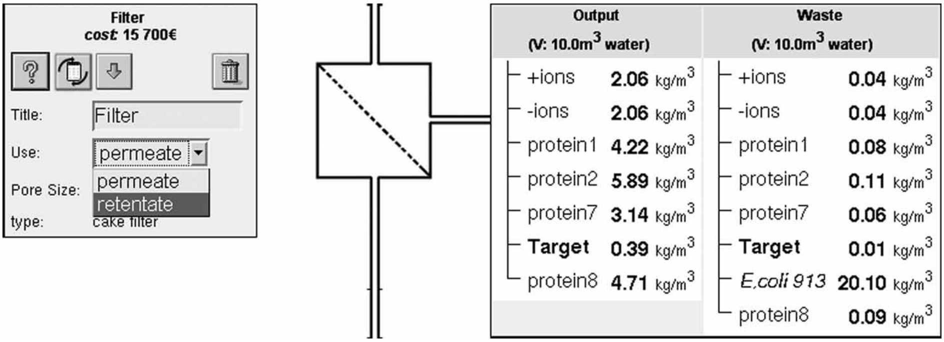


Title: reactor

Output (V: 10.00m^3 water)

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

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



Filter
cost: 15 700€

Title: Filter
Use: permeate
Pore Size:
type: cake filter

Output (V: 10.0m³ water)	
+ions	2.06 kg/m³
-ions	2.06 kg/m³
protein1	4.22 kg/m³
protein2	5.89 kg/m³
protein7	3.14 kg/m³
Target	0.39 kg/m³
protein8	4.71 kg/m³

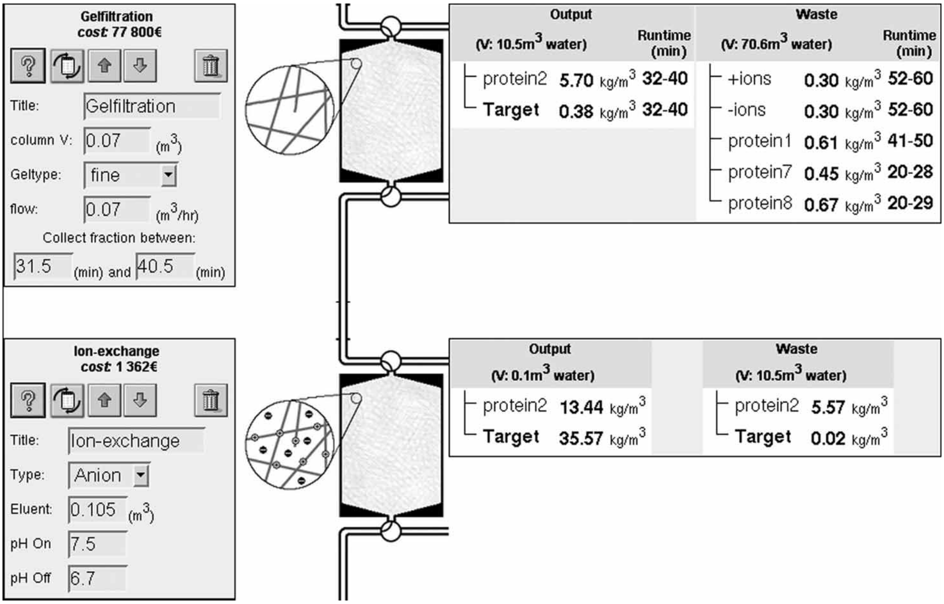
Waste (V: 10.0m³ water)	
+ions	0.04 kg/m³
-ions	0.04 kg/m³
protein1	0.08 kg/m³
protein2	0.11 kg/m³
protein7	0.06 kg/m³
Target	0.01 kg/m³
E.coli 913	20.10 kg/m³
protein8	0.09 kg/m³

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

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 10 m^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.

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 10 m^3. The gelfiltration unit can thus be much smaller in this situation, resulting in



Gelfiltration
cost: 77 800€

Title: Gelfiltration
column V: 0.07 (m³)
Geltype: fine
flow: 0.07 (m³/hr)
Collect fraction between: 31.5 (min) and 40.5 (min)

Output (V: 10.5m³ water)		Runtime (min)
protein2	5.70 kg/m³	32-40
Target	0.38 kg/m³	32-40

Waste (V: 70.6m³ water)		Runtime (min)
+ions	0.30 kg/m³	52-60
-ions	0.30 kg/m³	52-60
protein1	0.61 kg/m³	41-50
protein7	0.45 kg/m³	20-28
protein8	0.67 kg/m³	20-29

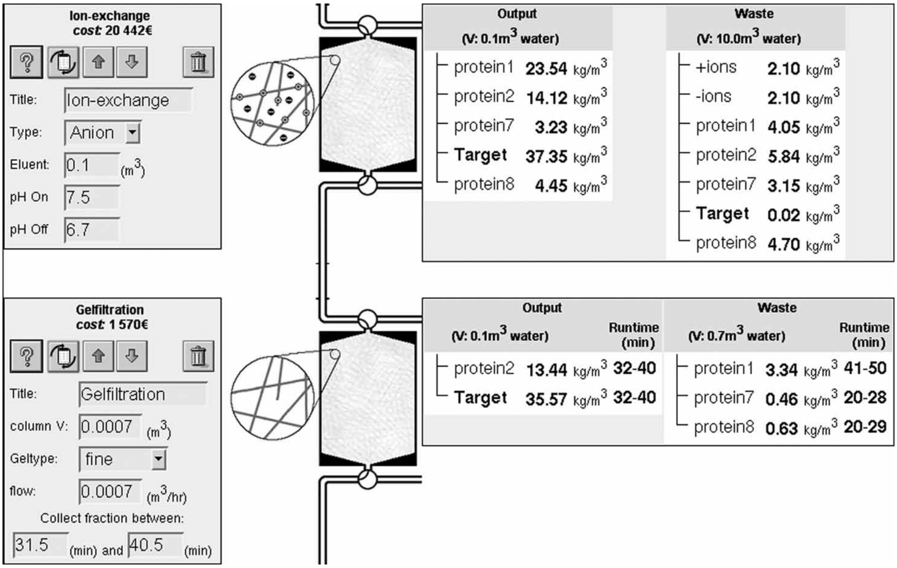
Ion-exchange
cost: 1 362€

Title: Ion-exchange
Type: Anion
Eluent: 0.105 (m³)
pH On: 7.5
pH Off: 6.7

Output (V: 0.1m³ water)	
protein2	13.44 kg/m³
Target	35.57 kg/m³

Waste (V: 10.5m³ water)	
protein2	5.57 kg/m³
Target	0.02 kg/m³

Figure 6. A volume of 10 m^3 water containing proteins and ions is first treated with a gelfiltration unit and subsequently with an ion-exchange unit.



Ion-exchange
cost: 20 442€

Title: Ion-exchange
Type: Anion
Eluent: 0.1 (m^3)
pH On 7.5
pH Off 6.7

Output (V: 0.1m^3 water)	
protein1	23.54 kg/m^3
protein2	14.12 kg/m^3
protein7	3.23 kg/m^3
Target	37.35 kg/m^3
protein8	4.45 kg/m^3

Waste (V: 10.0m^3 water)	
+ions	2.10 kg/m^3
-ions	2.10 kg/m^3
protein1	4.05 kg/m^3
protein2	5.84 kg/m^3
protein7	3.15 kg/m^3
Target	0.02 kg/m^3
protein8	4.70 kg/m^3

Gelfiltration
cost: 1 570€

Title: Gelfiltration
column V: 0.0007 (m^3)
Geltype: fine
flow: 0.0007 (m^3/hr)
Collect fraction between: 31.5 (min) and 40.5 (min)

Output (V: 0.1m^3 water)		Runtime (min)
protein2	13.44 kg/m^3	32-40
Target	35.57 kg/m^3	32-40

Waste (V: 0.7m^3 water)		Runtime (min)
protein1	3.34 kg/m^3	41-50
protein7	0.46 kg/m^3	20-28
protein8	0.63 kg/m^3	20-29

Figure 7. A volume of 10 m^3 water containing proteins and ions is first treated with an ion-exchange unit and subsequently with a gelfiltration unit.

a much smaller waste volume and a cheaper process. The student should realise 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.

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.

8. 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 recognise 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 (Jong and Joolingen 1998).

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.' (Posner 1997) 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.

9. 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 than 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.

References

ANDERSON, J. R., 2000, *Cognitive Psychology and Its Implications* (New York: Worth Publishers).

ASENJO, J. A. and MERCHUK, J. C., 1994, *Bioreactor System Design* (New York: Marcel Dekker Inc.).

BRANDT, D., HENNING, K., 2001, Perspectives of information and communication technologies for engineering education. *Eur. J. Ed.*, **26**(1), 63–68.

CABRAL, J. M. S., MOTA, M. and TRAMPER, J., 2001, *Multiphase Bioreactor Design* (London: Taylor & Francis).

CHANDRASEKARAN, B., 1990, Design problem solving: a task analysis. *AI Magazine*, **11**(4), 59–71.

CROSS, N., 2000, *Engineering Design Methods: Strategies for Product Design*, 3rd edn (Chichester, England: John Wiley & Sons).

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

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

Irandoust, S. and Sjöberg, J., 2001, International dimensions: a challenge for European engineering education. *European Journal of Engineering Education*, **26**(1), 69–75.

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

de Jong, T. and van Joolingen, W. R., 1998, Scientific discovery learning with computer simulations of conceptual domains. *Review of Educational Research Summer*, **68**(2), 179–201.

van Keulen, H., 1999, Design teaching views and practices in Delft. In 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), pp. 51–58, ISBN 1 86058 2087.

van Merriënboer, J. J. G., 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.

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

van't Riet, K. and Tramper, J., 1991, *Basic Bioreactor Design* (New York: Marcel Dekker Inc.).

Simon, H. A., 1996, *The Sciences of the Artificial*, 3rd edn (Cambridge MA: MIT Press).

http://www.fbt.eitn.wau.nl/, *go to the content showcase, follow* Try the Downstream Processing Design Case.

About the authors

Ir. Hylke van der Schaaf gained masters in Process Engineering at Wageningen Agricultural University in 1999. He is currently doing a Ph.D. in Process Engineering on the subject of the design of digital learning materials for process engineering.

Dr. ir. Marian Vermuë is Assistant Professor in Food and Bioprocess Engineering. She graduated in the subject Biocatalysis in Non-conventional Media in 1995. Her current field of expertise is the development of educational material for Process Engineering and as such she is involved in the project "Development of digital learning material".

Johannes Tramper is a Professor of Bioprocess Engineering at Wageningen University in Wageningen, The Netherlands. He received M.Sc. Degrees in Chemical and Biomedical Engineering grom Delft University of Technology, The Netherlands, and Purdue University, West Lafayette, Indiana, and the Ph.D. Degree in 1979 in Agricultural Sciences from Wageningen Agricultural University, Wageningen, The Netherlands.

Rob J. M. Hartog studied experimental Physics at the University of Amsterdam. After gaining his degree in 1974 he taught Physics in the Netherlands and in Surinam for the Dutch Ministry of Foreign Affairs. From 1979 to 1986 he lectured Physics and Pedadogy of Physics at the Teacher Training Centre in Nijmegen. Since 1986 he has been employed by Wageningen University and Research Centre. He has had several assignments in the fields of (Intelligent) Computer Assisted Instruction and functional design of information systems. Currently he is Programme Manager, Instruction Technology, for the School of Technology and Nutrition, and Assistant Professor of Information Systems. As IT programme manager, he is responsible for roughly 20 projects on ICT in Education in the domain of food and biotechnology. As Assistant Professor of Information Systems he directs the Wageningen Multimedia Research Center and gives lectures on decision support systems in food industry.