Jl. of Interactive Learning Research (2007) **18***(4)*, 533-554

Proteus: A Lecturer-Friendly Adaptive Tutoring System

OLIVIER D. T. SESSINK, HENDRIK H. BEEFTINK,
JOHANNES TRAMPER, AND ROB J. M. HARTOG
Wageningen University, The Netherlands
olivier.sessink@wur.nl
rik.beeftink@wur.nl
hans.tramper@wur.nl

Effectively targeting a heterogeneous student population is a common challenge in academic courses. Most traditional learning material targets the "average student," and is suboptimal for students who lack certain prior knowledge, or students who have already attained some of the course objectives. Student-activating learning material supports effective training. Adaptive systems may help to support learning for a heterogeneous group of students. Development of adaptive learning material, however, is usually a complex task not easily done by the average lecturer. An adaptive tutoring system is therefore designed that requires little knowledge and skills from lecturers: Proteus. Proteus provides adaptive navigation on a micro scale based on a set of closed questions. The questions are used both to measure, as well as to stimulate student progress. Entering questions in Proteus requires little effort from lecturers. Proteus is tested and evaluated in a Bachelor of Science (BSc) course with 91 students and is received very well.

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

The usual approach to target such a heterogeneous student population is to start at the "average" student level. For students who have already attained the learning objectives of the initial lectures this will be boring. For students who lack some of the course's prerequisites this will be difficult: they need to identify the gap between their knowledge and the lecture content themselves, and bridge the gap by means of self-study. Finding the right information without guidance, however, is not straightforward, and time consuming for learners (Pivec & Baumann, 2004). The prior knowledge and skills of the students should be determined both with respect to prerequisites as well as with respect to the learning objectives of the course. Students should then be individually trained in those areas of the prerequisites and learning objectives where training is found to be required.

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

Adaptive systems may provide the means to address such a heterogeneous student population: "Adaptive systems cater information to the user and may guide the user in the information space to present the most relevant material, taking into account a model of the users goals, interests and preferences" (Brusilovsky, Eklund, & Schwarz, 1998). Development of adaptive learning material, however, is often time consuming and requires specific knowledge and high-level skills from the author (Brusilovsky et al., 1998; De Bra & Ruiter, 2001; Dagger, Wade, & Conlan, 2005; Armani, 2005). This makes adaptive systems poorly accessible for lecturers outside computer science.

Requirements for a Lecturer-Friendly Adaptive System

Several authors have stressed the deep conceptual gap between authoring systems and authors (Mizoguchi & Bourdeau, 2000; Aroyo & Mizoguchi, 2004). A lecturer-friendly learning system should use concepts that match the intuition of the lecturers and require little effort for effective use (Murray, 1999). For adaptive systems, usability is mostly determined by the type of adaptive rules, the perceived complexity of the system, and the type of information required by the adaptive rules. Perceived complexity is the complexity of the system as far as the lecturer needs to understand it to effectively use the system. However, there is a design trade-off: increasing the usability of a tutoring system comes at a cost. Murray listed a design trade-off space for authoring of learning systems, based on breadth, depth, learnability, productivity, fidelity, and cost. In lecturer-friendly adaptive system, authoring learnability and productivity are the most important requirements.

An adaptive system in education has a model of the student called a *student model*, and a model of the taught domain called the *domain model*. An adaptive system can, furthermore, assemble content (adaptive presentation), or can select links between content (adaptive navigation; Brusilovsky, 2001). Adaptive presentation requires specially crafted content. This content is usually built from small blocks, called fragments, that have meta data defining their relations. The adaptive system uses the student model and the meta data to assemble the presented content from fragments. Adaptive presentation systems are for example C-Book (Kay & Kummerfeld, 1994) and 2L6770 (Calvi & De Bra, 1997). Adaptive navigation, however, does not require content built from fragments or meta data of fragments. Although many adaptive navigation systems use the electronic book metaphor, adaptive navigation has little requirements for the content. Adaptive navigation systems are for example Interbook (Bruzilowski, Eklund, & Schwarz, 1997), and Dynamic Course Generation ([DCG], Vassileva, 1997).

The type of adaptive rules may contribute to the perceived complexity. In some adaptive systems the lecturer should provide specific instructions for each update of the student model, using a special purpose specification language (De Bra, Aerts, Smits, & Stash, 2002; Esposito, Licchelli, & Semeraro, 2004). This allows very detailed and customizable updates, but requires understanding of this specification language by the lecturer. Other adaptive systems use rules based on statistical theory, such as rules based on Bayesian statistics (Henze & Nejdl, 1999; Conati, Gertner, & Van Lehn, 2002) that require the lecturer to build a Bayesian network.

Requirements for Training

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

To support the learning process effectively, and improve retention and retrieval of knowledge, student-activating learning material should be used (Anderson, 1995). Student activation is for example achieved if students have to give answers in order to proceed through the learning material. If closed questions are used, a computer may effectively interact with the student to support the learning further.

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

Design Outline

In this article we describe the design of Proteus, an adaptive system that requires little effort and knowledge from the lecturers. Proteus provides adaptive navigation on a micro scale through a set of closed questions. Answers to these closed questions provide explicit knowledge to Proteus. The knowledge is explicit, because it is obtained by direct questioning of the student, as opposed to implicit knowledge that could be obtained while observing the actions of a student while learning (Brusilovsky, 1994). Usage of explicit knowledge to update the student model reduces the uncertainty in the student model (Beaumont & Brusilovsky, 1995). The questions in Proteus activate the student and stimulate the development of the student's mental model. Proteus selects questions such that students will be trained individually in specific areas where training is found to be necessary. Proteus requires only little meta data for each question. The required meta data are defined using concepts that are intuitive for lecturers. Proteus is used and evaluated in Bio-Process Engineering education.

TYPICAL STUDENT SESSION

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

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

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

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

After submitting an answer, the student receives feedback whether his answer was correct or not (Figure 1). If not, the student will get feedback

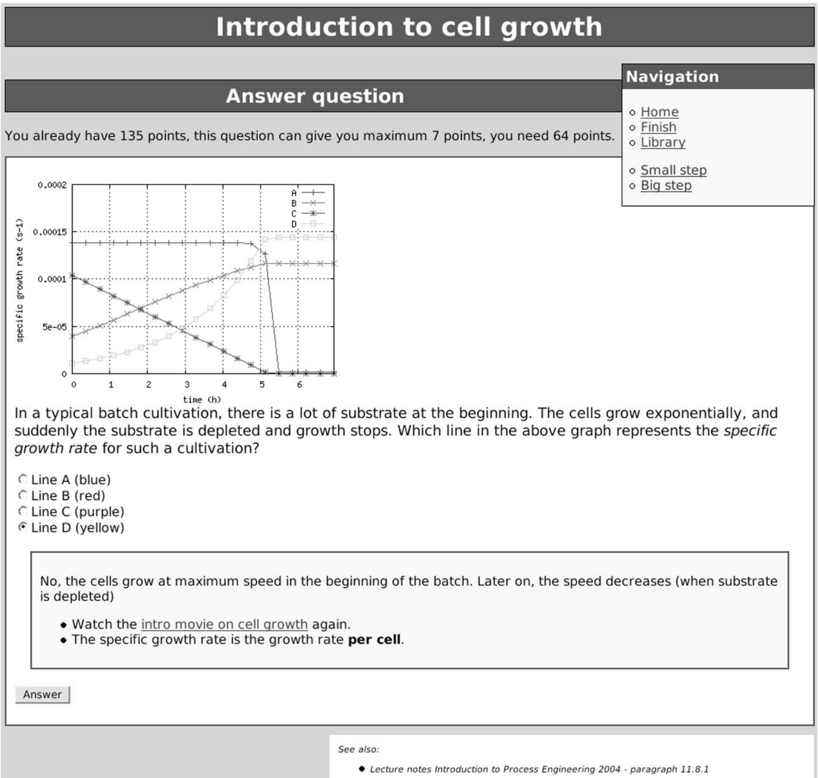

Figure 1. A typical question in the "Introduction to cell growth" module. The student's answer is wrong, and the figure shows the feedback on this answer.

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

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

TYPICAL LECTURER SESSION

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

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

After the learning objectives have been defined in Proteus, questions are added. To add a question, the lecturer selects the learning objectives that are relevant for this question. For each of those learning objectives, a "prerequisite level" may be set for this question (Figure 3). The prerequisite level is the minimal level the student should have for this learning objective before this question may be presented to this student. The question, furthermore, should have "exit levels" for at least one learning objective. The exit level defines which level students may reach by answering this question correctly. An exit level higher than the target level of a learning objective thus

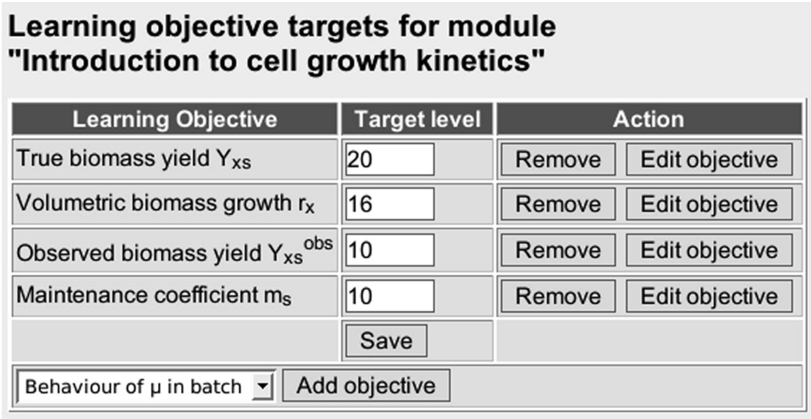

Figure 2. Choosing learning objectives and defining "Target levels"

means that a student may finish this learning objective by answering the question correctly.

After the lecturer has entered prerequisite levels and exit levels, Proteus shows information how this question is placed relative to other questions. The interface shows the number of "introductory" questions and the number of "follow up" questions (Figure 3). The introductory questions are introductory relative to the present question because they have an exit level that at least

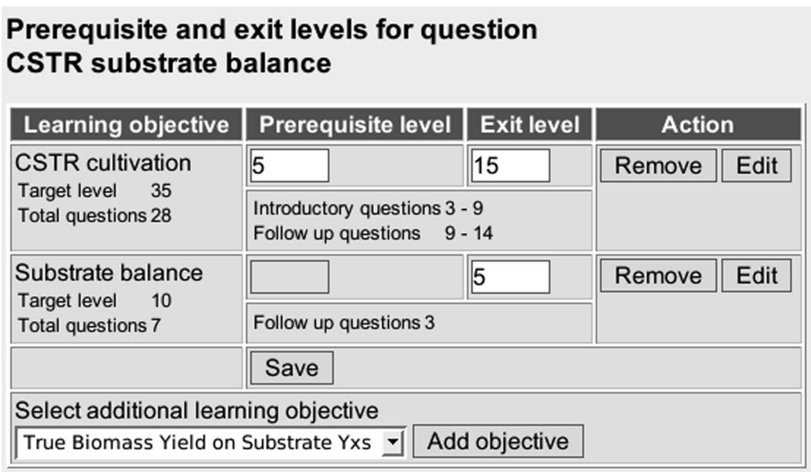

Figure 3. The prerequisite and exit levels for a question in the lecturer interface. One prerequisite, and two exit levels have been defined.

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

STUDENT AND DOMAIN MODELING

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

Together, the learning objectives for the module form an n-dimensional knowledge space, where *n* is the number of learning objectives. Each student *s* has a level for each of these learning objectives *o*: $L_{o,s}$ (Figure 4a). In order to finish the module, a student has to cover the complete knowledge space, and thus attain the target level T_o for each learning objective. A model in

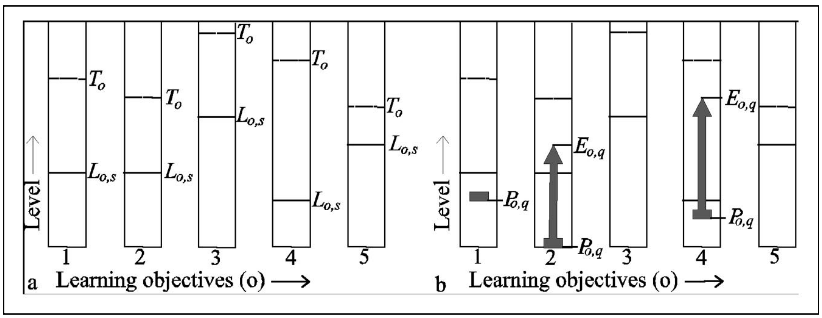

Figure 4. A five dimensional knowledge space. In figure a each learning objective has a target level T_o defined. The current student level $L_{o,s}$ for one student is shown for each objective. In figure b a question is shown. This question has a prerequisite level $P_{o,q}$ defined for learning objective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

which the student model evolves to the domain model is called an "overlay model" (Brusilovsky, 1994). A student raises his level for a learning objective by answering questions about that learning objective.

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

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

ADAPTIVE RULES: UPDATING THE STUDENT MODEL, AND SELECTING QUESTIONS

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

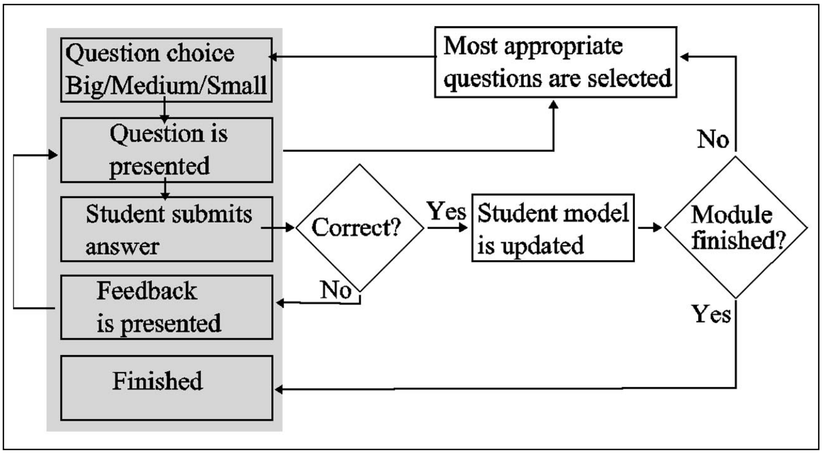

Figure 5. The flow of actions in Proteus, the gray area involves student interaction

Table 1
Symbols and Their Identifiers,
o for Learning Objective, *s* for Studentand *q* for Question

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

Updating the Student Model

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

$$L_{o,s,post} = \max(L_{o,sprior}, E_{o,q}) - f1_{o,q,s} ?r_{o,q}$$

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

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

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

$$f1_{\mathrm{o,q,s}} = \frac{N1_{s,q} - 1}{N_{s,q} + f2_{o,q,s}}$$

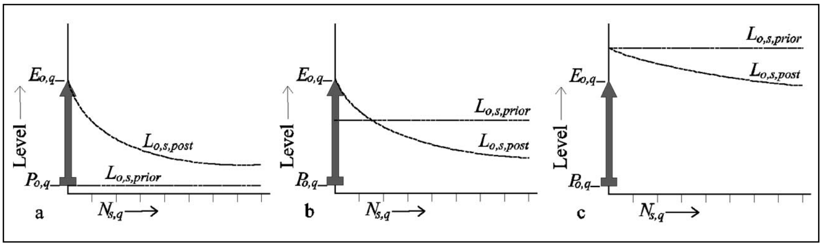

Figure 6. The behavior of the update function that calculates the new student level $L_{o,s,post}$ as function of the number of tries $N_{s,q}$ for three situations. In a, the student level $L_{o,s,pre}$ equals the question prerequisite level $P_{o,q}$. In b, the student level $L_{o,s,pre}$ is close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

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

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

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

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

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

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

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

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

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

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

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

Selecting Questions

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Question Development

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

Measuring Student Progress

The ideal question to measure student progress would be answered immediately correct by all students who have attained the exit levels defined for this question. In reality questions are never ideal, but there should be a good match between the question content and the levels defined for the question. If the prerequisites are met, the number of tries should only depend on the learning objectives that have exit levels defined. The number of tries should not be affected by other learning objective. The question should, furthermore, not have any prerequisite learning objectives other than defined by the prerequisite levels. Lack of modularity in the design of questions is thus a risk. Although these requirements may sound very obvious, in reality it occurs that questions have implicit learning objectives that have been overlooked by the author. For example, students could all fail a question that targets the learning objective "exponential growth," because the author assumed students knew how to solve a differential equation. If it is known that many students have problems solving differential equations, either there should be a learning objective "solving differential equations" in Proteus, which should be the prerequisite for the question about "exponential growth," or the question should be redesigned such that it can be answered without knowledge of differential equations.

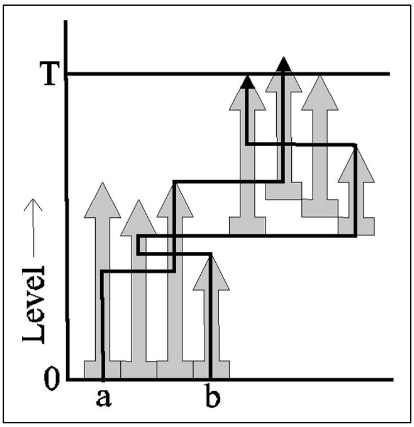

Figure 7. Prerequisite and exit levels of eight questions, projected on a single learning objective. Students a and b extend their knowledge space with different questions

Stimulating Student Progress

There are several reasons why the system stimulates student progress, ranging from the adaptive behavior of Proteus to the content of individual questions. Closed questions can be activating learning material; to proceed through the learning material, students actively have to answer the questions. Elaboration or practice is found to improve retention of knowledge (Anderson, 1995).

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

The questions themselves are annotated with links to relevant paragraphs in the book and the lecture notes, as well as hyperlinks to online theory. This allows students to access the theory with very little effort. The theory is thus directly accessible for the student. The application of just-in-time information presentation in the adaptive system reduces extraneous cognitive load (Kester, Kirschner, van Merriënboer, & Baumer, 2001).

All questions provide feedback on the students' answer, to assist the student in restructuring his knowledge (Kulhavy, 1989). Lack of modularity in the question design may reduce the effectiveness of the feedback. The feedback is designed following the recommendations of Narciss and Huth (2002). If students answer the question correctly, feedback is provided to show why this answer is indeed correct, providing both verification and elaboration. If students answer the question incorrectly, specific feedback is provided that explains why this particular answer is incorrect, again providing verification and elaboration, and hints are provided how to approach the question, providing elaboration. The hints are adjusted with the number of tries a student needs to answer the question. The more wrong answers, the more hints are given, and the more specific and concrete the hints will be. To avoid students losing confidence, most questions will, in the end, show the correct answer. Some lecturers might worry this can be abused by students, however, if a student needs a large number of tries, Proteus will not increase his student levels up to the exit levels defined for the question.

Usage and Evaluation

A module "Introduction to cell growth kinetics" was developed in Proteus. This module corresponds to eight hours of study load for students, which includes for example reading the lecture notes, and exam preparation. Thirteen (13) learning objectives were defined in Proteus. The learning objectives had target levels ranging from six for the smallest and easiest learning objective until 35 for the largest and most difficult learning objective. Proteus was filled with 96 closed questions with feedback. Once created, Proteus needs prerequisite and exit levels for each question. Adding all 96 questions to Proteus was completed within four hours.

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

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

ANALYSIS AND DISCUSSION

Observations and Tracking Data

The assistants observed that most students started the module without using the lecture notes. After the first or second question where students needed multiple tries, students realized that this precluded a maximum score, and started studying the lecture notes if they were unsure whether or not their answer was correct.

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

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

Questionnaire

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

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

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

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

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

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

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

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

In 2004, 15% of the students referred to the difficulty of the questions, of which 12% was a negative comment. Several students complained the questions were too difficult. Questions should only be presented when students have acquired the prerequisites. So either the adaptive behavior did not work for these students, or the prerequisite levels for these questions were not set well, or the prior knowledge of these students did not even meet the initial requirements for the module. Students mostly complained that the "easy," or "medium" option in the module sometimes presented a question that was very difficult in their perception. Because these labels correspond to the training potential β, the same question that is initially presented as "difficult" might at a later stage become

Table 2
Analysis of the Open Question Answers, What Percentage of the Answers Mentioned a Certain Category, and if it was a Positive (+) or Negative (-) Remark

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

"medium" or even "easy." The labels have therefore been changed to "big step," "medium step," and "small step" after the evaluation in 2004.

About 12% of the students in 2004 and 15% of the students in 2005 had problems with the duration of the module. Analysis of the tracking data, however, revealed that many of these students tried to finish the module in one session without a break. Most of these students needed four to five hours to finish the module, which is not very long for the module, but is very long for a single session without breaks.

Ten percent (10%) of the students in 2004 referred to the concept of gaining points, of which half had a positive remark, and the other half a negative remark. Several students with a negative remark reported it was demotivating them to see only the points they lacked, and not the points they already attained. After the evaluation in 2004 the presentation of the points has been changed to show both the number of points students already have attained as well as the number of points students still need.

Seven percent (7%) of the students in 2004 and 6% of the students in 2005 have a negative remark on the point gain mechanism. Students stated that it was unclear to them how the rules were working. An error was later found in some of the questions that probably contributed to some of the 2004 remarks. And, although it is not the intention that students should know how the rules are working, introducing the students to the adaptive behavior and the update rules might increase their confidence with the system.

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

The answers on the multiple-choice questions show that students have a positive view on the module. Especially the "useful" rating is high; in 2004, 39% of the students gave the maximum possible score, in 2005, 58% gave the maximum score. At Wageningen University all courses are evaluated with a standard questionnaire, including the question "This course is useful." In the last year, 802 courses were evaluated, of which 22% received a rating of 4.1 or higher, and only 5% received a rating of 4.4 or higher.

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

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

Table 3
Average Answers on the Multiple-Choice Questions and Their Standard Deviation

Questions about understanding were rated 1=poor ...5=good, questions about statements were rated 1=disagree ...5=agree, exceptions are shown.

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

Future Developments

The closed questions in Proteus currently use a nonstandard XML based format, which was developed for quick development of case-based digital learning material. With the release of the IMS Question and Test Interoperability (QTI) Specification version 2.0 (Innovation, Adoption and Learning, Global learning consortium, [IMS], 2004), the QTI specification can replace our nonstandard format without the loss of functionality. This would enable the use of existing sets of QTI questions in Proteus.

Deploying the adaptive rules in a standard learning management system would improve interoperability further. A first approach for adaptive rules using learning standards is described by Cheniti-Belcadhi, Braham, Heinze, and Nejdi, (2004). At this moment we have not found a way to implement the adaptive rules described in this article conform SCORM 2004. Database functionality for learning objects is probably required to deploy these rules in a learning management system (Sessink, Beeftink, & Hartog, 2005).

Proteus needs to measure the student progress. Only learning objectives that allow questions to be designed such that the number of tries is a good indicator of the student progress can thus be used in the adaptive system. It is less effective for "design," or "collaborative" learning objectives, or "whole task" assignments (van Merriënboer, 1997).

CONCLUSION

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

Addressing a heterogeneous student population effectively with Proteus was found to be very successful. A module developed in Proteus was used twice by in total 143 students in a BSc course, and received very well. Different students followed different tracks through the questions. The quickest students needed 17 questions and two hours to finish the module. Students who required much training needed up to five times more time and up to four times more questions to finish the module. After finishing the module, students answered a questionnaire, in which students gave very positive remarks on the module, and rated the module usefulness on average 4.1 on a 1-5 scale.

References

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

Armani, J. (2005). VIDET: A visual authoring tool for adaptive websites tailored to non-programmer teachers. *Educational Technology & Society, 8*(3), 36-52.

Aroyo, L., & Mizoguchi, R. (2004). Towards evolutional authoring support systems. *Journal of Interactive Learning Research, 15*(4), 365-387.

Beaumont I., Brusilovsky, P. (1995). Adaptive hypermedia: From ideas to real systems. In H. Maurer (Ed.), *Educational multimedia and hypermedia 1995* (pp. 93-98). Charlottesville, VA: Association for the Advancement of Computing in Education.

Bra, P. De, Aerts, A., Smits, D., & Stash, N. (2002). AHA! Version 2.0, more adaptation flexibility for authors. In M. Driscoll & T. C. Reeves (Eds.), *Proceedings of World Conference on E-Learning 2002* (pp. 240-246), Montreal, Canada. Norfolk, VA: Association for the Advancement of Computing in Education.

Bra, P. De, & Ruiter, J.P. (2001) AHA! Adaptive hypermedia for all. *Proceedings of the AACE WebNet Conference* (pp. 262-268), Orlando, FL. Norfolk, VA: Association for the Advancement of Computing in Education.

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

Brusilovsky, P., Eklund, J., & Schwarz, E. (1997). Adaptive navigation support in educational hypermedia on the world wide web. *Proceedings of INTERACT97, The 6th IFIP World Conference on Human-Computer Interaction* (pp. 278-285), Sydney, Australia. New York: Chapman & Hall.

Brusilovsky, P., Eklund, J., & Schwarz, E.W. (1998). Web-based education for all: A tool for development adaptive courseware. *Computer Networks and ISDN Systems, 30*(1-7), 291-300.

Brusilovsky, P. (2001). Adaptive hypermedia. *User Modeling and User-Adapted Interaction, 11*(11), 87-110.

Calvi, L., & De Bra, P. (1997) Using dynamic hypertext to create multi-purpose textbooks. In T. Müldner & T. C. Reeves (Eds.), *Proceedings of ED-MEDIA - World Conference on Educational Multimedia, Hypermedia & Telecommunications* (pp. 130-135), Calgary, Canada. Norfolk, VA: Association for the Advancement of Computing in Education.

Cheniti-Belcadhi, L., Braham, R., Henze, N., & Nejdl, W. (2004, October). A generic framework for assessment in adaptive educational hypermedia. Paper presented at the IADIS WWW / Internet 2004 Conference, Madrid, Spain.

Conati, C., Gertner, A., & VanLehn, K. (2002). Using Bayesian networks to manage uncertainty in student modeling. *User Modeling and User-Adaptive Interaction, 12*(4), 371–417.

Dagger, D., Wade, V., & Conlan, O. (2005). Personalisation for all: Making adaptive course composition easy. *Educational Technology & Society, 8(*3), 9-25.

Esposito, F., Licchelli, O., & Semeraro, G. (2004). Discovering student models in e-learning systems. *Journal of Universal Computer Science, 10*(1) 47-57.

Henze, N., & Nejdl, W. (1999, September). Bayesian modeling for adaptive hypermedia systems. Paper presented at the GI-Workshop Adaptivität und Benutzermodellierung in interaktiven Softwaresystemen, Magdeburg, Germany.

IMS (2004). *Question and Test Interoperability (QTI) specification*. Retrieved June 13, 2007, from http://www.imsglobal.org/question/

Kay, J., & Kummerfeld, R.J. (1994, October). *An individualised course for the C programming language*. Paper presented at the Second International WWW Conference, Chicago, IL.

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

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

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

Mizoguchi, R., & Bourdeau, J. (2000). Using ontological engineering to overcome common AI-ED problems. *International Journal on Artificial Intelligence in Education, 11*, 1-12.

Murray, T. (1999). Authoring intelligent tutoring systems: An analysis of the state of the art. *International Journal of Artificial Intelligence in Education, 10*, 98-129.

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

Pivec, M., & Baumann, K. (2004). The role of adaptation and personalisation in classroom-based learning and in e-learning. *Journal of Universal Computer Science, 10*(1) 73-89.

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

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