

Self-instructive course on Principles of databases

Ivana IVÁNOVÁ, Wim F. FERINGA, Rolf A. de BY, Bas RETSIOS, The Netherlands

Keywords: self-instructive, course, learning object, educational applet, design, development

SUMMARY

The self-instructive course on *Principles of Databases* is offered in distance education and covers the regular module of ITC's degree programs. An important and intended side-effect of the self-instructive approach is the reduction of staff input effort during the course run. A careful course design, with self-explanatory content, should enable students to behave self-instructively during their study. The main learning objective is to make the students understand the logic and principles of database use more than to teach how to operate on a database in a certain database management system (like Microsoft Access). Exercises are in part shaped as educational applets, which enable the student to enter the answer and receive explanatory feedback on every part of the answer. To build such educational applets (which can 'understand' mathematical expressions and formulas, and can also provide translation into SQL syntax) has been one of the biggest challenges in the design of our distance education course on principles of databases. The presented course applies an instructional course design using the ICARE system. Learning objects are self-instructional, allowing students to proceed at their own pace, with and extensive direct feedback for (self-)assessment based on results obtained. The feedback provides the necessary repetition, or possibly advice for further study.

Self-instructive course on Principles of databases

Ivana Ivánová, Wim F. Feringa, Rolf A. de By, Bas Retsios

ABSTRACT

In this paper, we present our ideas on the development of a self-instructive course on *Principles of databases*. We discuss, the motivation and objectives of the self-instructive course's design. It respects the didactic and pedagogical principles for the distance education course curriculum development. Emphasis of the course design has been put on the 'self-instructiveness'. The most important course design principles are discussed, highlighting amongst others, the introduction of new tools not yet introduced in the environment, where the course is offered (being ITC).

INTRODUCTION

Geoinformation professionals should be aware of databases as one of the fundamental technologies in their field. Databases are typically used to organize very large, well-structured data collections for multiple user groups and purposes. This is especially important to organizations in which the information provisioning (internally as well as externally) is a critical success factor to the organization's mission. But database technology can also be extremely useful in smaller, one-off single-user projects with a short life-cycle. This presents the rationale for the course.

After completion of the course, the students will understand when to apply the database technology and when not. An important side-effect of the course is that the student will rejuvenate her/his knowledge of mathematical logic, which is fundamental not only for database use but also for data manipulation in Statistics, GIS and Remote Sensing. Completing the course successfully implies an exemption for the module by the same name, for regular Master or M.Sc. courses at International Institute for Geo-Information and Earth Observation (ITC).

According to the strategic plan of ITC's E-learning project group (ITC, 2005), when the same courseware is used for 'face-to-face' and distant education, adaptation or re-development of the existing courseware is needed. In the case of the self-instructive course on *Principles of databases*, that we discuss here, it means the adaptation of the existing material to a format suitable for distance education must take place. Again, according to that strategic plan, "this requires that the courseware is less lecturer-dependent ; the expertise of the lecturer that is needed for regulation of the learning process, guidance of the exercises and provision of feedback is integrated in the courseware". These expectations are met by the design, set-up and structure of the self-instructive course under discussion.

1. STRUCTURE OF THE COURSE

One of the tools to ensure higher flexibility is the possibility to choose the order of course components within a lesson. The content of the course is divided into four units; every unit has three lessons and lessons where lesson is the main learning object. Lesson have six types of components: (1) introduction (study guide), (2) lecture, (3) demonstration, (4) exercise, (5) self-test and (6) the reader. For assessment and monitoring purposes the so-called unit test is created - it is executed in the Learning Management System (LMS) environment and serves as a support to the final score (a motivation factor for student to take the exam) and as a tool for monitoring student

progress, for tutors. A discussion board, which runs on the LMS and is facilitated by tutor, serves as another stimulant. A general overview of all this is provided in Figure 1.

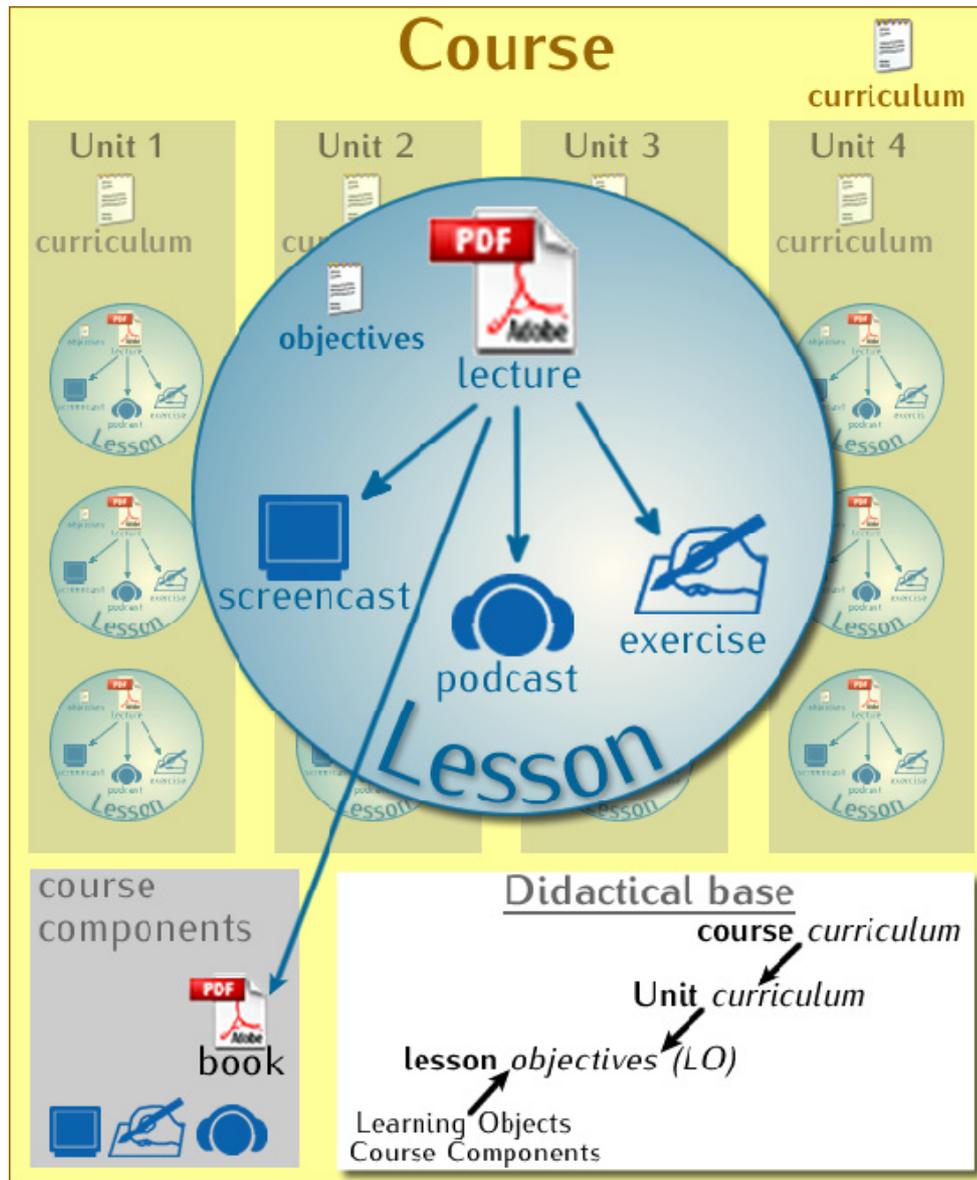


Fig. 1: The course structure and lesson contents, and their interdependencies.

The content of the course has been developed independently from the LMS in a way that the course material is re-usable, maintainable, and independent of software versions. Since the course will be opened in different environments, html navigation is used to ensure a strictly standardized structure and appearance. The materials and course content must also be manageable with a low-end technical infrastructure, meaning that there should be no need for high bandwidth Internet, or expensive software at the student's side. The standardized learning environment is supported with an extension of the current learning management system used at ITC.

2. APPLIED DIDACTIC PRINCIPLES

Distance education course comprises learning episode during which, educator and the educated are typically not at the same location. The basic premises for the choice of a self-instructional didactic approach was to offer to the student the possibility within the duration of the course to study at her/his own pace, in her/his own time, and at her/his location.

Our course on *Principles of databases* applies the didactic approach known as the ICARE (Introduce, Connect, Apply, Reflect, Extend) system supported by principles of instruction theory (Pastor, 2000). ICARE is a five part system that borrowed aspects from various taxonomies, including Gagné's *Nine events of instructions*, Merrill's *Component display theory*, and Bloom's *Taxonomy of higher order learning* (Bannon, 2007).

Robert Gagné's work is classified as an instructional design theory more than a learning theory. It applies a 'teacher-centered approach', because of focus on outcomes resulting from instructions. Gagné defined a nine-step process for a systematic instructional design. This process is called *Nine events of instructions*, in which event is one of the following (Gagné, 1977):

1. Gain attention
2. Inform learners of objectives
3. Stimulate recall of prior learning
4. Present the content
5. Provide learner guidance
6. Elicit performance
7. Provide feedback
8. Assess performance
9. Enhance retention transfer

David Merrill's *Component display theory* (CDT) provides a list of prescriptions to design instruction for different kinds of instructional outcomes (Merrill, 1996). CDT classifies learning along two dimensions: content (fact, concepts, procedures and principles) and performance (remembering, using and generalities). The theory specifies four primary presentation forms: rules (expository presentation of a generality), examples (expository presentation of instances), recall (inquisitory generality) and practice (inquisitory instance). Secondary presentation forms include prerequisites, objectives, helps, mnemonics, and feedback.

Benjamin Bloom's *Taxonomy of higher order learning* is a classification of the different objectives and skills that educators set for learners (learning objectives). The taxonomy includes the following categories (Martin, 2001):

- Knowledge
- Comprehension

- Application
- Analysis
- Synthesis
- Evaluation

The course is designed to be self-instructive, which means that the content of the course acts as a teacher. This may just sound obvious, but in the creation of content material for a 'distant' students it is quite different to apply the instructional principles mentioned above as compared to creation for a 'face-to face' students. The material in a lesson must explain itself. It also has to provide everything necessary for the student to behave in a self-instructive manner, and gather the knowledge aimed at. The approach towards 'distant' students differs from that towards 'face-to-face' students, and materials used in both approaches may complement each other.

The *Principles of databases* course is offered as a set of stand-alone course components, accessible in an order as preferred by the student. The course set-up and structure (see Section 1) naturally suggest an ideal path through the material, in which lecture is the central course component. A lecture acts as 'knowledge glue' that helps the student in constructing her/his new knowledge. This approach arises from the characteristics of the presented topic (*Principles of databases*), since it introduces principles (axioms, theorems, and base skills). The expected entrance knowledge level in the field of databases is low, therefore, it is less suitable to allow students (which do not yet master the principles) create their own opinions, through application of collaborative pedagogical methods. Collaborative learning techniques (discussions, collaborative presentation, etc.) are more suitable for advanced students in a studied subject.

The content of the course is self-explanatory, meaning that the student should find everything s/he needs for comprehension of the course content in the provided materials. This approach allows the teacher to act more as a tutor than a sage during the course run. This significantly decreases the staff time required to run the course. It is often neglected that such a significant cut back of delivery time requires increased effort in the course's design and development. Although the time invested by the tutor during the course run is diminished, the students' performance remains a tutor's interest. The use of a discussion board in the LMS, will be stimulated and must be carefully monitored. Students will be motivated to pass tests after a content block (after each unit, see Section 1) and test results contribute to their final score for the course.

3. COURSE DEVELOPMENT - DESIGN AND SET-UP

Before the course design, we defined the following points of departure for the course *Principles of databases*:

- It is an e-learning course and should be offered as short course in distance education mode.
- The course is first offered in September, 2008 and will last six weeks.
- The course offers high flexibility while following it - a student can study anywhere (online/offline), anytime (not only at specific times) and at his/her own pace without a prescribed route. Suggestions for the course road map to follow are obviously provided.
- The course covers the core module *Principles of Databases* of ITC's regular degree programmes.
- Completing the course successfully implies exemption from the respective module at ITC in context of a Master or M.Sc. course.

- The entrance level of course participants equates to completed BSc education, without specific knowledge in databases, but preferably with a professional background or orientation in any of the academic disciplines of ITC.
- The student is expected to have good basic computer skills.
- The course material is also available for e-learning in a campus setting (to enhance classical learning by individualizing learning) and for joint, decentralized programmes.
- The LMS for the course is Blackboard™ – the course content will be independent from the LMS and be nested into LMS as a web page (to ensure fully standardized structure of course elements, navigation, and appearance of course material)
- The material and course conduct must also be manageable with a low-end technical infrastructure (without need for high bandwidth internet, or expensive software at the client side). This is supported with the Blackboard Backpack™ solution.
- The course material is neither "free" nor "open source", i.e., the source material is not made available nor is the course offered at no costs. It is offered only as an educational service to registered participants.
- The time needed to complete the course is approximately 140 hours (equaling 5 ECTS) for the average course participant.
- The technical solution of the courseware should make the course material re-useable, maintainable (independent of software version) and easy to update. It should even be possible to provide students with updates while they are taking the course, in case errors or omissions are identified.
- Support will be provided to install and test the software required for the exercise prior to the start of the distance course. Fully functional software is a requirement for entering the e-learning course.

4. DESCRIPTION OF THE COURSEWARE

The web's fast development and its increasingly wide access to many areas of the world and higher percentages of the population have prompted educators to design and create learning materials specifically for web use. A simple conversion from the analogue material to a digital equivalent was perhaps considered acceptable, in the childhood years of the web, but this is no longer the case. Web authoring tools have come of age, and course materials where the purpose of use suggests, have become interactive and often dynamic, i.e., responding to user input.

Learning is more than just reading, more than just following the didactic roadmap from A to B. Learning is also doing, finding one's own directions, at one's own pace. Development of course material is a time-consuming task, often with many people involved and with different techniques in use. It is consequently rather costly. A logical consequence is that the developed materials must be re-used over and over, and in different situations (meaning in different courses).

At the end of the 1990s, the term "Learning Objects" (LO) slowly found its way in the world of education. LOs are (Wiley, 2002): instructional components that can be reused a number of times in different learning contexts. The Wisconsin Online Resource Center (WTCS, 2007) describes an LO as:

- The most basic building block of a lesson or activity,
- searchable,

- usable in any learning environment,
- allowing to be grouped or to stand alone by itself, and
- transportable from course to course and program to program.

In addition to this, we felt that for our course it is important that the LO is a "self-contained, reusable entity with a clear learning aim that contains at least three internal changing and editable components: content, instructional activities (learning activities), and context elements" (Chiappe, 2007).

Another crucial LO characteristic is the addition of metadata, which makes the LO findable and then retrievable in an easy way. The metadata description should comply with international standards for describing resources, e.g. the Dublin Core Metadata Initiative's *DCMI Metadata Terms* or the IEEE's *IEEE Standard for Learning Object Metadata* standard.

In our design of the course, we did not see an LO as (according to WTCS) the most basic building block of a lesson, but we adopted Chiappe's definition (see Figure 1). The lesson itself has clearly stated objectives, has well-defined and limited content, and can be assessed. All elements used to build such an LO are seen as course components. Some of these components are not independent, and only make sense in combination with their specific reference component (e.g. a demonstration that explains a certain action described in the lesson). Other components can easily be reused in other lessons in another context (e.g. a reader or exercises).

4.1 Lecture

The etymology of 'lecture' dates back to the 14th century and means 'action of reading, perusal and also that, which is read or perused' (Oxford English Dictionary, 2008). Usually, the lecture serves to put forward the lecturer's point of view on the topic under discussion. In our course, the lecture serves as the main component of the LO, where the "Lesson" is the LO. In our approach, the content of a particular lecture acts as a 'knowledge glue' in a specific topic. By this, we mean the following. The lecture *glues* together ideas presented elsewhere in a course (reader, exercise, demonstration), it *adds* connections, *shows* the application of the discussed theory, and *relates* all the course materials referring to a discussed topic (see Figure 2).

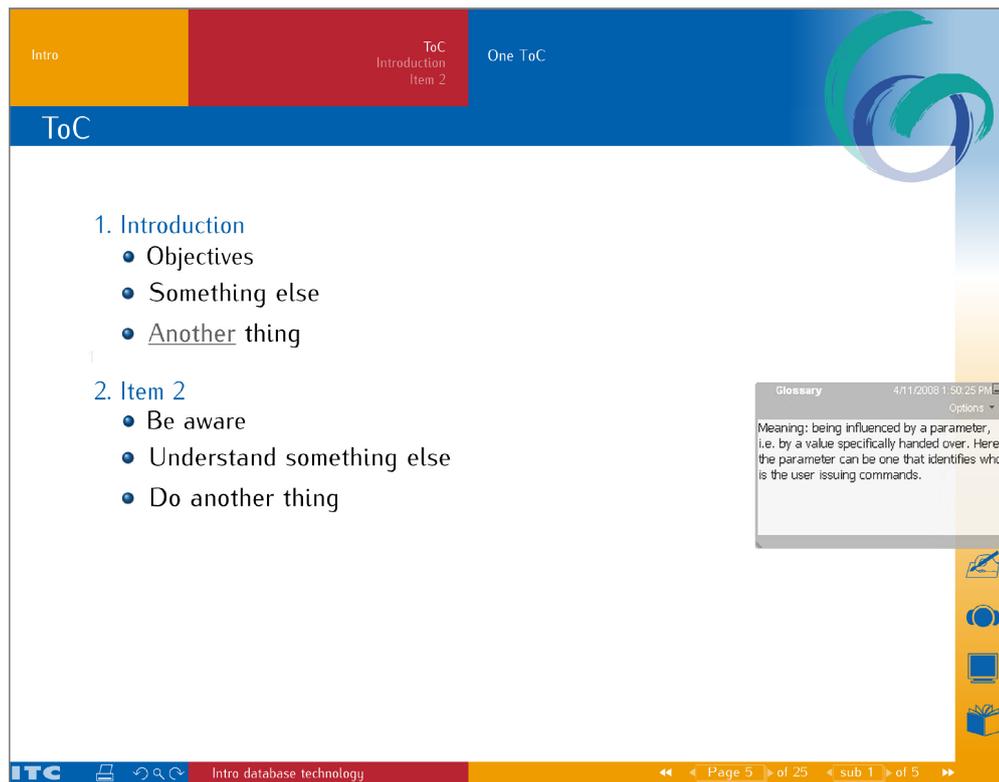


Fig. 2: The design of a lecture

Each lecture is shaped as a PDF file, that can be used on any operating system platform used by the student, with the software to read the files (Adobe Acrobat Reader) is free. Acrobat Reader (AR) itself has built-in functions to search and navigate over pages. An advantage is that external files such as Adobe ShockWave (SWF) and audio files (various formats) can be embedded. Also hyperlinks to webpages, local files or within the same document can be made. JavaScript can be used to add extra functionality, such as the pop-up windows for glossary terms or footnote material. As we make intensive use of all these features, the PDF format is the best format for distributing our educational content.

In the design of lectures, the search and page navigation functions of AR are used, but for consistency and clarity reasons we decided to open the PDF document in full-screen by default, which hides the Acrobat toolbar. The student will face a user interface similar to the one illustrated in Figure 2. Navigation can be done on a page-by-page basis, for which at the right hand bottom page forward-backward buttons are offered. Some lecture pages have a so-called built-up set of subpages (offering a kind of animation) and on such cases, a second forward-backward set of buttons appear, that allows the student to step through these animated subpages. It is important that the student knows "where" s/he is in the lecture, so for this purpose the page number is shown as well as the total number of pages for the lecture. For the animation subpages within a page, 'subnavigation' is offered. Page numbers however do not really give useful information; it is more useful to show the structure of the lecture (providing its chapters, sections and subsections). The main chapters are shown in the orange area at the left hand top (in Figure 2). When clicking on one of the main chapter names that name will be emphasized, in the red area the main sections will show as well as the start page of that section, shown in the content area (having a white background). For some of these sections, also a sub-section can be available, and it shows in the blue area at the top. The lecture has to be structured in such a way that never more than 7 - 7 - 7 entries in the top navigation will occur (i.e., no more than seven chapters per chapter, no more than seven sections and per section no more than seven sub-sections). In the coloured bar on the right

hand side four different icons (may) appear. These icons stand for different types of added content. An icon only shows when its respective content is available, so that the student's attention is drawn to it. The icons stand for (top to bottom) the availability of an *exercise*, a *podcast*, a *demonstration* or reference to a page of the *reader*, which accompanies this course. An exercise is a description of a task that the student has to do (do an internet search, execute a task in MS Access, use the Educational Applet (see Section 4.2)). Podcasts can be e.g. dialogs about a stated problem, or an oral introduction to the lesson, or part of a lecture. A demonstration shows actions that a student has to do when using an application (see Section 4.3).

The design is using the house style colours of ITC. Colours are also used in the (white) content area to emphasize text elements and to indicate a certain use (see Figure 3)



Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Fig. 3: Colours used in a lecture's text

Red colour stands for an alert, to let the student know that this is an important term. Grey underlined indicates that the word or phrase is linked to the glossary or to a footnote explanation. A mouse-over will open a pop-up window (see Figure 3) with the detailed description. Blue is the main title and/or a reference, blue underlined is a reference to an internal page. A green piece of text finally is reserved for examples, they are completely in green as well as the graphic elements used.

4.2 Educational applet

Within the course on *Principles of databases*, the active use is being made of a component that has been dubbed as *educational applet* (EA) takes place. Before we discuss this applet, it is useful to reiterate three of the important learning objectives in the course:

1. Learn how to specify data manipulation requests on an existing database,
2. Learn how to execute these requests against that database, and
3. Gain or regain familiarity with the basics of mathematical logic and set theory.

The latter objective has a purpose wider than that of the field of mastering database technology: logic and set theory are not only fundamental for understanding databases, they also provide equally important foundations for the fields of GIS, remote sensing and computer programming. Since this course on database principles is the first in the curriculum, the exposition to logic and sets that it provides serves multiple purposes.

The educational applet is a piece of software that allows the student to enter an expression in a formal language that the applet subsequently parses for correctness, after which it replies about its findings. Thus it is a tool with which the student can learn to specify and get replies on her/his tries. The formal language is a dialect of mathematical logic that is geared to database manipulation. This is all best explained through a simple example.

Suppose the database holds a set of parcel records, and we want to obtain an overview of those property owners (cadastral identifiers and date of last real estate transaction) in the village of

Lonneker who are paying the high real estate tax tariff. The mathematical expression that formulates this wish is (and see Figure 4):

```
[[ p ∈ Parcel
   p.taxation > 0.18 ∧ p.municipality = 'Lonneker' •
   p.cadastreId, p.owner, p.dateLastSale ]]
```

Fig. 4: Example of mathematical expression

The student will learn how to write up such a specification, step by step. The first line is a variable declaration, the second formulates a tuple selection expression, and the last indicates an output tuple definition. Together, they make up a complete but abstract database request. With the educational applet, s/he can exercise skills on each of these parts separately and eventually also on the whole sentence construction. Once the formal mathematical sentence has been constructed, it can be rewritten into the SQL language which essentially any database system can execute. And this rewriting process can also be learned by using the applet, in a slightly extended version. The applet provides feed-back to the student about her/his attempt when the student asks for it. At present, this feed-back is rather rudimentary, but plans exist to improve this.

So, what is the applet exactly and how is it used? Technically speaking, the applet is a small interpreter of the formal specification language, but fitted with extra functionality for digesting a problem statement, and one or more solutions to the problem. It is a stand-alone Java application, packaged as a JAR file. In the background it is aware of the formal language, which has been defined in a simple Backus-Naur-Form grammar. For instance, five rules in syntax are:

```
<Root> ::= <Query>
<Query> ::= <BagQuery>|<SetQuery>|<Query><UNION><Query>
<BagQuery> ::= [[<QueryBody>]]
<SetQuery> ::= {<QueryBody>}
<QueryBody> ::= <VariableDeclarations>|
                <VariableDeclarations>|<LogicalExpression>|
                <VariableDeclarations><BULLET><OutputTuple>|
                <VariableDeclarations>|<LogicalExpression><BULLET><OutputTuple>| ...
```

There is no intent to get into details here, but the last syntax rule matches the earlier example. The full syntax is approximately 100 rules. A nice feature of the applet is that it can be asked to parse for an expression that must be a <Root>, or a <SetQuery> or perhaps a <LogicalExpression>. This will depend on the didactic purpose of the exercise, and the problem statement given to the student. It helps us to build a didactic strategy.

In the early stage of the course, the student will be asked to formulate a condition (i.e., a <LogicalExpression>) that expresses that a parcel is under the highest tax rate. The teacher will provide the applet with (1) a natural language sentence that formulates that exercise, (2) the type of formal expression that is expected as answer (here: <LogicalExpression>), and (3) one or more correct answers to the exercise. The applet will then interact with the student attempting the exercise, and verifying her/his tries.

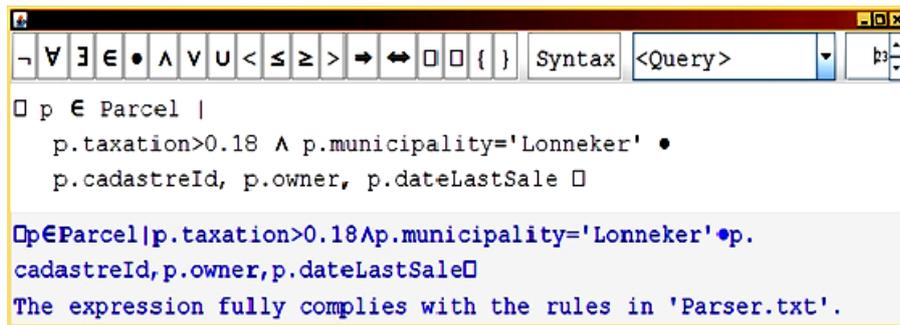


Fig. 5: Example of exercise using EA

Figure 5 illustrates the user interface of the applet, at time of writing. The top pane provides function buttons, with the left hand 60% devoted to mathematical symbols. Their use results in the symbol being inserted in the (white) specification pane, just below the top pane. The Syntax button allows the user to verify her/his current specification. The button marked <Query> indicates the type of expression that the student is expected to enter. The option list at the top right allows resizing of text. The bottom pane in grey, finally, shows the feed-back, here positive, of the applet on the provided specification. What is not included in this version of the applet is the actual problem statement. The plan is to include this as a PDF wrapper page that embeds the applet. This wrapper page will be seen as a separate exercise document, which will be hyperlinked into the overall course structure.

A specific challenge in developing the applet is that multiple correct answers to a single specification problem usually exist, and that an exhaustive listing of those is infeasible. Thus, it is necessary to work out a small equivalence theory, using rewrite rules to normalize any given specification. For instance, if the specification text above would have used variable q throughout instead of variable p , it would have been an equally correct specification. This is what the applied normalization procedures try to attempt. They are correct but not complete, and thus we allow the teacher to provide multiple correct answers, against which the applet can all check.

A more advanced version of the applet, will not only provide an exercise for the student to write the specification, but will continue in another pane, with the actual translation into correct SQL code. The student should also learn how to do this, although it is a fully automatable process, and the student should understand that s/he has solved the problem if s/he comes up with the correct mathematical specification.

In summary, the applet provides an interactive environment in which the student can exercise high-level specifications for logic, set constructs, and in combination query expressions. It can be used without teacher interaction, and as often as the student requires. Moreover, the teacher as content provider has an easy, light-weight job to generate more and more exercises. Due to the versatility of the applet, those exercises can be placed at different levels of complexity, allowing the build-up of a rich exercise set over time.

4.3 Other course tools components

Introduction to lesson - this course component uses a study guide structure, which is a clear description of the objectives and learning activities of the lesson. The Introduction to the lesson also suggests a path along which the student may proceed. The suggested path offers the lecture as a central course component.

Demonstrations - these course components are part of the lecture and exercise in LO in which it is embedded (or accessed separately). Its main purpose is to expand the theoretical knowledge. We use the 'demonstration' concept for screencasts. A screencast is a digital movie in which the setting is a partial or complete computer screen, and in which audio narration describes the on-screen action (Udell, 2005). The narration can be provided by audio recording or text boxes; in our course, we use both. The content of a demonstration, as we use it, can show the student how the particular theoretical principle is applied within a real environment. As an example, see the screenshot of a screencast, which demonstrates the selection of the desired data from a database using SQL - this is content of Lesson 2.2 of the course (see Figure 6)

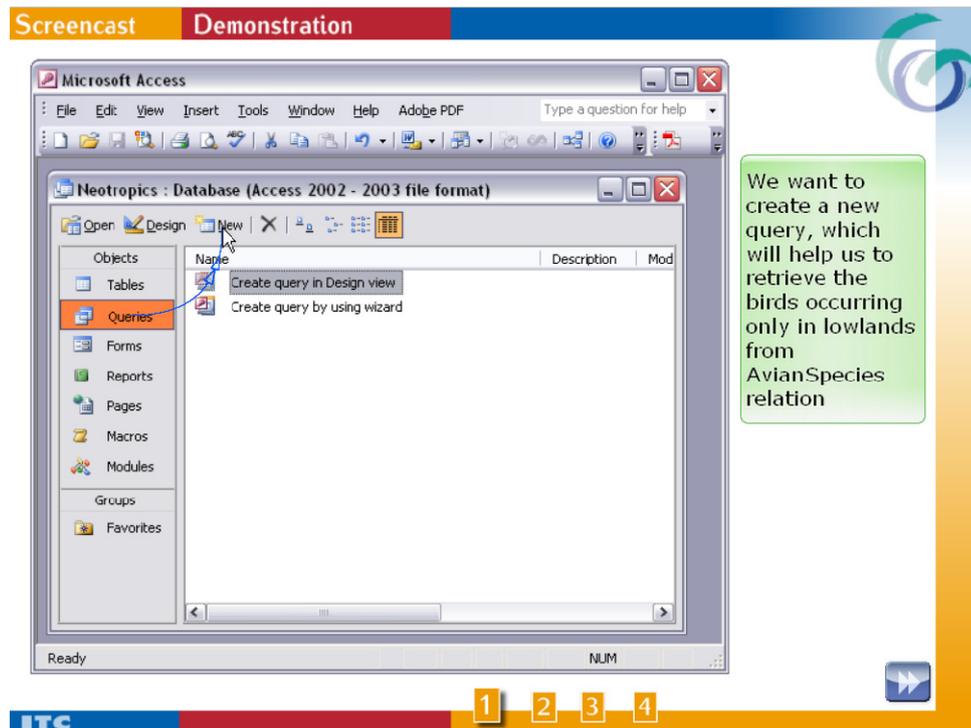


Fig. 6: Example of a demonstration

Self-tests - this course component provides a student the possibility to assess her/his knowledge typically after completion (but, since the order for using the courseware is only suggested, also before or during the lesson) the study of the lesson's content. Self-tests are not monitored (although this option will be interesting when upgrading to a newer version of the LMS used at ITC, which would allow automation of the scoring distribution).

Additional links - this course component will serve as a mean for further study to expand the knowledge gathered during the course starting with the suggested possibilities. Additional links can point to external (outside the course content) web-sources, books, articles, case-studies and so on.

Reader - this course component supports other course components and serves as a reference from the lecture and exercises (Exercise book). To use both books for on-screen reading, these are adjusted to have screen layout, being interactive with maintained cross references (from and to the lecture).

Unit tests - this course component is a part of the LMS and serves as a tool for assessing the student's knowledge level of content of a unit. The achieved score contributes to the final mark of the course, therefore, the student is motivated to take the test.

Exam - the *Principles of databases* course study is finalized by examination. The exam is a part of LMS and is the means to obtain the final grade. Any student, who passes the exam successfully obtains the certificate of completion of the course. Completing the course successfully implies exemption for the respective module at ITC in the context of a Master or M.Sc. course.

5. DISCUSSION AND CONCLUSION

One of the main objectives in a design of a self-instructive course on *Principles of databases* was to lower the tutors input during the course run, but do not to let the student drown in the stack of lecture notes, readers, articles and software manipulation instructions. The design of a course, which would incorporate the premises as stated in Section 3, requires carefully applied a clear, didactic and pedagogical system (see Section 2). How does our course set-up comply with a chosen didactic approach? An overview of this compliance is given in Table 1.

Table 1 Principles of databases course design and its compliance to ICARE System didactic principles

ICARE System	Principles of databases course
Introduce	Introduction to the lesson
Connect	Lecture, Demonstrations
Apply	Demonstrations, Exercise
Reflect	Self-test
Extend	Additional links

The courseware consists of learning objects and components (see Section 4). All these separate components as well as the LOs have to be stored in a (local) repository, where they can be found for other use.

For the content authoring process, i.e., the write-up by educators of the teaching materials and the subsequent formatting and styling of the materials into hyperlinked and multimedia-enriched courseware, we wanted the process to be controllable and manageable. We also wanted to be in a position of being able to address errors in the content rapidly and generate a new version of the affected courseware component on-the-fly. Such allows us to provide students taking the course with improved courseware components, which is certainly important for a first run.

All of this was translated into the requirements of strict separation of content from presentation style, and associating the responsibilities for each to different staff. Many proprietary software solutions, such as MS Powerpoint, do not accommodate this strict separation very well, and the consequence is that courseware generated by collaborating educators often suffers from style and presentation inconsistencies. This then normally requires much effort in before-press clean-up, which by itself reduces the repeatability of the production process substantially, as well as that it lengthens the production cycle.

We found resolve in a technique in which courseware authors write-up the content in a declarative manner, denoting structure to the materials, without the need to mess with layout. For instance, all they need to produce a hyperlink to the online database book is a statement like `hyperlinktomainbook(name_of_the_section)`. Similar declarative statements are available for other inclusions like podcasts and demos, and further structuring statements to indicate animated pages, examples, in-page illustrations, numbered lists and so on are also provided. These declarations are put in a plain text file that can be edited with any text editor (Notepad could do, though we use a better alternative).

All of this high-level content is subsequently read by a compiler program, which collects all the components, interprets the declarations of structure in the plain text file, and creates a fully equipped PDF document out of it. This cycle takes less than 30 seconds, and so, though the procedure is not WYSIWYG, the final product is always only a mouse click away. We accidentally believe that not working in WYSIWYG mode is an advantage, as exposure to the lay-out tends to distract content authors too much, where they should focus on content structure, not lay-out.

The designers of the course believe and hope that the set-up of the course, as described above, will lead to an enjoyable learning experience and minimal student drop-out. The course has all the attributes to be successful and may serve as an example for other distance education courses developed at ITC.

6. REFERENCES

- Bannon, J. (2007). Comprehensive reflection, John Bannon's portfolio on a web of Department of Educational Technology, San Diego University. Retrieved April 13, 2008, from <http://edtec.sdsu.edu/portfolios/key1/examples/bannon/jbannon/compref.htm>
- Chaippe, A. (2007). Que es un objeto de aprendizaje? - What is a learning object?, Retrieved April 13, 2008, from <http://andreschiappe.blogspot.com/2007/09/que-es-un-objeto-de-aprendizaje-what-is.html#links>
- Gagné, R. M. (1977). The conditions of learning. New York: Rinehart and Winston Hold, 1977, ISBN 0-03-049646-2
- Harper, D. (2001). "lecture". In Online Etymology Dictionary. Retrieved April, 13, 2008, from <http://www.etymonline.com/index.php?search=lecture&searchmode=none>
- Martin, J. (2001). Bloom's learning domains. In: B. Hoffman (Ed.), Encyclopedia of Educational Technology. Retrieved April 13, 2008, from <http://coe.sdsu.edu/eet/articles/cdt/index.htm>
- Merrill, D. M. (1996). Instructional Transaction Theory: Instructional Design based on Knowledge Objects. Educational Technology, 36(3), 30-37. Retrieved April 13, 2008 from <http://cito.byuh.edu/merrill/text/papers/IDTHRYK3.PDF>
- Oxford English Dictionary (2008). Oxford University Press 2008. Retrieved April 13, 2008 from http://dictionary.oed.com/cgi/entry/50131233?query_type=word&queryword=lecture&first=1&max_to_show=10&sort_type=alpha&result_place=1&search_id=Igb5-oxrVhI-2951&hilite=50131233
- Pastor, N. (2000). The ICARE system. In B. Hoffman (Ed.), Encyclopedia of Educational Technology. Retrieved April 13, 2008, from <http://coe.sdsu.edu/eet/Articles/icare/start.htm>
- Udell, J. (2005). What is screencasting? In: O'Reilly's Digital Media, Retrieved April 13, 2008, from <http://digitalmedia.oreilly.com/pub/a/oreilly/digitalmedia/2005/11/16/what-is-screencasting.html?page=1>
- Wiley, D. A. 2002, Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy, Retrieved April 13, 2008, from <http://www.reusability.org/read/chapters/wiley.doc>

WTCS (2007). Learning objects. In Wisconsin Online Resource Center Library. Retrieved April 13, 2008, from <http://www.wisc-online.com/about.asp#defined>

BIOGRAPHICAL NOTES

CONTACTS

Dr. Ivana Ivánová, Wim F. Feringa, Dr. Ir. Rolf A. de By, Ir. Bas Retsios
Department of Geo-information Processing
International Institute for Geo-Information and Earth Observation
Hengelosestraat 99
P.O.Box 6
7500 AA Enschede
THE NETHERLANDS
Tel. +31 (0)53 4874 444
Fax +31 (0)53 4874 400
Email: ivanova@itc.nl
Web site: <http://www.itc.nl>