Wuttke, Heinz-Dietrich; Henke, Karsten:

Learning management systems: coupled simulations and assessments in a digital systems course

URN: urn:nbn:de:gbv:ilm1-2014210093

Published OpenAccess: September 2014

Original published in:

Interactive technology and smart education : ITSE. - Bingley : Emerald (ISSN 1741-5659). - 6 (2009) 2, S. 97-107.
DOI: 10.1108/17415650910968107
URL: http://dx.doi.org/10.1108/17415650910968107
[Visited: 2014-09-02]

Interactive Technology and Smart Education

Learning management systems: Coupled simulations and assessments in a digital systems course

Heinz#Dietrich Wuttke Karsten Henke

The current issue and full text archive of this journal is available at
www.emeraldinsight.com/1741-5659.htm

Learning management systems

Coupled simulations and assessments in a digital systems course

Heinz-Dietrich Wuttke and Karsten Henke
University of Technology Ilmenau, Ilmenau, Germany

Abstract

Purpose – The content, provided in learning management systems (LMS), is often text oriented as in a usual textbook, extended by some animations and links. Hands on activities and experiments are not possible. The paper aims to give an overview about the concept to couple smart simulation and assessment tools with an LMS to provide a more explorative approach to the learning content.

Design/methodology/approach – Interactive components, such as smart design tools and online laboratories are added to an LMS that allow exploring the learned content additionally to text parts interactively. The objective of this teaching concept is to empower the students to solve complex design tasks for digital systems and to validate the results. This requires on the one hand site knowledge about the mathematical background of the algorithms used by design tools, and on the other hand, experience from as much as possible examples. Commercial tools are too complex for teaching purposes and hide mostly the algorithms they use for the several design steps. That is why we have developed smart special tailored tools for each single design step that should be learned during a lesson.

Findings – These smart tools are very useful to support the process of understanding and learning by doing. Learners can explore the several design steps with own examples and get immediately feedback about the correct solutions.

Practical implications – The connection to an LMS allows us to record all students' relevant actions in the design process and to evaluate the student or to give individual tailored hints.

Originality/value – The paper introduces a new teaching concept that allows exploring the learned content interactively additionally to text parts.

Keywords E-learning, Simulation, Learning methods

Paper type General review

1. Introduction

Engineering students should be able to develop new technical systems. They have to combine parts to create a new whole and to validate the results critical. To reach this goal, it is very important to have a mix of different phases in the learning process. Students should get new information from text books or lectures but also reflect the learned subjects by own activities. In these active phases, they need individually tailored hints during e.g. hands on experiments, provided by tutors. In that way, learning outcomes for teaching subjects, like "design of ..." can be immediately assessed by tutors. During hands on trainings in traditional learning scenarios, the level of knowledge can be proved directly: The students' experimental results fulfill the requirements or not. Other traditional methods to assess the higher levels of knowledge are oral examinations or project works. The final task of such a project work is usually an assessed oral presentation with discussion and feedback.

To reach the same results and to asses higher levels of knowledge in e-learning scenarios is much more difficult. Assessment possibilities, implemented in learning

The authors would like to thank the Thuringian Ministry for Science, Research and Art (TMWFK/TKM) that partly supports the project. Furthermore we would like to thank our students Sven Hellbach, Andreas Degenhardt, Stefan Prinke and Norman Wagner for programming the presented applets.

Interactive Technology and Smart Education
Vol. 6 No. 2, 2009
pp. 97-107
© Emerald Group Publishing Limited
1741-5659
DOI 10.1108/17415650910968107

management systems (LMS), cover only lower levels of knowledge in relation to Blooms taxonomy. In this paper we will discuss a new approach to teach and verify knowledge also at higher levels in e-learning scenarios by coupling interactive simulations to LMS. The importance of interactive simulations in the teaching process is also pointed out in Jeschke *et al.* (2008), Cherner *et al.* (2006) and many other publications.

The paper is organized as follows: First we give a brief introduction to the objective of our teaching concept, followed by an overview of the design steps for digital systems that are supported by our smart interactive e-learning tools. As an example we discuss the teaching concept for programmable devices including the appropriate assessment components of our assessment system. In the next section we describe the interaction between this system and an LMS. Concluding remarks and references complete the paper.

2. Teaching design of digital systems

On the one hand site, teaching "design of digital systems" is related to a lot of theoretical stuff. The function and the structure of digital circuits can be described by means of Boolean algebra and automata theory. On the other hand site, learners should be able to synthesize real designs that fulfill the required functionality. So we have to teach both, solid theoretical basics and practical design competencies.

Our teaching concept, published in Wuttke and Henke (2002) and Henke *et al.* (2007), starts with the design of combinational circuits. Different means, describing the same function, are suitable in different steps of the design process. After learning several description methods it is difficult for students to understand interrelations between them. That's why our concept includes some problem based parts as well as possibilities for "learning by doing".

In addition to the usually functionality of a LMS we have realized two concepts for "learning by doing": an access to a remote laboratory, were students can design, verify and implement digital circuits and control systems and a collection of interactive tools. Using these tools, the students can explore their knowledge and get new ideas. This paper concentrates on the simulation tools.

The interactive tools follow the concept of "living pictures" (Wuttke and Henke, 2002) and are implemented as JAVA-applets. That way, they run platform independent on any browser platform and are accessible via the internet free of charge.

How we can use these applets to generate knowledge at the "synthesis level"? This level is characterized by Bloom (Jeschke *et al.*, 2008) as the ability to combine parts to create a new whole, where that whole is not immediately apparent before creation. Such kinds of tasks are typical in the design process of digital circuits. Before learners are able to combine parts of a circuit to create a more complex one, the parts have to be taught. This can be done in a traditional way with text/picture combinations as chapters in a text book or by e-learning courses. To combine the learned parts creatively, interaction is required. This is the point in the teaching process, where our interactive tools are used. They extend the text part with interaction possibilities.

3. Interactive exploration tools

Typical "living pictures" are highly interactive and have "the big picture" (a kind of summary at the end of a chapter in a text book) as graphical user interface. That way, learners get familiarly with the handling of the tool and meaning of the symbols and buttons very easy. Whilst the text book contains the theoretical background, and can only display one example of a special function, the interactive tool is able to explain any function with a given number of variables.

Figure 1 shows an example of a "big picture" from a text book and how it is directly transferred into the design of a graphical user interface of an applet. Teaching subject in this case are programmable logic devices (so called PLDs). Students should learn the main differences of such devices: their structure, their programming and their limitations. The user interface shows a programming matrix, a truth table and for each function an input line to edit Boolean expressions.

All parts of the picture are interactive, e.g. points in the programming matrix can be clicked, and expressions can be edited and values in the truth table can be changed by mouse clicks. Examples, given in the text book, that way can be explored interactively. Additionally, own examples can be created and explored. Learners can interact with this interface by direct manipulation of formulas or values and make own experiments. In this example of a "living picture" students can understand how to program PLDs (read only memories (ROM), programmable logic arrays (PLA) and programmable array logic (PAL)). Students can compare the differences in programming the same function with different PLDs. We will explain this example more in detail in the next chapter.

Concerning to the learning goal, the developed tools support the design of control systems by using Boolean equations and finite state machines and parallel automata and help to verify the design. We have developed them for each step of the design process of digital circuits.

Table I gives an overview of the design steps and the developed, special tailored applets. That kind of applets we use also for demonstration purposes during lectures.

By using these tools it is much easier to teach in a problem based manner. We give the students an interesting, practical task and show them how to solve it with the taught methods and the use of the tools. Because the tools are tailored to the learning process, they are easy to use and have a self explaining user interface. The functionality is limited to the actual discussed design step and small-size tasks. Thus, the students can concentrate on the actual design step and are not distracted by a confusing number of features that design tools normally provide.

4. Applet for exploring programmable devices

The following chapter describes in detail the use of the introduced applet to give a better understanding of the functionality and the interaction possibilities.

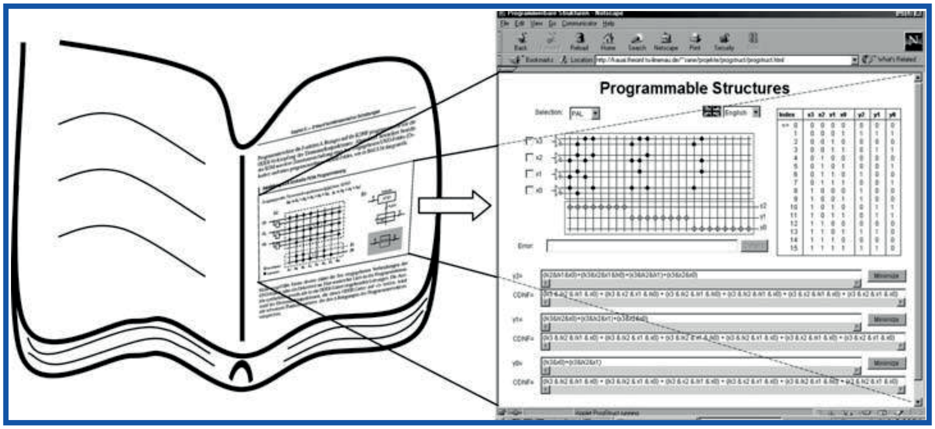
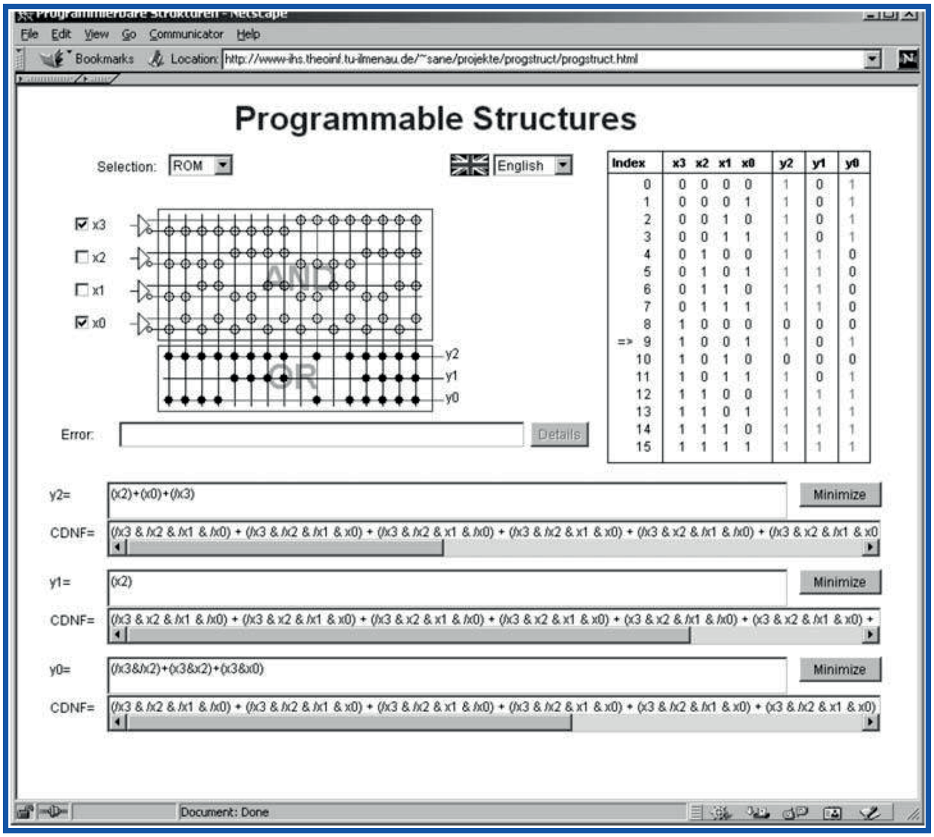


Figure 1.
The "big picture" as a user interface

Table I.
Design steps and tools

Design step	Living picture
Functional description	Boolean set algebra
Transfer into a synthesizable formal description	Boolean expression algebra Normal forms
Optimization of the design	Minimization with Karnaugh maps and the method of Quine/McCluskey
Structural design	Graphical circuit editor Programmable structures
Sequential circuits	Automata design Flip-Flop simulation
Verification	Hazards detection FSM analysis and simulation

At the upper left part of the user interface (see Figure 2) learners can select the programmable logic device. They can choose between a ROM, a PAL or a PLA. Each programmable device consists of an AND-matrix and an OR-matrix. They differ in the programmability of these matrices. Up to three functions can be programmed in the programming field or in the truth table or can be edited in the form of Boolean expressions.


Programmierbare Strukturen - Netscape
File Edit View Go Communicator Help
Bookmarks
Location: http://www-ihs.theoinf.tu-ilmenau.de/~sane/projekte/progstruct/progstruct.html
English
☑ x3
☐ x2
☐ x1
☑ x0
AND
OR
y2
y1
y0
Details
Minimize
Document: Done

Programmable Structures

Selection: ROM

Index	x3	x2	x1	x0	y2	y1	y0
0	0	0	0	0	1	0	1
1	0	0	0	1	1	0	1
2	0	0	1	0	1	0	1
3	0	0	1	1	1	0	1
4	0	1	0	0	1	1	0
5	0	1	0	1	1	1	0
6	0	1	1	0	1	1	0
7	0	1	1	1	1	1	0
8	1	0	0	0	0	0	0
=> 9	1	0	0	1	1	0	1
10	1	0	1	0	0	0	0
11	1	0	1	1	1	0	1
12	1	1	0	0	1	1	1
13	1	1	0	1	1	1	1
14	1	1	1	0	1	1	1
15	1	1	1	1	1	1	1

Error:

y2= (x2)+(x0)+(/x3)

CDNF= (/x3 & /x2 & /x1 & /x0) + (/x3 & /x2 & /x1 & x0) + (/x3 & /x2 & x1 & /x0) + (/x3 & /x2 & x1 & x0) + (/x3 & x2 & /x1 & /x0) + (/x3 & x2 & /x1 & x0

y1= (x2)

CDNF= (/x3 & x2 & /x1 & /x0) + (/x3 & x2 & /x1 & x0) + (/x3 & x2 & x1 & /x0) + (/x3 & x2 & x1 & x0) + (x3 & x2 & /x1 & /x0) + (x3 & x2 & /x1 & x0) +

y0= (/x3&/x2)+(x3&x2)+(x3&x0)

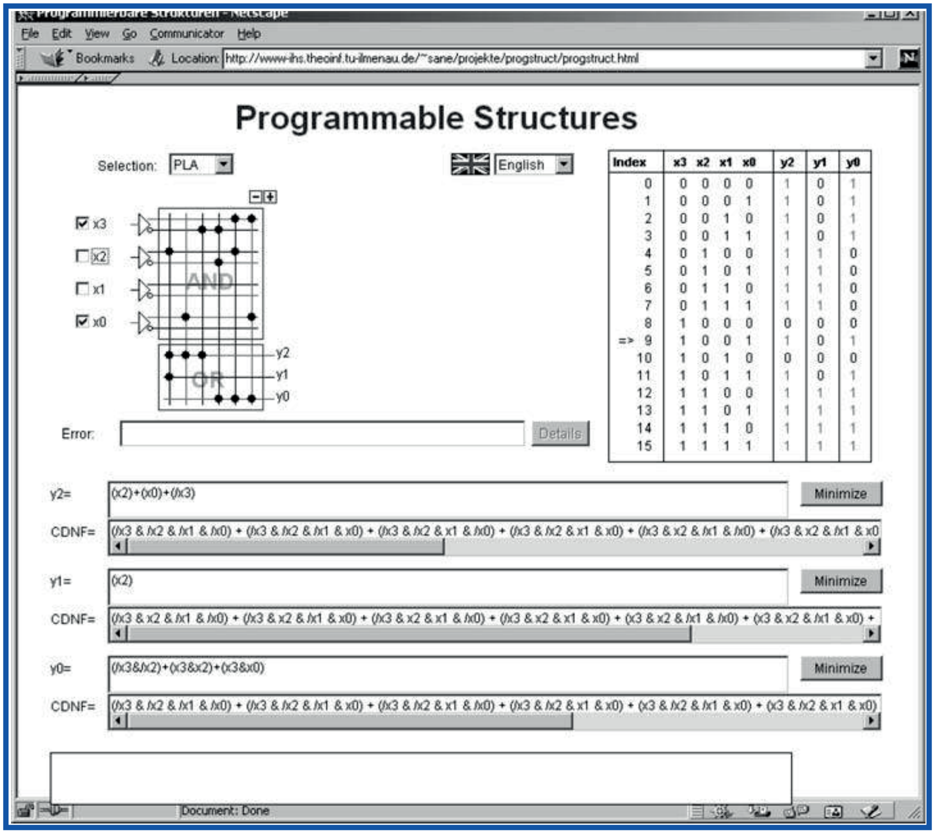
CDNF= (/x3 & /x2 & /x1 & /x0) + (/x3 & /x2 & /x1 & x0) + (/x3 & /x2 & x1 & /x0) + (/x3 & /x2 & x1 & x0) + (x3 & /x2 & /x1 & x0) + (x3 & /x2 & x1 & x0)

Figure 2.
ROM realization

The programmability of the line connections in the matrices is symbolized by crossing lines without connections. Fixed connected lines are symbolized by unfilled circles in the middle of crossing lines.

To understand the functionality of the chosen PLD, it is possible to assign the values “1” or “0” to the input variables x_0, x_1, x_2 and x_3 and watch the red colored, active elements in the circuit. Figures 2 and 3 show the same programmed function and the situation, where $x_0 = 1$, $x_1 = 0$, $x_2 = 0$ and $x_3 = 1$. In this situation the outputs of the three functions are $y_0 = 1$, $y_1 = 0$ and $y_2 = 1$. Ones the functions are programmed; it is also possible to compare the same functions and assigned values with realizations in the other PLD-types. Thus, students can see different realization principles and compare the effort of resources. Another possibility to compare different kinds of realizations is to watch the differences between a minimized and a canonical form of the function.

Figure 2 shows the programming of a ROM. In a ROM device the AND-matrix is fixed (symbolized by unfilled circles in the middle of crossing lines) and realizes a decoder-functionality. Below the fixed AND-matrix, a programmable OR-matrix can be programmed interactively. Clicking at any cross point of the lines in the OR-matrix installs a connection (filled circle) between a line of the decoder and the crossing output line of one of the output variables y_0, y_1 or y_2. That way an elementary product-term (logical AND connection) is added to the output function. The sum-of-products-term that realizes the output function of the connected output variable y_0, y_1 or y_2 is


Programmierbare Strukturen - Netscape
File Edit View Go Communicator Help
Bookmarks
Location:
AND
OR
y2
y1
y0
Details
Minimize
Document: Done

Location: http://www-ihs.theoinf.tu-ilmenau.de/~sane/projekte/progstruct/progstruct.html

Programmable Structures

Selection: PLA

English

☑ x3
□ x2
□ x1
☑ x0

Index	x3	x2	x1	x0	y2	y1	y0
0	0	0	0	0	1	0	1
1	0	0	0	1	1	0	1
2	0	0	1	0	1	0	1
3	0	0	1	1	1	0	1
4	0	1	0	0	1	1	0
5	0	1	0	1	1	1	0
6	0	1	1	0	1	1	0
7	0	1	1	1	1	1	0
8	1	0	0	0	0	0	0
=> 9	1	0	0	1	1	0	1
10	1	0	1	0	0	0	0
11	1	0	1	1	1	0	1
12	1	1	0	0	1	1	1
13	1	1	0	1	1	1	1
14	1	1	1	0	1	1	1
15	1	1	1	1	1	1	1

Error:

y2= (x2)+(x0)+(/x3)

CDNF= (/x3 & /x2 & /x1 & /x0) + (/x3 & /x2 & /x1 & x0) + (/x3 & /x2 & x1 & /x0) + (/x3 & /x2 & x1 & x0) + (/x3 & x2 & /x1 & /x0) + (/x3 & x2 & /x1 & x0

y1= (x2)

CDNF= (/x3 & x2 & /x1 & /x0) + (/x3 & x2 & /x1 & x0) + (/x3 & x2 & x1 & /x0) + (/x3 & x2 & x1 & x0) + (x3 & x2 & /x1 & /x0) + (x3 & x2 & /x1 & x0) +

y0= (/x3&/x2)+(x3&x2)+(x3&x0)

CDNF= (/x3 & /x2 & /x1 & /x0) + (/x3 & /x2 & /x1 & x0) + (/x3 & /x2 & x1 & /x0) + (/x3 & /x2 & x1 & x0) + (x3 & /x2 & /x1 & x0) + (x3 & /x2 & x1 & x0)

Figure 3.
Minimized functions in a PLA

simultaneously refreshed and described in a so called canonical disjunctive normal form. Thus the programmed functions are simultaneously shown in three views: as the programming matrix, as the truth table at the right hand side, and in the form of Boolean expressions in the edit lines below these two views.

The function can also be defined by clicking at the output values in the truth table or by editing the Boolean expressions. Any function with no more then four input variables can be defined. Syntax errors are indicated in the "error"-line. The "detailed"-button at the end of this line opens an extra window, describing the error in the notation of the last edited Boolean expression. The correct notation can be derived from examples, generated by clicking at the matrix or the truth table.

In a PAL device the OR-matrix is fixed and has a pre-defined number of connections to the AND-matrix. Opposite to the ROM scheme (see Figure 2) the upper left part of the applet is in that mode changed to an AND-matrix that can be programmed interactively. Clicking at any cross point in the AND-matrix installs an AND-connection between the input line and all other input variables connected to the same vertical line. That way a min-term is generated, which is fixed connected to an output line in the OR-matrix. (A min-term is a product-term that includes one or only some of the input variables or their complements). Here students learn that PALs are able to realize minimized functions in a disjunctive normal form.

The programmed functions are simultaneously shown in the truth table at the right hand side as well as in the form of Boolean expressions in the edit-able lines below the matrices.

For PAL devices, the minimized terms can be programmed. If a non-minimized function is inserted and the user changes the PLD-type, the applet will automatically minimize the function and display the minimized form, if the user returns to the PAL device view. To get this view, it is also possible to press the "minimize" – button.

The function can also be defined by clicking at the output values in the truth table or by editing the Boolean expressions in the same manner as described for the ROM-programming.

In a PLA device both matrices, the OR-matrix and the AND-matrix, are programmable. Figure 3 shows the programming of a PLA. The AND-matrix can be programmed by clicking a cross point in the matrix. That way a product-term that includes only the connected input variables (or their complements) can be defined at the same vertical line. New empty lines for further product-terms can be added by the "+" – button in the upper left corner of the AND-matrix and deleted by the "–" button.

The assignment of the product-terms to the output function (the sum-of-products) can be programmed in the OR-matrix. The same sum-term can be used in each function, if applicable. The programmed functions are simultaneously shown in the truth table at the right hand side and in the form of Boolean expressions in the edit lines below the matrices. For PLA devices, minimized terms can be programmed.

The described applet supports one step in the design process of a digital circuit, as it is required in a laboratory exercise. For preparing the solutions of laboratory tasks other applets support further steps of the design, as listed in Table I.

Based on the PLD-applet we have furthermore derived a number of interactive tasks that the students can use for an initial, intermediate or final assessment of their knowledge. These tasks are described in the next section.

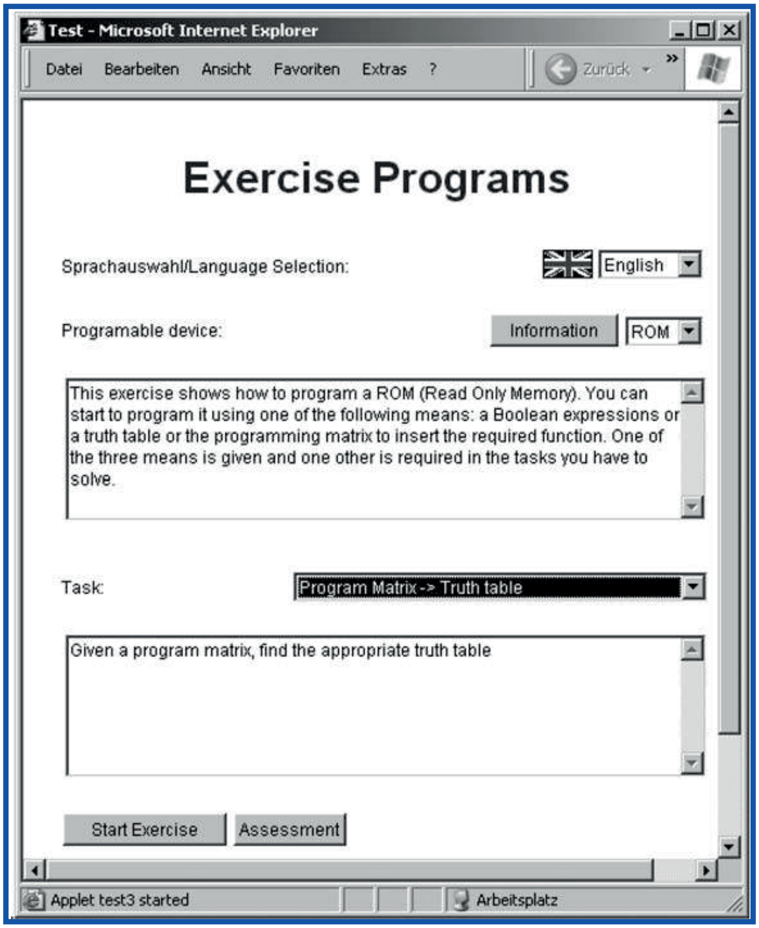
5. Assessment components

To derive interactive tasks from the described applet, the simulation component of the living picture was modified by a programmer. The applet was modified to write the simulation results in a separate data structure instead of displaying results of experiments immediately. In this way, most of the code of the applet could be reused. Only components for task generation and assessment have to be added. This kind of reuse requires special knowledge about the program code as well as appropriate programming skills and tools.

In these interactive tasks we present a function, described in one of the three above mentioned views of a programmable structure (as truth table, as Boolean equation or in the form of programmed matrices). The students' task is to find an equivalent description of the function in one of the other two views.

Figure 4 shows the initial task description. After starting the exercise, an example is randomly generated, where a function is given in one of the three forms. The task for the student is to determine the same function in another view.

Figure 5 gives an example, where the function is generated in the form of a programmed ROM structure ("required function").



Exercise Programs

Sprachauswahl/Language Selection: English

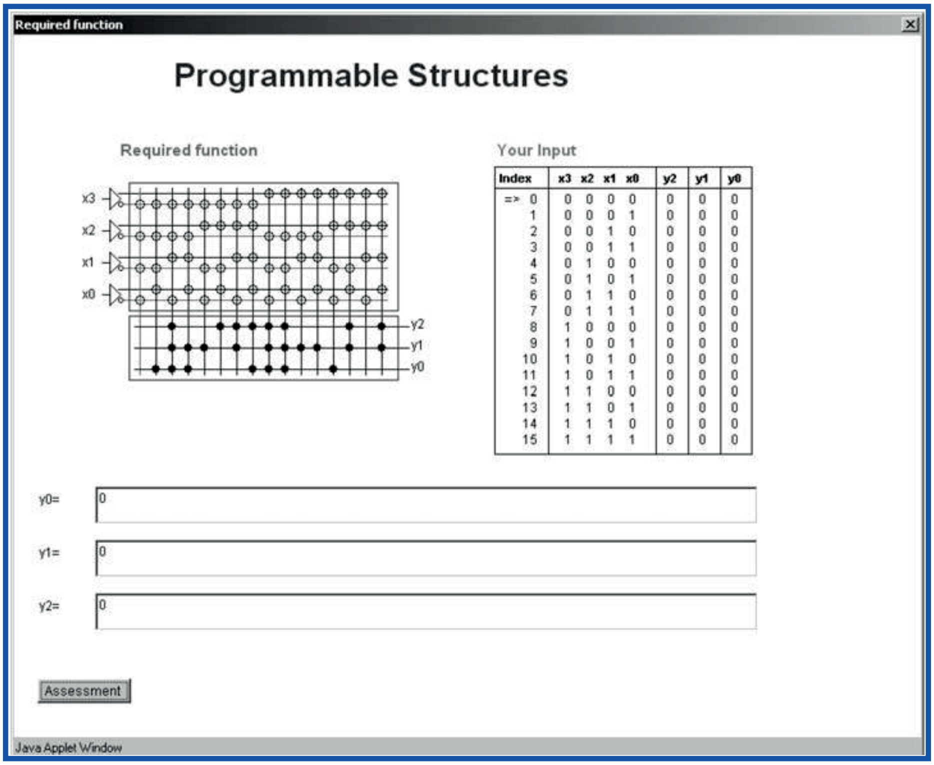
Programable device: ROM

This exercise shows how to program a ROM (Read Only Memory). You can start to program it using one of the following means: a Boolean expressions or a truth table or the programming matrix to insert the required function. One of the three means is given and one other is required in the tasks you have to solve.

Task: Program Matrix -> Truth table

Given a program matrix, find the appropriate truth table

Figure 4.
Screenshot of the task description



Required function

Programmable Structures

Your Input

Index	x3	x2	x1	x0	y2	y1	y0
=> 0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0
2	0	0	1	0	0	0	0
3	0	0	1	1	0	0	0
4	0	1	0	0	0	0	0
5	0	1	0	1	0	0	0
6	0	1	1	0	0	0	0
7	0	1	1	1	0	0	0
8	1	0	0	0	0	0	0
9	1	0	0	1	0	0	0
10	1	0	1	0	0	0	0
11	1	0	1	1	0	0	0
12	1	1	0	0	0	0	0
13	1	1	0	1	0	0	0
14	1	1	1	0	0	0	0
15	1	1	1	1	0	0	0

y0= 0

y1= 0

y2= 0

Figure 5.
Screenshot for task solving

Thus the students' task is to analyze the programmed structure and to find out, which values in the truth table have to be set to the values "1" and "0". For this task, the truth table is clickable and the values of the output functions y_2, y_1 and y_0 can be changed by simple mouse clicks ("your input").

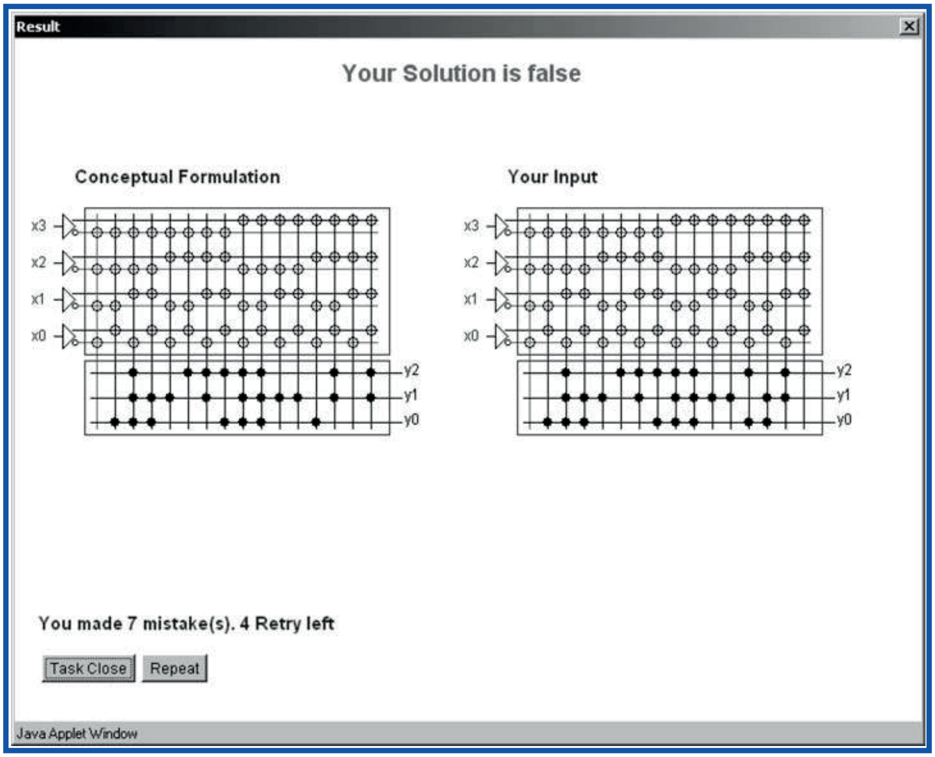
Opposite to multiple-choice-like questions, there are no hints to the right solution. Students have to combine the learned parts creatively.

If the student has found a solution he/she can get an assessment of the solution by clicking the appropriate button. The program counts the faulty output values and gives a feedback, showing the differences between the required function and the students' solution in the required view.

Figure 6 shows this intermediate result. In case of errors the student can repeat the solution up to five times. The trials are counted and displayed on demand as a statistic.

Such a kind of an applet is one component of our assessment system, called TExAS. TExAS stands for *Test, Examination and Assessment System* and is described in more details in Wuttke and Henke (2008) and Wagner (2007). This system supports online and offline tests. There are two kinds of assessment possibilities: low level tests like multiple choice and cloze questions and high level tests like described above.

The two kinds of assessment possibilities differ in the variety of possible answers as well as in the effort to produce new tasks. Inside the domain "programmable structures" the method, described above, is very comfortable because a very big


Result

Your Solution is false

You made 7 mistake(s). 4 Retry left

Figure 6.
Assessment window

amount of tasks (2^16 possible functions combined with three output variables and three different views) can be generated automatically. To generate such an amount of multiple-choice questions all variants have to be written in extra text files, a very boring task! Nevertheless, if we leave the domain of "programmable structures" another "living picture", listed in Table I, has to be varied by a skilled programmer.

The assessment system can be coupled to an LMS to extend the assessment possibilities. We have done this for the open source LMS "moodle" (Cole and Foster, 2007).

In the next chapter we describe the interaction between our assessment system and an extended LMS "moodle".

6. Extended learning management system

The assessment possibilities, provided by the applets, can only give a feedback to the actual solution and to the actual user. They are not suitable for a classroom use or any statistical evaluations. Such statistical and administrative possibilities offer LMS. That's why we coupled our assessment solution with an LMS. Thus, we can use the rich assessment possibilities of our software as well as the administrative support, given by the LMS.

We have chosen the "moodle" system because of its open and well published architecture. For the communication between the LMS and the applets we use the (Sharable Content Object Reference Model) Application Programming Interface (www.adlnet.gov). Therefore, the applets are embedded in shareable content objects.

Learning content in "moodle" is provided in the form of activity-modules. Thus we added three new kinds of activity-modules to include our modules in this LMS. The first one is a so called "AWiP-player", which is able to assemble a collection of tasks. The second one is a so called "Applet-player" to run the applets. The third one is for statistic purposes and collects the interaction data during a session. For more details see (Wagner, 2007).

Additional to these interactive teaching tools hands on experiments support the students learning process and the ability to solve creative tasks. Such hands on experiments in the subject of digital systems are remote controlled devices (hardware models of machine tools and the like). The students have to design the control algorithm and to implement it either in PLDs or as software solution.

The applets can be accessed also for learners that are not enrolled in a group in the LMS. Therefore a free web site can be accessed at www-ihs.theoinf.tu-ilmenau.de/forschung/projekte/sane/uebersicht_de.htm

References

Cherner, Y., Lotring, A., Klein, R. and Campbell, T. (2006), "Innovative simulation-based online system for learning engineering and training sailors' technical skills", *Conference Proceedings, Annual ASEE Conference*, *Chicago, IL*.

Cole, J. and Foster, H. (2007), *Using Moodle: Teaching with the Popular Open Source Course Management System*, 2nd ed., O'Reilly Media, Sebastopol, CA, ISBN-13: 978-0596529185.

Jeschke, S., Pfeiffer, O. and Richter, T. (2008), "User adaptive interactive courses in SCORM compliant learning management systems", in Auer, M. (Ed.), *Conference Proceedings IMCL 2008*, *Amman*.

Henke, K., Wuttke, H.-D. and Braune, T. (2007), "Virtual and remote labs in the educational process", in Auer, M. (Ed.) *Proceedings of the International Conference on Remote Engineering and Virtual Instrumentation, REV2007*, *Porto*, *25-27 June*, ISBN: 978-3-89958-278-9.

Wagner, N. (2007), "Konzeption und Umsetzung eines rechnergestuetzten Pruefungssystems", master thesis, TU Ilmenau, Ilmenau.

Wuttke, H.-D. and Henke, K. (2002), "Teaching digital design with tool – oriented learning modules ("living pictures")", *Proceedings of the 32nd ASEE/IEEE Frontiers in Education Conference*, *Boston, MA*, *6-9 November, CD-ROM*, ISBN: 780374444, pp. S4G 25-30.

Wuttke, H.-D. and Henke, K. (2008), "Pre- and post-tests as reusable information objects", *Communication & Cognition*, Vol. 41 Nos. 1-2, pp. 95-110.

Further reading

Prinke, S. (2006), "Integration eines remote labs in die Prozesse eines Lernmanagement-Systems", master thesis, TU Ilmenau, Ilmenau.

About the authors

Heinz-Dietrich Wuttke obtained his PhD and facultas docendi in Computer Science at the Ilmenau University of Technology. His research experiences and interests are in the fields of developing and organizing e-learning content and systems, remote laboratories and computer systems. He leads several national and international projects, has more then 50 publications in the last five years, and is a reviewer in international conferences and journals (e.g. editorial board member of the *International Journal of Advanced Technology for Learning*). He has a 20 years experience in teaching at a university and is supervisor of master and PhD students. He is also a member of national and international professional associations. For more details please visit: www.tu-ilmenau.de/fakia/Personen.1748.0.html

Karsten Henke obtained his PhD in Computer Science at the Technical University of Ilmenau. He is currently a member of the scientific staff at the Department of Integrated Hardware and Software Systems of the Institute Computer Engineering at the Ilmenau University of Technology. He has significant experience and expertise in distance learning development as well as web-based educational and training systems, methods and tools for the design and verification of embedded control systems for many years. Furthermore he is participating in projects funded by the European Union. He is Vice President of the International Association of Online Engineering and editorial board member of the *International Journal of Advanced Technology for Learning* and the *International Journal of Online Engineering*.