TECHNISCHE
UNIVERSITÄT
ILMENAU

Fachgebiet Grundlagen der Elektrotechnik
Fachgebiet Konstruktionstechnik

7. Workshop

„Multimedia für Bildung und Wirtschaft"

25. und 26. September 2003

TAGUNGSBAND

ISSN 1436 - 4492

mileET - Knowledge Based Assistance for Electrical Engineering Education

H. Garbe, V. Yakimchuk, C. Möbus, J.- P. Osterloh, H.- J. Thole, L. Weber (Kuratorium OFFIS e.V.), E. Wagner, TU Ilmenau

Abstract. MileET is an intelligent problem solving environment (IPSE) for a topic area of general electrical engineering that will be used at universities. It contains special teacher and student modes offering the ability to set up new tasks and to solve tasks assisted by the system. An underlying generative expert system enables mileET to offer knowledge based assistance. In order to build this generative expert system we classified the tasks according to their informal specified goals and elaborated a new task model (called Y-model). This Y-model, the base of our knowledge based assistant, describes the process of domain specific task solving.

1. Introduction

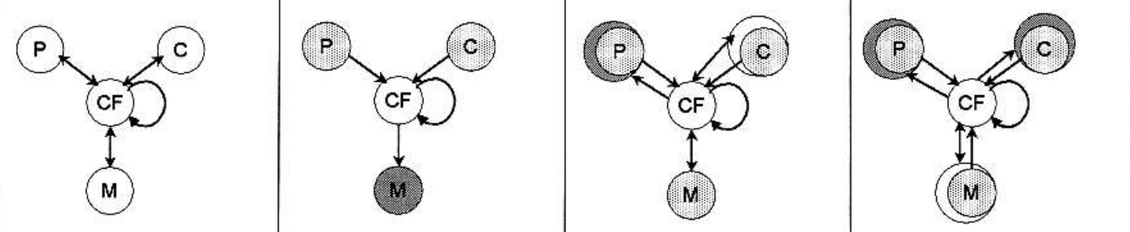
In the context of the BMBF-project mile (multimedia learning environments) [1] we provide an IPSE for a topic area from the basics of general electrical engineering [2, 3]. The IPSE should help the student to acquire procedural knowledge from this domain. The corresponding declarative knowledge can be acquired from the web-based learning modules of the partner universities. Our work is based on cognitive science research [4] and the development of IPSEs in various domains [3, 5, 6, 7] as well as on the competence of the partner universities in electrical engineering and their extensive task collections. The objective of our project is to develop a knowledge-based learning environment, which supports the learner throughout the whole process of problem solving of selected topics of the basics of general electrical engineering. To reach this aim a working environment is needed, that enables the user e.g. to edit circuits and equations. While solving tasks he should not be constrained to typical solutions, he should be as free as possible. The system is able to analyse even incomplete solution proposals and give adapted help.

2. Knowledge Representation and Hypotheses Testing

2.1 Y-Model

Analysing the task collections of the partner universities in detail we elaborated a model, called Y-Model, to represent the domain specific task solving process. This process can be seen as application of relations over the sets of circuits, parameters, mathematical descriptions, and electrical engineering concepts with their formulas, see Figure 1, on the left. The circuit topology is stored in the set C, symbolic and numeric parameter values of elements of the circuit in the set P, and mathematical equations in the set M. The elements in P, C, and M can be derived from each other by application of concepts and formulas (CF). This derivation represents the task solving process of an expert and can be produced by our IPSE by means of gmrs (see section 2.2).

In the considered tasks some parts of a circuit and parameters and mathematical equations are given and some are demanded to be build or calculated by the students. Depending on the task goal and the given data the appropriate Y-Type can be applied. Figure 1 shows the different Y-Types: the general model as well as the Y-Types for different tasks with examples. The sets initially given (P, C, or M) in a task are light grey and the goal sets (P,C, or M) are grey.



general Y-Model	Y-Type 1	Y-Type 2	Y-Type 3
The task solving process in general.	Example Task: Obtain the Kirchhoff current law equations belonging to the given circuit.	Example Task: Calculate equivalent resistor for group of resistors.	Example Task: Set up a circuit for measurement.

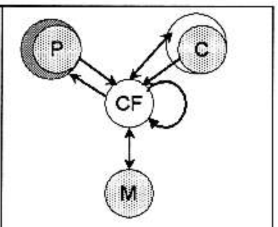
Figure 1. The Y-Model and possible tasks

2.2 Hypotheses testing with gmrs

To enable the testing of the hypotheses our IPSE contains a generative expert system, which is able to solve the tasks. Therefore, the essential electrical engineering knowledge has to be represented in the knowledge base of our system. For this purpose the goals-means-relations (gmr), a special type of prolog rules, are used. Gmr have been successfully implemented for varying domains [4]. The gmr assigns the means (solutions) to the goals (tasks and subtasks). The means are dependent on the domains. In our IPSE the means are parts of the three sets circuits (C), parameters (P) and mathematical equations (M) of the former described Y-Model. Additionally, there is a third parameter in the rules head. This is an informal text, that can be used to generate help in form of a trace of the solution steps of the program. A special Meta-Interpreter for handling the gmr-rules is implemented. Therefore, it is possible for the expert system to generate solutions for the tasks. In order to get many possible solutions the gmr-rules are very fine-grained. They can be combined to get different solution paths, also "unusual" ones. Owing to gmr the system is able to examine hypotheses of the students and to complete solution drafts if necessary.

```
/*  gmr(+GOAL, -MEAN)  */

gmr(transform(PCM_in), PCM_out, ['The circuit is simplified']) :-
    gmr(searchOne(ConnectionType, resistor), (Elements, Points)),
    gmr(getParameter(r, Elements), Parameters),
    gmr(replace(Elements, Points), [resistor, Name, Points, [R, I, U, P]]),
    gmr(calculate(ConnectionType, Parameters, R), Equation),
    gmr(transform (PCM_in_next), PCM_out).
```



Y-Type 2

Figure 2. gmr-rule example

The example (Figure 2) shows a gmr-rule for the equivalent resistor calculation. It is part of a gmr that simplifies the circuit. Tasks like this are described by the Y-Type 2, because the circuit and some parameters are given and some parameters have to be calculated. First, one has to find resistors of the given circuit that are connected according to the "ConnectionType" (e.g. parallel or series). The result of this step, the mean of this rule, is a part of the circuit (part of set C). As later on the resistance parameters of the resistors will be needed, the next step is to get them out of the set of parameters. Then, the resistors are replaced with one new resistor, the mean of this rule is a part of the circuit. The last step is to calculate the new resistance parameter; the mean of this rule is a mathematical equation as well as a parameter.

3. Working with mileET

MileET has got two different modes, one mode for students and one for teachers. As the teacher mode is an extension of the student mode, the student mode will be explained first.

3.1 Student Mode

Within the student mode students are supported while solving tasks. The working environment (Figure 3, Figure 4) consists of several areas. In the exercise area (Figure 3, on the left) the exercise settings and the information about circuit elements are shown; under Solution Hints, one can ask for possible solution steps generated by the system (Figure 4, on the left).

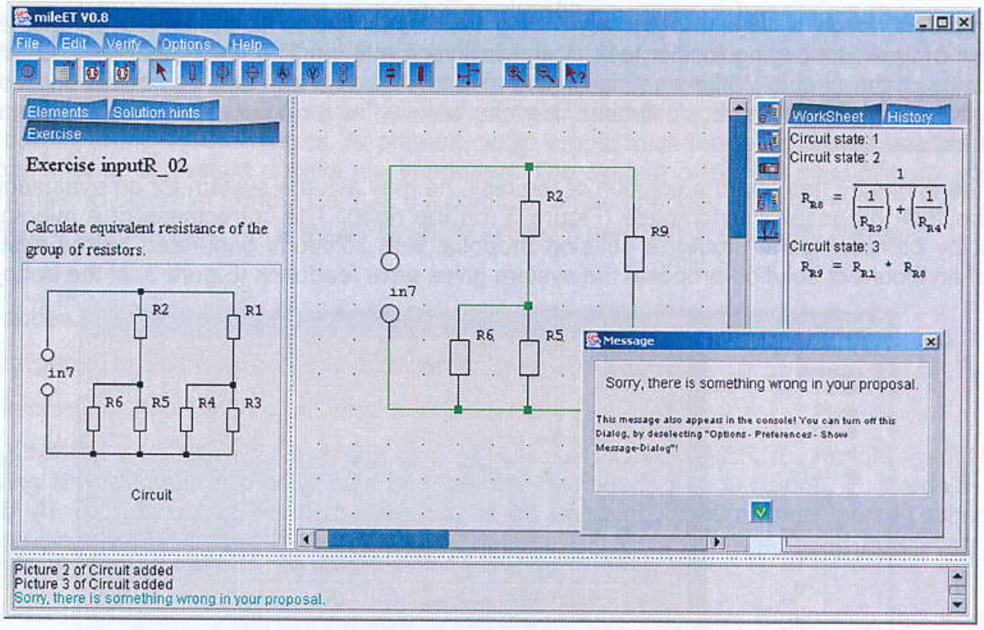


Figure 3. Incorrect Hypothesis (in the Worksheet)

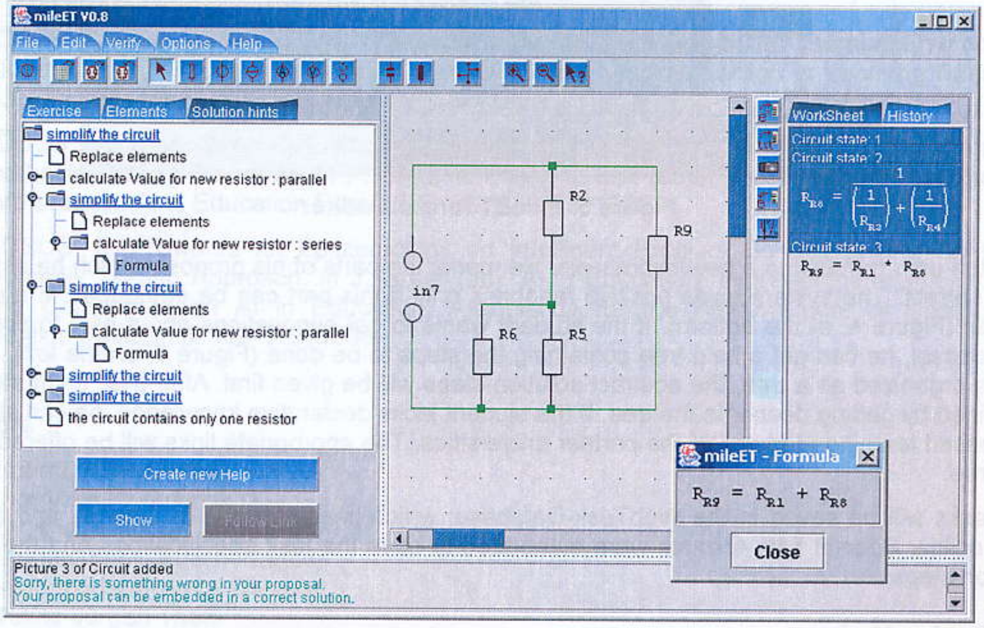


Figure 4. Embeddable Hypothesis (in the Worksheet) and Adaptive Help

In the middle of the working area the circuit editor is located. The toolbar above contains the tools, which can be used to manipulate the circuit. For instance new elements can be placed in the circuit editor or new connections can be drawn.

On the right side there is the worksheet area. This worksheet contains the solution proposal of the student and his hypotheses. If he thinks the circuit he has edited belongs to the solution of the task, he can copy it into the worksheet. Also he can write formulas with a special formula editor implemented for the system (Figure 5) and make textual annotations to his proposal.

The system's feedback to hypotheses testing can be seen at the bottom of the window and in a pop-up window. The user can turn off the appearance of this pop-up window so that the message appears at the bottom of the environment only.

If the student loads a task, the corresponding task specification is presented on the left and possibly a circuit belonging to this task is shown in the middle. The worksheet just contains the initial state of the circuit. While working on the task, the student can edit this circuit and copy the different states of it into the worksheet. He can also write formulas and annotations into the worksheet and save it.

When he thinks he has found a solution of the task, he may ask the system for an evaluation even if the solution proposal is incomplete (Figure 3, on the right). The correctness of a hypothesis is proved by comparing the student's solution proposal with solutions generated by the system. In case of an incorrect solution proposal the system gives error feedback (Figure 3, at the bottom).

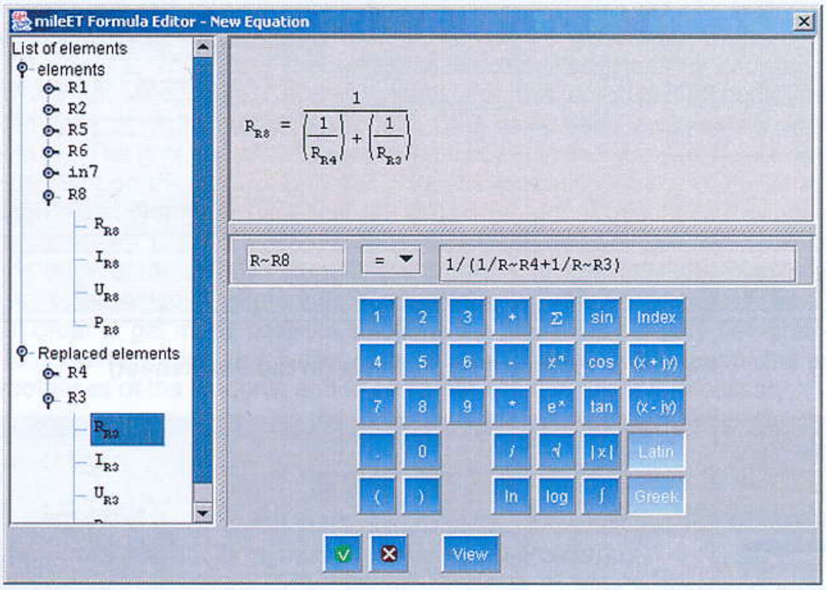


Figure 5. mileET formula editor

Now, the user formulates a new hypothesis. He marks the parts of his proposal which he assumes to be correct. The system gives positive feedback only if this part can be embedded in a correct solution (Figure 4, at the bottom). If the student wants to get suggestions about how to complete his proposal, he can get a help tree containing the steps to be done (Figure 4, on the left). As the help is organized as a tree, the abstract solution steps will be given first. After that, each step can be refined by getting deeper in the tree. If the student lacks declarative knowledge, he can start the web-based learning modules of the partner universities. The appropriate links will be offered in the help tree.

The tasks will be saved in the WebTask-Database, which is established in Ilmenau and can be used online. Special XML-formats were elaborated to save the task specifications and the user's solution steps.

3.2. Teacher Mode

To adopt the tasks to different courses and to set up new tasks, mileET provides a teacher mode. Using this teacher mode teachers can build new tasks for mileET and export them for deployment in the TaskWeb Database. In order do this no programming skills are needed. Also, students can use the teacher mode to set up their own tasks. The tasks (for examples Figure 3, on the left) can contain a freely editable text, circuits constructed using the mileET circuit editor and formulas written with the mileET formula editor. After the teacher has completed the text, the circuit, and the formulas he has to choose a task goal from a list of supported task goals (in the above example this goal is to determine an input resistance). With regard to the Y-Model this means that the task types can be chosen from a fixed (because in gmr implemented) set of task types. But the initial circuit and parameters for the tasks can be freely created and varied by the teacher. This is possible because the gmr does not work with fixed values but on the sets P, C, and M.

4. Conclusion

This paper described an Intelligent Problem Solving Environment for electrical engineering. The system supports the student throughout the whole task solving process. As there is a teacher mode that enables teachers to easily create new tasks it is adjustable to different courses, as long as the tasks belong to the covered topics. At present, tasks e.g. to such topics as basic electrical circuits, Kirchhoff's laws, equivalent circuits are supported. The system offers the possibility to adjust the environment to personal likings. So, e.g. the colours can be changed. At the moment, the system's language is english. Later on, it will be possible to switch between english and german.

References

[1] http://get-20.e-technik.tu-ilmenau.de/mile/

[2] lls.informatik.uni-oldenburg.de/projekte/et/index.html

[3] YAKIMCHUK, V., GARBE, H., MÖBUS, C., THOLE, H.-J., WAGNER, E., An Intelligent Problem Solving Environment in the Domain of Electrical Engineering, in U.Hoppe, F. Verdejo, J. Kay (eds) AI in Education: Shaping the Future of Learning through Intelligent Technologies, Proceedings of the 11th International Conference on Artificial Intelligence in Education (AIED) 2003, p. 518-520, 2003.

[4] MÖBUS, C., SCHRÖDER, O. & THOLE, H.-J., Diagnosing and Evaluating the Acquisition Process of Programming Schemata, in J.E. Greer, G. McCalla (eds), Student Modelling: The Key to Individualized Knowledge-Based Instruction, Berlin: Springer (NATO ASI Series F: Computer and Systems Sciences, Vol. 125), p. 211-264, 1994

[5] MÖBUS, C. et al., Towards an AI-Specification of Intelligent Distributed Learning Environments, AI Journal 1/03 "Knowledge Modeling and Knowledge Communication in Learning Scenarios", 2003

[6] MÖBUS, C., Towards an Epistemology on Intelligent Design and Modelling Environments: The Hypothesis Testing Approach, in P. Brna, A. Paiva, J. Self (eds), Proceedings of the European Conference on AI in Education EuroAI-ED '96, p.52-58, 1996

[7] MÖBUS, C., Towards an Epistemology on Intelligent Problem Solving Environments: The Hypothesis Testing Approach, in J. Greer (ed), AI in Education, Proceedings of AI-ED 95, 7th World Conference on AI in Education, Association for the Advancement of Computing in Education (AACE), p. 138-145, 1995

Authors

Dipl.-Inform. Hilke Garbe[1]
Dipl.-Ing. Vera Yakimchuk[1,2]
Prof. Dr. Claus Möbus[1,3]
Prof. Dr.-Ing. habil. Edwin Wagner[2]
cand.-Inform. Jan-Patrick Osterloh[1]
StW. Heinz-Jürgen Thole[1]
cand.-Inform. Lars Weber[1]

[1]Kuratorium OFFIS e.V.,
Escherweg 2, 26121 Oldenburg, Germany

[2]Technical University of Ilmenau
Faculty of Electrical Engineering and Information Technology
Institute of General Electrical Engineering
Helmholtzplatz 2, PF 10 05 65, 98684 Ilmenau, Germany

[3]Carl von Ossietzky University of Oldenburg
Department of Computing Science
Learning Environments and Knowledge-Based Systems
26111 Oldenburg, Germany

E-mail: mileET@offis.de