

Project ZERO: Developing Online Material for Mathematics Teacher Education

1. Starting Point and Aim of the Project

Beginners often find it hard to integrate new and unfamiliar mathematical concepts and methods into their knowledge from school. Solving problems and exercises in training groups therefore usually plays an important role in overcoming these initial obstacles. Such mathematical activities, however, will be successful only if their focus is on genuine problem solving rather than on skills and routines. As, on the other hand, the acquisition of routines may not be simply neglected, the idea arose making online available appropriate exercises in various formats and on various levels. With this supply students should be given the opportunity to control and assess themselves and to practice according to their own needs. In order to pursue this aim ZERO has been started in 1996/97 as a small project at the University of Flensburg. In what follows I intend to give a survey of the content and the sort of interaction to be found in it; then, looking behind the curtain, I shall put forward some of its driving wheels and levers.

2. Survey of the Contents

ZERO has its own web-site (<http://www.uni-flensburg.de/mathe/zero/zero.html>) on which are presented learning materials of three categories:

Lecture notes (links: „Veranstaltungen“, „Literaturhinweise“).—Some dozens of chapters from various courses have been transformed into webpages, a stock continuously maintained and to be extended in the future. Emphasis on logical structure was decisive in choosing HTML (though page layout is rather limited and display of formulae nearly impossible). The HTML files contain text, graphics, links (to references and remote sites), MATHEMATICA notebooks, and dynamic figures.

Exercises (links: „Übungsaufgaben“, „Aufgaben-Trainer“). —The collection of exercises currently (June 2001) contains round about 300 items covering the central topics from the introductory courses, e.g., natural numbers and induction, integers and divisibility, the concept of function, some elementary combinatorics, algebra, and geometry. Items concerning heuristics and didactics will be added later. The exercises are for the most part elementary. From a technical point of view three main types have to be distinguished:

- *paper-and-pencil-exercise*:
The learning task has no special format. The student may receive help (i.e., a hint at how to solve the problem) and get a solution on demand.
- *dynamic-geometry-module*:
A dynamic figure is presented together with a question or exploratory task. In some cases individual feedback is provided to different responses (actions).

- *computer-based-training-frame*:

A learning task is presented within a frame where the student makes his or her input which then will be analyzed and commented. This procedure may be iterated.

I am aware that none of these categories is a magic pill to cure our ills or is even free of shortcomings; and, of course, there are further formats (e.g., simulations) which could be used with at least equal benefit. However, it seems acceptable to offer a versatile mixture the quality of the whole depending heavily on the quality of its ingredients.

Supplements (links: „Figuren-Galerie“, „Mathematiker-Galerie“, „Denkzettel“).—This kind of content is organized in three sections: a gallery of mathematical objects (up till now mainly dynamic geometry), a gallery of mathematical people (with concise comments on their lives and work), and „denkzettel“, a sporadically growing collection of miniatures concerning the relation between mathematics and the ‚rest of the world‘ (e.g., art, literature, science, philosophy, pedagogy). The „denkzettel“s stand open for contributors from outside, and open up an experimental field beyond any kind of bureaucratic habits whatsoever, thus offering an opportunity for prospective teachers to develop a more flexible and as well intellectually as emotionally a richer relation to mathematics.

3. Outline of the CBT- Module („Aufgaben-Trainer“)

The „Aufgaben-Trainer“ (exercise-coach) can be downloaded from the ZERO-site. It is a program for testing and practicing currently comprising 120 exercises (arithmetic, algebra, no geometry) to be used offline. For online use 70 percent of the exercises have been prepared as CBT-style frames.

The software is based on an abstract adaptive tutoring system (DUAL) which has been developed since 1991 and described in greater detail in [7] and [8]. The system is built up as a three-tier application with the following layers: 1. a relational database model which contains the items to be presented as exercises, 2. a tutoring module which processes the inputs/actions of the student and decides what to do next, 3. a visual front-end serving as the communication interface between student and system. On a closer inspection, in the second layer, there are two components working together:

1. A MACROTUTOR decides, on the basis of data up till then produced by the student’s actions, which item to present next, thereby using a special teaching algorithm. There is a slot allowing such algorithms (called *teaching strategies*) to be selected on demand. Since the user gets access via a personal password the system can process his/her data individually.
2. A MICROTUTOR presents the frames (items) requested by the MACROTUTOR. The item ‚points‘ to the frame-type (e.g., multiple-choice, matching and free-text questions) and the external data needed for the presentation. So, the MICROTUTOR’s cyclic four-stage-process (fig. 1) can be started the result of which, finally, returns to the MACROTUTOR.

From the beginning, basic ideas with the design and development of the system had been a *logical description of instruction* (cf. [2], [6]) and the *separation of the components* (cf. [5], [7]). By consequence, the MICROTUTOR is able to do his work without the MACROTUTOR. Practically does this mean that programs (clients) may call individual frames from any DUAL-project outside—a signifi-

cant feature not shared by those authoring/tutoring systems which use a *combined structure* of content and logical data.

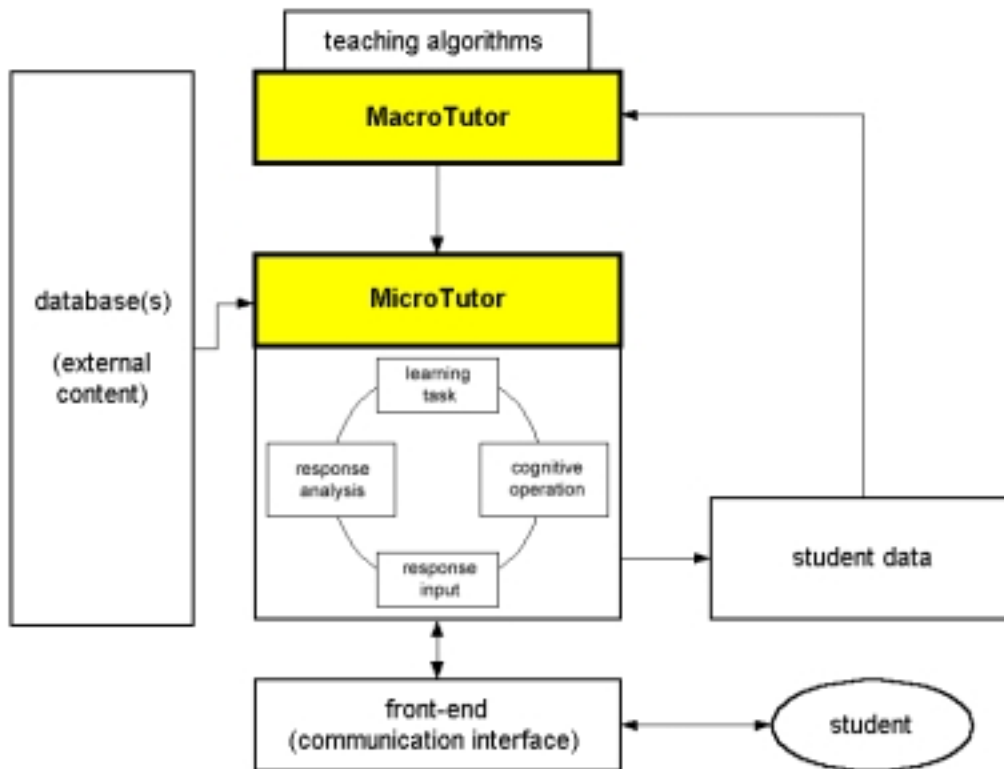


Fig. 1. Schematic Structure of the Tutoring System DUAL

4. Computer-Based-Training-Frames

In order to be able to present the various frames (exercises) from the „Aufgaben-Trainer“ on the web the MICROTUTOR had to be re-developed as a Java applet. The following types of frames are supported by the new module: true/false questions, multiple-choice-frames of both types ‚1 out of n ‘ and ‚ k out of n ‘ (with choices in random order), injective and non-injective mappings from one set of objects to another (fig. 2), free-text questions and fill-in-exercises with single- and multiple-string answers.

The frame-content is stored in a relational database and can be retrieved by the MICROTUTOR. A record of such frame-data not only contains texts and images needed for the display but also specific informations concerning the internal logic of the frame, e.g., a set of rules prescribing how to judge the student’s response. The MICROTUTOR uses two kinds of procedures: general schemes that apply to all types of frames, and specialized routines for the processing of type-dependent data.

Remark.—A short remark on response analysis shall illustrate this. As I have shown in [8], one uniform method applies for the various forms of matching and multiple-choice, whereas the evaluation of arbitrary alphanumeric inputs is an entirely different task requiring its own sophisticated routines. The possible answers of a free-text question, for example, have to be known in advance. By means of boolean and fuzzy operators, the author may define classes or patterns of strings and add them to the database (for details cf. [4], [8]). Judging a given answer means testing whether it belongs to one of these predefined classes or not. If not, then the answer has to be judged as ‚not yet understood‘ though it might be correct. This is a shortcoming all kinds of pattern matching have in common (even those using semantic methods, since no user can be prevented from intentionally entering nonsense). What I learned from experience is a twofold advice: to

expect, optimistically, reasonable inputs from students, and to choose the context of the problem so as to confine the range of potential inputs.

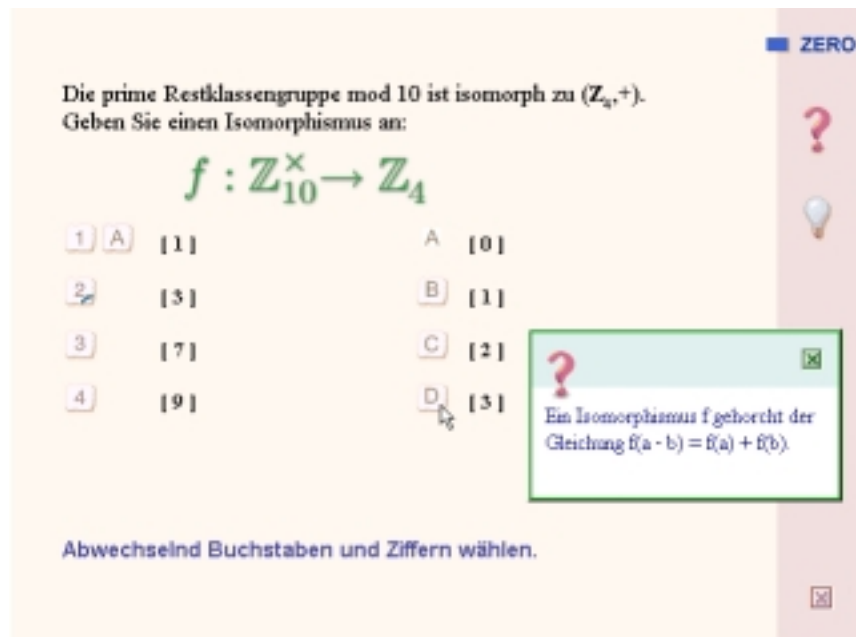


Fig. 2. Example of a Frame of Type ‚injective mapping‘

5. Dynamic-Geometry-Modules

These elements (interactive work sheets, exercises) have been developed using a special class library of dynamic geometry methods. Authors can rely on flexible facilities for ‚scripting‘ the figures and interfacing to Java/JavaScript. For more details cf. [3] and the contribution of T. Ehmke in this volume.

6. References

- [1] Alessi, S. M. / Trollip, S. R.: *Computer-Based Instruction. Methods and Development*. Prentice Hall: Englewood Cliffs, New Jersey 1985, 2nd ed. Allynand Bacon
- [2] Eckel, K.: *Instruction Language. Foundations of a Strict Science of Instruction*. Educational Technology Publications, Inc.: Englewood Cliffs, New Jersey 1993
- [3] Ehmke, T.: *Eine Klasse beweglicher Figuren für interaktive Lernbausteine zur Geometrie*. PhD Diss. University of Flensburg 2001. URL: http://www.uni-flensburg.de/mathe/homepg/schreiber/projekte/projekte_index.html
- [4] Nesbit, J. C.: Approximate String Matching in Response Analysis. In: *Journal of Computer-Based Instruction* 12 (1985), 71-75
- [5] Schreiber, A.: Bausteine für Lernprogramme: Beschreibung und Implementierung. In: *Informatik- Fachberichte Nr. 259*. Springer: Berlin, Heidelberg, New York 1990, 333-348
- [6] Schreiber, A.: Review of [2]. In: *Zentralblatt für Didaktik der Mathematik* 22 (1990), 114-116
- [7] Schreiber, A.: Eine Didaktik-Umgebung für Adaptives Lernen (DUAL). In: *Grundlagenstudien aus Kybernetik und Geisteswissenschaft / Humankybernetik* 33 (1992), 25-31
- [8] Schreiber, A.: *CBT-Anwendungen professionell entwickeln*. Springer: Berlin, Heidelberg, New York 1998