Prof. Dr. Alfred Schreiber
University of Flensburg, Germany

Project ZERO: Developing Online Material for Mathematics Teacher Education

Summary. This paper reports about a project dealing with the conception and production of supplementary learning material for mathematics teachers. It surveys the various types of courseware-modules presented herein online (e.g., dynamic geometry, computer-based-training-like frames, paper-and-pencil-exercises), and discusses their specific purpose and use. Emphasis is put on the problem of how to embody appropriate functions that provide the opportunity to evaluate user inputs – thus enabling an author to give "local" feedbacks to the student.

1. Starting Point and Aim of the Project

Nowadays many university lecturers are used to provide texts and supplements for downloading from their websites. With respect to mathematics, however, there still are further-reaching specific requests. As is well known, beginners often find it hard to integrate new and unfamiliar mathematical concepts and methods into their „old“ 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 successfull only if their focus is on genuine problem solving rather than on skills and routines. On the other hand, to ensure that the acquisition of routine skills is not neglected, appropriate exercises are given on-line so that students can self assess and then 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 (summer 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.

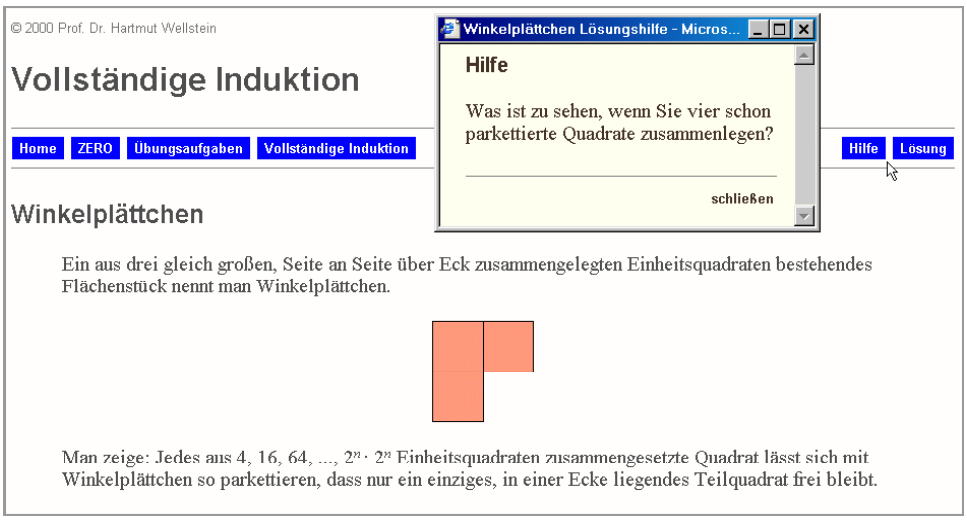

Fig. 1. *Example of a paper-and-pencil-exercise. A help-window is opened.*

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“, „Denkzettel“).—This kind of content is at present arranged in two sections: a gallery of mathematical objects (up till now mainly dynamic geometry), 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, history, pedagogy).

A gallery of mathematical people (with concise comments on their lives and work) will be added in the future.

The "denkzettels" are open to contributors from outside, and open up the opportunity for people to experiment without any bureaucratic constraints, thus offering an opportunity for prospective teachers to develop a more flexible as well intellectually and emotionally richer relationship with 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]. Here I will comment on the system features only so far as it is required for the understanding of the following.

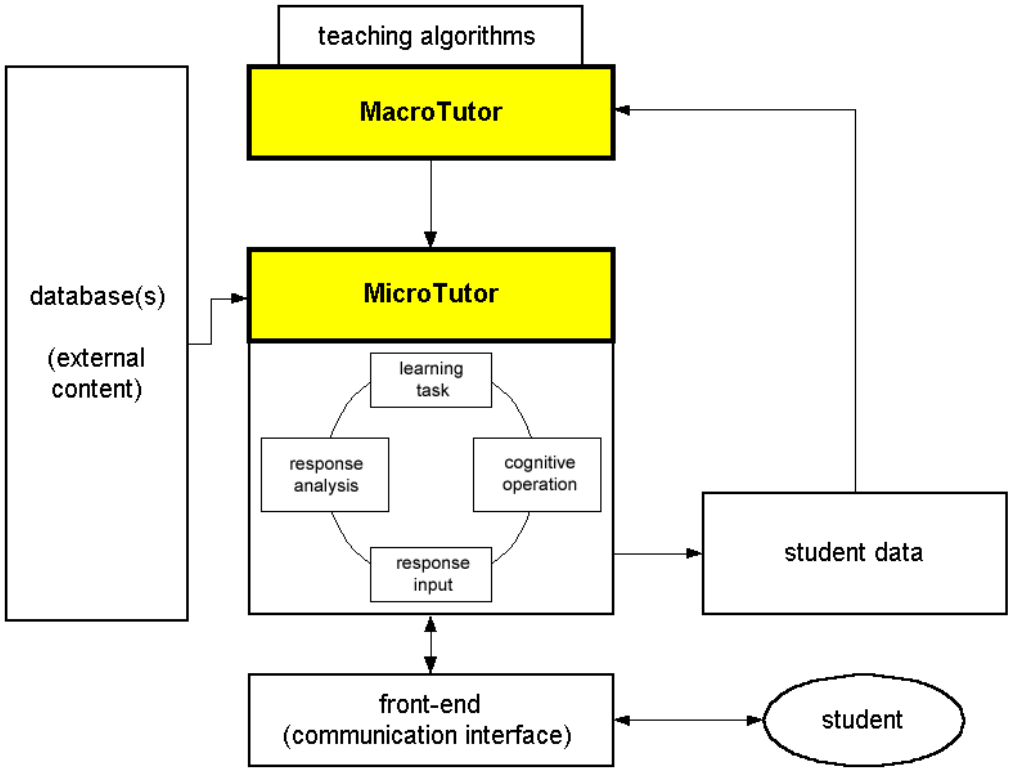

Fig. 2. *Schematic structure of the tutoring system DUAL*

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 (fig. 2). On a closer inspection there are two components working together in the second layer:

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 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 significant feature not shared by those authoring/tutoring systems using a *combined structure* of content and logical data.

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. 3), free-text questions and fill-in-exercises with single-string 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.

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], [9]). 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.

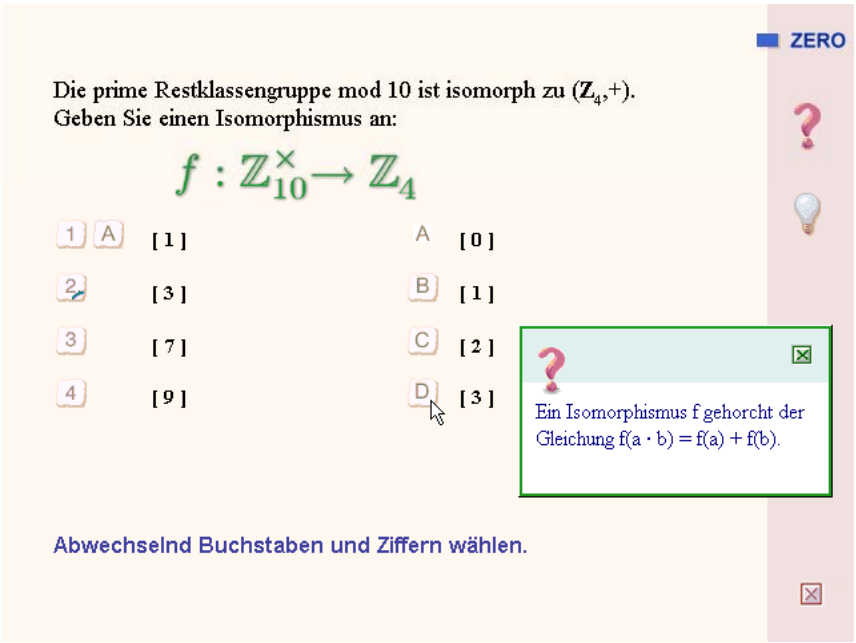

Fig. 3. *Example of a frame of type 'injective mapping'*

5. Dynamic-Geometry-Modules

Nowadays, most geometry programs are able to save their figures as files that can be displayed in a web-browser. One of the first systems which from the start served as an on-line tool was D. Joyce's GEOMETRY-APPLET. Partly, it served as a basis of a project [3] that aimed at providing a large class of dynamic objects which both can be more flexibly generated and easily be published on the web. I will only shortly point out the essential features of this approach:

1. A figure comes into existence by a sequence of commands written in GEOSCRIPT (a mixture of markup codes and Joyce's parameters) and stored in a separate file. So, the focus is on an author defining a geometric figure in order to produce, e.g., some interactive working sheet. The result will be viewed by an applet (named GEOMETRIA).

2. You may extend GEOSCRIPT by adding your own class-libraries (without having to compile GEOMETRIA'S sources again). The interfacing is done by a central class `Measure` which, among other things, is able to both identify and control the state of an object. Thus authors get access to the system's internal functions as well as to the full power of Java. JavaScript may be used, too.

3. An object not only can be built up by compass-and-ruler-methods or with the help of geometric mappings, but also directly through analytical expressions. Even an arbitrary point set

(represented as a two-dimensional boolean array) may control the dislocation of draggable points. Furthermore, by means of an external interface, recursive methods may be used, e.g., in order to produce *variable* fractal objects (fig. 4).

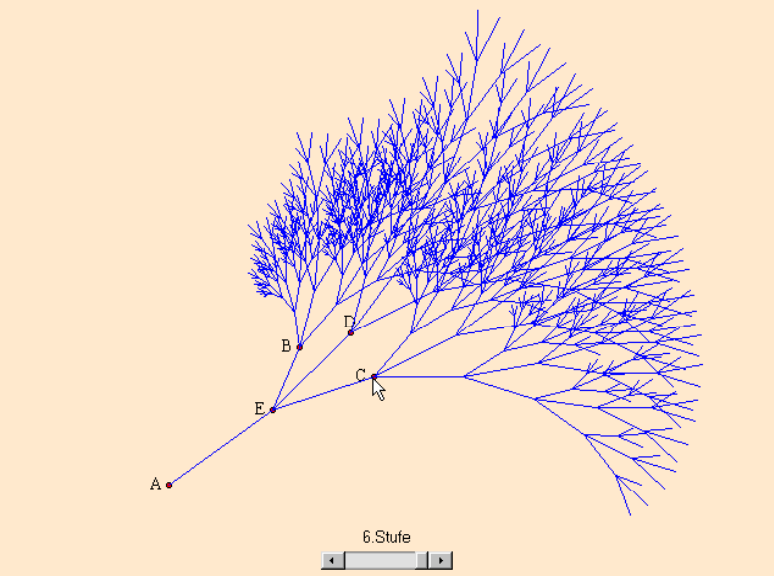

Fig. 4. *A fractal tree which can be varied by the points A,B,...,E*

4. Computer-based training functionality: Some sort of frame-structure can be added to a figure thus allowing certain of its internal states to be recognized. As a student's input corresponds to a predefined type of action belonging to a given set, the author is able to provide individual comments on these inputs (fig. 5).

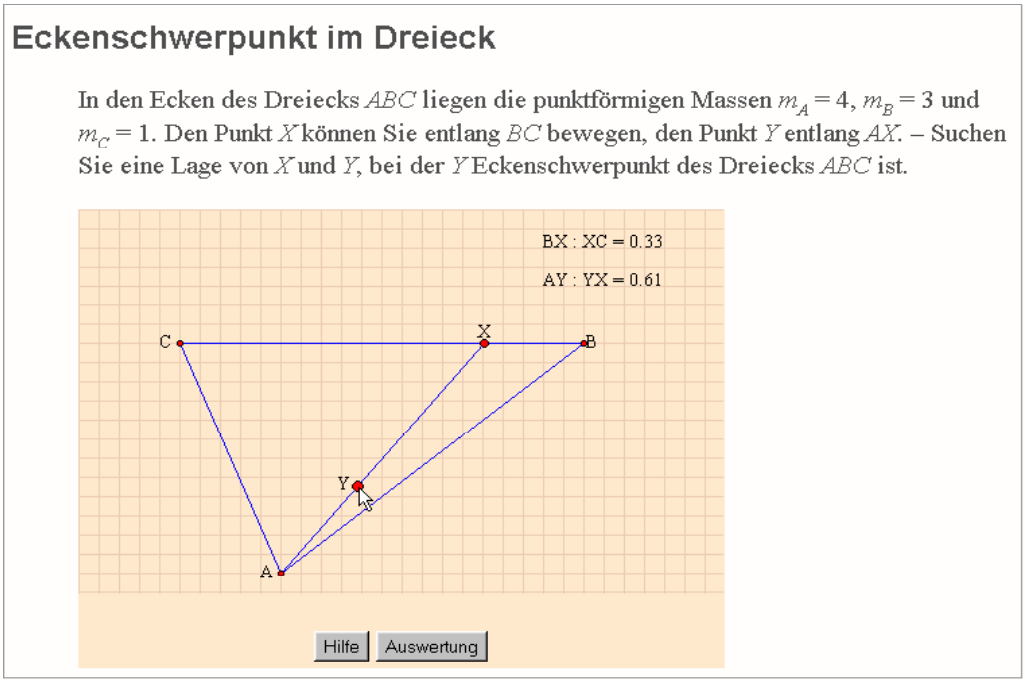

Fig. 5*. Dynamic figure including help and input evaluation*

Finally, it should be mentioned that EUKLID DYNAGEO (created by R. Mechling) is able to export its objects as editable GEOSCRIPT files. For more details cf. [3] and also the GEOMETRIA website http://www.geometria.de.

Valuable contributions to the project have been received from Hartmut Wellstein (selected chapters from a geometry course, a lot of exercises, and some „denkzettels“), and from Timo Ehmke (GEOMETRIA APPLET, and examples for the gallery of figures). Over the years I have been helped and encouraged by many people. To all these I am very grateful.

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

[9] Schreiber, A.: Ein logischer Rahmen für die Antwortanalyse in Lehrprogrammen. To be published in: Grundlagenstudien aus Kybernetik und Geisteswissenschaft / Humankybernetik