TEE - The Electronic Exercise

Rüdiger Krauße, Hermann Körndle

Psychology of Learning & Instruction
TU Dresden
D-01062 Dresden, Germany
ruediger.krausse@psychologie.tu-dresden.de
hermann.koerndle@psychologie.tu-dresden.de

Abstract: The Electronic Exercise (TEE) is an experimental adaptive learning system which uses knowledge structures to apply an intelligent tutoring algorithm and especially the representation of its surmise relation via HASSE diagrams to navigate within the learning material, to visualize the learning progress and convey the structure itself to the learner. TEE is implemented via dynamic HTML and comes with an editor tool for the structure and the test exercises. Thus an author or teacher can build its own TEE system easily.

1 Introduction

The Electronic Exercise (TEE) is a small experimental learning environment that was designed and implemented at the TU Dresden, Germany. It takes advantage of knowledge structures in various kinds of ways. The knowledge is broken up into small units (elements, building stones, nodes) and every unit has its test exercises which query the contents. These nodes are linked by the well known surmise relation: an expert or teacher surmises that a certain node cannot be mastered until the learner has successfully dealt with other nodes. In other words, these nodes secure sufficient pre-knowledge for the considered node [AL99].

The most obvious application of this knowledge structure in TEE is the HASSE diagram of the surmise relation. It is visible permanently and can be used to navigate within the learning material. In addition it displays the learning progress by the colours of the nodes. Furthermore the structure serves as a base for an intelligent tutoring mode, which first establishes the knowledge state of the learner and subsequently supports a systematic extension of this knowledge by providing new units. The advantage of TEE is quite evident: in addition to all the usual benefits of systems using knowledge structures, the knowledge domain itself is structurally presented to the learner. This makes the actions of the system understandable and thus avoids being “lost in hyperspace”.

However, this leads to a rational upper limit of nodes (about 30, because of the readability of the HASSE-diagram) and the skills are mapped to learning material/knowledge structure which allows no OR-connection of skills to surmise another node. Therefore, TEE can merely be considered as an experimental application, which is very useful to familiarise oneself with the idea of knowledge structures and links theory with practise. Since authors can fill it with their own learning material and exercises, TEE can be understood as a framework.

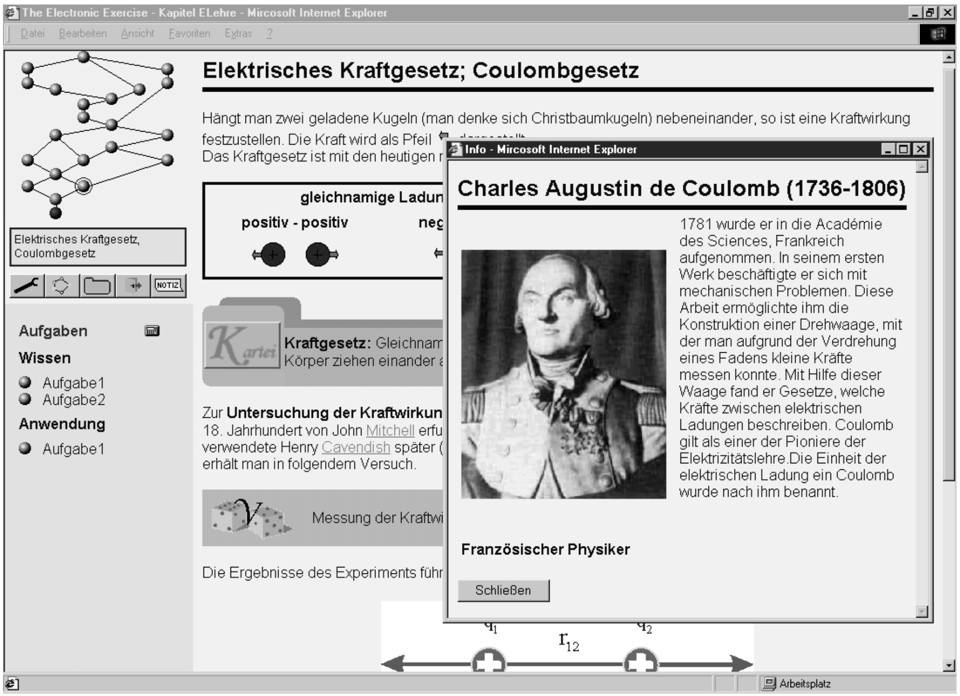

Figure 1: Screenshot of TEE with learning materials

2 TEE: How it works

Since TEE is implemented by dynamic HTML it can be used at every computer with an internet browser of at least 4th generation. The main part of the screen is occupied by the working area, in which the learning material, the exercises or a large diagram of the surmise relation is displayed. At the left side a smaller version of the diagram is always visible and a state window provides various information, control features and access to corresponding exercises. Both diagrams can be used for navigation – a mouse click at a node opens its learning material and displays test exercises. Thus, it works as a two-dimensional table of contents. Naturally it shows the position of currently displayed material.

The learning materials have to be written by the author and will be compiled into web pages with more or less multimedia content. Their design and content should enable the reader to acquire the knowledge of the corresponding node and to answer the test exercises correctly. Vice versa, these exercises have to be designed to check exactly the newly obtained facts and to assure that the learner has understood the contents of the unit. These exercises fulfil two important purposes. Specifically, they are typical learning tasks with hints, explanations and informative feedback which help the learner to adopt knowledge by initiating and supporting active and intensive information processing [PKN04]. Generally, they serve as tests to check whether a node is mastered.

The learning progress is visualised by node colours in the diagram: if a node was mastered it is coloured blue. Nodes below this node provide prior knowledge and can therefore also be supposed as mastered and turn light blue. If the learner fails to achieve a certain amount of points when solving the test exercises, the node is considered as not mastered and turns red. All nodes whose knowledge depends on the knowledge of this node also turn light red. In addition all not mastered nodes, with completely attained prior knowledge are marked yellow – as suggestions of topics to learn next. So the knowledge state of learners is not represented by a position within the diagram (which would be the case, if it would be a diagram of the knowledge space), but a colour pattern of nodes. The learner has the opportunity to use the learning environment in different modes, which vary from full access to all nodes via restricted access (only blue or yellow nodes accessible, see above) to random tests and paths trough the whole knowledge domain.

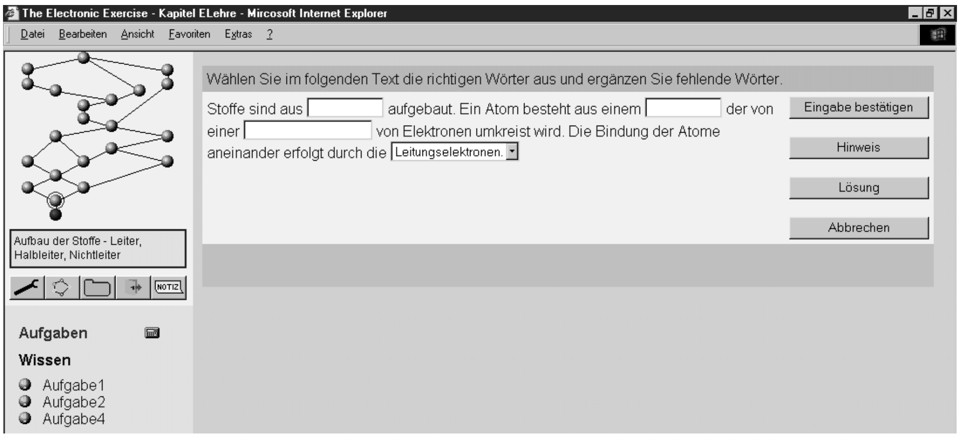

Figure 2: Screenshot of TEE with exercise

The most interesting mode is the intelligent tutorial mode that has extra benefits from the knowledge structure. The aim is to explore the knowledge of learners quite quickly and then to present adequate new learning material. During the exploration phase the learner has to solve a number of test exercises. So the program can find out which nodes the learner can already master and which he still has to learn. To shorten this investigation process to a bearable amount of exercises the surmise relation is used intensively. A half-split like algorithm decides which node has to be checked next to obtain a maximum amount of information, thus allowing the surmise of colour states for most of the further possible nodes. After the state of all nodes is checked or surmised the learn phase starts. The program displays sequentially all non-mastered nodes, always testing the learning progress and caring, whether the previous nodes are already mastered. This way the learner can acquire the complete knowledge domain without having to deal with reading material he is already familiar with.

Summarising, the graphical representation displays the structure of the knowledge domain to the learner, offers different ways to work with the material, allows the single handed choice of a learning sequence and logs the course and the success of dealing with the various knowledge elements.

Closely viewed, two mechanisms of learning act at two different levels in TEE: learning tasks offer informative feedback [NH04] to support their complete solution and the acquiring of the knowledge of the assigned node. If these feedback procedures cannot guide to the aimed (local) learning goal, the learner needs to attain the related pre-knowledge. This is managed by an outer control loop onto the nodes itself which helps to achieve the general learning goal. The learning task loop can handle many slips and errors and the learner stays inside a node until he is "catapulted" into another segment of knowledge structure. Therefore the learning within TEE is much more thematically coherent in comparison to other learning management systems which only provide one loop for controlling the learning process. So the course of learning may be build up individually from very different learning steps, which is not the case within the above mentioned learning management systems with a set of pre-structured learning sequences considered to be optimal which have to be defined by the authors. Additionally, these two feedback loops help the authors of learning applications, because the planning and implementation of the sequencing can be omitted respectively it only has to be carried out for the structure of the knowledge domain.

3 Technical Implementation

TEE is implemented by dynamic HTML using JavaScript intensively. It uses internet browser features that have been available since the 4th browser generation. It sticks completely to client side dynamic and can therefore be used offline (i.e. on CD). Despite this it is possible to store the personal performance data on the computer (cookie) on the internet (on a server) or as a file (on disk).

4 Author's View

The first step for an author is to divide the knowledge domain into smaller units (the nodes) and order them according to the surmise relation. The implicit way to obtain this structure is to put questions to experts or teachers. In one of our TEE applications we used a reduced structure about fraction calculus which was acquired by such an interrogation process [Do93],[BD97]. Unfortunately, this method of determining a knowledge structure is tedious and time consuming. The explicit way is to build up the structure manually and to support the choice of relationships by arguments [AL99]. This requires some experience, but as TEE suggests taking fewer nodes, this can be done with little effort and after a few attempts the final structure is usually drawn. The next step is to create or collect the learning materials and to combine them into small tutorials each covering a node. This can be done with any website editor/generator. Furthermore adequate test exercises have to be designed. They have to be typed into the EF-Editor [PKN05], an exercise editor which compiles the exercises into HTML files that can be used by TEE. Using this editor most of the exercises can be generated without programming knowledge. The author only has to specify the main contents of an exercise and can combine various input formats (multiple choice, free text, drag and drop etc.). Moreover hints and specific feedback can be added.

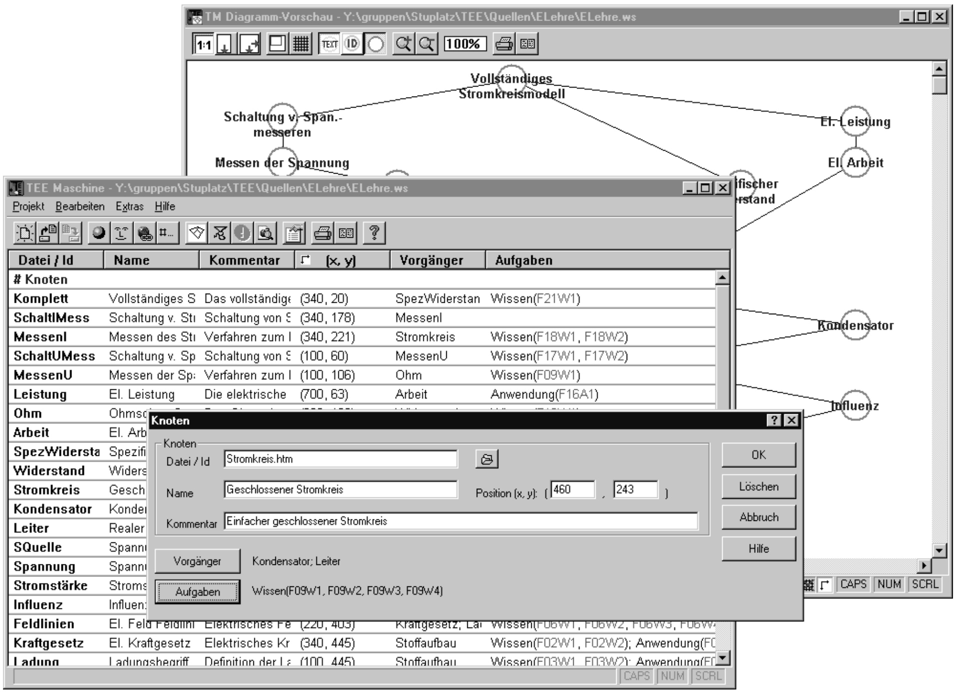

Figure 3: Screenshot of the TEE-Maschine

The final step is to join all these parts by the program TEE-Maschine. A diagram of the surmise relation has to be drawn; the learning material has to be inserted along with the exercises and the conditions when a node is considered to be mastered. If there are more exercises specified than actually needed to decide whether a node is mastered TEE will always display only a random selection of exercises. This means that the same exercise will not be repeated by the learner if he fails to master a node at the first attempt. Finally the TEE-Maschine compiles everything into a complete TEE environment.

5 Outlook

Recently TEE has been used randomly and in its experimental state. There have been projects in schools where the pupils built TEE systems within a school project. They could also choose to develop a knowledge domain, which would accompany the lessons for several weeks, with every pupil having to design one node (for further information see http://studierplatz2000.tu-dresden.de/toolkit). TEE was also used at TU Dresden in multimedia lessons for teacher students. The majority of these people said they learned most by the processes of thinking about the structuring of the knowledge domain and constructing the exercises. Therefore there is no real experience how a novice learner views the system, except one study with adult learners, which generally welcomed this unfamiliar representation of learning material. Unfortunately, most authors didn't structure their knowledge domains according to a surmise relation, but by using the diagram in the intuitive way of a mind map. The tools TEE-Maschine and EF-Editor are freely obtainable for any non-commercial use (http://studierplatz2000.tu-dresden.de).

6 References

[AL99] Albert, D.; Lukas, J.: Knowledge Spaces - Theories, Empirical Research, and Applications. Mahwah, New Jersey, Lawrence Erlbaum Associates, 1999.

[AS99] Albert, D.; Schrepp, M.: Structure and Design of an Intelligent Tutorial System Based on Skill Assignments. In (Albert, D.; Lukas, J. Eds.): Knowledge Spaces - Theories, Empirical Research, and Applications, Mahwah, New Jersey, Lawrence Erlbaum Associates, 1999; pp. 179-196.

[BD97] Baumunk, K.; Dowling, C.E.: Validity Spaces for Assessing Knowledge about Fractions. In: Journal of Mathematical Psychology, 41, 1997; pp. 99-105.

[Do93] Dowling, C.E.: On the irredundant generation of knowledge spaces. In: Journal of Mathematical Psychology, 37, 1993; pp. 49-62.

[NH04] Narciss, S.; Huth, K.: How to design informative tutoring feedback for multimedia learning. In (Niegeman, H.M.; Brünken, R.; Leutner, D. Eds.): Instructional Design for Multimedia Learning, Münster, Waxmann, 2004; pp. 181-195.

[PKN04] Proske, A., Körndle, H. & Narciss, S.: The Exercise Format Editor - A multimedia tool for the design of multiple learning tasks. In (Niegemann, H.; Brünken, R.; Leutner, D. Eds.): Instructional design for multimedia learning, Münster, Waxmann, 2004; pp. 149-164.

[PKN05] Proske, A., Körndle, H. & Narciss, S.: The Exercise Format-Editor – Supporting the Systematic Construction of Interactive Learning Tasks. In this volume.