

EDUCATIONAL DESIGNER

JOURNAL OF THE INTERNATIONAL SOCIETY FOR DESIGN AND DEVELOPMENT IN EDUCATION



A designer speaks: Peter Boon

*Freudenthal Institute for Science and Mathematics
Education
University of Utrecht
The Netherlands*

Designing Educational Software for 3D Geometry

Abstract

This paper describes the design process of a collection of educational Java applets for fostering spatial abilities in primary and secondary education.

It describes the process in which I tried to exploit the unique possibilities of software in realizing learning activities with a clear add-on value in reaching the learning objectives.

I will not only use the viewpoint of an educational designer who is inspired by the possibilities of software, but also that of a software designer and programmer who is inspired by educational needs and ideas to modify existing software features or create new ones.

Introduction

There are several different approaches for the design of technology-rich educational materials. An educational designer can use existing software for activities that are additional to paper lesson materials. In that case the software is used as a ready-made resource as part of the educational design. In the design process I describe in this article, the software was not ready-made, but part of the process of testing, change and improvement that is typical for educational design.

The theoretical foundation of our educational software design is mainly based upon the theory of Realistic Mathematics Education (RME). The main idea of RME is to create opportunities for students to come to regard the knowledge they acquire as their own knowledge. Therefore, contextual problems which students recognise as relevant and real, and which also evoke productive solution strategies, play an important role. We designed a large collection of applets (small, single-purpose computer programs) which support learning that fits in this approach. In this design work we discovered and articulated how applets can be helpful (cf. Boon, 2006)

When designing software together with other educational materials, one should keep in mind that building software from scratch for every single educational application is not very realistic (especially in the case of three-dimensional (3D) graphics software). It is important (1) to use resources that are already available; (2) to keep in mind other educational applications for the designed software; and (3) make it flexible and customizable.

In the design projects described, I chose to build on previous work from an earlier project ([figure 1](#)), where I had already developed a lightweight 3D library to show and manipulate 3D objects and nets as interactive illustrations on the screen.

This pre-existing software enabled me to design the 3D applets discussed in this article in a reasonable time frame. In fact, the software not only formed the initial programming resources, but also the source of inspiration for the educational design described here.

Figure 1: 3D applets from Freudenthal Institute



The software design served three projects, which increased the available resources (time, money), though this also made demands on the architecture of the software. The different parts had to be easily adaptable and reusable, not only to make them fit the requirements for the different projects, but also to provide the flexibility that is vital in the process of testing and (re)design.

Finally, this article is not merely a description of the process and the design principles used, but is also an attempt to discover the structure of a process that was in fact quite chaotic, and to discover and articulate the design principles that were, at least partly, used subconsciously during the process.

The context of the design

The software design addressed the subjects Geometry and Measurement, and took place in three different projects:

- The [WINST project](#), a project for vocational education (students of 12-16 years) (WINST is a Dutch abbreviation of “Mathematics in Science and Technology”).
- The TAL project in which learning-teaching trajectories were developed to offer a global pathway of the development of mathematical understanding in students in grades K to 6. ([Heuvel-Panhuizen, M. van den & Buijs, K., 2005](#)).
- The [WisWeb project](#) for secondary education, for developing and testing web-based activities (applets) for a range of subjects in close cooperation with teachers.

In these projects prototypical lesson material on Measurement and 3D Geometry was developed and tested. Although the students we aimed at were quite different in age and level, a common obstacle for them was their weak ability to make the connection between 3D objects and 2D representations of these objects. We expected that interactive representations of 3D objects on the 2D screen could work as a bridge from the real 3D world to the static and more abstract 2D representations on paper and back.

Programming these virtual 3D objects that could be manipulated with the mouse was just a first step. Designing tasks that use these resources in a productive way was another part of the process and was far from trivial. It would have been possible to present these tasks on paper and use the software as additional tools or interactive illustrations, but we wanted to exploit the medium more extensively, and tried to define tasks within the software to create game-like features. Beside this, we wanted to design more advanced interactions like 3D construction facilities to support richer (and more open-ended) tasks.

The process of the design

In our educational design for the vocational students, we focussed on the subject “Technical Drawing”, which is an application of the mathematical subject Geometry and Measurement. The educational objectives were to develop (1) the ability to interpret the 2 dimensional projections and construct the 3D object; and (2) the ability to draw the 2D projections of given 3D objects. [Figure 2](#) is an example of such a drawing.

[Figure 3](#) represents a (paper and pencil) task for students that is meant to support this learning goal. The student work that was observed in the project showed that gaining insight into the meaning of the different representations was problematic.

Figure 2: Technical drawing

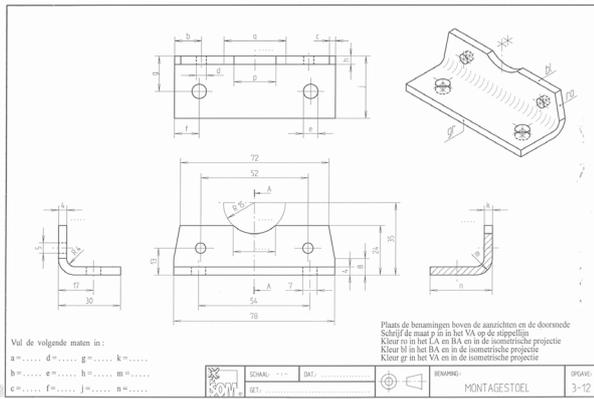
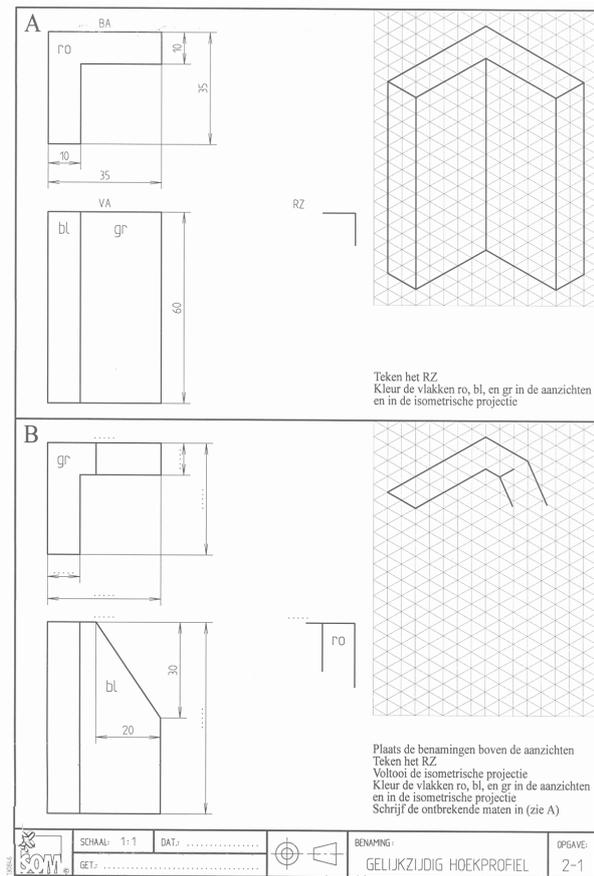


Figure 3: Technical drawing student exercise

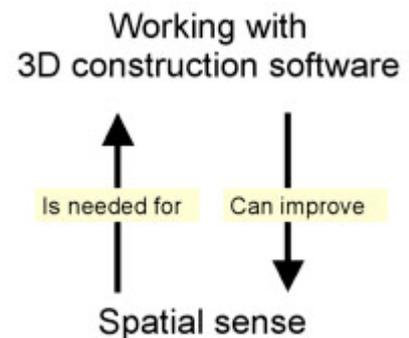


To improve this situation we made use of the features of the applet “Drawing Cube” (Applet 1). In this applet a ‘3D’ cube is shown on the screen. The cube can be rotated with the mouse and lines and planes can be constructed by clicking on the vertices. The idea was that quick constructing facilities in a (virtual) 3D environment, combined with the possibility to rotate the object to get the different views, could establish a firmer connection between the different representations. To make a stronger link to the format of the technical drawings, we adapted the software and designed a feature to toggle between the technical drawing and the 3D object. With this prototype we worked with students.

Note: Most of the applets discussed here can be run live from the online version of this article at: www.educationaldesigner.org/ed/volume1/issue2/article7

Several problems occurred:

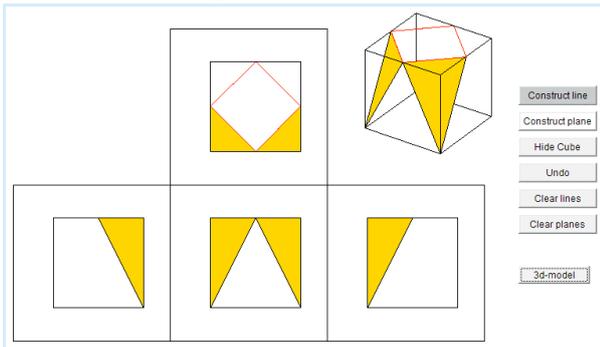
- Constructing in the virtual 3D model appeared to be quite difficult for the students. Although we thought the constructing environment to be quite user-friendly, apparently quite some spatial insight was needed to be able to operate the environment.
- The objects that can be constructed with “Drawing Cube” are not very ‘real’ for the students. The objects, cubes with lines and planes, are more math-like and there is little connection with the common technical subjects that were used in the other learning materials.



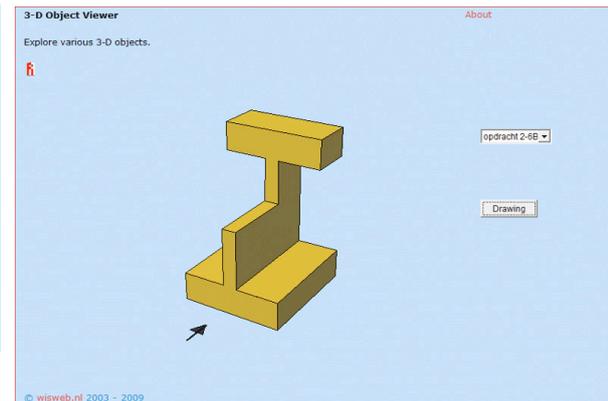
We decided to go back to the objects that were used in the paper materials and made virtual 3D versions of those objects that could be manipulated on the screen, with a toggle to the 2D technical drawing. We left out the construction facilities that appeared to be too difficult for the students.

This led to the second prototype: “Profile Viewer” ([Applet 2](#)). This applet was used as an interactive illustration within the paper tasks. We observed the students working with this prototype and investigated the problems they met. This gave us ideas for the design of new useful tasks.

Applet 1: Drawing Cube



Applet 2: Profile Viewer



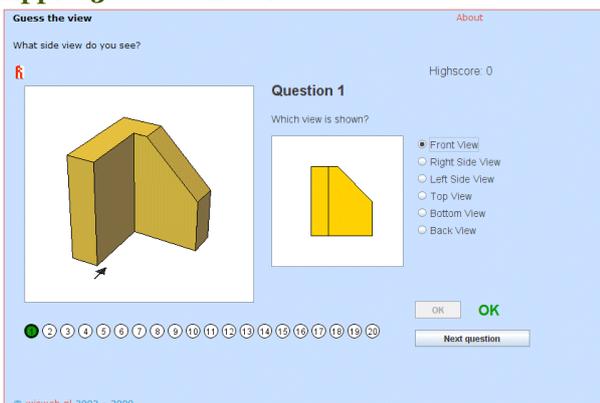
The next step was the applet “Guess the View” ([Applet 3](#)) in which both representations were shown on one screen instead of separate ones, to make the connection between the representations easier. Moreover, a task in which the student had to determine a sequence of different views was designed within the applet. We implemented this task as a game in which scores can be achieved and feedback is given.

The virtual 3D representation could help to find the requested view by rotating the object with the mouse.

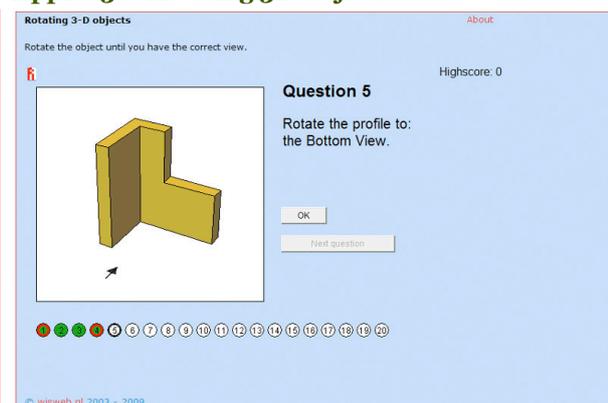
Among the students we noticed many differences in the abilities to play this ‘game’. For a group of weak students who had difficulties with the task, we designed the applet-game “Rotating 3D-Objects” ([Applet 3b](#)) that gave extra support and practice in finding the different views by applying the proper rotation in the 3D model.

For this group, this turned out to be a suitable preparation for “Guess the View”.

Applet 3: Guess the View



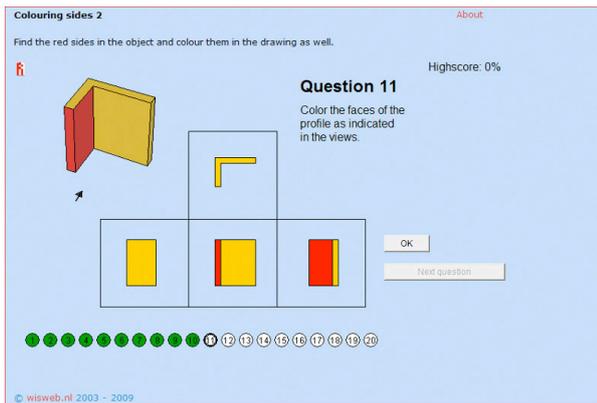
Applet 3b: Rotating 3D Objects



In a third game “Colouring Sides” ([Applet 4](#)) we extended the activity to a more sophisticated level. Students had to connect corresponding parts in the two representations. This applet also provided more of a challenge for the better students in this group.

The three ‘games’: “Rotating 3D-objects”, “Guess the view” and “Colouring Sides” turned out to be a successful addition to the subject “Technical drawing” in vocational education, and were also appreciated in primary as well as secondary mathematics education. They became popular applets on the WisWeb website.

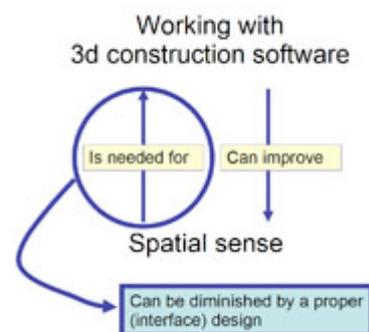
Applet 4: Colouring Sides



Extending the design to richer interactions

In the design process described above we initially wanted the software to support richer interaction by allowing students to make their own constructions within the 3D and 2D representations.

The initial construction tool (“Drawing Cube”) that we made for the group of vocational students, turned out to be too difficult to operate with. We knew we had to design a tool with a user interface that did not need knowledge of advanced spatial concepts (like crossing and intersecting lines or determination of planes by three vertices), but could use more basic ones. This challenge was fed by requirements from the TAL project where designers wanted a tool that could be used by young children (from the age of about 5 to 6).



We started the design with observing young children playing with the applets that had been designed for vocational education. We customized “Rotating 3D Objects” and changed the objects to so-called ‘block buildings’, with which they were more familiar (this kind of object was used in the curriculum as well). Instead of the names of the views, we used silhouettes of these views and the task was to rotate the block building so that the view that was asked for would become visible.

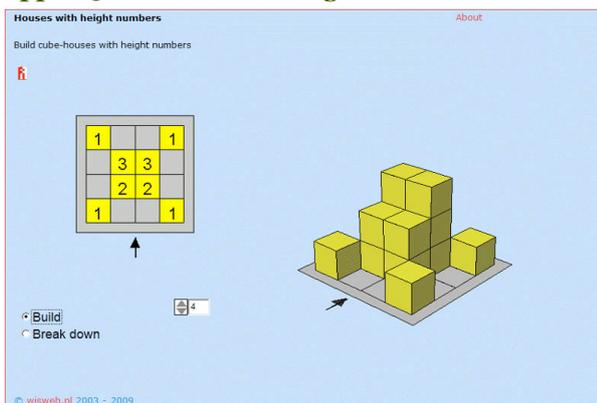
Alongside this game we worked in a more open way and asked children questions about the objects on the screen and let them discover properties by using the ‘drag to rotate’ feature.

In the children's play and reasoning we observed that the objects on the screen did not always lead to a proper 3D-experience of the object, but that in most cases this could be 'called up' by using the 'drag to rotate' feature. This observation had important implications for the interface of the 3D construction tool we wanted to design. We had to offer an environment in which a proper 3D experience of the representations on the screen is not a precondition for the constructing activity, but an ability that can grow and develop during the constructing activity.

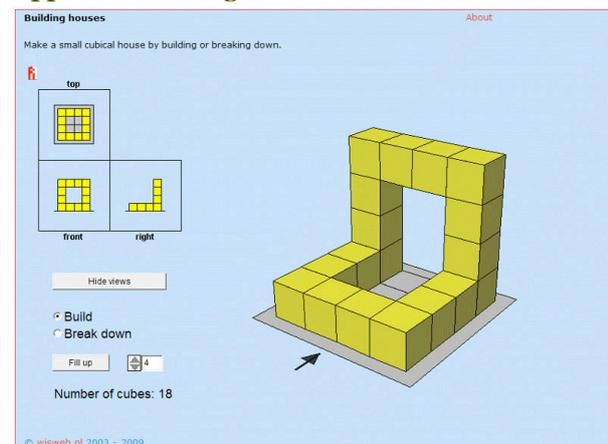
A first prototype was "Houses with Height Numbers" ([Applet 5](#)) in which block buildings could be constructed. Clicking in the top view representation results in adding new blocks in the 3D-model. This worked in a way, but the two separate representations were a bit confusing, although in special tasks this confusion could be used to create powerful learning experiences.

The applet "Building Houses" ([Applet 6](#)) was a break-through. We achieved an interface that met the requirements in quite a satisfactory way. Cubes can be added or removed just by clicking in the 3D model on the grid or on the cubes that are already in place. The result is immediately visible and with this feedback, together with the drag-to-rotate feature, the students could really explore 3D space on the screen, and even young children showed they were able to 'build' their own 3D experiences using the computer.

Applet 5: Houses with Height Numbers



Applet 6: Building Houses

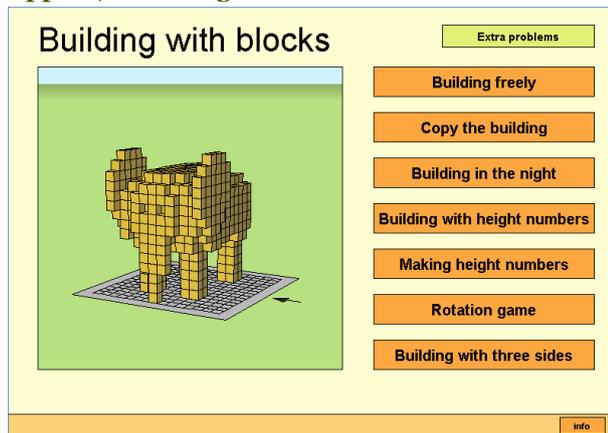


The next step was designing tasks for the students. We did not want to prevent students (and teachers) to freely use the tool for their own ideas and constructions, but some clear tasks presented in a game-like way appeared to be very attractive for students. Moreover, as a learning resource on the Web, the tasks presented are a means to communicate the ideas and possibilities to a bigger audience beyond the projected educational context.

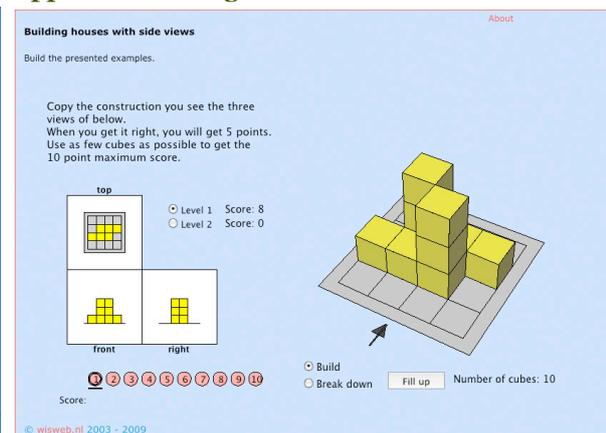
The final product for the TAL project is the applet “Building with Blocks” ([Applet 7](#)). This applet is in fact a compilation of tasks using the same tool, customized in different ways. Most of these tasks emerged during the design process described above. Some of the activities, for example “Copy the building” can be checked by the computer and can be played as a game. Other activities, like “Extra Problems” are more open. And of course the construction tool (“Building Freely”) is included for constructing your own buildings.

With the software being implemented as a package of customizable components, it was not too difficult to adapt it, and to implement challenging tasks for students in secondary education. The applet “Building Houses with Side Views” ([Applet 8](#)) is a nice example of a challenging game on a higher level.

Applet 7: Building with Blocks



Applet 8: Building Houses with Side Views



To a more mathematical interface for making 3D-constructions

The last design in this software collection is “Building Programs”. It is the same block building tool as in the previous examples, but extended with an entirely different interface for making constructions. The user interaction is a kind of programming that makes use of the 3D coordinate system, as well as other mathematical concepts like intervals and variables.

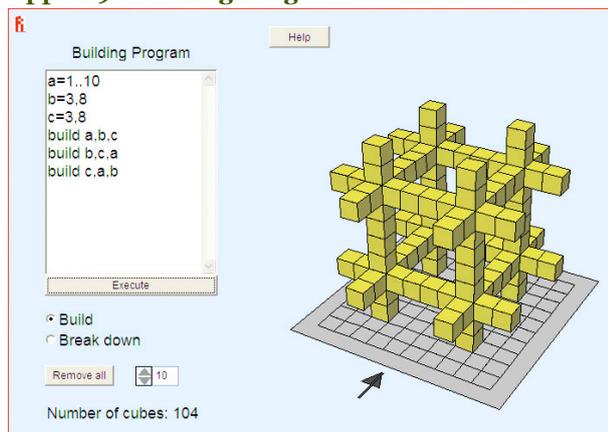
Programming activities within mathematics education already have a long tradition, mainly associated with “turtle graphics” using a subset of the Logo programming language, but also with other accessible programming languages like Basic. In the applet “Building Programs,” a very different approach was tried out, based on a paradigm known in computer science as “functional programming”. In functional programming, concepts such as functions and variables have a much closer correspondence to their mathematical namesakes than they do in more common “imperative” or “procedural” approach used in, say BASIC. In “Building Programs”, I tried to present the essence of functional programming is an accessible way, without the need to learn a lot of technical information.

Compared with the intuitive way of ‘building by clicking’ in the other applets, the more

formal language that is used here makes the interaction more advanced. On the other hand, once the user has mastered this language and the concepts used, it provides a powerful and efficient way of building. I think this is a metaphor for mathematics in general. Difficult to master, but once mastered it offers you a tool for solving problems that are hardly solvable without.

What I like in this design, is that it shows the power of mathematical concepts like ‘variable’ in a very direct way. In the standard math curriculum it often appears to be very difficult to show the power of mathematics to students. As teachers we make nice promises for the future to justify the hard labour needed to master algebra, but I doubt many students will really reach this “promised land.” Perhaps this kind of IT design can be elaborated on further to improve this situation.

Applet 9: Building Programs



Reflections on educational software design and new developments

In projects for educational design where software design is involved, we have to deal with the specific problems and characteristics of software development. Educational design processes are highly iterative and often not very structured. Since software development is time consuming and therefore quite expensive, it is reasonable to expect that design processes in which software design is involved are better structured and planned. Careful and detailed specification of the software to be designed seems inevitable. On the other hand, creativity is an important aspect of successful educational design; we are not just implementing ready-made ideas in a determined way, but we want to allow the initial ideas to grow and mature in a process of prototyping, testing and improving. In the examples in this article we even saw ideas jump from one project to the other. So, even when software design is involved, the processes should be kept flexible.

Several approaches can help to overcome this dilemma:

- Use existing resources to build on (that are flexible enough to make the necessary adaptations).
- If possible, use higher level programming or prototyping tools that already provide much of the elementary material (e.g. a programming tool like Flash already has support for many of the graphical necessities and animations).
- Try not to focus exclusively on one educational design, but keep in mind other possible educational applications for your product.
- Make the software configurable (to be able to meet other possible educational

applications).

- Design the software as a collection of reusable components and packages.

An important issue is creativity on different levels. The interface of “Building Houses” that supports construction activities in such a user friendly way, could not have been designed without creative input from the technical programming level. In many software design contexts, the work is split up so it can be carried out by different specialists. The programmer gets the specifications from functional designers via technical designers and is supposed to just implement this. In our educational software design projects, interdisciplinarity is more common than specialism.

It turns out to be a highly efficient and effective way of working if the technical people have knowledge of the educational goals and principles. During the programming a slight modification of the requirements can have major consequences for the code and thus for the time needed for the implementation. On the other hand, a programmer can, from his point of view, also bring in new possibilities. Understanding the goals, combined with a creative attitude, the programmer can have a crucial role in improving not only the efficiency of the design process, but also the educational design itself.

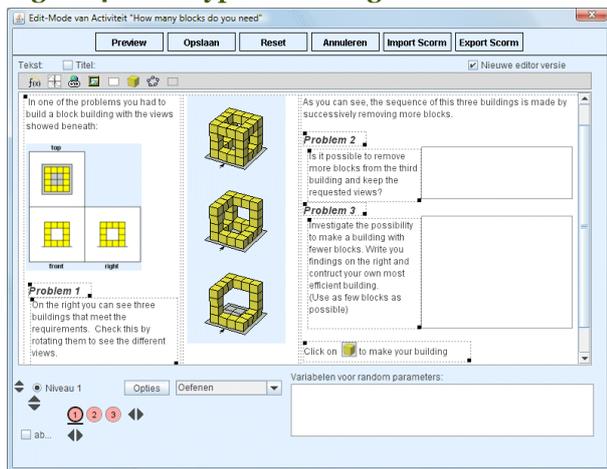
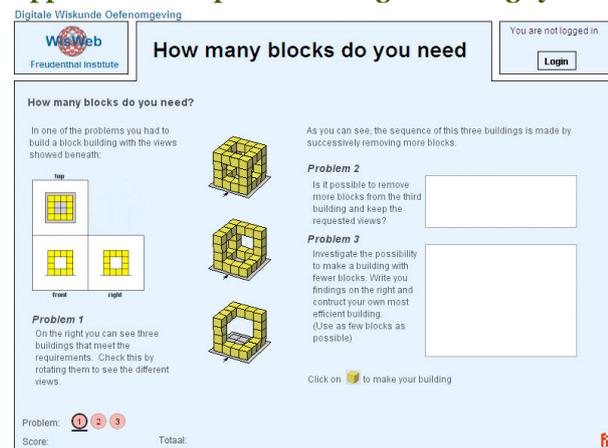
Another promising way of supporting flexibility in the design of technology rich materials is the design of authoring environments. Recently I have been working more and more in this direction.

It is a way of making a new division of labour between programmers and education designers in the development of educational software. Programmers can focus more on the design of high quality interactive components and take the time that is needed for this, while designers without programming skills can combine and customize these components and design interactive tasks more easily. Especially in a context where digital learning environments are becoming mature as a new medium for learning and working, these authoring tools are becoming important.

[Figure 4](#) shows a screen of a prototype of an authoring tool I designed. It works like a text editor. Besides text, many types of interactive components can be used. These interactive components are based on (existing) applets and, in this environment, can be easily customized to fit the designed tasks. The ‘document’ that is made in this way can be exported as either a new applet or as a so called SCORM-package that can be used in a digital learning environment. (SCORM is a standard format for digital learning materials.) By embedding these digital activities in a learning environment, it becomes possible to store students’ results and even make them visible for teachers.

[Applet 10](#) shows a task built using the authoring tool.

This new approach in educational software design opens new ways for integrating tools and tasks and helps to make the new medium evolve, not only to a flexible environment for educational designers, but also to a more adequate mathematical working environment for students.

Figure 4: Prototype Authoring Tool**Applet 10: Example built using authoring system**

References

Boon, P. (2006). *Designing Didactical Tools And Microworlds For Mathematics Educations*. In: Hoyles, C., Lagrange, J.-B., Son, L. H., Sinclair, N., *Proceedings of the 17th ICMI Study Conference*. Retrieved from http://www.fi.uu.nl/isdde/documents/software_boon.pdf on April 21, 2009.

Heuvel-Panhuizen, M. van den & Buijs, K. (2005). *Young children learn measurement and geometry. A learning-teaching trajectory with intermediate attainment targets for the lower grades in primary school*. Utrecht: Freudenthal Institute, Utrecht University / SLO, 356 pp. Retrieved from http://www.fi.uu.nl/publicaties/nieuws/tal_en/welcome.html on April 21, 2009.

WINST-project (website). Freudenthal Institute, Utrecht University. Retrieved from <http://www.fi.uu.nl/winst/> on April 21, 2009. (Dutch)

WISWEB (website). Freudenthal Institute, Utrecht University. Retrieved from <http://www.fi.uu.nl/wisweb/> (Dutch) <http://www.fi.uu.nl/wisweb/en/> (English) on April 21, 2009.

About the Author

Peter Boon is a senior java programming specialist and mathematics curriculum developer and researcher. He combines his technology expertise and his mathematics teacher background for developing rich learning content that can be easily used by students and teachers. A wide range of applets have been programmed for algebra and geometry as well as a variety of basic skills. Other areas of his interest are computer science education, learning content management systems and the development of authoring tools for teachers and education designers for creating digital mathematical learning materials.

He is a member of the ISDDE Executive.