Applets for learning school algebra and calculus
Experiences from secondary school practice with an integrated learning environment for mathematics

André Heck¹, Peter Boon², Christian Bokhove³ and Gerard Koolstra³

¹ Universiteit van Amsterdam, AMSTEL Institute, heck@science.uva.nl
² Universiteit Utrecht, Freudenthal Institute, p.boon@fi.uu.nl
³ St. Michaël College, {cbokhove,gkoolstra}@stmichaelcollege.nl

Abstract. In recent years, the use of ICT tools for learning of mathematics received a lot of attention. New computer programs include digital chapters for existing schoolbooks, Java applets, dynamic software for geometry and statistics, computer algebra systems, computer aided testing and assessment systems, and many more. In this paper we will focus on the development and use of Java applets for learning and practising algebra and calculus at secondary school level. We have explored in what ways applets can help students to develop algebraic understanding and algebraic skills. Especially applets that provide feedback with scoring facilities and that are fully integrated in a learning environment with which students and teachers are familiar have great potential. Of course, teachers should be able to create exercises and tests based on their own ideas: the customizable SCORM compliant Java applet generator makes this possible. We will discuss our experiences from development work and classroom tests during the GALOIS project.

1 The GALOIS project

Since December 2004 the GALOIS project aims to realize an integrated learning environment for learning and doing mathematics in which

– students can practise mathematics anytime and anywhere;
– mathematical assignments can be generated ‘at random’, so that the amount of exercise material becomes unlimited;
– activities and answers of students are stored;
– ‘intelligent’ feedback on students’ work is given automatically.

We wish to reach these goals by using open source technologies and open standards. We try to translate and extend existing technologies into concepts that work in the practice of school mathematics. We also make ICT-rich learning materials and try them out in the classroom. In this paper we describe our classroom experiences with exercise material at secondary school level. We discuss the following topics in the given order:

– the design and use of mathematical Java applets from the Freudenthal Institute, called WisWeb applets [1],
– the use of the ‘Digitale Wiskunde Oefenomgeving’ [Digital Mathematical Exercise Environment] developed at the Freudenthal Institute [2];
– the use of the SCORM standard [3] to make teaching and learning with WisWeb applets fully operational in an existing virtual learning environment, in this case the open source system Moodle [4];
– intelligent feedback in the JavaScript-based testing and assignment environment DITwis [5], which is also integrated in an existing VLE, in this case Moodle again, via the SCORM standard;
– the design and use of a customizable SCORM compliant applet generator for learning school algebra and calculus.

By the way, GALOIS is a Dutch acronym for Geïntegreerde Algebraïsche Leer-Omgeving In School [Integrated Algebraic Learning Environment in School]. The Dutch Ministry of Education, Culture and Science financially supports this innovation project in the Vooruit! framework until 2007. For more information we refer to the website www.stmichaelcollege.nl/galois/index.php?lang=en.

2 Design and use of WisWeb applets

In the framework of the projects WisWeb and Welp [1] a large number of Java applets for learning mathematics have been developed by a team of researchers and curriculum developers at the Freudenthal Institute in collaboration with teachers at secondary schools. The last three authors of this paper were involved from early days, tried out prototypes of these little computer programs in the classroom, and improved the software and accompanying instructional materials. Knowledge on how to use the software to enrich the learning was built up in this way. The use of applets for learning mathematics was expected to contribute to visualization of mathematical concepts, exploration of context situations (including purely mathematical contexts), and to bridging the gap between informal knowledge and formal mathematics. The Realistic Mathematics Education (RME [6]) approach was the key theoretical framework for the design of the software and of the instructional material, and for the flanking educational research. Considerable effort was put into the design and field testing of applets for students to develop algebraic understanding and skills, because this topic is for ages an important topic in mathematics education, but at the same time it is an obstacle for many a student. This kind of applets were used and further developed in the GALOIS project.

Two kinds of applets can be distinguished: model applets, which help to develop mathematical understanding (concepts), and exercise applets, which support the development of mathematical skills. In model applets visual representations are used as models for mathematical concepts. Within these representations students can work on the basis of their own ideas and experiment freely. An example of a model applet is the ‘Geometric Algebra 2D’ applet shown in Figure 1. It allows adding and multiplying numbers and variables by combining line segments to new line segments and rectangles. The lengths of the line segments and
the areas of the rectangles are displayed as expressions. By splitting and joining rectangles students can develop insight in equivalence of various expressions.

![Fig. 1. The ‘Geometric Algebra 2D’ applet.](image)

A model applet such as the ‘Geometric Algebra 2D’ applet has an open nature; it allows students to work with it on the basis of their own constructions and productions. The ICT tool is only expected to help students to mathematise the situation on the basis of their activities. Activity sheets may guide students in making progress from informal to more formal mathematical activity. We refer to [7, 8] for further reading about the concept of the emergent-modeling design heuristic, in which the objective is to help students model their own informal mathematics activity and gradually develop this model into a model for more formal mathematical reasoning.

In the second class of Java applets that have been developed, the exercise applets, this open structure is replaced by a more closed structure. In exercise applets, a number of tasks of certain type are given to the students, their work is checked and feedback is provided. Figure 2 illustrates the ‘Area Algebra’ applet, in which the area model is used to verify equivalence of algebraic expressions. The main task is to find the correct expressions for areas and edges of rectangles. The applet is an interactive version of the well-known geometrical representation of the distributive law of algebra. A formula like $3(y + 2)$ is treated as the area of a rectangle with length 3 and width $y + 2$. This results in a stepwise formula with a part equal to $3y$ and a part equal to 6. Students intuitively learn and see that $3(y + 2)$ is equivalent to $3y + 6$. Of course, this model has some limitations when it comes to negative numbers, but then it can easily be replaced with a simple table. Feedback in this applet is direct: green symbols indicate which part of the answer is correct, and red, orange and green circles around the assignment.
numbers indicate wrong, incomplete and correct answers. This stimulates the students to go on and try another one, and to get as many correct answers as possible. Also very important, the stored feedback provides both the student and the teacher detailed information about where and what kind of mistakes have been made. The screen shot shows an example: maybe the student had simply forgotten to fill out a text box in the rectangle; maybe he or she had worked in a too formal manner and written 32 with $3 \cdot 2$ in mind, just as one writes $3y$ when the product is actually meant.

![Area Algebra Applet](image)

**Fig. 2.** The ‘Area Algebra’ applet with immediate feedback.

The method of completing squares for solving quadratic equations links up well with the above area model. It is supported in the exercise applet ‘Quadratic Equations’ shown in Figure 3. The applet presents a quadratic equation that is to be solved step by step. With the red buttons on top, the student can select an operation that is to be performed on both sides of the equation. The goal is to simplify the equation and eventually end with $x = \text{solutions}$. With this applet, a student has to think and decide what operation to perform. Two versions of the applets are available: one in which the computer performs the calculation at each step, thus allowing the student to concentrate on the solution strategy, and one in which the student has to do the calculation himself and the program only verifies whether intermediate expressions are still equivalent. A ‘gaming’ feature can be added to the second version by letting students earn points for entering correct intermediate results. Using help or letting the computer perform an intermediate step costs points. The idea is that changing a set of mathematical exercises into a game with various levels motivates students to work hard.
As was noted before, the Java applets have been published on the WisWeb-site (www.wisweb.nl) and have been field tested extensively at secondary schools in collaboration with teachers and educational researchers [9–11]. We quote some findings on the use of school algebra applets [11]:

“It was found that the use of applets has add-on value: they are fun and motivate students; they allow students to work at their own level of thinking and thus better address individual difference between students; the visual, interactive and dynamic features of applets makes the mathematics more easy to understand; thanks to the calculation power of the applets one can focus on the mathematical concepts and models; students are more creative and get more self esteem; the applets form a model students can fall back on; the practice and feedback features are much more powerful than pencil-and-paper exercises. However, to take fully advantage of these opportunities of applets, applets should be integrated in the daily mathematics class routine. Using the computer should not be a voluntary thing that is extra beyond regular math class.”

The last issue played an important role in the GALOIS project. The successes with mathematical applets and other ICT tools actually lead to higher expectations. For a teacher it would be nice if he or she could review after a computer aided lesson what the students actually did, what progress they made, which problems arose during learning and to which mathematical subjects attention must be paid in the next lessons. Also, if assignments were given outside of class, this mechanism of ‘student tracking’ would aid a teacher in determining student ability. An even more basic advantage would be the possibility of checking whether a student has actually done his or her homework. At secondary school level — with students at age between 12 and 18 years — this added bonus is often overlooked.

All this would enable teachers to use their actual ‘contact time’ more efficiently. On the other hand, students also benefit from monitoring student performance: a digital archive of computer based activities gives these activities a
firm place in the curriculum and improves the learning process while it enables students to review mistakes made and it stores feedback. This systematic approach within one learning environment enhances the already positive effects of direct, instantaneous feedback.

Many Virtual Learning Environments (VLE) store student work, and it seems that the use of VLEs in secondary education in the Netherlands is on the rise. Nonetheless, with these environments, schools still have to add their own educational tools and lecture notes. Reuse of ICT-components and exchange of learning materials with colleagues without conversion problems promotes this. One could even argue that a VLE only deserves its name when one can actually learn within the system, which implies that an integrated learning environment actually has content. A logical next step in creating an integrated learning environment for mathematics is to see whether mathematics programs can communicate with standard VLEs to store user feedback and aid sharing of content. For this we used in the GALOIS project the SCORM standard.

3 The DWO math environment

The DWO math environment [2] is a web-based application in which a student logs in on a central server, works with an applet, and automatically receives feedback from this applet. Student results are stored so that students can stop working in the DWO and return later, without losing any results. Teachers can monitor the progress of individual students, but also of entire classes (Figures 4 and 5 show examples).

At the secondary school St. Michaël College, the DWO math environment was used for example in the second grade (12 to 14 year olds) in combination with written material, replacing two chapters on algebra. This first learning unit gave students a first acquaintance with mathematical expressions and above all the important mathematical concept of ‘variable’. It also aimed to instil some ‘symbol sense’ and the ability to rewrite mathematical formulas. One of the applets used was the ‘Area Algebra’, which we described in the previous section (Figure 2). Students were given the opportunity to go through the instructional materials at their own pace, not only at school but also at home. Because the DWO math environment stores the results and feedback of the exercises of all students, the teachers could not only see which assignments students answered incorrectly, but also watch the mistakes in detail.

As a matter of fact, just the knowledge that most pupils did not have any problems with certain subjects was valuable information in practice, too: the teachers could pay more attention to the things the students did not know or had not mastered well enough, instead of teaching about things the students already knew. Of course, in the classes there were differences in the progress that students made because they had different levels of ‘talent’ for the subject. Individual results from students in one class (Figure 4) and in other classes (Figure 5) helped with tracking student performance individually as well as globally, and at a more detailed level than in traditional classroom instruction.
To correctly interpret the results shown in Figures 4 and 5, it is important to know that in this case teachers had given the students homework to make the first two parts of the applet, resulting in the highest attainable score of 42%. In this case five out of seven classes score higher than the expected percentage. Upon even closer inspection 25% of all students correctly made more than two thirds of all assignments, and 5% of all students even scored 100%. This is a clear sign that the applets highly motivated the students, even beyond the requested activity level.

Fig. 4. The results of all students in one class using the ‘Area Algebra’ Applet.

Fig. 5. The results of all 2nd grade classes using the ‘Area Algebra’ Applet.
In summary, use of the DWO helps the teacher to get good insight into the learning processes of pupils and to make grounded choices in the construction of his or her lessons. With the possibilities of such an interactive environment the polarity between classroom and individual teaching has to be reconsidered. Teaching can shift from a classroom approach, with instruction and discussion, to a more individualistic approach with custom feedback. To students and teachers alike, the DWO is seen as a clear and intuitive environment with not too many options and features. Administration of the DWO is in the hands of an external organization (in this case the Freudenthal Institute) and teachers do not have to concern themselves with the technical side of things.

It also is possible to construct new assignments within a couple of Java applets. This feature aids a teacher in making custom made assignments for every school level within a given subject. To evaluate the performance of students it is also possible to give an exam within the DWO environment. A teacher makes a new module that can be accessed during a limited time span. In this module buttons for feedback and marking have been removed, so now a student has to do everything himself. A teacher does have access to the marking button to facilitate the marking of these assignments.

There are some disadvantages, too. When a school already uses a VLE, with all kinds of communication facilities such as email and bulletin board available, it could be considered cumbersome that yet another environment is added to the already long list. Especially when more and more persons start learning and teaching with the aid of computers, the necessity of an integrated and uniform learning environment, becomes more and more apparent. In some cases it is preferable that administration of the system lies with the own organization (for example when one wants to indicate whether parts of a learning module belong to a basic or an advanced track). These pros and cons of the DWO environment were also mentioned during interviews with teachers and students. We refer to [12] for details about practical problems, user wishes and other remarks that came up on usage of the DWO math environment in classroom experiments.

4 Using SCORM to monitor student performance with applets in an existing VLE

A big advantage of using an existing virtual learning environment with which students and teachers are familiar is that many things run very smoothly: the users know how to log in on the system, how to send emails to groups of student or to individuals, how to work with other built-in communication tools, and how to inspect students’ results and export them into the school administrative system. There are a couple of requirements when aiming to integrate applets into an existing learning environment:

- An applet and the VLE have to be able to communicate with each other;
- It should be possible to share the content in various VLEs, for it could very well be that existing programs and learning environments change over the years at school;
– It should be possible to use this mechanism for other applets or computer programs.

The use of existing standards seems to be the best method to reach these goals, with a clear distinction between the role a VLE plays and the role the program plays. This distinction also means that suppliers of learning environments can operate independently from the suppliers of educational programs, but with the certainty that both work well together. A well-known and often used model is the Sharable Content Object Reference Model, abbreviated as SCORM [3]. It aims to integrate several specifications into one standard describing technical requirements for storing digital learning materials and the interaction between VLE and these materials. SCORM 2004 consists of seven different standards, including requirements for

– metadata (What type of material? What is the target audience? Who has made this material? How can use this material? etc);
– constructing SCORM packages;
– for navigation through the package;
– for ‘running’ the package in a particular VLE.

Many system suppliers already support the SCORM model, e.g., Moodle [4] and Blackboard [13]. Within the GALOIS project we adapted applets and exercise programs so that they would be SCORM compliant. We intended to use SCORM compliant packages in Moodle, which is called MichelangElo at the St. Michaël College, but by using a standard we wanted to ensure that our results could be used in any SCORM compliant learning environment.

Fig. 6. The ‘Herleiden’ [Rewriting] applet in Moodle.
One of the applets that we made SCORM compliant is the ‘Herleiden’ applet, shown in Figure 6. This applet was actually used in a fourth grade class to practise rewriting of given expressions and to reinforce symbol sense. Examples of given assignments concern expanding and rewriting mathematical expressions like $2-3(2a+2b)-2c+a-(a+2)$, $(a+b)(b+c) - 2ac$, and $(2x+y)^2 - (2x-y)^2$. Many students had problems with this type of assignments, despite the attention given to formula manipulation in the years before and the successes at that time. A common mistake in teaching and learning: not only does one have to teach a certain mathematical skill or ability, it also has to be maintained. Practice makes perfect, but in this case it would be helpful if students could practise a lot without giving the teacher an unnecessary extra workload. A web-based exercise program makes this possible. In our school situation, students could practise rewriting formulas during computer workshops at school as well as at home. Students’ scores and results were stored in the system, which enabled the teacher to see whether students have a good command of the algebraic skill of removing brackets and combining like term.

Within MichelangElo it is also possible to export results to text and spreadsheet format. With slight modifications they can even be exported into existing school administration systems. When giving assignments, there naturally are students who do not do their homework. Low scores easily reveal this. Moodle’s internal messenger can be used to address these students. Problems are also discovered easily, especially because ‘the road to an answer’ is also stored in the VLE [14]. For example, some students still made the common mistake to rewrite $(2a + b)^2$ as $4a^2 + b^2$. This misconception was addressed in the next lesson by using the area model to explain why this equivalence doesn’t hold. We noticed in practice that the students — at the difficult age of 16 — were highly motivated by this novel way of practising an otherwise difficult and boring subject; they found the applets attractive and motivating. The short-term effect of using the applets in the VLE was that students seemed to get a better understanding of the structure of formulas and that the teachers could monitor student performances and address common mistakes more easily. All this happened within a VLE that was already in use at school and that both teachers and students were familiar with.

5 Intelligent feedback

A teacher must be all ears and have eyes in the back of one’s head during classroom work when he or she puts effort in interactive education that takes individual differences of students into account. Occasionally — mostly when working with small groups or in a face-to-face situation — one gets the feeling of having realized ‘customized education’ in which differentiation based on the needs of the students came into being. Normally, a teacher is forced to aim at the average student or the interaction is merely a random check. Issues like a large number of students in a class, a limited number of contact hours, time pressure, and so on, often make things more complicated.
Whether ICT can make the difference here is not clear yet. Of course, a modern teacher benefits nowadays from all kinds of computer programs and Internet (for lesson material and communication), but the optimism that ICT can contribute to effective teaching and learning is not shared by everybody. It is tempting to state that an online exercise environment enables students to practise mathematics independently and as much as needed, without increasing the work load of a teacher. But often the interaction is rather primitive and limited: answers are marked correct or incorrect without any explanation or further information. The student is left with questions like “What’s wrong? Should I type a comma or a decimal point (or the other way around)?”, “Is my answer not accurate enough (or too accurate)?”, and “Did I make a mistake and, if so, what kind of error: a careless mistake, a reading error, or a conceptual misunderstanding?” When they call in a teacher he or she may hope to find out quickly where the shoe pinches.

An interesting question is to what extent the expertise of a mathematics teacher, i.e., the knowledge about mathematics, teaching and learning, and about the students, can be built into an exercise environment in a such that student receive useful feedback when they make a mistake. We refer to this as ‘intelligent feedback’ and ‘tailored feedback’.

In education, feedback is inextricably bound up with learning processes [15]: It is usually meant to give students insight in their own learning process or functioning, to confirm, strengthen or transform their knowledge, and to give teachers insight in the competencies of their students so that they can make their teaching more effective. In this paper we restrict ourselves to ‘local feedback’, i.e., to feedback that is based on the typed answer or another interaction provoked by a question or task. Different forms of local feedback can be distinguished and often more than one of these forms is present in a single reaction. Some examples:

- **conceptual**: explanation or elucidation of a particular reaction (comments on the accuracy of the given response, instructions of how to input mathematical expressions, etc.);
- **clarifying**: supportive remarks concerning content (hints, references to learning materials, etc.);
- **evaluative**: information about total score, time spent on the task, and so on;
- **affective**: for example, an encouragement or a indication of correctness of intermediate results;
- **advisory**: think of suggestions for further practice.

We speak about intelligent feedback when it is based on an analysis of the student’s answer or action and when it goes beyond a simple correct/incorrect sign or a mark. For example, an incorrect answer is annotated with a short explanation of what might have gone wrong. Also, a hint might be given or a student gets a second chance. Another example is the highlighting of a correct, but mathematically inferior answer (e.g., $\sqrt{4}$ and $\sqrt{50}$ are usually simplified to 2 and $5\sqrt{2}$, respectively). Intelligent feedback is a detailed reaction based on the following.
1. expert knowledge about the mathematical content;
2. a model in which common mistakes and actions are rubricated;
3. knowledge about learning and instruction so that the most suitable feedback can be chosen.

In the GALOIS project we have experimented with tailored feedback for exercises in algebra and arithmetic. For an algebraic example we refer to [16]. Table 1 shows various forms of feedback on typical answers to an arithmetic question.

**Question:** Your bicycle tyre has a puncture. Each minute 8% of air escapes. What percentage has escaped after 11 minutes? Round your answer to the nearest integer.

<table>
<thead>
<tr>
<th>Response</th>
<th>Feedback</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>60</td>
<td>Excellent!</td>
<td>correct answer</td>
</tr>
<tr>
<td>60.04</td>
<td>Fine, but too accurate</td>
<td>too accurate answer</td>
</tr>
<tr>
<td>60,04</td>
<td>Your answer cannot be interpreted; use the decimal point</td>
<td>not recognized by parser</td>
</tr>
<tr>
<td>59 or 61</td>
<td>Close to the correct answer. A round-off error?</td>
<td>probably a round-off error</td>
</tr>
<tr>
<td>40</td>
<td>This amount of air is left</td>
<td>well-known mistake</td>
</tr>
<tr>
<td>133</td>
<td>Confusing addition of 8%?</td>
<td>1.08** = 2.33; well-known mistake</td>
</tr>
<tr>
<td>88</td>
<td>No, this is not the way it goes. Use the growth factor.</td>
<td>notorious mistake (8 times 11)</td>
</tr>
<tr>
<td>other</td>
<td>Wrong answer; have again a close look at the section on exponential decay</td>
<td>God knows what went wrong.</td>
</tr>
</tbody>
</table>

Table 1. Example of tailored feedback on students' responses.

This form of feedback is not in the future. Tailored feedback exists in the computer algebra based testing and assignment environment STACK [17] and in the custom-made JavaScript-based testing and assignment environment called DITwis [5]. At first, DITwis was developed as an independent means to practice and test certain concepts without teacher’s intervention. However, we thought it would be useful to see whether integration of DITwis into an existing VLE would be beneficial. Again we made this program SCORM compliant. We used the system in a third grade (trigonometry) and a fourth grade (combinatorics and probability) class. The advantages of integration soon became apparent, as well as (temporarily) some disadvantages. Many of these findings overlap with the aforementioned ones, with a positive emphasis on: better insight in student work and problems that students face; the possibility of practising anytime, anywhere; the added bonus in the communication facilities of a standard VLE.

6 Design and use of a customizable SCORM compliant applet generator

It would be even better if an applet could be configured within the learning environment in such a way that custom assignments could be made. This pos-
sibility has been successfully explored in the last phase of the GALOIS project in 2006-2007. In this section we discuss a prototype of a customizable SCORM compliant applet generator for learning school algebra and calculus.

The primary design goal was to enable teachers to create exercises and tests for their students similar to the existing WisWeb applets, but set up according to their own ideas. WisWeb applets dealing with the same mathematical topic vary with respect to feedback, ranging from marking correct/incorrect only to symbols indicating (in)correctness of intermediate results, and with respect to marking of answers (e.g., different marks for fully simplified answers and correct, but not completely worked out responses). We wanted to leave the decision on the level of feedback and the marking system to the teacher who creates algebraic exercises and tests. Furthermore, teachers should be able to create SCORM compliant applets so that they can use them in any virtual learning environment that supports this standard. In this way, teachers could exchange their materials more easily. Since teachers are not computer programmers, the process of creating SCORM packages containing Java based exercises should be as simple as possible. Teachers should be able to concentrate on the mathematical contents and instructional design instead of spending time and effort on technicalities. A teacher should also be able to look at the end result through the eyes of students. Keeping in mind that most teachers have no programming experience, the process of creating randomized questions should be kept simple, too.

Let us first look at a real example of an exercise on computing derivatives, which was used in a test for 4th grade pre-university students. Figure 7 is a screenshot of a prototype of the customizable SCORM compliant applet generator.

![Fig. 7. Creating an exercise with the applet generator.](image-url)
When this exercise together with all the other exercises in the test is saved, the program generates a SCORM package that can be uploaded into a VLE that supports this standard. That’s all! In such a learning environment, marking of the students’ work is automatically dealt with. Slight modification of the Moodle environment and other VLEs allowed us to save and reuse the state in which a student leaves the applet. The advantage of this in school practice was that a student could temporarily stop working at the computer and return later to the tasks, that a student could go back to an earlier task and improve his or her work, and that a teacher could not only see the marks of the students, but also inspect their work and see how they had done the exercises. Figure 8 shows a screen shot of how the applet generated looks like when used in Blackboard. The example shows that the randomly generated exercise was not completed yet, but was already marked with 6 points; combination of like terms would lead to the full score of 10 points.

Let us examine in more detail the implementation of the question shown in Figure 8. It naturally includes more than just the question posed to the student. The minimum information which needs to be entered to create a viable question is the text of the question, the teacher’s provided answer, and the marking. For simple questions this is entirely straightforward. Having mastered this, a teacher can seek to include randomly generated parameters, a more complicated expression generation, a tool menu, and more sophisticated marking of (intermediate) expressions entered by a student. We detail the fields in the applet generator in this section to illustrate the authoring of questions.

Question variables are of the form parameter = value, separated by a semi-colon when they are specified in one line. The value can be a sequence of integers separated by comma’s or a range of values (denoted as a starting value and an
end value, separated by the ellipsis operator ..). When a question is instantiated, a random choice from possible integer values will be made. It is possible to use previously defined question variables when making a new one. For example, i=1..5; j=i^2 will lead to a random choice of i and j such that j = i^2 and i=1..5; j=1..i will lead to a random choice of i and j such that i ≥ j:

Initial formula is the expression or equation posed to the student in the work area. You use this field for example, when you do not want the formula in the question text. In the given example, the function whose derivative must be calculated is specified in the question text.

Question text is the actual question or exercise posed to the student. It is more or less a compulsory field which the teacher must enter for a valid question. It may contain text and formulas. In Figure 7 you see that the formula (invoked by pressing the F button) is a mathematical expression surrounded by hash signs (#). When the question is instantiated, all parameter values are randomly chosen, their values are substituted in the parts that are between hash signs, and basic simplifications (e.g., combining like powers, adding numbers, removal of superfluous zeros and ones in expressions — think of replacements such as x + 0 → x, 1x → x, x^1 → x, and so on —) are automatically carried out. Apparently, the following values were chosen for the parameter in the given example: a = 4, b = 2, q = 2, and d = 4. More than one pair of hash signs may be present in a formula to allow maximal flexibility in the formula generated. An example says more than words. Assume that a, b, and n are question variables leading to nonzero natural numbers. In Table 2 are listed formulas with 1, 2, and 4 pairs of hash signs leading to different formula types (a concrete example is given for a = 4, b = 6, and n = 12).

<table>
<thead>
<tr>
<th>Formula</th>
<th>Explanation</th>
<th>Sample</th>
</tr>
</thead>
<tbody>
<tr>
<td>#a/#n + #b/#n</td>
<td>The task ‘as it is’</td>
<td>(\frac{4}{12} + \frac{6}{12})</td>
</tr>
<tr>
<td>#a/n + #b/n</td>
<td>Simplification of fraction, but no addition</td>
<td>(\frac{1}{2} + \frac{1}{2})</td>
</tr>
<tr>
<td>#a/n + b/n</td>
<td>The final answer</td>
<td>(\frac{5}{6})</td>
</tr>
</tbody>
</table>

Table 2. Example of the use of hash signs in formula generation.

Correct answer is another compulsory field that the teacher must enter; a checkbox is used to indicate that the answer is an equation instead of an algebraic expression. By using pairs of hash signs, the type of answer that is considered as correct in a randomized question may be specified (as we have seen before). But there exist other ways of specifying the level of correctness. When the box Equivalence is checked, any student answer that is equivalent to the teacher’s answer is marked as correct and a number of points is scored. When the box Rewriting is checked, additional points can be earned by the student when he or she has simplified an intermediate result. In the given example, extra points are given when the answer is a polynomial in expanded form, i.e., without brackets. Special forms that have been implemented in the prototype of the applet generator are expressions in which
– no brackets are left in polynomial expressions (e.g., \((x + 1)^2 \rightarrow x^2 + 2x + 1\));
– powers with the same base have been combined into one power \((x^2 x^3 \rightarrow x^5)\);
– a formula has been rewritten such that no powers with fractional or negative exponents occur \((x^{-2} \rightarrow \frac{1}{x^2} \text{ and } x^{\frac{1}{2}} \rightarrow x^{\sqrt{2}})\);
– a rational expression has been rewritten into a single fraction \((\frac{1}{2} + \frac{1}{3} \rightarrow \frac{5}{6})\);
– logarithms have been combined as much as possible into a single logarithm \((2 \ln 3 + \ln 8 \rightarrow \ln 72)\);
– only simple logarithms appear in a formula \((\ln 72 \rightarrow 2 \ln 3 + 3 \ln 2)\).

When the box **Exact** has been checked, a student can score extra points when he or she enters exactly the answer that the teacher had in mind. When the box **Steps enabled** has been checked, a student can enter intermediate expressions in the work field, gets feedback on correctness, and may even score points for correct intermediate expressions.

**Levels** of sets of exercises can be specified in the left bottom corner of the user interface of the applet generator. In the example shown in Figure 7, the different levels have been named after the differentiation rule that must be applied. But one can also divide the exercises in a test into sets of easy and difficult exercises. The number of levels and the number of questions at each level can be set by pressing the up/down and left/right arrows, respectively.

**Title** of the exercise is optional. It can only contain plain text and no formulas. Like in the question field, you can only use alphabetic characters and digits. Use of accents and HTML coding is not provided (yet).

![Fig. 9. Creating an exercise about addition of two fractions.](image-url)
The dedicated use of hash signs in randomized questions and the sophisticated processing of (intermediate) student’s answers are very convenient for intelligent marking of responses. Such intelligent question generation and marking of answers is not easily implemented in computer algebra based systems like STACK [17] or Maple T.A. [18]. Making an exercise in steps is not possible in these systems either, unless it has been explicitly implemented by the author of the exercise. The following example shows how easy it is to implement an addition of two randomly generated fractions and to give different marking when a student is able to

- write the fractions such that they have a common denominator (2 points);
- rewrite the sum correctly as a single fraction, but not with the smallest possible denominator (5 points);
- find the exact teacher’s answer (10 points).

Figure 9 shows the applet generator in action. Figure 10 shows a correct answer to the question. The marking changed in the intermediate steps from 2 points to 5 points, and finally to the full score of 10 points.

**Fig. 10.** Creating an exercise about addition of two fractions.

The main drawback of the generated Java applets is that there is no computer algebra system behind it (yet). Thus, equivalence testing is done numerically. Although this may lead to erroneous results — after all, inequivalent algebraic expressions may evaluate to the same numerical result — in practice it works quite satisfactory.
7 Conclusion

Existing mathematical tools become even more powerful and useful when results and performance of students can be tracked online. Use of a common standard, in this case SCORM, enables the sharing of content, but also of storing student performance. Students know that they can be and are monitored by their teacher. Together with other features in virtual learning environments, especially for communication, students are stimulated to do their homework, or even more than that. Students do not seem to mind this ‘big brother is watching you’ concept. On the contrary, they seem to appreciate the fact that a teacher has good insight into common mistakes and can use this knowledge to address certain problems more effectively. They also seem to find that computer practice prepares them better for written exams than making assignments from a book. Computer aided learning is also seen as a more ‘fun’ way of practising otherwise boring subjects. Direct feedback adds to this appreciation as well as the fact that small mistakes can easily be corrected. Using direct feedback, giving students hints and enabling a ‘trial and error’ approach makes it possible for students to explore different strategies for solving a given problem. But exercise applets without these facilities have also been developed.

In our opinion, digital assignments and diagnostic testing add to the motivation and performance of students in mathematics education. By making use of computers, the extra time involved for teachers is acceptable. Contact time can effectively be reorganized because students that need extra attention are signalled earlier. One of the success factors of the use of ICT would be that all teachers in an organization agree on the vision they have on mathematics education. Ideally, less experienced teachers would be supported by experienced teachers and experiences are shared alike to evaluate and improve the use of ICT in the classroom. From technological point of view, SCORM contributes to the realization of meaningful ICT-rich education. The customizable SCORM compliant Java applet generator allows teachers to create exercises and tests for their students in an easy way and based on their own ideas. These tests can be exchanged with fellow teachers because of the use of a widely used standard. Field tests at school confirm that all works well in practice, although many wishes for improvement of the software arise at this stage of the development work. Linking the applets to a backbone computer algebra system and having more question types at one’s disposal are two of the envisioned extensions. The developed ICT tools have a surplus value for mathematics education at secondary school, and probably as well for other educational levels: experiments at the University of Amsterdam in remedial math for freshmen and in math intensive courses for first year chemistry students take place at this moment.

References

1. Information on the projects WisWeb and WELP, Java applets, and instructional materials can be found at the website www.fi.uu.nl/wisweb/en/
2. To get an idea of the DWO math environment login as guest at the website www.fi.uu.nl/duo/en
3. Information about SCORM can be found at the website www.adlnet.org
4. Modular Object-Oriented Dynamic Learning Environment (Moodle). Website: www.moodle.org
13. Blackboard website: www.blackboard.com
14. This requires a slight modification of the VLE. This modification for Moodle 1.5.3 can be downloaded from www.galoisproject.nl. Newer versions are still under development.