Abstract

The paper describes the implementation of an on-line math lesson featuring a system of interactive self-assessment and computing utilities. The lesson is SCORM-compliant so that it can be inserted in any LMS that supports SCORM 1.2. Randomly generated exercises and tests, powered by Java applets, are offered to the user. Data for exercises and tests are randomly generated. Both single-concept and comprehensive tests can be generated at runtime. A number of parameters (such as the difficulty level, problem types in a test or the test grading system) can be controlled at design time.

1. Introduction

The idea of supporting (or even replacing) traditional classroom instruction by software systems available on the Internet can be seen at work in a fast-growing number of education institutions and corporate organizations. What the overwhelming majority of computer-mediated learning systems lack when confronted with in-class teaching, is the teacher-student interaction. There is probably no substitute for an experienced teacher when it comes to the crucial issue of assessment of a student’s state of knowledge and of his or her readiness to proceed from one math topic to another. As powerful tools as the computing technology can offer, the student is put in a self-study situation anyway. Particularly sensitive to the absence of that kind of factor is every math-related instruction process. The issue of an effective system of knowledge assessment is certainly the key one when it comes to math e-learning. In addition to exercises that could guide the student through the concepts and methods of a topic, a versatile system of tests is necessary to help the student get the correct estimation of his or her readiness to proceed with the study. We know of one Web-based e-learning system that is very close to satisfying the requirements discussed above. The system is ALEKS (www.aleks.com). It offers several courses on math-related subjects and has amazing possibilities as far as the student’s knowledge measurement is concerned. It should be mentioned at this point that ALEKS’s power is based on a formal theory of knowledge assessment (see [2]).

Partially inspired by ALEKS, the authors of this paper decided to experimentally explore the possibility of creating an Internet math lesson (e-lesson), covering a single topic from the secondary-school level mathematics curriculum. The main idea was to see to what extent it was possible to enhance the lesson functionality by a flexible set of self-assessment utilities at low cost.

2. Project initial assumptions

As we mentioned above, one of the project goals was to find out, by experiment, how a single math topic could be offered on the Internet in the way giving the final user a reasonably large and flexible set of self-assessment tools at a relatively low cost of their creation. The main idea was not so much to give a detailed, full treatment of the math topic in question, but rather to create a kind of a remedial-type tutorial incorporating an effective system of exercises and tests. Additionally, we wanted the final user’s system and software requirements limited to affordable range.

The following summarizes the project assumptions:

- **Student’s preparedness:**
  - Prior (but not necessarily very recent) contact with the topic.
  - Standard math prerequisites as in the high-school curriculum that are necessary to study the topic.
  - The computer literacy (knowing how to operate windows applications and a Web browser).

- **Student’s computing environment:**
  - An arbitrary operation system capable of running a graphical Web browser. No broadband connection to the Internet was assumed.
  - The browser’s ability to make use of Java 2 Runtime Environment SE, version 1.4. The e-lesson web pages would be designed in such a way that the JRE software on the
The issue of how the self-assessment part should be working and what kind of technology tools to use for the creation of the lesson was of special importance. The lesson’s mathematical content was to be “wrapped” in a framework of exercises and tests, offering the student:

1. Enough room for practice.
2. Tests functionality that would help the student with identifying the current state of his or her knowledge as accurately as possible in a self-study environment.

Accordingly, the following approach was taken:

- All the tests included in the lesson would be based on random generation of data rather than on a static database of problems. (Database driven selection of other problems could be added later).
- The tests would be implemented in the form of Java applets. The Java technology was chosen for several reasons. It features powerful programming possibilities with no need for a programmer to reach for commercial development environment. It can interact with Maple [3] or Mathematica [4], the feature that could be utilized in future extensions of the project. Finally, the fast-growing popularity of Java Servlet Pages technology on the Internet seemed to be a promising perspective for yet another direction of development.

As far as the explanation/illustration part is concerned, we decided to use the GIF graphical format for the publishing of mathematical expressions and graphs. GIF Format is commonly used for publishing graphics on the Web, and is well supported by all kinds of browsers. The well-known powerful graphical capabilities of Mathematica were utilized to generate GIF format of math expressions and graphs. Later, in the course of the project, Mathematica turned out to be a perfect choice for that task. It took some experimenting and programming skills at the beginning to get satisfactory results, but eventually the required output was obtained at an amazingly low time cost. It is the latter two components on the list above that that turned out to be the most valuable and attractive part of the lesson. Powered by appropriately parameterized Java applets, problems and tests show an amazingly high degree of variability and flexibility. Different data are selected at random each time an individual problem (exercise) is visited. Problems and tests with strictly predefined data are available as well. A variety of additional features can be easily controlled at design time with no need to modify the applets source code. Those features include (but are not limited to):

- Problem difficulty level (e.g., a Web designer can make decisions as to the range of roots in a generated “solve a quadratic equation” problem or as to whether rational numbers are allowed as roots in addition to integers).
- Time assigned to student for the solution.
- Availability of solution presentation by the system.
- Number of problems in a test.
- Types of problems in a test; a test can consist of problems of one type or of problems of varying types; the presence and the number of respective problem types in a test can be fully predefined, partially predefined, or entirely random.
- Test time and access to solutions; the functionality of tests has been enhanced by the mechanism that prevents the time countdown from being interrupted; even closing the applet window by the student and re-opening it again

3. Organization of the lesson

The math content of the lesson is organized according to the standard order of Quadratic Function topic exposition. From that viewpoint, the lesson is divided into several units. A single unit corresponds to one or more concepts included in the topic. From the point of view of math contents the following material is covered:

- Introduction to the notion of a quadratic function.
- Canonical form.
- Zeros of quadratic functions.
- Graphing.
- Quadratic inequalities.

The lesson has been given the Web form, in which the explanation/illustration materials blend with computing tools, exercises and tests. Four types of components of the resulting Web structure can be named from the functionality viewpoint:

- The exposition/examples component (in a near-book form, additionally equipped with a convenient system of links to reminders, auxiliary explanations and illustrations).
- The calculator component, offering four graphing calculators; the calculators are intended to facilitate the student’s exploration of concepts;
- The problem component, offering randomly generated interactive exercises of 14 types.
- The test component, including randomly generated tests; each test can consist of a collection of problems of various types. The problem types can be predetermined by a Web designer or randomly generated.

It turned out to be the most valuable and attractive part of the lesson. Powered by appropriately parameterized Java applets, problems and tests show an amazingly high degree of variability and flexibility. Different data are selected at random each time an individual problem (exercise) is visited. Problems and tests with strictly predefined data are available as well. A variety of additional features can be easily controlled at design time with no need to modify the applets source code. Those features include (but are not limited to):

- Problem difficulty level (e.g., a Web designer can make decisions as to the range of roots in a generated “solve a quadratic equation” problem or as to whether rational numbers are allowed as roots in addition to integers).
- Time assigned to student for the solution.
- Availability of solution presentation by the system.
- Number of problems in a test.
- Types of problems in a test; a test can consist of problems of one type or of problems of varying types; the presence and the number of respective problem types in a test can be fully predefined, partially predefined, or entirely random.
- Test time and access to solutions; the functionality of tests has been enhanced by the mechanism that prevents the time countdown from being interrupted; even closing the applet window by the student and re-opening it again
does not stop the countdown from continuing in the background; the timer simply resumes at the point where it was as the window was last being closed.

- Grading system in a test; each problem in a test can be assigned its score; the final grade for the test is computed based on a few predefined grading systems.

4. Computing and self-assessment tools

Every individual lesson unit is provided with a link to the corresponding problem section where the student can find interactive, randomly generated problems related to the unit material. Besides, three additional parts are included in the Web structure:

- Calculators.
- Tests related to individual units of the lesson.
- Final test, offering problems of various types.

4.1. Computing tools (calculators)

Four types of computing utilities (calculators) accompany the lesson. Each calculator works as a Java applet that can be either embedded as an object on a Web page or launched in a separate window by pressing a button on a Web page. The following calculators are available:

The canonical form graphical calculator. The applet window is equipped with three sliders for controlling the values of the square term coefficient and the coordinates of the vertex of the parabola. Changing the position of a slider modifies the graph of the quadratic trinomial. The canonical form corresponding to chosen values is displayed.

The general form graphical calculator. The applet window provides edit fields for entering the coefficients of a trinomial. The graph is then displayed.

The quadratic equations calculator. The applet window provides edit fields for entering the coefficients of a trinomial. The number and the values of the trinomial roots are then found and displayed.

The quadratic inequalities calculator. The applet window allows the student to explore inequalities of the type $ax^2+bx+c > d$ (as well as the three remaining inequality types). The student is provided with edit fields for entering a trinomial coefficients and the right-hand side of the inequality, as well as with a drop-down list of symbols corresponding to the required inequality type (e.g. “less than”, “less than or equal”). The applet then graphs the parabola, the horizontal line given by the right-hand side of the inequality, and then graphically shows the solution on the x-axis. Additionally, the algebraic form of the solution is displayed.

4.2. Exercises

The student is offered 14 types of exercises, often referred to as “problems”, too. Below we give a complete list of their types, together with a brief indication how a student is asked to enter his or her input.

1. Solve the given quadratic equation. (Check the appropriate box indicating the number of roots. Enter the roots in the fields provided in the window).

2. Given the trinomial, find the coordinates of the vertex of the corresponding parabola. (Enter the appropriate values in the fields provided in the window).

3. Find the canonical form of the given trinomial. (Enter the appropriate values in the fields provided in the window).

4. By looking at the location of the vertex and at the direction of parabola, select the only expression matching the graph. (Check the box beside the appropriate expression).

5. Give an example of a trinomial whose graph is shown. (Enter the coefficients in the fields provided in the window).

6. The figure shows the location of the vertex and of another point on a parabola in the coordinate system. Find the corresponding quadratic function formula. (Enter the coefficients in the fields provided in the window).

7. The figure shows the locations of three points on a parabola in the coordinate system. Find the corresponding quadratic function formula. (Enter the coefficients in the fields provided in the window).

8. Solve the given quadratic inequality. (Check the box beside the appropriate solution type. Enter the appropriate values when applicable).

9. Given the quadratic function formula, graph the function. (Use the graphical tools provided in the window).

10. Solve the inequality $f (x) < g (x)$ where $f$ is the given quadratic function and $g$ is the given linear function. Then use the graphical tools provided in the window to graph both functions. Use the graphical tools to mark the solution on the x-axis.

11. Without solving the quadratic equation, determine the evenness of its roots. (Check the appropriate box indicating one of the three possible cases).

12. Without solving the quadratic equation, determine the sum, the product, the arithmetic mean and the geometric mean of its roots. (Enter the appropriate values in the fields provided in the window).
13. Solve the given bi-quadratic equation. (Check the appropriate box indicating the number of roots. Enter the roots in the fields provided in the window).

14. A system of one quadratic and one linear equation (in variables x,y) is given. Solve the system. (Check the appropriate box indicating the number of solutions of the system. Enter the x-values and the y-values in the fields provided in the window).

As we said before, every individual problem is powered by a Java applet that can be embedded as an object in a Web page or can be launched in a separate window through a button on the page. The actual way of an applet’s behavior can be controlled at design time. An individual applet randomly selects numerical data for the problem. The applet window displays the problem formulation together with the user interface controls necessary for the solution to be entered by the student. Additionally, the applet window can (but not necessarily needs to) be equipped with the following elements:

- **New Test** button, generating an exercise of the same type but with different data. A Web designer can decide how many problems are to be generated in that way (in particular, an unlimited number of problems can be generated).
- **Solution** button, making the applet display a full step-by-step solution of the problem.
- **Submit** button, making the applet verify the solution and display the grade.
- **Help** button, showing a brief explanation how to operate the applet window.

The time given to the student for the solution can be left unlimited or can be restricted to the amount chosen by a Web designer. In the latter case, a timer placed in the applet window displays the time countdown. As soon as the timer expires the student can no longer continue editing or entering the solution for the currently given problem. (The student can still ask for another problem, though).

4.3 Tests

Exercises can be combined to yield tests. An individual test is powered by a Java applet that again can run either as a Web-page-embedded object or in a separate window (invoked from within a Web page).

The main power of tests lies in their versatility. Tests devoted to one particular problem type as well as comprehensive tests are available. In addition to the difficulty of individual problems that the test comprises, a variety of test parameters can be controlled at design time:

- The test total time.
- The number of problems in the test.
- The maximum score for a single problem in the test (different problems can carry different scores).
- The grading system – three types of grading are offered for the designer, with percentages of scores transformed into respective letter grades.
- The number of problems of a specific type appearing in the test – each such number can be predefined or the probability distribution can be given, e.g. “for the last two problems on the test, randomly select from types 3, 5, 7, 8 with equal probabilities”.

The tests have this additional functionality:

- The student cannot submit the test done until he or she visits every problem on the test.
- Grading procedure is made available for the student.
- The correct solutions are made available to the student as soon as the test is submitted.
- The tests are fool-proof in that they are resistant to the student’s deliberate attempt to extend the test time; once the test applet starts running and the time countdown begins, even closing the test window does not stop the countdown.

5. Graphing-type exercises

Two particular types of exercises are worth mentioning: types 9 and 10 (see the Exercises section above). Those are exercises that require the student to come up with a picture. Special graphing utilities are available the applet window, so that the student can draw various objects in the coordinate system:

- A parabola (specifically, the student can perform a two-step operation: pointing the location of the vertex followed by pointing the location of another point of the parabola).
- A straight line.
- A half-line.
- Segments of all 4 topological types: open, closed, left-open, right-open.

A virtual eraser is additionally provided, as well as a choice of line styles and colors for the student to use. The exercise grading is based on the number, the type and the location of graphical objects that the student has submitted as the solution. Penalty (negative) scores are given if superfluous objects appear. The correct solution itself is also available in the form of a step-by-step presentation.

6. SCORM-compliance

The lesson has been designed to meet SCORM 1.2 requirements [1]. Once the lesson is placed in a SCORM-
recognizing LMS, it is the LMS environment that controls the user navigation. In that case, an LMS also keeps track of the user progress. Prerequisites have been defined for individual parts of the lesson, forcing the user to pass specific tests in order to be allowed to proceed with the material.

7. Conclusion

The experiment obviously demonstrates that it is perfectly possible to create an on-line math tutorial armed with effective self-assessment tools. Even though the outcome may seem far from a complete educational product, it is the authors’ belief that a step has been taken in the right direction both in terms of logic structure and the technology used. A larger project - the creation of a full on-line remedial math course - is being planned.

9. References


