FiM, a Test & Assessment Environment complementing the Mathematical Education by Drill & Practice Exercises and Examinations

Thomas Risse
Hochschule Bremen, University of Applied Sciences
risse@hs-bremen.de

ABSTRACT

The ever decreasing level of mathematical skills of students beginning their engineering studies calls for radical countermeasures. To gain time for developing high-order competencies we have to unload crude drill & practice exercises to a computer program. We propose an open source learning platform in combination with an open source computer algebra system, CAS, in order to generate custom-made exercises via using CAS-scripts, which in turn assess the performances of the learners. The coupling of the CAS with the learning platform allows the problems generated by the scripts to be stated and the processing of the assessment and the related feedback to the learner for later evaluation. This division of labour allows for efficient programming of (classes of) exercises and for fully utilising platform's book-keeping, evaluation, and statistics features. We will discuss and illustrate our approach by relevant examples, illustrate experiences and provide an outlook on how to spread the usage of FiM and on further developments.

Key words: e-learning, web-based learning, test & assessment, mathematics, computer algebra system, CAS

1. Tendency of Students Mathematical Skills to Fall and Countermeasures

Over the years students at the start of their engineering degree courses show more and more severe deficits in their mathematical skills – deficits as monitored by many observers at least in Germany, and to our knowledge, for example also in Austria and the UK. These deficits comprise difficulties to calculate and to give rough estimates as well as problems to model, troubles to check plausibility etc. At the same time students have little background knowledge in science. Especially in the mathematical education there seems to be a need for crude drill and practice exercises which we consider necessary to enable students to work their way from simple calculus (with most of the time unrealistic problems) to more demanding tasks like modelling and simulating real problems.

Students should train their basic, low level mathematical skills by running lots of exercises. Such exercises have to be marked, and there has to be some reasonable feedback to the student in case of failure.

The high demand for such training cannot be met by traditional means. There are not enough resources to do all the marking and to give the necessary feedback. Therefore, a computer program is needed to generate parametrisable questions and problems which the students have to answer and to solve – a computer program which at the same time assesses the answers and solutions providing a right/wrong decision and some feedback on errors. (The question which types of mathematical skills can be developed by which type of computer assistance is addressed in general by Risse [8] and for graphical input by Risse [9].)

2. FiM – a Fitness Program in Mathematics not only for Engineering Students

We designed such a system as an addition to the open source learning platform ILIAS, Ilias [3] and used a CAS like Octave [7] or proprietary MATLAB/Maple, TheMathworks [5] as the symbolic mathematics engine. (In the following, ILIAS stands for any suitable learning platform and CAS for any suitable computer algebra system.) Students formulate their answers using the prevalent syntax of mathematical expressions as in MATLAB/Maple or in any other programming language.

We demonstrate how by scripting a wide variety of parametrisable problems can be generated and how the CAS checks the correctness of the solutions of the students.

2.1 Programming FiM-Scripts by Using a Template

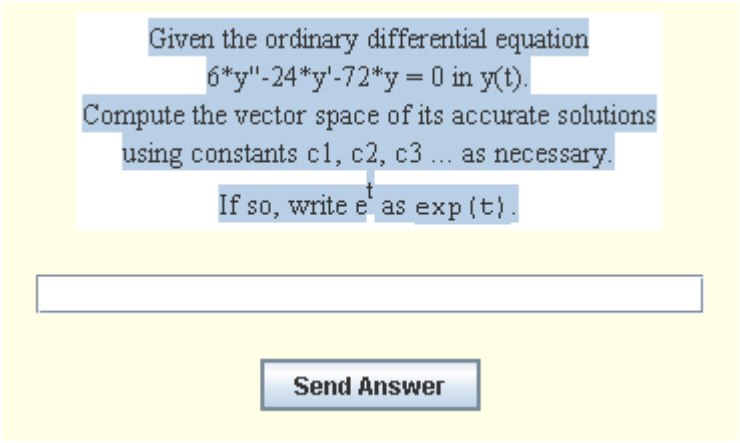
Together with some supporting global functions we use a template for FiM-scripts to facilitate the programming of exercises. Here, we list its generic structure in a pidgin octave/MATLAB m-file to show how scripts work as well as how the learning platform and the computer algebra system are coupled.

In principle, when no answer is passed to a FiM-script it produces an exercise by generating variable names or constants and parameters at random. However, when an answer is passed, the FiM-script checks the correctness of the answer and produces suitable feedback.

```
function result = Testname(answer)
if (nargin == 0)                % if in answer no argument is given then generate test
        % generate symbol names etc
        result = textoftest;    % parameters or coefficients at random
        % and then, construct text of test question
else    % if in answer an argument is given then assess test
        if (answer is mathematically correct)
                if (answer has the correct form)
                        result = 'ok';
                else
                        result = 'false';       % or some other appropriate feedback
                end
        else
                result = 'false';               % or some other appropriate feedback
        end
end
```

Of course, in this template for the sake of clarity we omitted e.g. the management of the functions workspace by the session ID, the localization (the same exercise in different languages), and the discrimination of drill, self test and examination modes. Additionally, we skipped the analysis of the answer of the learner necessary to generate specific feedback.

However, due to its importance, we give an example of feedback generation: the learner is asked to solve a second order ordinary differential equation with constant coefficients. The parameters of this equation might at random be generated to produce the following exercise.



Then FiM first checks whether the answer of the learner is a solution of the differential equation, secondly it checks whether there are the two fundamental solutions provided by the learner and thirdly it makes sure that the answer spans the whole vector space of solutions. If the answer fails any of these conditions the script generates a corresponding feedback, given that the learner does not perform the test in examination mode.

3. Communication between ILIAS and the CAS

The coupling between the learning platform ILIAS and the computer algebra system provides the necessary services to the client application, i.e. the learner. The following figure illustrates the different interacting programs.

ILIAS allows the so called tests to be implemented by JAVA applets, Schottmüller [10]. We implemented the communication between the JAVA applet on the client and the JAVA interface program on the server by Remote Message Invocation, RMI. The access to the CAS octave or MATLAB/Maple is implemented using the JMatLink library, see Müller [6].

1. Client: By choosing an exercise the learner makes the JAVA applet to call the corresponding script on the CAS server by RMI.
2. Server: The JAVA interface program calls the script on the CAS server which generates parameters at random and produces the exercise as a string. The JAVA interface program returns this string to the JAVA applet by RMI.
3. Client: The JAVA applet presents the exercise to the learner.
4. Client: The learner provides his or her answer as a string. The JAVA applet sends this string by RMI to the m-file which generated the exercise on the MATLAB/Maple server.
5. Server: The script on the CAS server checks the answer for correctness and produces appropriate feedback as a string which again the interface program sends to the JAVA applet by RMI.
6. Client: The JAVA applet presents the result to the learner. The learning platform ILIAS does all the bookkeeping for later evaluation and statistics.

4. Potential and Limits of FiM

There is practically no limit to the areas which can be addressed by FiM-questions and problems: algebra, calculus, statistics, etc. comprising literally all areas of scientific computing. We verified the potential of FiM by implementing a lot of the exercises of a typical exercise book for first year students, and twenty exercises for students in their third semester as well.

Right now, the pool of exercises comprises well over a hundred exercises for students about to commence their engineering degree courses and those in their third semester. For the beginners, topics cover term manipulations, fractions, powers, percentages etc. – for the students in the third semester there are Fourier series, multi dimensional analysis, i.e. functions of several variables with typical applications like optimization and integration, and stochastics, i.e. probability calculus and statistics.

All exercise parameters like constants and variable names can be generated at random: programming one script thus generates a variety of exercises. By initialising the seed of the pseudo random number generator depending on the date of some test or examination it is even possible to generate exactly the same questions and problems for a group of students who are scheduled to be under examination on this very date.

4.1 A Real Open Source Version of FiM

Substituting all MATLAB/Maple by octave [7] together with GiNaC [2], s.a. Bauer, Frink, Krekel [1] in the future, will render the FiM system a complete open source solution for the generation and assessment

of question/answer type tests – competing with commercial, proprietary solutions like MapleTA of Maplesoft [4] for example.

5. Experiences, Expectations and Promotion of FiM

The necessity of reasonable support of the drill & practice part of mathematical education drives our development of the technical aspects of FiM. The means offered by FiM are regularly employed when the next generation of first year students has to brush up their mathematical skills, and when we teach the third semester students each semester. FiM proves that custom-made exercises and tests can be efficiently generated by programming CAS scripts in such a way as to offer opportunity for training and to provide helpful feedback especially to weak students.

At the same time we co-operate with teachers of a handful of schools in Bremen who weekly specify exercises which we implement quickly enough so that their students can work on these tailored exercises. In this way, the ongoing learning process of students in secondary schools is supported too.

References

[1] C. Bauer and A. Frink and R. Krekel: GiNaC; J. Symbolic Computation, 2002, Vol 33, pp 1-12, www.ginac.de/csSC-0004015.pdf

[2] GiNaC: GiNaC is Not a CAS; www.ginac.de

[3] ILIAS Learning Management (Integrated Learning, Information- and co-operation System); www.ilias.de

[4] Maplesoft: Maple T.A. – Online Testing and Assessment; www.maplesoft.com/products/mapleta

[5] The MathWorks: MATLAB, the Language of Technical Computing; www.mathworks.com

[6] S. Müller: JMatLink – Connect Matlab and Java; http://jmathlink.souceforge.net

[7] Octave: 'GNU Octave is a high-level language, primarily intended for numerical computations mostly compatible with Matlab'; www.octave.org

[8] Th. Risse: E-Tests – What Can be Assessed in E-Learning Applications; ICL 2002, Villach, in M. Auer, U. Auer (Eds.): Interactive Computer Aided Learning – Blended Learning; kassel university press, 2002, ISBN 3-933146-83-6 www.weblearn.hs-bremen.de/risse/papers/ICL2002

[9] Th. Risse: Assessing Graphical Sketches in e-Learning Systems; ICL 2005, in M. Auer, U. Auer, R. Mittermeir (Eds.): Interactive Computer Aided Learning – Ambient and Mobile Learning; kassel university press, 2005, ISBN 3-89958-136-9 www.weblearn.hs-bremen.de/risse/papers/ICL2005

[10] H. Schottmüller: ILIAS Test & Assessment Documentation; www.ilias.de/docu/data/docu/lm_data/lm_8782/Test_Assessment_38.html