

A Poor Mans TA – Testing and Assessing Mathematical Skills by CAS-Scripts

Thomas Risse

*Institute of Informatics and Automation,
Hochschule Bremen, University of Applied Sciences, Germany*

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 a combination of an open source learning platform together with an open source computer algebra system, CAS, in order to generate custom-made exercises by CAS-scripts which in turn assess the answers of learners. By coupling a CAS and a learning platform the learning platform states the problem generated by a CAS script and processes the scripts assessment of and its feedback to the answer of the learner, saving it for later evaluation.

This division of labour allows for efficient programming of (classes of) exercises and for fully utilising the learning platforms bookkeeping, evaluation, and statistics features.

We will discuss and illustrate our approach by relevant examples.

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. These deficits comprise difficulties in calculating and in giving rough estimates as well as problems in modeling and troubles in checking plausibility, etc. At the same time students have little background knowledge in science. Especially in the mathematics 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 modeling 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 parameterisable 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 of which types of mathematical skills can be developed by which type of computer assistance is addressed in general by Risse (2002) and for graphical input by Risse (2005).)

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 (1997)), and used MATLAB/Maple (The Mathworks (1994-2008)) as the symbolic mathematics engine. (In the following, ILIAS stands for any suitable learning

platform and MATLAB/Maple for any suitable computer algebra system.) Students formulate their answers using the prevalent syntax of mathematical expressions as in MATLAB/Maple or in any programming language. We demonstrate how by scripting, a wide variety of parameterisable problems can be generated and how MATLAB/Maple checks the correctness of the solutions of the students.

Programming FiM-Scripts by Using a Template

Together with some supporting functions we use a template for FiM-scripts to facilitate the programming of exercises. Here, we list its generic structure as a pidgin 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 normally generating variable names or constants at random. However, when an answer is passed, the FiM-script checks the correctness of the answer and produces suitable feedback.

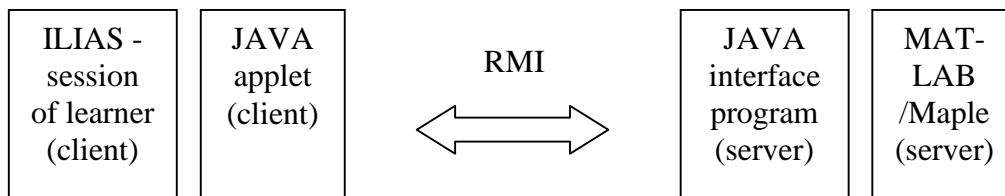
```
function result = Testname(answer)
if (nargin == 0)           % if no argument is given then generate test
                           % generate symbol names
    result = textoftest;   % parameters or coefficients at random
                           % and then, construct text of test question
else                       % if 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 the management of the functions workspace by the session ID, the localization (the same exercise in different languages), and the discrimination of exercise and examination mode.

Communication between ILIAS and MATLAB/Maple

The coupling between the learning platform ILIAS and the computer algebra system MATLAB/Maple 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, cf. Schottmüller (2007). 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 MATLAB/Maple is implemented using the JMatLink library, see Müller (2005).



1. Client: By choosing an exercise the learner makes the JAVA applet to call the m-file on the MATLAB/Maple server by RMI.
2. Server: The JAVA interface program calls the m-file on the MATLAB/Maple 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 m-file on the MATLAB/Maple 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.

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, see Schäfer, Georgi & Trippler (2006). Right now, the pool of exercises comprises well over seventy exercises for students about to commence their engineering degree courses. Topics cover term manipulations, fractions, powers, percentages etc.

All exercise parameters like constants and variable names can be generated at random: programming one script can thus generate 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.

A Real Open Source Version of FiM

Substituting MATLAB/Maple by octave/GiNaC (octave (1998), GiNaC (2001), Bauer, Frink & Krekel (2002)) 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 solutions like MapleTA (Maplesoft (1988-2008)) for example.

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 will be employed at the latest when the next generation of first year students has to brush up their mathematical skills. 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, their ongoing learning process is supported.

References

Bauer, C., Frink, A. and Krekel, R. (2002) GiNaC. *J. Symbolic Computation* (2002) 33: 1–12. Accessed via www.ginac.de/csSC-0004015.pdf (February 2008).

GiNaC (2001) “GiNaC is Not a Cas.” Accessed via www.ginac.de (February 2008).

Ilias (2007) “ILIAS Learning Management (Integriertes Lern-, Informations- und Arbeitskooperations-System).“ Accessed via www.ilias.de (February 2008).

Maplesoft (1988-2008) “Maple T.A. – Online Testing and Assessment.” Accessed via www.maplesoft.com/products/mapleta (February 2008).

The MathWorks (1994-2008) “MATLAB, the Language of Technical Computing.” Accessed via www.mathworks.com (February 2008).

Müller, S. (2005) “JMatLink – Connect Matlab and Java.” Accessed via <http://jmatlink.sourceforge.net> (February 2008).

Octave (1998) “GNU Octave is a high-level language, primarily intended for numerical computations mostly compatible with Matlab”. Accessed via www.octave.org (February 2008).

Risse, T. (2002) E-Tests – What Can be Assessed in E-Learning Applications. ICL 2002, Villach 25.9.–27.9.2002. In Auer, M. and Auer, U. (Eds.): *Interactive Computer Aided Learning – Blended Learning*. Kassel university press. Accessed via www.weblearn.hs-bremen.de/risse/papers/ICL2002 (February 2008).

Risse, T. (2005) Assessing Graphical Sketches in e-Learning Systems. ICL 2005, Villach 28.9.–30.9.2005. In Auer, M., Auer, U. and Mittermeir, R. (Eds.): *Interactive Computer Aided Learning – Ambient and Mobile Learning*. Kassel university press. Accessed via www.weblearn.hs-bremen.de/risse/papers/ICL2005 (February 2008).

Schäfer, W., Georgi, K. and Trippler, G. (2006) *Mathematik-Vorkurs – Übungs- und Arbeitsbuch für Studienanfänger*. Teubner, Stuttgart.

Schottmüller, H. (2007) „ILIAS Test & Assessment Documentation.“ Accessed via www.ilias.de/docu/data/docu/lm_data/lm_8782/Test_Assessment_38.html (February 2008).