

Exercise templates with Sage

Pedro Cruz^{1,2}, Paula Oliveira¹, Dina Seabra^{1,2}

¹ Universidade de Aveiro, Campus Universitário de Santiago, 3810-193 Aveiro, Portugal

² Center for Research and Development in Mathematics and Applications, Campus Universitário de Santiago, 3810-193 Aveiro, Portugal

E-mail: {pedrocruz,paula.oliveira,dfcs}@ua.pt

Abstract

As the literature points out, students can benefit from the study of a detailed answer of an exercise after trying to solve it in a traditional pen-and-paper way. This encouraged the creation of a new software package for the growing community of Sage users. Thus, this package aims at enabling the author to add, change and delete parameterized exercise templates, with detailed answer, from one or more personal databases. Moreover, it allows the production of new exercises in a booklet style or exam form.

This paper describes the package, presenting some examples of its functionalities.

2000 Mathematics Subject Classification. 97U50. 97U30.

Keywords. exercise template; Sage.

1 Introduction

MEGUA [12] is a software package for Sage [10], developed by the authors of this paper. It is a computer algebra system that works in Linux, virtual linux for MS-Windows, and Web environment, allowing a working group to take advantage of the Sage Notebook sharing capabilities. With this package an author can create, edit and share his own database of parameterized exercise templates and an user can search, select and produce L^AT_EX and PDF files from the exercises on a database. Given a database, a teacher can produce text (as a PDF document) for class work, exams or homeworks. We remark that computer assessment is not implemented since other projects at our university deal with it [11].

Some ideas are shared from other works. In [1] it is presented a work that uses Web access to a database of calculus problems using L^AT_EX. Project STACK [3, 4] is a system for computer aided assessment (CAA) and teaching system for mathematics using Maxima, where the authors had particular attention to the teacher's needs in authoring questions of their own. The 'mathbook' project [9] uses T_EX to read a database of math exercises for a special class of students as well as providing solutions and instructions for teachers. Using the R language, the system described in [2] permits the automatic generation of standardized statistical exams which is especially useful for large-scale exams. It's important to mention works like [8] where exercises are generated according to an algorithm and not by an author.

The authors of this paper teach first-years mathematics at University of Aveiro, in Portugal, to a broad spectrum of courses, like tourism, finance, engineering and sciences, each of them requiring different levels of mathematics on their curricula.

The university entrants for these courses undertook different studies in mathematics, and some of them elected not to undertake any mathematical studies at all in the final years of their secondary

schooling. Usually the teaching method consists on presenting and explaining the material to be learned followed by solving specific application exercises. Assessment is again done with exercises, frequently of the “same type” or using generalizations from standard class exercises.

This context inspired us to produce a tool to improve the following issues:

- Students with different backgrounds: each course curricula is designed considering some mathematics prerequisites. However, students that apply for those courses do not always fulfill them. There are students with different backgrounds thus requiring some subjects to be presented in a different way: starting from simple concepts to more complex ones. Usually teachers suggest these students the study of previous topics on their own and there are also preparation courses available. This issue has been studied in several countries [7] where an analysis of the effect of “remedial programmes” is presented. The tool we are developing could help teachers to prepare a “remedial programme” to students with a defective mathematics preparation. These students usually need more practice of the same type of exercises. This is an advantage of parameterized exercises implemented in MEGUA: a student can practice until he feels comfortable, because generating different exercises is quite immediate.
- Detailed answers: we believe that presenting full, detailed answer of an exercise, not just a short summary or numerical solution, is of utmost importance for first years’ students. This is in accordance to the study in [5] where we can read: “This article demonstrates the feasibility and effectiveness of teaching several mathematics skills by presenting students with carefully chosen sequences of worked-out examples and problems – without lectures or other direct instruction.”
- Viewpoints and misconceptions: during a lecture a teacher may become aware of new difficulties preventing his students to understand what he is teaching, so he has to put up strategies to overcome these difficulties. The possibility of changing a mathematical text is in accordance, for example, with [6] that warns that “the presentation of mathematics material (...) could contribute to creation and strengthening of students misconceptions”.
- Variability and concepts: a concept (property, theorem, application and so on) can be understood by a person but its application to a variety of situations must be practiced. From the same exercise model one can derive other instances to cover different aspects of the same problem.

We developed MEGUA, a software package for Sage computer algebra system, because Sage concentrates many mathematical packages on a single tool available through prepared Python language. Typesetting of mathematical notation is done with \LaTeX . Summarizing, to use MEGUA package an author needs knowledge of \LaTeX and basic Python programming. Sage functionalities can be easily found in documentation for basic mathematical subjects.

The following listing is an example of an exercise template in \LaTeX :

```
%PROBLEM Some Name Here
What is the primitive of  $\int a^x + \ln(x)$  ?
%ANSWER
We can present several steps and result is
```

$s.\text{prim} + C$ for $s \in \mathbb{R}$.

LISTING 1.1. Exercise template in L^AT_EX

This example has two parameters ‘ap’ and ‘bp’, this one decorated with ‘bp@()’ to produce ‘(bp)’ if value is negative. Then, using a Python declaration like

```
class E28E28_primdirect_001(Exercise):
    def make_random(s):
        s.ap = ZZ.random_element(-4,4)
        s.bp = ZZ.random_element(-4,4)
    def solve(s):
        x = SR.var('x')
        s.prim = integrate(s.ap * x + s.bp)
```

LISTING 1.2. Python for the Exercise template

An exercise has a name, like ‘E28E28_primdirect_001’, and two functions ‘make_random’ and ‘solve’: the first creates a random exercise and the second evaluates values for the solving details. Sage specific functions are: `ZZ.random_element`, `SR.var` and `integrate`, where the first generates integer random elements, the second creates a symbolic variable and the third integrates the function.

The following section presents more details about MEGUA design options and uses. We conclude with a brief summary of case studies and possible future directions.

2 MEGUA package

The following are MEGUA design implementation options:

- **Currently for pen-and-paper work**

We design this package to help the traditional pen-and-paper study by allowing a student to read detailed and complete answers to problems. It wasn’t given priority to computer evaluation of a student answer.

- **Authors can create and edit one or more local databases**

One user can create several databases of exercises, for example, one for each subject he teaches. On the other hand, this separation allows an author to decide what he wants to share with colleagues.

- **Several output formats**

MEGUA is designed to export selected exercises to different templates. Currently, it produces L^AT_EX files (and the corresponding PDF files) from a set of exercises, with or without solution, and an HTML index of database contents.

- **Classification**

Each exercise developed must be classified. It has a summary containing a description of the objectives and its mathematical classification (we are using MSC identifiers). Besides this, it

has a label in format ‘section; subsection; subsubsection’ to be placed in a booklet as well as to be referenced in the index of the database it belongs. Finally, the exercise has a suggestive name (rather than an exercise number) so that students can easily remember it.

- **Designed for translations**

When an author produces an exercise, he is writing L^AT_EX text and also producing Sage/Python code; all this information is stored as utf8 strings in a database. The exercise text (problem and answer) is separated from the programming part that determines how variables are replaced by values (or formulas), thus, if we want to use the same exercise in another language, we only need to translate the text since the programming code is the same.

- **Operating System**

For the time being, stable versions of Sage run on a Linux machine (stand alone or server) using the Web browser or the command line. There are virtualizations for running Sage in MS-Windows machines. The MEGUA package is ready to work in a command line shell or with a Web browser through Sage Notebook.

- **Team work**

Sage Notebook provides a natural way to work over the Web for team work where users are able to share worksheets. We are using a worksheet for each exercise template for better organization. Using it via Linux command line one has means well known to programmers: we keep each text file as an exercise template, everything organized in directories and a versioning software ([14]) can be used to share work and keep records of changes.

2.1 Session example

We start by opening a new file (or new worksheet on notebook). It is necessary to declare the package

```
from megua.all import *
meg = MegBook('/home/pedro/calc-exercises.sqlite')
```

LISTING 1.3. MEGUA package declaration

The second line opens or creates a database ‘calc-exercises.sqlite’ at ‘/home/pedro’ directory. Then, below in the file (or in another cell of the notebook) we declare the exercise:

```
txt = r'''
%SUMMARY Matrices; Types of Matrices
About matrix types and transposition.
%PROBLEM Type of a matrix
Which is the order of the matrix given by:
    $$ A = matrixA ? $$
Write down its transpose.
%ANSWER
Matrix $A$ has $nr$ rows and $nc$ columns so its order
is $nr$ \bftimes $nc$. Transposing a matrix consists on
using rows from the first matrix as columns in the
```

second matrix, for instance, column 1 in AA will be row 1 in the transpose of AA . The transpose is a matrix with nr columns and nc rows, so its order

```
is $nc \times nr$
$$ matrixATranpose $$
```

```
class E12X34_matrices_001(Exercise):
    def make_random(s):
        s.nr = ZZ.random_element(2,5)
        s.nc = ZZ.random_element(2,5)
        s.matrixA = random_matrix(ZZ, s.nr, s.nc)
    def solve(s):
        s.matrixATranspose = s.someMatrix.transpose()
'''
meg.save(txt)
```

LISTING 1.4. Full template

The `txt=r'''` part defines a string in which we write all \LaTeX code and Python class. MEGUA parses this `txt` string and tests its content. Afterwards, if errors are not encountered, it is stored into the database using `save`.

In the example, the parameters used in the text part are ‘matrixA’, ‘matrixATranspose’, ‘nr’ and ‘nc’. These parameters are going to be replaced by values defined in ‘make_random’ and ‘solve’. The advantage of separating parameters instantiation in two parts is that one can assign it’s own set of values to parameters, and for that set, request the evaluation of output values. This topic is described in more detail in the next section.

2.2 Parameters

As remarked in the previous section, in the exercise text one uses variables that are to be replaced by numeric values or formulas. These variables can be letters or letters and numbers, like ‘x11’ or ‘yslope’.

Some rules of transformation were added in MEGUA, allowing that the variable assignments are written in a specific desired format. To illustrate this we’ll present some examples.

Suppose variable `name1` has the assignment `-12.123456`. If, within the exercise the variable is called, it will appear with different formats, depending on the rule we use:

parameter	after substitution	comment
<code>name1</code>	<code>-12.123456</code>	straight replacement
<code>name1@()</code>	<code>(-12.123456)</code>	if negative use <code>(·)</code>
<code>name1@f(2.3g)</code>	<code>-12.1</code>	use of <code>printf C</code> language notation

Also some functions defined on Sage may be used to create new variables, as `name2@s\{sin\}` for example. Suppose variable `name2` has the assignment `-34.32`. Then `name2@s\{sin\}` will return `0.42857465435` (`sin(name2)`)

Another functionality implemented was what we call decision variables, for instance,

`name3@{ "John", "Peter" }`. These variables are used to generate different texts in the same exercise.

If `name3=0` the expression `name3@{ "Jonh", "Peter" }` will be replaced by `John`; if `name3=1` the expression will be replaced by `Peter`.

2.3 Using exercises

MEGUA implements these possibilities:

- Produces a \LaTeX file for a single exercise instantiated with some random set of values. This can be done with ‘new’ command:
`meg.new('Exercise_Key')`
- Produces a booklet from a selected set of exercises.
- Produces exams, even different exams for each student, with a selection of parameterized exercises.

All these tools can export to a \LaTeX file to be modified and compiled elsewhere.

2.4 Creating booklets or exams

Having a database of exercises allows us to select exercises and produce a \LaTeX document, using a template like Listing 1.5.

```

\documentclass{ article }
\begin{document}
{{ put_here("E26B05_polynomial_001") }}
{{ put_here("E26B05_exponential_001") }}
{{ put_here("E26B05_powers_001") }}
{{ put_here("E26B05_logarithm_001") }}
{{ put_here("E26B05_rational_001") }}
\end{document}

```

LISTING 1.5. Booklet template

In this example, five exercises will be written on a work file (worksheets, exams, homework, ...), using random values. What is exposed from each exercise can also be controlled: only the problem text, problem and answer and other details. It is also possible to specify particular values for each exercise.

3 Conclusion

Students can be engaged on homeworks if they know they will find the detailed answers later as a mean to correct their mathematical reasoning. This mechanism was used in a few courses at University of Aveiro, and students appreciation was very positive. Some of them explicitly stated that it was a great help on their preparation for exams and during assessments improved answers were found.

We know that there are other software tools, like STACK [4], that have achieved this same results but, to our best knowledge, this is the first candidate to develop a tool for Sage that allows

the creation of author database(s) of parameterized exercises using Sage directly in command line or notebook mode.

This tool also proved to be useful to teachers since exercises and their detailed answers are available online inside campus, thus, permitting a rapid preparation of study materials.

Future work: Nowadays we are testing other input/output formats to export files either for learning platforms like Moodle or HTML, using MathJAX to represent mathematical formulae.

Also new exercise templates are being thought, as multiple choice, true/false and short answer, to be included in computer assessment systems for mathematics and science in general (like [4, 13]).

Regarding the exercises themselves, and its construction, we are introducing new random elements either in text or in mathematical functions, using decision variables, thus avoiding repetition of the same model of exercise.

The enlargement of our databases of exercises is a concern to the authors: more users, more topics covered.

We also plan to embed MEGUA in Sage Notebook so that a student can call for an exercise inside a notebook worksheet and solve it as a homework or an exam.

Acknowledgements

This work was partially supported by *FEDER* funds through *COMPETE*–Operational Programme Factors of Competitiveness and by Portuguese funds through the *Center for Research and Development in Mathematics and Applications* (University of Aveiro) and FCT (Portuguese Foundation for Science and Technology) within project PEst-C/MAT/UI4106/2011 with *COMPETE* number FCOMP-01-0124-FEDER-022690.

The authors would like to thank Mathematics Department, the STIC (Services for Technologies of Information and Communication) and to the Sage group.

References

- [1] Kalle Saastamoinen, Jaakko Ketola, Tuukka Kurppa and Liisa Torikka, *Enabling Web access to a database of calculus problems using L^AT_EX, PHP and L^AT_EX₂HTML*, TUGboat Proceedings of the 2003 Annual Meeting 24(1), 2003.
- [2] B. Grn B and A. Zeileis, *Automatic Generation of Exams in R*, Journal of Statistical Software 29(10) (2009), 1–14.
- [3] C. J. Sangwin and M. J. Grove, *STACK: addressing the needs of the “neglected learners”*, Proceedings of the WebAlt Conference, 2005.
- [4] C. J. Sangwin, *Assessing elementary algebra with STACK*, International Journal of Mathematical Education in Science and Technology 38(8) (2007), 987–1002.
- [5] Xinming Zhu and Herbert A. Simon, *Learning Mathematics From Examples and by Doing*, Cognition and Instruction 4(3) (1987), 137-166.
- [6] Ann Kajałndera and Miroslav Lovrićb, *Mathematics textbooks and their potential role in supporting misconceptions*, International Journal of Mathematical Education in Science and Technology 40(2) (2009), 173–181.

- [7] Cristina Varsavskya, *Chances of success in and engagement with mathematics for students who enter university with a weak mathematics background*, International Journal of Mathematical Education in Science and Technology 41(8) (2010), 1037–1049.
- [8] Ana Paula Tomás, José Paulo Leal and Marcos Aurélio Domingues, *A Web Application for Mathematics Education*, Conference ‘Advances in Web Based Learning - ICWL 2007’, United Kingdom. Lecture Notes in Computer Science 4823, Springer-Verlag, 2008.
- [9] Pter Szab and Andrs Hrask, *Managing a math exercise database with L^AT_EX*, TUGboat, Proceedings of EuroTEX 2006, 27(1), 2006.
- [10] W. A. Stein et al., *Sage Mathematics Software (Version 4.8.0)*, The Sage Development Team, 2012, <http://www.sagemath.org>.
- [11] Paula Oliveira and Sabrina Silva, *An overview of PmatE: developing software for all degrees of teaching*, Proceedings of the International Conference in Mathematics Sciences and Sciences Education, June 11-14, University of Aveiro, 2006.
- [12] Documentation: <http://megua.readthedocs.org>. Code: <http://code.google.com/p/megua/>
- [13] Moodle, version 1.9 (2007) <http://moodle.org>.
- [14] Versioning on wikipedia: http://en.wikipedia.org/wiki/Software_versioning