PASSAROLA: High-Order Exercise Generation System

José João Almeida[1], Isabel Araújo[2], Irene Brito[2], Nuno Carvalho[1],
Gaspar J. Machado[2], Rui M.S. Pereira[2], Georgi Smirnov[3]
[1]Centre of Informatics, [2]Centre of Mathematics, [3]Centre of Physics
University of Minho
Braga, Portugal

Abstract—**In order to be robust and achieve multi-domain coverage, exercise generation systems usually work with answers of simple types (*e.g.* multiple-choice, Boolean, integer, or file comparison). In this paper we describe an exercise generation system *PASSAROLA*, a simple, yet powerful, language that anyone with no computer science background, can use to develop exercises, that include a collection of heterogeneous objects, and allows the usage of complex elements. Its main characteristic features are the use of simple reusable templates, simple and rich types, rich notation and syntax (LaTeX based) for questions, solutions, and answers, transformations and calculations, external calculators.**

Keywords - e-learning, exercise generation, randomization.

I. INTRODUCTION

When implementing the Bologna education reform in Portuguese universities, the number of contact hours of the courses decreased (considerably in some cases), therefore increasing the need of a more self-responsible learning by the student. This means that the student has to work by himself (*i.e.* outside lectures and examples classes) on a regular basis. In practice this implies that the lecturer is supposed to plan the students' work, in principle weekly basis [1]. We present an intelligent interactive system called *PASSAROLA*, name inspired by the Portuguese writer Saramago on his book "Memorial do Convento", a system that allows lecturers to describe exercises that are composed together to build complete exams (including step-by-step solution and assessment). The system surpasses the multiple choice questions assignments most e-learning environments are limited to, allowing the creation of more complex exercises in completely different domains (such as Music, Maths, Physics, and Geography), using almost any kind of object (graphs, sound, source code, music score, etc). It uses a transformation system based on a universal methodology – dynamic programming – that allows the reuse of sub problems already programmed. It is a free open source system, easy to use and with a large scope of applications.

Recently, a number of powerful exercise generating systems have been proposed (see, e.g [2-8]), and several e-learning systems have been developed (see, *e.g.* [9-13]). Also, a significant number of methodological studies in e-learning were done during the last years (see, *e.g.* [1,14]). Publications on these topics are surveyed in [15,16].

This paper introduces *PASSAROLA*, and illustrates several examples of exercises created using this system for different courses (*e.g.* Music, Chemistry, and Geography). These examples also show that exercises implemented with this system, include the question, formulated in a way that the student concentrates only on the essential. It can also provide a step by step solution and validation that helps the student to identify mistakes if the answer is not correct. When properly used by lecturers it has proven to be a powerful tool for a student to auto regulate his study for a given course and indeed provide him the capability of learning to learn.

II. *PASSAROLA*

Several robust exercise generating systems (most of them distributed as proprietary software) that use templates are currently available. However, usually these systems require a specialist, or a person with a computer science background, to implement exercises. Another shortcoming of most popular systems is that the exercises can only be solved using simple type answers, like boolean or multiple choice questions. One of *PASSAROLA*'s design goals is to provide a simple, yet powerful, language that anyone with no computer science background, can use to describe exercises, that include a collection of heterogeneous objects, and allows the usage of more complex elements to provide answers, than simple boolean buttons, or input text areas.

The *PASSAROLA* project and framework was created to make it possible to build your own exercises:

- using simple reusable templates, pre-defined templates as often as possible;
- using simple and rich types;
- using rich notation and syntax (LaTeX based) for questions, solutions, and answers;
- using and defining transformation steps and calculations;
- using external calculators (functions and tools);
- create your own user defined reusable functions and tools;

- programming your own templates, processor, and evaluations if required;

In order to do that, a Domain Specific Language (DSL) *PASSAROLA* was created for implementing and modeling: (1) rich exercises generation, (2) their step-by-step resolutions and explanations, (3) evaluations, from simple to complex comparisons to specific algorithm or functions.

The main *PASSAROLA* characteristics are:

- to provide random choices from static or dynamically calculated lists, allowing to control the complexity and constrain the domains to reasonable values;
- to provide coupled choices (selection of corresponding elements from two or more lists);
- to use LaTeX as the main authoring system to build documents (exercises and tests), allowing the reuse of huge amounts of styles and pre-defined common blocks;
- to use templating techniques to simplify LaTeX creation;
- use previously computed elements as instances to describe more complex transformations and formulas, to simplify exercises, results, resolution steps, etc;
- to use external languages and tools to perform specialized transformation and calculations (*e.g.* Maxima, Perl, TikZ, and abcm2ps), this allows the use of more suitable languages for specific domains, increasing exercises expressive power and overall language efficiency;
- possibility of writing auxiliar functions in Maxima and Perl;
- the system can be used locally or via web application;
- uses table objects to describe choice lists or coupled lists, in order to simplify the exercises description;
- distributed using an open-source license, growing with user contributions and needs.

A general idea of *PASSAROLA* syntax is illustrated in the Fig. 1.

```
1  #title: Collective nouns
2  #let:
3     colect = [ team,   bed,     pack    ];
4     elem  = [ horses, oysters, coyotes ];
5     colect ~ elem;
6  #question:
7     What is the name of a set of
8     \textbf{#elem}?
9
10 #result:
11    #colect of #elem
12
13 #verification
14    #colect type=icstr;
```

Figure 1. Example using *PASSAROLA* syntax.

Comments:

- line 3: select a random colect from the list;
- line 5: colect and elem are coupled elements;
- line 7: the exercise text (in LaTeX) #elem is replaced by the random chosen value;
- line 13: evaluation section;
- line 14: #colect type=icstr – the validation function is a ignore-case string comparison of the value submitted with the value of colect.

In this example there is no suggestion, resolution or calculations, no user provided functions, etc.

III. EXERCISE GENERATION PROCESS

To explain the essentials of the exercise generation process let us consider a quadratic equation

$$ax^2 + bx + c = 0,$$

where the roots of this equation are given by the formula

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

It is clear that a random generation of the coefficients a, b, and c, with high probability will give us an equation without rational roots and such an exercise is not suitable for middle-school students. A pretty exercise can be generated making use of the Viéte theorem. Namely, we randomly generate a rational coefficient a and rational roots x_1 and x_2, *i.e.* the answer, and calculate the coefficients b and c as follows:

$$a(x - x_1)(x - x_2) = ax^2 - a(x_1 + x_2)x + ax_1x_2$$

$$= ax^2 + bx + c = 0.$$

Determining the discriminant $D = b^2 - 4ac$ we also generate the solution to the exercise. In general, a solution to an exercise (not necessarily mathematical one) can be represented as a sequence of "operators" (logical steps) A_k applied to data "vectors" x_k in order to deduce the next data vector $x_{k+1} = A_k x_k$. Schematically the solution can be represented as the following diagram:

$$x_0 \to x_1 = A_1 x_0 \to \cdots \to$$

$$x_{k+1} = A_{k+1} x_k \to \cdots \to x_n = A_n x_{n-1}.$$

If the operator A_k is "invertible", then the vector x_{k-1} can be reconstructed from the vector x_k. The exercise generation usually starts from an intermediate data vector x_k. Moving to the left we generate the exercise and moving to the right we obtain the answer:

$$x_0 \leftarrow x_1 = A_1 x_0 \leftarrow \cdots \leftarrow$$

$$x_{k+1} = A_{k+1}x_k \to \cdots \to x_n = A_n x_{n-1}.$$

The whole chain gives us the solution to the exercise. A natural question arises: "Is it possible to treat non mathematical exercises in the same manner?" The answer is: "Yes!" Moreover, from the logical point of view there is no difference between exercise generation process in "exact" and other sciences. To clarify this point, consider a simple example. Assume that the information we use to generate exercises is collected in various databases. For example, consider the relations

$$\text{Rivers(name, length, tributaries, towns)}$$

and

$$\text{Towns(name, population, drainage basin)}$$

(instances of these relations are given in Table 1 and Table 2).

TABLE I. RIVER INSTANCE EXAMPLE

name	length	tributaries	base level
Tagus	1038km	Zêzere, etc.	Lisbon

TABLE II. TOWN INSTANCE EXAMPLE

name	population	drainage level
Covilhã	51797	Zêzere

The sequence of logical steps

Covilhã←Zêzere→Tagus→Lisbon,

obtained from the above relations can be used to formulate a question like: "Where precipitation fallen in Covilhã reaches the Atlantic Ocean?" The answer is "In Lisbon" and the solution is "Covilhã belongs to the Zêzere drainage basin. Zêzere is Tagus' tributary. Tagus' base level is in Lisbon."
This example illustrates that the exercise generation principles coming from mathematics, can also be used in other academic areas. Provided that the information is represented in an adequate way this allows the identification of logical relations between knowledge elements.

IV. CASE STUDIES

In this section we present a few examples illustrating the features of *PASSAROLA* in different domains of knowledge.

A. Providing a Great Variety of Generated Texts

Automatic generation of different exercises means not only to vary the problem numerical data, but also the text of the exercise in order to stimulate the abstract thinking skills of the students. The following simple example illustrates how this can be done using *PASSAROLA*:
A bag contains 30 blue balls. How many yellow balls should be introduced in the bag, so that the probability of extracting a blue ball is $\frac{5}{8}$*?*

In this exercise the words *ball*, *yellow*, and *blue* were randomly chosen from the lists presented in Fig. 2 and Fig. 3.

```
s L1=[balls, chips, raffles, gums];
```

Figure 2. List of objects.

```
s L2[2]=[yellow, blue, white, brown, red,
green, black];
```

Figure 3. List of colors.

B. Generating Exercises With Controlled Graphics

For some exercises it is useful to illustrate the solution of a problem using a controlled graphic which varies with certain parameters. As an example, consider the following exercise from Mechanics:

A projectile is launched at an angle of a degrees to the horizontal, with velocity v m/s, from a height h m (see Fig. 4). Calculate the time of flight, the maximum height, and the total horizontal distance traveled by the projectile (neglecting air resistance and considering the origin of the coordinate system at the initial position of the center of the ball).

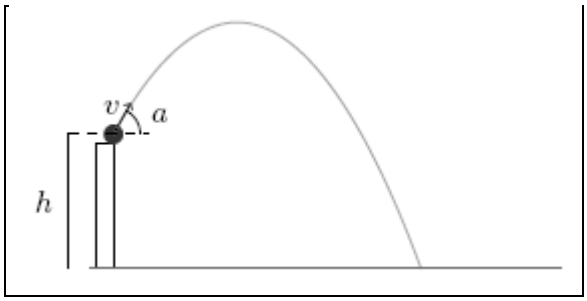

Figure 4. Trajectory of the projectile.

In this exercise, it is interesting to see how the given parameters a, v, and h, will influence the trajectory of the projectile, which in this case is a parabola, determined from the equations of motion: $x = v\cos(at)$, $y = v\sin(at) - 0.5gt^2$, where g is the gravitational constant. Also, a realistic graphic (of course, with an adapted scale $1:100$) can be helpful, in the sense that one can have an idea about the expected solution before solving the exercise (*e.g.* just by observing and comparing lengths) and judge later if the obtained solution is plausible. For these reasons it makes all sense to create a dynamic graphic having a different aspect each time the exercise is generated. This can be done using the TikZ package, which allows the generation of graphics in LaTeX. Here we explain only the main idea of this process, in particular how to draw the trajectory of the projectile varying with the sorted parameters. First of all, one establishes variation intervals for the parameters, here assumed to be integer-valued: $v = [1..10]$, $a = [0..75]$, $h = [1..5]$. Then using the equations of motion, the coordinates of the points are calculated where the projectile reaches its maximum height, (x, y), and the maximum distance, (w, z). The trajectory of the projectile, a parabola, can then be drawn in TikZ using the command given in Fig. 5, indicating the starting point (0,0)

the vertex (x, y) and the final point (w, z), where the projectile finishes its trajectory. Note that the operator # is used to substitute the calculated values of the coordinates in the TikZ instructions. Other varying elements of the graphic can be constructed in the same way.

```
\draw (0,0) parabola bend (#x,#y) (#w,#z);
```

Figure 5. Tikz command.

C. Music: Chord Recognition

In order to generate music data (staves and audio) we use ABC music notation [17]. In Fig. 6 it is presented a sample of ABC notation and in Fig. 7 the corresponding result.

```
M: 2/4          % time signature
K: C
L:1/8           % base time unit
CE Gz | [C4E4G4] |]
```

Figure 6. ABC example.

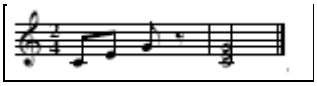

Figure 7. ABC result.

In this way, we can define *PASSAROLA* chord recognition exercise given in Fig. 8.

```
1  #let:
2   a[3] = [[C4E4],[C4G4],[E4G4],[C4~F4],[C4E4G4]];
3   b    = [ 1, 2, 3 ];
4   c    = [ #a[0], #a[1], #a[2]];
5   b ~ c;
6   commonabc = {{% all score except the final chord
7     M: 2/4
8     K: C
9     L: 1/8
10    CE Gz| }};
11
12 #question:
13 Double click the red note and choose the
14 correspondent score.
15 \inline_abc[width=0.01mm,midi]{ #commonabc #c|]}
16
17 1) \inline_abc[scale=0.5]{ #commonabc #a[0]|]}
18 2) \inline_abc[scale=0.5]{ #commonabc #a[1]|]}
19 3) \inline_abc[scale=0.5]{ #commonabc #a[2]|]}
20
21 #resolution:
22 the correspondent score is #b
23 \inline_abc[scale=0.5]{ #headabc #c|] }
```

Figure 8. *PASSAROLA* code for chord recognition.

Comments:

- line 2: select randomly three chords;
- lines 3 to 5: choose a chord and save its number;
- line 15: define a score with an invisible size of 0.01mm and a midi attachment red note;
- lines 17 to 19: show the three previously selected chords.

This *PASSAROLA* module specification generates the exercise in Fig. 9.

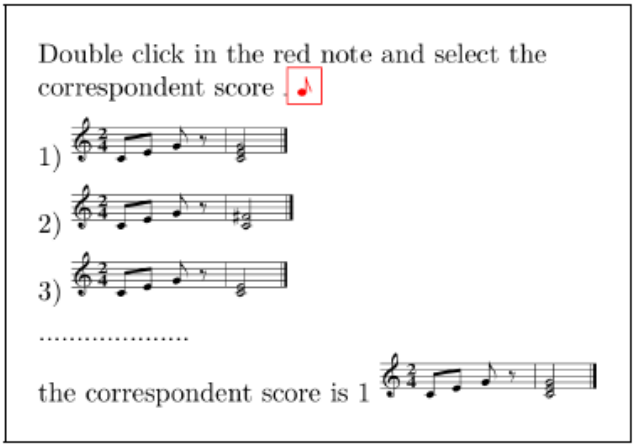

Figure 9. *PASSAROLA* exercise generated for chord recognition.

As we can see, it is possible to build exercises with audio and images derived from a random selection of the textual description of chords and music. In order to do this, we just need a textual syntax to define the final objects and tools that translate and include them in LaTeX.

V. CONCLUSION

In this work we presented an exercise generation system based on a DSL language. Some decisions taken in the DSL design process, proved to be very important. Namely, *PASSAROLA* provides:

- division of a complex problem in several sub-problems, reusing parts;
- possibility of embedding a language code (for example, Perl and Maxima) inheriting a large amount of modules and functions, and providing a strong expressive power for user specific functions;
- a very powerful way of building a large variety of exercises via LaTeX -based templates (LaTeX has a very rich set of packages that interfaces with large set of external tools like gnuplot, graphviz, and ABC);
- table support and specific table importing notation to deal with tuples lines, providing a separation between processing and data, and constituting a mechanism for contributions of non expert users (it is easy to add one more line to a table).

Some components of *PASSAROLA* were not covered in this document.

ACKNOWLEDGMENT

This research was financed by FEDER Funds through Programa Operacional Factores de Competitividade - COMPETE and by Portuguese Funds through FCT – Fundação para a Ciência e a Tecnologia, within the Projects PEst-C/MAT/UI0013/2011 and PEst-OE/EEI/UI0752/2011.

REFERENCES

[1] R. M. S. Pereira, I. Brito, G. J. Machado, T. Malheiro, E. Vaz, M. Flores, J. Figueiredo, P. Pereira, and A. Jesus, New e-learning objects for the Mathematics courses from Engineering degrees: Design and

Implementation of Question Banks in Maple T.A. using LaTeX. International Journal of Education and Information Tecnologies, Vol. 4, pp. 7–14, 2010.

[2] M. V. Belmonte, E. Guzmán, L. Mandow, E. Millán, and J. L. Pérez de la Cruz, Automatic Generation of Problems in Web-based Tutors, in Jain, L. C., Howlett, R. J., Ichalkaranje, N. S. & Tonfoni, G. (eds.), Virtual Environments for Teaching & Learning. World Scientific Publishing, Series on Innovative Intelligence 1, pp. 237–281, 2002.

[3] A. Gurtovoi, A. Guerman, and C. Santos, Sistema de Ensino Baseado no Computador: Gerador Automático dos Exercícios. In Proceedings of the Engenharia'2005 – Inovação e Desenvolvimento Conference. Covilhã, 21–23 November, 2005.

[4] A. P. Tomás and J. P. Leal, A CPL-based tool for Computer aided generation and solving of Maths exercises, in Dhal, V. & Wadler, P. (eds.), Proceedings of the Fifth International Symposium on Pratical Aspects of Declarative Languages, PADL'2003. Springer-Verlag, lecture notes in Computer Science 2562, pp 223–240, 2003.

[5] A. P. Tomás, N. Moreira, and N. Pereira, Designing a Solver for Arithmetic Constraints to Support Education in Mathematics, in Artificial Intelligence Applications and Innovations, AIAI 2006. Springer and IFIP, IFIP Series, Vol. 204, pp. 433–441, 2006.

[6] P. Oliveira, Gerador de Exercícios de Matemática. VI encontro AFI integrated in XII Encontro de Professores. University of Aveiro. 27th–29th October, 2011.

[7] C. J. Sangwin, Assessing higher mathematical skills using computer algebra marking through AIM. In Proceedings of the Engineering Mathematics and Applications Conference (EMAC03, Sydney, Australia), pp. 229--234, 2003.

[8] C. J. Sangwin and M. J. Grove, STACK: addressing the needs of the "neglected learners". In Proceedings of the First WebALT Conference and Exhibition January 5-6, Technical University of Eindhoven, Netherlands, pp- 81–95. Oy WebALT Inc, University of Helsinki, ISBN 952- 99666-0-1, 2006.

[9] G. Dettori and D. Persico, Fostering Self-Regulated Learning through ICT. Institute for Education Technology-National Research Council, Italy Elissa, 2011.

[10] A. Guerman, C. Santos, R. Costa, A. Raposo, A. Mendonça, and C. Lopes, Plataforma SEMENTE: Tecnologias de Informação para o Ensino na UBI. In Proceedings of the Engenharias'07 – Inovação & Desenvolvimento Conference. Covilhã, 21–23 November, 2007.

[11] M. Siegel, S. Derry, J. B. Kim, C. Steinkuehler, J. Street, N. Canty, C. Fassnacht, K. Hewson, C. Hemlo, and R. Spiro, Promoting Teacher's Flexible Use of the Learning Sciences through case- based Problem Solving on the WWW: A Theoretical Design Approach, in Fishmann, B. & O'Connor-Divelbiss, S. (eds.), Fourth International Conference of the Learning Sciences. NJ: Erlbaum, Mahwah, pp. 273–279, 2000.

[12] A. P. Tomás, J. P. Leal, and M. A. Domingues, A Web Application for Mathematics Education, in Advances in Web Based Learning, ICWL 2007. Springer-Verlag, lecture notes in Computer Science 4823, pp. 380–391, 2007.

[13] P. Cruz, P. Oliveira, and D. Seabra, Exercise Templates with Sage. Tbilisi Mathematical Journal, in press.

[14] E. Millán, L. Descalço, G. Castillo, P. Oliveira, and S. Diogo, Using Bayesian Networks to improve knowledge assessment in Journal Computers and Education, Vol. 60, January, pp. 436–447, 2013.

[15] G. Goguadze, ActiveMath – Generation and Reuse of interactive exercises using domain reasoners and automated tutorial strategies. PhD Thesis, Universität des Saarlandes, 2011.

[16] R. Queirós and J. L. Leal, A survey on e-Learning content standardization, in Information Systems, E-learning, and Knowledge Management Research, CCIS, Vol. 278, pp. 433-438, 2013.

[17] G. Gonzato, Making music with ABC plus. http://abcplus.sourceforge.net/ABCGuide, 2012.