

Methodology for the development of an application of manipulation of exercises and associated documents

Ângela Oliveira

Escola Superior de Tecnologias – Castelo Branco

E-mail: angelaoliveira@est.ipcb.pt

Nelma Moreira

DCC-FC&LIACC, Universidade do Porto

E-mail: nam@ncc.up.pt

Abstract

Through the years, teachers accumulate a considerable amount of documents to support their activity, such as drills, assignments, homework's or exams. The basic content of those documents are exercises, that normally are kept disperse and unorganized.

Computers and new Web technologies can be of valuable help in manipulating and organizing those learning resources.

In this work, first we created a methodology that it originated a language definition and create a prototype that is seats in the beginner base from E-learning, but with entry of functionalities that will be allowing teachers hold information available and organized.

The objective is to become more efficient the manipulation of exercises and exams.

XML language allows keeping structure of information with bigger similarity with real documents in opposition to the traditional databases.

Combine XML flexibility and an oriented multiplatform programming language Python, that it is easy to use, it allows an excellent specify in this type of applications.

1 - Introduction

The education of one determined substance compels to the evaluation of the same one. All the teachers if come across with innumerable periods of evaluation, which imply the accomplishment of evaluation tests.

The resource the informatics half, to teach the type of substances all, as well as long-distance education, have come to gain each time more relevance in the academic world and become necessary to adapt the traditional methods, techniques and technologies to develop not conventional and evolved applications, of form to automates how much in such a way possible the tasks of "routine".

2 – Problem definition

The Problem that if presents in education institutions is the elaboration and evaluation of tests, this process that demands to the teachers much time. The professor, a long this career goes, carrying through innumerable tests of evaluation, that although nowadays, to be stored in digital format, if find distributed by some documents, not having, many times notion of the amount and diversity of exercises that had been accumulated to the long one of the time.

He will be advantageous to use informatics solutions, that allow to congregate the material, as it relates Jamie Dinkelacker, in its article "" Placing Our Stuff So We Can Find It Later: Meta-Learning Essential"[1], and in the analogy that makes with a flock of birds in fly, showing that the coordination and existing cooperation can be used shape systems of data, using an organization in the filing-cabinets, placing them where they can easily be acceded and to eliminate the confusion created for unnecessary copies or repeated filing-cabinets, as well as the definition of names for folders and sub-folders where the filing-cabinets must be kept.

3 - Methodology

The definition of the methodology and the modulation of the information for manipulation of exercises of support to the lessons and documents of trainings and evaluation, are arrested with the reality, many times distinct in the education institutions. However all the information is centred in an only aspect the production of exercises, that allow the teacher to test the evolution and the knowledge acquired for the pupils.

Thus, the definite methodology has as starting point the exercises and is illustrated in Figure 1:

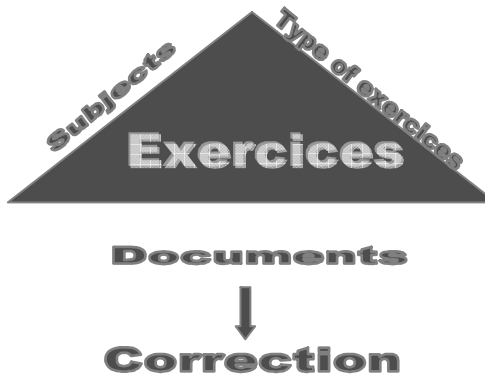


Figure 1 – Methodology

The idea of this methodology seat, in a set of exercises that must be grouped for subjects, being considered the following stages:

Subjects Definition - Define a set of subjects in accordance with the exercises to manipulate.

In accordance with the substances to teach identify a main subject and if necessary different sub-subjects.

The exercises must belong to a subject (unit of stuff), for example in accordance with the ACM Computing Classification System [2].

Each exercise must obey a classification in order to allow the definition of different levels.

Each subject could be principal node without sub-subjects, where exercises are only included, or to have a limitless structure of sub-subjects.

Example: For exercises of Probability, the main subject would be "Probabilities" and consonant the chapters where this substance if divides to consider necessary sub-subjects: Simple, Conditional Probability, etc.

The definition and classification of subjects allow the adequacy of the number of sub-subjects according with the preferences of each teacher.

Exercise Attribution Type - Given that a significant amount of information associated with the

exercises, it is necessary to organize and to distinguish clearly each type of form exercise to structuralized the relative information to each exercise.

Basis on GerExa [3] modulation had been identified the follow types of exercises, which can be divided in exercises of automatic correction:

- Multiple choice
- Multiple reply
- Correspondence of columns
- True / False
- Calculated reply

Exercises of not automatic correction:

- Development

Beyond the subject, the one that each exercise must belong, must be attributed one of the definite types.

Exercise Construction - Each type of exercises, must allow an adequate construction in accordance with its particular specifications, being defined an adequate interface to each distinct type of exercise.

In the exercises of automatic correction, it will be inserted when construction of the exercise which correct answer in the case of the exercises of multiple choice or multiple reply, the true reply in the case of True/False exercises, the result of the calculated reply and the correspondence correct of the columns, in the respective type of exercise.

Generate documents of evaluation/ support - On the basis of the existing exercises to construct the different types of documents: tests of evaluation or trainings.

This construction will must to be the most flexible, it could be:

Completely automatic - on the basis of choices done for teacher, for example in subject or subjects, or the type of exercises, the document is generated;

Attended - the choice could be done exercise by exercise;

Half-automatic - allow to generate automatically a set and to make a choice of the produced set.

Correction - the students will be able to answer the tests of evaluation and trainings, which are the only submitted documents to correction:

Automatic Correction - classification answers to questions that do not need the supervision of teacher.

Attended Correction - teacher will be able to make the correction of questions that not allow automatic correction in order to complete the evaluation of students.

4. Some aspects of Language

Based in methodology, was defined a language XML[4, 5, 6, 7], having as objective the construction of a structure that stops besides supporting the types of defined exercises, allows that they contain some elements, as they are mathematical formulas and images, for example. Thus, considering the type of information to treat, and the main characteristics of the XML, like:

- Portability
- Flexibility
- Development of flexible applications for the WEB

There are many references to languages in the same area, especially defined by IMS [8] and SCORM [9], being extensive languages, become simpler to manipulate a language elaborated to the "measure" in reality to treat.

XML Language will be described using DTDs [10], for each one of its components: Subjects, Exercises, Tests, Meta date, Teachers and Pupils.

In this article will be make reference to must important: Subjects and Exercises.

Choice of the DTD's, appears to present a form concise what it is intended for each one of documents XML, without specifying types of data, a time that all validations of types will later go be effectuate, for a programming language.

Subjects

Exercises must belong to a unit, following ACM Computing Classification System [2] specification. Each exercise must obey a classification that in defined language XML, each user determines, considering the necessary level number. The representation of the following DTD, represents the element subject:

```
<!ELEMENT node (isComposedby|isRelatedTo|hasNote)*>
<!-- node id CDATA #IMPLIED
      label CDATA #IMPLIED -->
<!-- isComposedBy (node+) -->
<!-- isRelatedTo (node+) -->
<!-- hasNote (#PCDATA) -->
<!-- hasNote type CDATA #REQUIRED -->
```

Figure 2 – Subject DTD

Exercises

Each exercise will have "id_exercício" to identify it, will have "id_node", that it represents the subject to which it belongs and it will be associated the Meta date, represented for "id_md".

It will have an optional statement that contains text, mathematical formulas, fragments of described programs in a language and/or image, and some questions. The representation of element exercise is illustrated in Figure 3:

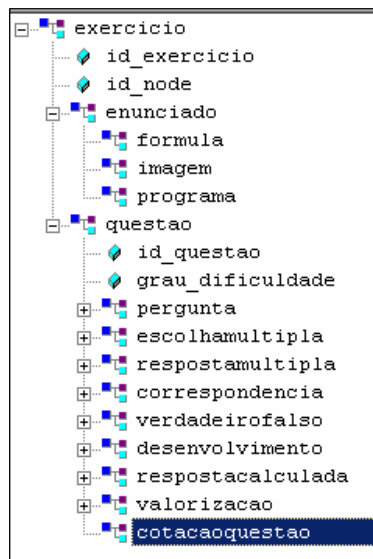


Figure 3 - Representation tree of element "exercise"

Each one of the definite types of question is dealt with separate form, as it describes after that:

Multiple Choice

For multiple choice, the minimum insertion of three options, being this configurable option, being able to contain for beyond the habitual text, mathematical formulas and/or images, having each one of the options an only identification. As an only option exists correct, was specified that the reply of this option it will have logical value one and all the others will have the logical value zero.

```
<!ELEMENT escolhamultipla (op,op,op,(op)+,(resolucao)?)>
<!ELEMENT op (#PCDATA|formula|imagem|programa)*>
<!ATTLIST op
  op_id (1|2|3|4|5) "1"
  resposta (0|1) "0"
  resolucao (0|1) #IMPLIED>
```

Figure4 - DTD Multiple Choice

Multiple Response

In this type of questions at the very least three choices can be inserted.

Also for this type of exercise, it was considered, the fact of if to have true answers and false answers, following project of the questions of multiple choice, the true answers are designated with logical value one and the false ones with logical value 0, as much to indicate an correct answer as also to facilitate the automatic correction.

This type of question is illustrated in Figure 5:

```
<!ELEMENT respostamultipla (alinea,alinea,alinea,(alinea)+,(resolucao)?)>
<!ELEMENT alinea (#PCDATA|formula|imagem|programa)*>
<!ATTLIST alinea
  op_id (1|2|3|4|5|6|7|8) "1"
  resposta (0|1) "0"
  resolucao (0|1) #IMPLIED>
```

Figure 5 - DTD Multiple Response

Columns Correspondence

That represents the questions of columns correspondence type. These questions can have at least three columns and five at maximum.

Logic functioning of this type of exercise, implies that the choice of the same number of columns 1 and columns 2, to effect the correspondence between them.

Thus, for column 2 an identification attribute was defined, so that if it can fix the numeration of a column and effect the correspondence with the values of the other.

Thus being the field resolution considered for the reply of the students, it appears only in column 1.

Following Figure 6 described it:

```

<!ELEMENT correspondencia
(col1,col1,col1,(col1)?,(col1)?,col2,col2,col2,(col2)?,(col2)?,(resolucao)?)>
<!ELEMENT col1 (#PCDATA|formula|imagem|programa)*>
<!ATTLIST col1
    col_id (1|2|3|4|5) "1"
    resposta (1|2|3|4|5) "1"
    resolucao (1|2|3|4|5) #IMPLIED>
<!ELEMENT col2 (#PCDATA|formula|imagem|programa)*>
<!ATTLIST col2
    col_id (1|2|3|4|5) "1"
    resposta (1|2|3|4|5) "1">

```

Figure 6 - DTD Columns Correspondence

Calculated Answer:

This type of questions needs the elements for solution and reply, who, will only appear in the tests. In Figure 7, they can observe the DTD:

```

<!ELEMENT respostacalculada ((respostacal)?,(resolucaocal)?)>
<!ELEMENT respostacal (resultado|(dominio)?)>
<!ELEMENT resolucaocal (#PCDATA|formula|imagem|programa)*>
<!ELEMENT resultado (#PCDATA|formula|imagem|programa)*>
<!ELEMENT dominio (aplicacao,metodo,testes)>
<!ELEMENT aplicacao (#PCDATA)>
<!ELEMENT metodo (#PCDATA)>
<!ELEMENT testes ANY>

```

Figure 7 - DTD Calculated Answer

Replay to this type of question, it can be in future, a reply produced or calculated for an application auxiliary.

For questions of evolution and true/false, specification is simpler, as it can observe in respective the DTD's, at Figure 8 and Figure 9, respectively:

```

<!ELEMENT verdadeirofalso EMPTY>
<!ATTLIST verdadeirofalso
    resposta (V|F) "F"
    resolucao (V|F) #IMPLIED>

```

Figure 8 - DTD True/False

```

<!ELEMENT desenvolvimento ((respostadesenv)?,(resolucaodesenv)?)>
<!ELEMENT respostadesenv (#PCDATA|formula|imagem|programa)*>
<!ELEMENT resolucaodesenv (#PCDATA|formula|imagem|programa)*>

```

Figure 9 - DTD Evolution

5. Some aspects of the definite Prototype

Programming language of support to the application that was chosen is the Python[11], for the reasons that if start to describe.

Characteristics of language are:

- Simple syntax;
- Easy to understand.

Python will be used in development of this application, because satisfies programming requirements and appears as an experience in use "a new" language, for WEB.

To processing XML documents exist two standard API's: DOM and SAX, [11, 12, 13, 14, 15] for beyond others. Python works efficiently with both.

Other technologies exist to manipulate XML documents directed for the characteristics of related language, in particular considered the ELEMENTTREE [16, 17].

Functionalities of Archetype

To implement any application its functionalities are necessary which. Thus prototype allows:

Teacher:

- Insert, Edit, Delete:
 - System of Classification;
 - Exercises;
- At Random or Assisted Production
 - Exercises
 - Tests of Evaluation;
- Correct and Solve Tests;

Students:

- Solve Training and Evaluation Tests.

Class Diagram

Specification of that it intends is clearer, was elaborated in UML [18], a diagram, that allows the graphical visualization of the classes to consider in this development. Diagram is a domain model, therefore is not enclosed the referring methods to each class:

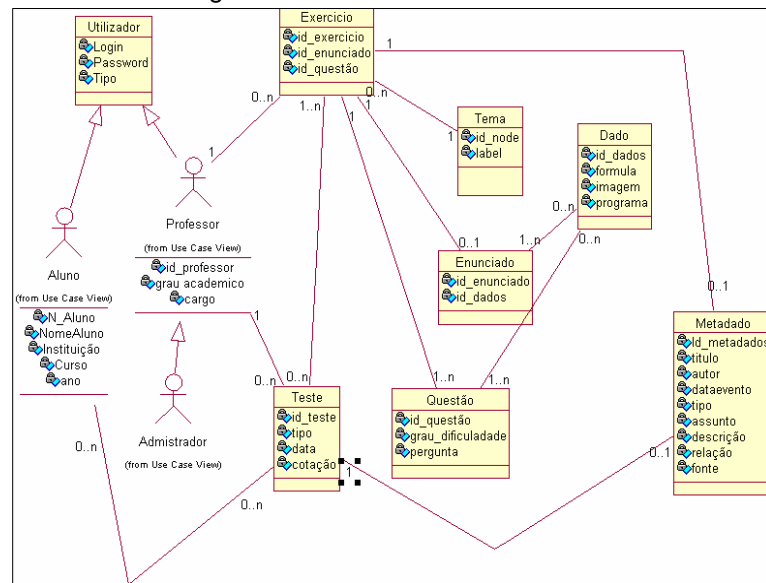


Figure 10 – Class Diagram

Subjects


Layout presents an initially empty tree, with a level of appointed top "Subjects", from which it can effect the insertion of new subjects, as it is illustrated in Figure 11:




Figure 11 – Initial Tree

This structure is similar to Explorer of a system of filing-cabinets, as Windows Explorer.

Structure in tree was constructed, appealing to a Javascript - "dtree.js" [19] already existing, to which was added the code necessary to reach intended solution.

In part of prototype that the user makes use, step following is to insert new "subjects" in tree, that can to be "subjects leaf" ("subjects" that does not contains "sub-subjects"), represented by  or

"subjects folder" ("subjects" that contain "sub-subjects"), represented by  .

In Figures 12 and 13, are illustrated the insertion of new "subjects", as "subject's folder":

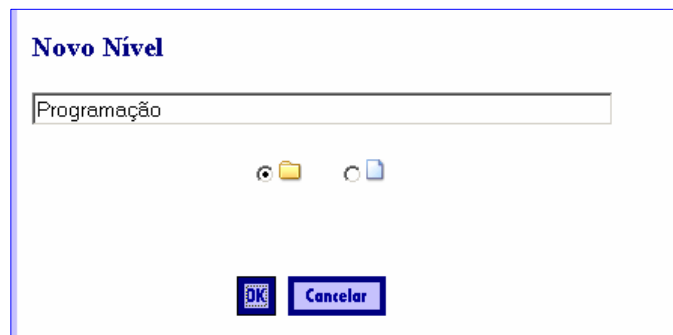


Figure 12 – Insertion Subject

Result is:

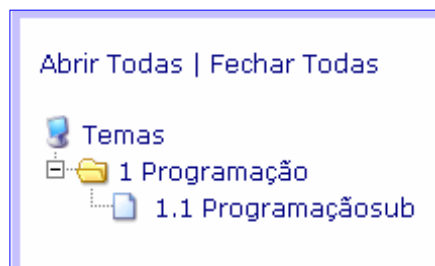


Figure 13 – Insertion Result

In insertion of a "subject folder" "sub" is created in simultaneous as "subject leaf" with name increased of suffix, or either, the concatenation of strings "name of the subject" + "sub", a time that was stipulated that levels do not exist folder, that do not have at least one "subject leaf". The name was generated for prototype and can later modify.

In following *Figure 14* an insertion, in a more complete tree, is illustrated:

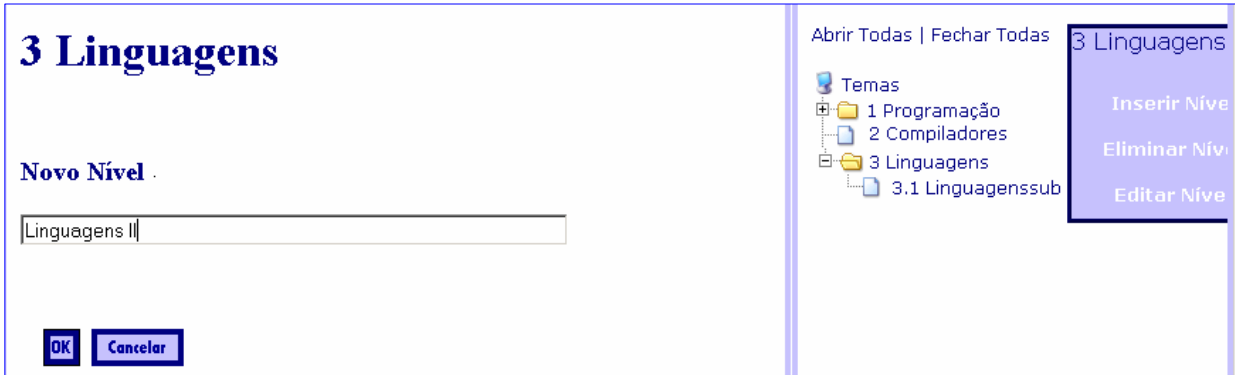


Figure 14 –Subject Insertion

Exercises

The basic functions of manipulation had been defined:

- Insertion;
- Edition;
- Elimination;
- Consulting.

To effect this manipulation a similar tree of subjects was considered, but have the operations that say respect to the exercises, as it is illustrated in *Figure 15*:



Figure 15 – Exercises Manipulation Menu

The following step will be insertion of question, as well as the respective correction. In the case of questions of multiple choices, the true reply when of the insertion of the data of the exercise, as it is illustrated in *Figure 17*, following is designated.

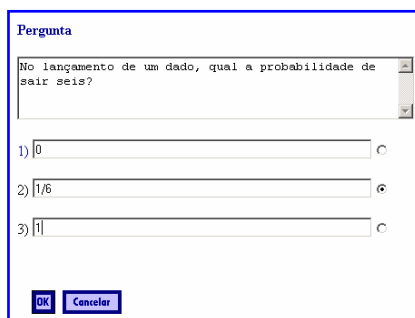


Figure 16 – Exercise Insertion

To consult all the inserted exercises, exist a list where the exercises of each subject are presented, as it illustrated in Figure 18:

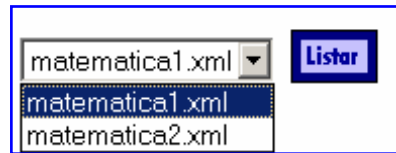


Figure 17 – Choice exercises

Result of listing in previous list will be:

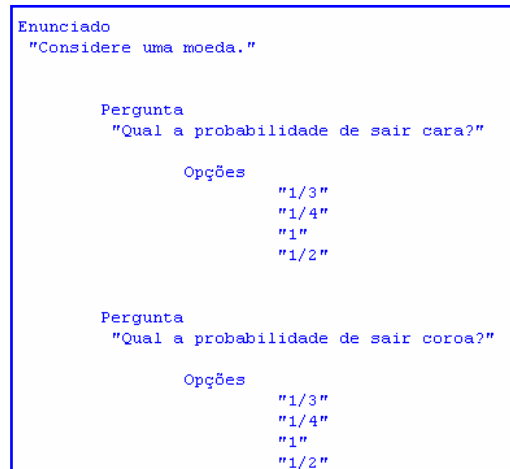


Figure 18 – Consulting Exercise

Edition operations and elimination are similar to consultation. After listing was determined what exercise it can be edited or eliminated.

6. Conclusion

With this work one intends demonstrate that methodology centered in exercises, allows definition of a particular language.

The XML prevents resource to Database Systems and shows functional, making use of inherent advantages, nominated in that it says respect to portability of documents and development of flexible applications for WEB.

References

- [1] Jamie Dinkelacker "Placing Our Stuff So We Can Find It Later: A Meta-Learning Essential", IEEE Computer Society, Learning Technology, in WWW, http://lfff.ieee.org/learn_tech/issues/october2003, October 2003.
- [2] ACM Computing Classification System, Disponível por WWW em <http://www.acm.org>, February 2005.
- [3] Oliveira, Ângela and Moreira, Nelma, "GerExa: manipulating exercises and exams in XML", III International Conference on multimédia & ICT's in Education, Cáceres, Spain, June 7-10th 2005.
- [4] Anderson, Richard et al., "Professional XML", Wrox Press, 2000
- [5] J.Ramalho, P. Henriques, "XML&XSL", FCA, 2002

- [6] Pedro Coelho, "XML – A nova Linguagem da WEB", FCA, 2000
- [7] A. Danguì, "Perspectivas sobre o Desenvolvimento de Aplicações Baseadas em Java e XML", 2003
- [8] IMS Global Learning Consortium, in WWW <http://www.imsproject.org/> September 2006.
- [9] SCORM, in WWW <http://www.adlnet.org/technologies/SCORM/index.cfm>, December 2005.
- [10] DTD, in WWW em <http://www.w3.org/TR/REC-html40/sgml/dtd.htm>, January 2006.
- [11] A. Downey, J. Elkner, C. Meyers, "How to Think Like a Computer Scientist – Learning with Python", Green Tea Press, 2002.
- [12] C.A. Jones; F. L. Drake, Jr., "Python & XML", O' Reilly, 2002.
- [13] DOM vs SAX, in WWW, <http://developerlife.com/saxvsdom/default.htm>, April 2006.
- [14] DOM & SAX, Linux Magazine, "XML processing with Python", 2001
- [15] DOM & SAX, in WWW em <http://www.devx.com/xml/Article/16922/1763/page/2>, April 2006.
- [16] ELEMENT TREE, in WWW <http://effbot.org/zone/elementtree.htm>, April 2006.
- [17] ELEMENT TREE, in WWW <http://www.xml.com/pub/a/2003/02/12/py-xml.html>, April 2006.
- [18] UNIFIED MODELING LANGUAGE (UML), in WWW <http://www.uml.org>, July 2006.
- [19] javascript, in WWW www.destroydrop.com/javascript/tree/, July de 2005.