QGM – A SYSTEM TO IMPROVE MATHS e-LEARNING

José Casimiro Pereira[1] and Joaquim Sousa Pinto[2]
[1] *Instituto Politécnico de Tomar, Quinta do Contador, 2300-313 Tomar, Portugal*

[2] *Universidade de Aveiro, Campus de Santiago, 3810-193 Aveiro, Portugal*

Keywords: Learning Objects, LOM, MathML, XML.

Abstract: The main goal of Projecto Matemática Ensino (Mathematics Teaching Project) – PmatE – is to promote the learning of mathematics. To achieve that, PmatE has been developing since 1990 an internet project especially designed to increase the interest of students in the discipline. The core of the Project anchors on the so-called "Questions Generator Model" (from the Portuguese "Modelo Gerador de Questões"). Each "model" is a tool based on parameterized expressions, which usually allows thousands of different questions to be generated (different formulations of the same "model") formally equivalent in terms of pedagogic content. The scope of this paper is to redesign the former "model" using the IEEE LOM's structure to define the learning object and redefine the model structure itself to allow for independence of the technological platform used to support the Project.

1 INTRODUCTION

The Mathematics Department of Aveiro University (Portugal) has been developing a project named *Projecto Matemática Ensino* (Mathematics Teaching Project) – P**mat**E since 1990 which aims at promoting the learning of mathematics using new information technologies. The central focus of this project is a structure called *modelo gerador de questões* (questions generator model) (Vieira, Carvalho et al., 2004).

Each *questions generator model* (QGM) is a piece of software that generates questions about a predefined theme, based upon scientific and learning objectives. These questions are randomly generated with parameterized expressions; the parameters depend on the age and educational level of their users. For each question, there are at least four possible answers, again randomly chosen among a set of possible answers.

Matching a question to an answer a true or false statement is obtained. All questions may be true or false. The user's task is to answer correctly to each statement. The QGM's may be organized in "competition mode" or in "evaluation and learning mode". In the competition mode, a collaborative competition is suggested. In the learning and evaluation mode, the cooperation among schoolmates and teacher is an intrinsic aspect of the process (Isidro, Pinto et al., 2005). Whatever the mode, the evaluations to be performed by the students will use, approximately, 20 models.

The randomness of the QGM's is the key of this project. In fact, for the same QGM thousands of different formulations with the same pedagogical and scientific value can be formulated. This feature enables two computers running side-by-side to have different questions and a different set of possible answers while using the same QGM. Combining the previous statement with the fact that the time to perform the evaluation is restricted, makes almost impossible for two students to practise any form of plagiarism (Pinto, Oliveira et al., 2007).

To materialise this project, P**mat**E has continuously developed an information system that offers a set of contents related with the several disciplines of Mathematics. Portuguese and Biology QGM's are also available recently. These learning materials are, however, prisoners of the technology used in their creation. They are all coded in Visual Basic 6.0 and stored in ActiveX dynamic link libraries (dll's). The content of these libraries are used to generate the questions that will be presented in a browser, using the ASP technology (Isidro, Fernandes et al., 2003). Such technological

Pereira J. and Sousa Pinto J. (2009).
QGM – A SYSTEM TO IMPROVE MATHS e-LEARNING.
In *Proceedings of the First International Conference on Computer Supported Education*, pages 259-265
DOI: 10.5220/0001976302590265

dependency implies that sharing or reusing QGM's is very difficult, if not impossible.

The next two pictures show the system look and feel. They represent two distinct formulations of the QGM with ID 76.

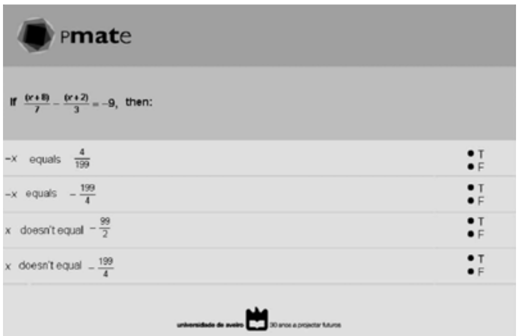

Figure 1: Using QGM number 76 (Pinto, Oliveira et al., 2007).

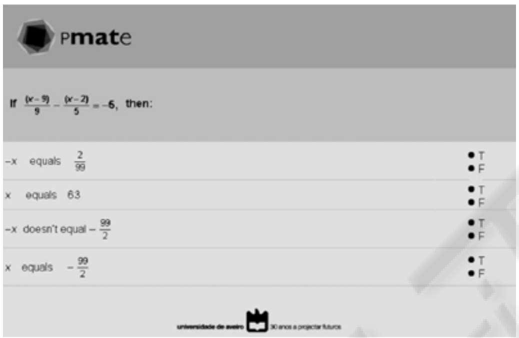

Figure 2: Another formulation of QGM number 76 (Pinto, Oliveira et al., 2007).

The pictures above show the power of the QGM's. To make it possible for the QGM's to be shared throughout Portuguese and international networks of Learning Objects, P**mat**E is promoting their adaptation. To do that, we are dealing with two distinct concerns: the QGM's must become independent from the technological platform and must comply with the international standards used to describe Learning Objects.

In this article, a description is given of the efforts made to transform the QGM's. First, we present the QGM LOM structure and the arrangements made to adapt it. Then we present the new QGM structure and syntax and finally we present the efforts done to test it.

2 LEARNING OBJECTS

There is no consensus on the definition of Learning Objects (LO) (Daniel and Mohan, 2004). Therefore, we will use the IEEE definition: "a learning object is defined as any entity, digital or non-digital, that may be used for learning, education or training" (IEEE, 2002).

According to that definition, we can consider that the QGM's defined and built by P**mat**E are suitable to be viewed as *learning objects*.

2.1 Learning Objects Metadata (LOM)

Before we start, just a few introductory questions: What is a Learning Object? In a simple way, a Learning Object is the conjunction of metadata about a learning object and the learning object itself. What is metadata? Metadata is knowledge about knowledge (Taylor, 2003)) or information about information. Why is metadata so important? The major reason is that with such information we are able to understand the usefulness of a learning object.

In 15 July 2002, the IEEE published the final "Draft Standard About Learning Objects Metadata" (IEEE, 2002). This was the end of a long journey that started in 1997 with the efforts of the IMS Project, a non-profit consortium of US institutions of higher education (Consortium, 2006). They aimed to develop a specification for learning content metadata and an open market-based standard for online learning. One year later, IMS and ARIADNE (Alliance of Remote Instructional Authoring and Distribution Networks for Europe) proposed together to the IEEE, a new proposal of the Learning Objects Metadata. After four years of development, the final release was published (IEEE, 2002). This final release proposed a structure to represent the metadata about an LO. This strongly hierarchical XML-based structure has nine sessions:

- *General* – describes the general information of the LO;
- *Lifecycle* – specifies the actual state of the LO and its evolution;
- *Meta-Metadata* – specifies the knowledge about the LOM, not about the LO;
- *Technical* – describes the technical requirements and technical characteristics of LO;
- *Educational* – describes the pedagogical and educational characteristics of the LO;

- *Rights* – describes the copyrights and terms of use of the LO;
- *Relation* – describes the kind of links that this LO could have with other LOs;
- *Annotation* – in this session comments on the educational use of the LO can be provided;
- *Classification* – specifies the classification system of the LO, if any.

Together they form the *LOMv1.0 Base Schema*.

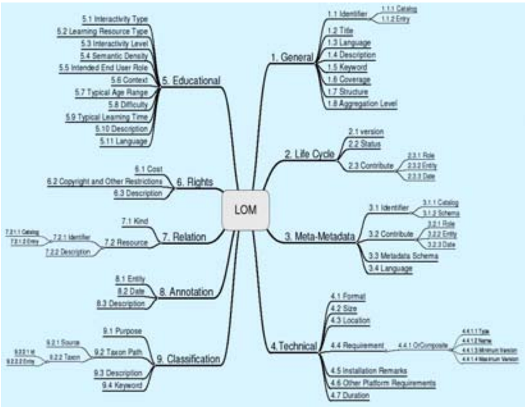

Figure 3: The LOM structure (Barker 2005).

Every element has sub-elements, and several of them have, again, new sub-elements. All the elements have their predefined data type. There are two kinds of data types: aggregate data elements and simple data elements. Simple data elements have individual values, defined by scope and data type. For aggregate data elements, *LOM v1.0 Base Schema* does not define individual values.

Whenever possible, the *LOM v1.0 Base Schem*a restricts the values a user can choose to a predefined list. This is useful to increase the interoperability of the LOM within several systems.

For IEEE, LOM can be defined as *strictly conforming* if it complies with all the rules and has no extended elements or, as *conforming*, if it has extended elements. Extended elements should have their own *namespace*, instead of using the LOM's *namespace*.

2.2 The LOM Models

As mentioned above, the QGMs are captive of the technological solution that had made them grow. Bearing that in mind, P**mat**E started to redefine the structure of the QGMs. This new model follows the Learning Object architecture: the *QGM* has two distinct sections:

- the metadata about the QGM, described using the LOM structure;

and

- the QGM itself.

The metadata section will contain the necessary to process information to allow compliance with the LOM's structure. Almost every LOM elements will be used.

The QGM internal structure will be completely rebuilt. The new structure is one of the scopes of this article and will be further discussed in Session 5.

2.3 The <bibliography> Element

Besides the LOM architecture, it accomplishes almost everything. However, during the process of QGMs implementation in LOM, we faced a problem: where to place the bibliographic references necessary to produce and support scientifically a QGM?

Our first attempt was to describe the bibliographic issues in the `<annotation>` element of the LOM's structure. Nevertheless, that solution conflicts with the nature of the LOMs – the interoperability between several metadata systems.

Therefore, we decided to create a new element (the 10th) in the LOM's proposed structure – the `<bibliography>` element. Our goal is to define the `<bibliography>` element accordingly to leading software in this area that represents the bibliography. This procedure will allow to export/import bibliographic information to/from LOM's structure.

This new element follows the Portuguese standard for bibliographic references. We also studied other formats used to describe bibliographic references, in XML: EndNote (EndNote, 2007) and BibTex (Patashnik and Feder, 2006; Gundersen and Hendrikse, 2007).

2.3.1 The <bibliography> Structure – 1

The `<bibliography>` element has one sub-element `<item>` that could be used once or several times. Each `<item>` element represents one bibliographic reference. This element has one attribute named *id*, which is mandatory. The attribute aims to identify uniquely each `<item>` element.

This element is then divided into 9 (nine) sub-elements: *type, authors, titles, publisher, edition Number, updatedDate, normalizedNumber, accessed URL, comment*. All elements can be used only once. The `<type>`, `<authors>` and `<titles>` elements are mandatory. The others are optional.

The `<type>` element identifies the class of bibliographic item and has a list of predetermined options. Those options are almost every types defined by (EndNote, 2007): *Ancient Text, Audiovisual Material, Book, Book Section, Chart or Table, Computer Program, Conference Paper, Conference Proceedings, Dictionary, Encyclopedia, Figure, Journal Article, Magazine Article, Manuscript, Newspaper Article, Online Multimedia, Personal Communication, Report, Thesis, Unpublished Work, Web Page*. To allow other types not specified in that set, we have the option *Other Type*.

The `<authors>` element represents the bibliographic reference authors. The `<authors>` section has, at least, one sub-element, named `<author>`. If the bibliographic reference has more than one author, one `<author>` element must be filled by author. The order of the `<author>` element in the `<authors>` section follows the order used in the source. We consider that the first author is the most important.

The `<titles>` element stores the title of the bibliographic item. It has four sub-elements: *title, chapter, pages and language*. As the name suggests, `<title>` represents the title of the bibliographic item. This element is mandatory. <chapter> and <pages> represent the sections of the book that is important to be noted. If the bibliography is something written, it should be important to express the `<language>`. The data type of this element is *LangString*.

The `<publisher>` element represents the editor's name of the bibliography. It has three sub-elements: *editorName*, *editionYear* and *local*. The `<editorName>` element expresses the editor's name of the bibliography. If `<publisher>` element is used, this sub-element is mandatory. `<editionYear>` and `<local>` represents the date and local where the bibliographic item was published. The data type for `<editionYear>` is *DateTime*.

`<editionNumber>` expresses the edition's number of the bibliographic item.

`<updateDate>` records the date of last update of the bibliographic item. Its data type is *DateTime*.

The `<normalizedNumber>` element is the place to represent the kind of standard number that is used to identify the bibliography. This element has an attribute named *type*. This attribute uses a list of predetermined options corresponding to six different standards: *ISBN*, *ISSN*, *LCCN*, *PCN*, *CIP* and *DOI*.

If the bibliography was accessed in the Internet, the element `<accessedURL>` expresses that url. This element has two sub-elements: *url* and *visitedDate*. If `<accessedURL>` is used, the two sub-elements are mandatory. As their names suggest, *url* represents the URL of the bibliography, and *visitedDate* represents the date when the reference was visited. For this last sub element, the data type is *DateTime*.

The last element, `<comment>`, allows the possible recording of comments about the bibliographic item. Its data type is *LangString*.

The default data type is *text*.

2.3.2 The <bibliography> Structure – 2

The next code presents the <bibliography> structure:

```
<bibliography>
  <item id="unique identifier">
    <type>type of bibliography</type>

    <authors>
      <author>Name of first
           Author</author>
      <author>Name of
           Author</author>
    </authors>

    <titles>
      <title>Book's title</title>
      <chapter>Chapter's name</chapter>
      <pages>consulted pages</pages>
      <language>Book's language</language>
    </titles>

    <publisher>
      <editorName>Editor's name</editorName>
      <editionYear>
         <dateTime>2008</dateTime>
      </editionYear>
      <local>Edition place</local>
    </publisher>

    <editionNumber>Edition number</editionNumber>

    <updatedDate>
       <dateTime>2008-12-20</dateTime>
    </updatedDate>

    <normalizedNumber type="ISBN">ISBN Number
    </normalizedNumber>

    <accessedURL>
      <url>URL</url>
      <visitedDate>
        <dateTime>2008-12-20</dateTime>
      </visitedDate>
    </accessedURL>

    <comment>
      <string language="en">possible comment about
         this bibliographic item</string>
    </comment>
  </item>
</bibliography>
```

Figure 4: The <bibliography> structure definition.

2.3.3 <bibliography> Example

Here is an example of the use of this new element:

```
<bibliography>
  <item id="01">
    <type>Web Page</type>

    <authors>
      <author>Duval, Erik</author>
      <author>Wason, Tom</author>
      <author>Hodgins, Wayne</author>
      <author>and others</author>
    </authors>

    <titles>
      <title>Draft Standard for
             Learning Object Metadata
      </title>
      <language>en</language>
    </titles>

    <publisher>
      <editorName>IEEE</editorName>
      <editionYear>
        <dateTime>2002</dateTime>
      </editionYear>
    </publisher>

    <editionNumber>IEEE 1484.12.1-2002
    </editionNumber>

    <updatedDate>
      <dateTime>2002-07-15</dateTime>
    </updatedDate>

    <accessedURL>
      <url>http://ltsc.ieee.org/wg12/
           files/LOM_1484_12_1_v1_Final
           _Draft.pdf
      </url>
      <visitedDate>
       <dateTime>2008-12-06</dateTime>
      </visitedDate>
    </accessedURL>
  </item>
</bibliography>
```

Figure 5: A <bibliography> example.

2.3.4 The <bibliography> Validation

Since this is an element that was added to the LOM's structure, to allow the validation of such element it was necessary to write a new Schema and integrate it with the Schema provided by IEEE (IEEE, 2005). With that integration, we are now able to perform a total validation of the new LOM's structure.

To support this extension, we defined a new namespace: *http://PmatE.ua.pt/lomExtended.xsd*.

4 MATHML

MathML is the acronym for *Mathematical Markup Language*. This markup language is especially suitable for representing mathematical notation. It can represent the structure and capture the content of a mathematical expression (Ausbrooks, Bos et al., 2007). MathML, like other markup languages, is an XML file. It is possible to write MathML documents with two distinct notations: presentation and content. *Presentation markup* is a more pleasant way to write but this version only encodes the notational structure. If it is necessary to encode the functional structure of an expression, the *content markup* is the best option. Nevertheless, this version is very hard to code. When both presentation and content markup are required, MathML allows the mix of the two markups.

The QGMs started to be mathematical learning objects, so the use of the MathML occurred naturally. With the new version of the QGMs, MathML gained a new importance. As will be explained in the next section, MathML will be used to represent mathematical expressions and to represent the validation expressions of the QGMs. Here the representation is necessary, but the knowledge about what is represented is crucial. That was the reason why we use the MathML *content markup* version.

5 A QGM AS LEARNING OBJECT

The new QGM uses the XML as a description language. With that, we intend to create a structure that is suitable to be represented by a browser or perhaps used in another system. By writing their learning materials in XML, P**mat**E promotes the content exchange with several platforms of Learning Objects. At this moment, P**mat**E is making an effort to diversify the subjects of their QGMs. The majority of them are about mathematics, but the number of QGMs in physics, biology and Portuguese are increasing. This structure is independent of the QGM subject.

A QGM has three sections:

- parameters (`<dominioParametros>`),
- text (`<texto>`), and
- answers (`<respostas>`).

```
<modelo>
  <dominioParametros>
  ...
  </dominioParametros>
  <texto>
  ...
  </texto>
  <respostas>
  ...
  </respostas >
</modelo>
```

Figure 6: A QGM structure.

The `<dominioParametros>` element has several sub-elements. Each one describes the set of possible values to each variables and the options that are required to use the QGM. All values are represented in MathML.

All elements have several attributes, needed to format the values they represent. We use the `<![CDATA[` element to isolate the MathML code and avoid inconsistencies with the Schema.

```
<operador id="op2_aberto" nOpcoes="1">
  <opcao varDependente="false"
         calcular="true">
     <![CDATA[<math>
        <cn>1</cn>
     </math>]]>
  </opcao>
  <opcao varDependente="false"
         calcular="true">
     <![CDATA[<math>
        <cn>-1</cn>
     </math>]]>
  </opcao>
</operador>
```

Figure 7: Example of a <operador> element.

The `<texto>` element contains the text used to be presented as *question* in the normal usage of the QGM. If the *question* has options, they are represented also in MathML.

The last section, `<respostas>` contains the set of possible *answers* that can be presented to the user when a QGM is instantiated. Since each question always has four possible answers, this set must have at least four answers. The necessary information to each answer is stored in three sub-elements: answer text (`<textoResposta>`), validation (`<validacao>`) and additional objectives (`<objAdicional>`). The `<textoResposta>` and `<validacao>` sub-elements use MathML. In `<textoResposta>` the MathML is used to prepare the information that will be presented to the user. In `<validacao>` the MathML is used to implement the validation expression that will determine whether the response of the user is valid or not.

6 THE QGM PARSER

The entire system required has three modules:

- the LOM's editor;
- the QGM editor;
- and the QGM parser.

So far, only the first version of the QGM parser is available.

The parser has the following architecture:

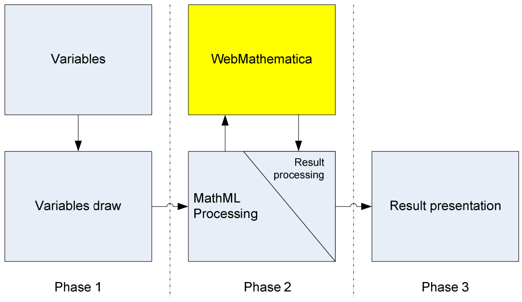

Figure 8: Parser architecture.

As referred, a QGM Learning Object uses XML as description language, and has a set of possible answers. In order to instantiate a new question and present it to a user, a parser is required. Such parser will select the possible answers and the difficulty level according to some parameters such as the user age.

Figure 9 shows the instantiation of the QGM number 290. It is possible to see the MathML code required to produce the answers that a user has to choose. Figure 10 shows the instantiation of the QGM number 1594 about the Portuguese language.

Figure 9: Parser result of math QGM.

Figure 10: A Portuguese language QGM.

The parser uses an XML Schema to validate the QGM and XSLT files to extract the information.

A QGM is a very complex document, with many options that the parser has to choose. Most of these options are coded in MathML. However, MathML is a complex language too and making a good parser for it would be very difficult. So, we decided to extend our own parser with WebMathematica. This leading software in the math's domain would be accessed through a *web service* and would be responsible for validating the choices made by the parser. The code produced in step 1 of the parser is complete but step 2 was impossible to test so far due to the high cost of WebMathematica's license. To test phase 3 we were able to solve the MathML manually.

7 CONCLUSIONS

At the beginning of this work we were confronted with two problems: How to pass on our long-term experience to Portuguese and international networks of Learning Objects? How to make our Learning Objects independent from the technological structure?

Although the P**mat**E project was older than the LOM proposal, the QGM metadata have evolved due to the efforts of its staff. Therefore, the adjustment of QGM's metadata to the LOM structure occurred straight. The only problem found was the lack of bibliography references in the LOM reference model. That limitation was overcome by adding the `<bibliography>` element.

The definition of the QGM structure through XML makes it freely available for other uses. In fact, now with a XSLT file it is possible to extract information from them rapidly. That was not possible with the older structure.

Another major advantage of this new architecture is the speed of development. Actually, due to the volume of material to be developed, it took us two to three months to be able to generate questions with a QGM. With the QGM editor, it will be possible to reduce that time by 90%, because we do not have to code and we can try it immediately.

In the whole work we had a remarkable setback: the inability to test our parser with the WebMathematica. Despite this setback, we believe that our new structure will be the future and it will work.

REFERENCES

Ausbrooks, R., B. Bos, et al. (2007). "Mathematical Markup Language (MathML) Version 3.0." Retrieved 2008-03-01, from http://www.w3.org/TR/2007/WD-MathML3-20071214/.

Barker, P. (2005, 2005/01/05). "What is IEEE Learning Object Metadata/IMS Learning Resource Metadata?" cetis standards briefings series Retrieved 2009/02/09, 2009, from http://metadata.cetis.ac.uk/guides/WhatIsLOM.pdf.

Consortium, I. G. L. (2006, 2006/08/31). "IMS Meta-data Best Practice Guide for IEEE 1484.12.1-2002 Standard for Learning Object Metadata." Retrieved 2008-12-27, 2008, from http://www.imsproject.org/metadata/mdv1p3/imsmd_bestv1p3.html.

Daniel, B. K. and P. Mohan (2004). A Model for Evaluating Learning Objects. IEEE International Conference on Advanced Learning Technologies.

EndNote (2007). EndNote X1.0.1.

Gundersen, V. B. and Z. W. Hendrikse. (2007). "BibTeXML." Retrieved 2008/Abril/28, 2008, from http://bibtexml.sourceforge.net/.

IEEE. (2002, 1992-07-15). "Draft Standard for Learning Object Metadata." Retrieved 2006-10-06, from http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf.

IEEE. (2005). "Draft Standard for Learning Technology — Extensible Markup Language (XML) Schema." Retrieved 2007-06-30, from http://ltsc.ieee.org/wg12/files/IEEE_1484_12_03_d8_submitted.pdf.

Isidro, R., R. Fernandes, et al. (2003). EquaMat2002: Descrição da Infra-Estrutura do Sistema de Informação. 3º Congresso Luso-Moçambicano de Engenharia, Maputo - Moçambique.

Isidro, R. O. G., J. S. Pinto, et al. (2005). SA3C - Platform of Evaluation System and Computer Assisted Learning. WSEAS Transactions on Advances in Engineering Education, Athens - GR, WSEAS.

Patashnik, O. and A. Feder. (2006, 2006). "BibTeX." Retrieved 2008/Abril/28, 2008, from http://www.bibtex.org.

Pinto, J. S., M. P. Oliveira, et al. (2007) "TDmat - mathematics diagnosis evaluation test for engineering sciences students." Internation Jornal of Mathematical Education in Science and Technology, Volume 38, DOI: 10.1080/00207390601035476

Taylor, C. (2003). "An Introduction to Metadata." Retrieved 2007-09-07, from http://www.library.uq.edu. au/iad/ctmeta4.html.

Vieira, J. C. D., M. P. Carvalho, et al. (2004). Modelo Gerador de Questões. Conferência IADIS Ibero-Americana WWW/Internet, Madrid - Espanha.