

Authoring Learning Contents, Assessments and Outcomes in an Integrated Way

Israel Gutiérrez Rojas (igutierrez@inv.it.uc3m.es)
Carlos Delgado Kloos (cdk@it.uc3m.es)

Telematic Engineering Department
Universidad Carlos III de Madrid
28911 Leganés (Madrid/Spain)

Abstract—Developing learning material is a tedious task mainly due to two factors: their variety and their cycle of continuous update. Based on the utilization of DocBook for developing material, a solution is proposed to solve both problems. The proposal is to establish a special markup inside the DocBook format that permits to define multiple kinds of learning material in the same macro-document.

Keywords- DocBook, authoring, learning material, learning assessment, learning outcomes

I. INTRODUCTION

In the context of learning material development there are two aspects that make this task cumbersome for the authors. The first one is the variety of the materials. Authors have to create units of learning for the students, scripts for the classes, test for assessment purposes, practical sessions, simulations, problem sets, flashcards (what we are calling *learning pills*), etc. Besides, all these documents can contain multimedia materials. Due to this variety of learning materials, their production becomes quite complex. The second problem is the cycle of updating materials. It is not easy to update all the references to a concrete subject that has changed. This is particularly a great problem in the technology environment because updates are more often than in other knowledge areas.

Some authors use standard formats for developing learning content. The DocBook format [1] is a schema maintained by OASIS [2] for books and papers that can be also used for the development of learning content. The objective of this paper is to present a special markup used in the DocBook format to provide it with a series of features that can solve the main problems of material authors. This markup also takes into account other aspects in a preliminary stage, like learning outcomes and objectives management and generation of evidences that prove these learning outcomes.

II. STATE OF THE ART

In [3], the advantages of using DocBook format files as a single source for authoring learning material have been presented. The use of XSLT transformations is presented as the main solution to the necessity of different formats to printed material (PDF) and web material (HTML). The packaging of these types of learning content using standards is studied in several articles, like [4] and [5]. In [5], the concept of reuse of existing learning documents is introduced, including several types of formats like Latex, DocBook and Word.

The idea of merging different types of information in the same shape is not innovative. Donald Knuth developed this same idea in his conception of Literate Programming [6]. This paradigm consists on writing programs in the order of demanded order and flow, not attached to any computer needs. Developed programs mix programming code and documentation. Literate program was subjected of a series of processing in order to obtain two outputs: one file with the code to be compiled and another one with the program documentation. This processing follows the same procedure as a demultiplexer (from one input, a set of outputs is obtained). One of the main ideas of this document is to present a mechanism that improve the work of content authors because they could, like in the literate programming paradigm, write learning material in any order as the ideas come to their minds.

The idea of defining a DocBook markup for including various types of content was also used in [5]. A variety of learning content can be extracted from the unique DocBook source (including the course itself) if the author follows a concrete markup during the authoring process: instructor annotations, slides including highlights of each section and extract chunks of learning material as reusable learning objects, that can be packaged following standards and integrated in Learning Management Systems (LMSs).

The rest of this section consists of the study of the main formats and tools used in this development. These are DocBook v4.5 as the basis format for material authoring, XSLT for obtaining individual material from the defined macro-document (as presented in next section) and ADA [7] as the tool used by authors in Carlos III University to produce learning multimedia materials.

A. DocBook

DocBook is a schema well suited to books and papers maintained by the DocBook Technical Committee of OASIS [8]. The DocBook Technical Committee maintains both SGML and XML versions of the DocBook DTD and XML versions in several other languages. It was originally intended for writing technical documents like books and papers, but it can also be used for other types of documentation like learning content for a course.

Initially, DocBook was a SGML document type but due to the emergence of XML a XML DTD was also developed. DocBook is a semantic language that provides authors with the possibility of publishing material in a variety of formats like

HTML or PDF. Dozens of organizations are using DocBook for millions of pages of documentation, in various print and online formats, worldwide.

DocBook provides a large number of semantic elements, which are divided into three broad categories as specified in [9]: structural, block-level, and inline.

Structural elements specify structural information like set (of books), book, navigational components (table of contents, references, index, list of titles), title, chapter, preface, glossary, appendix, section and so on. These include also meta-information elements like author, title, publisher, etc.

Block elements are elements like paragraph, lists, admonitions, examples, figures, tables, questions and answers and so forth. Not all of these elements can contain actual text directly (like xref component). Sequential block-level elements are expected to be rendered one below another.

Inline components are elements like emphasis, hyperlinks, phrase, footnote, mathematics, etc. They wrap text within a block-level element. These elements do not cause the text to break when rendered in a paragraph format, but typically they cause the document processor to apply some kind of distinct typographical treatment to the enclosed text, by changing the font, size, or similar attributes.

Finally there is a set of common attributes attached to all DocBook elements. The main attributes according to the development of the markup of this paper are:

- Role attribute contains a string used to classify or subclassify an element.
- The condition attribute provides a standard place for application-specific purpose.
- ID is an identifying string for the element. It must be unique at least within the document and must begin with a letter.
- Remap contains an element name or similar semantic identifier assigned to the content in a previous markup scheme.
- XrefLabel attribute holds text to be used when a cross reference is made to the element.

B. XSLT and XPath

XSLT [10] is a language for transforming XML documents into other XML documents. The original document is not changed; rather, a new document is created based on the content of an existing one. The new document is serialized by the XSLT processor in standard XML syntax. XSLT relies upon the W3C's XPath language for identifying subsets of the source document tree, as well as for performing calculations.

XPath [11] is a language for addressing parts of an XML document, designed to be used by both XSLT and XPointer. XPath also provides a range of functions, which XSLT itself further augments. This reliance upon XPath adds a great deal of power and flexibility to XSLT.

C. ADA

ADA stands for "Agile and Distributed Authoring" and is an authoring tool developed in the Telematic Engineering Department of Carlos III University. It is used for material production and publication in complex contexts: large number of resources (multimedia materials) and several authors working in a distributed environment.

ADA is also intended to deal with the authors' problem of continuous update of learning materials and it provides mechanisms to solve this problem. For this purpose, it uses a distributed repository of material. If some parts of the material are changed, ADA will include these changes in all the subsequent produced and published documents.

III. DEFINITION OF THE PROPOSED MARKUP

In order to deal with the material authors' problems using DocBook, the inclusion of new features has been proposed. These DocBook add-ons can be implemented in two ways:

- Extending elements and attributes of DocBook format. That is, defining new tags as needed for the new behaviours that were required. For this purpose, a new namespace has to be defined and a DTD or XML Schema.
- Using a special markup maintaining the DocBook format taking advantage of general attributes. There are some attributes like condition that are application-specific and are not taken into account in the DocBook document processing.

The second option has been chosen because maintaining the DocBook format without extensions provides us with a series of advantages like the compatibility with DocBook editors and the validation of the XML can be achieved without extending the DocBook DTD or XML Schema. Thus, the methodology used has been to define a set of markup procedures using DocBook elements and attributes in order to accomplish the proposed objectives.

As commented before, a series of new elements have to be added (using its internal markup) to the DocBook format in order to improve the management of learning material in the same document. That is, the author will be able to develop different learning materials in the same document; thus, the creation process has been improved because the author can reflect whatever comes to his/her mind in the same document and just in time.

The author has to follow a set of markup instructions in order to indicate what type of material he/she is creating. A great macro-document is the result of this process, which will be post-processed conveniently in order to obtain the set of documents that the author intended to create. XSLT technology has been used for this post-processing purpose, so a series of templates (one for each learning material type) have been developed. In the development, the considered learning material types have been: units of learning (student lessons), assessments, learning pills and problem sets.

The markup defined for the DocBook format provides the following features:

- Possibility of indicating the learning outcomes associated to a content section.
- Definition of variables that can be evaluated all across the document. This feature alleviates the problem of continuous material update.
- Introduction of multiple choice and true/false questions that can be used in assessments. These special questions can include also other metadata like hints, duration or indication of incompatible questions.
- Possibility of defining learning pills in the macro-document.
- Creation of sets of open questions from this same macro-document.

All these features are explained in detail below.

A. Learning outcomes indication

Nowadays, one of the main required practices in the e-learning environments is the use of competences. One of the European projects that deal with this affair is the ICOPER project. ICOPER [12] is an eContentplus Best Practice Network that started its work in September 2008. As part of its objectives, ICOPER will provide a *Reference Model* and mechanisms to ensure European-wide user involvement, cooperation, and adoption of standards in the educational framework. To accomplish this goal, the project will systematically analyse the specifications and standards available and in use, to draw conclusions on their validity. In the context of the ICOPER project, an effort is under way to relate learning outcomes and assessment results through evidence records.

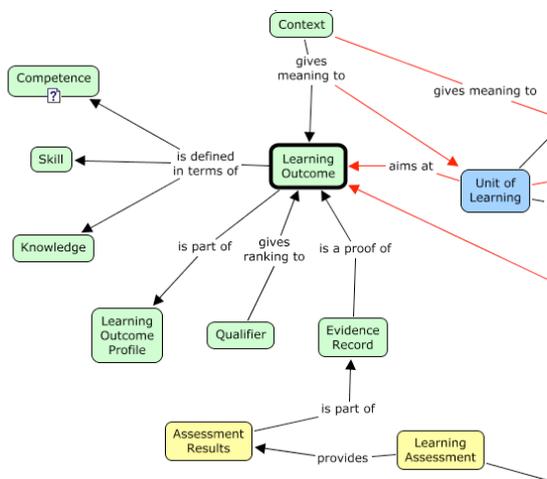


Figure 1. ICOPER Reference Model detail

As part of ICOPER Reference Model (see Figure 1), a conceptual map modelling key concepts for learning scenarios is being developed capturing key concepts and related specifications, but emphasizing on a competence-driven approach as well as the pedagogic underlying layout and the European-wide scope. It is important to define a set of concepts in the domain of e-learning, on one hand to clarify the

terminology used and, on the other hand to establish the relationships between these concepts.

Following the European Qualifications Framework (EQF) learning outcome is a combination of competencies a learner should have (when successfully finishes a learning unit, course or curriculum) which include knowledge, skills and personal, social and/or methodological abilities. Learning outcomes are divided in three categories: knowledge, skills (know how to solve problems and tasks) and competences (proven ability to use knowledge, skills and personal, social and/or methodological abilities, in work or study situations and in professional and personal development). Thus, in the following references to this subject the term “learning outcomes” will be used (that includes competences meaning).

One of the goals of this study is to provide a methodology to attach learning outcomes to courses. At a lower level, these learning outcomes are composed by learning objectives that can be attached to a chapter (or section of it). As explained before, this part of the development is in a early stage and only partially implemented. The DocBook markup used for learning outcomes is to define a paragraph (para) attached to the metadata of the section or chapter (i.e. section/sectioninfo/abstract) using the attribute role with value “learning outcome”. The competence itself will be defined as text inside this paragraph. In the post-processing of the macro-document the learning outcomes will be presented in the teacher lesson (but not in the student). In future developments, the questions obtained from this chapter/section that the student response correctly will provide evidence record that is a proof of a learning objective (and then of a learning outcome).

B. Cross-reference definitions

In the DocBook format there are some mechanisms used to reference parts of the content from some points of this same document. The *xref* tag is used for this purpose indicating the ID of the referenced element by means of the attribute *Endterm*. To include text the tag *xi:include* should be used, using attribute *href* to indicate the source file and *xpointer* to select a part of this file.

As this macro-document will be divided in several documents, it is necessary to define a way for not missing these references. Thus, the elements that will be referenced from other part of the macro-document will use the attribute role with the value id. In post-processing tome, this markup allows to get all this referenced elements and move them to all the generated documents. They will be placed as metadata of the chapter or section (i.e. sectioninfo/abstract).

This mechanism for cross-referencing text is a great solution for the problem of continuous material update because it allows not changing any reference to a concrete subject that has changed, but update just the original reference.

C. Assessments questions

DocBook format only allows one way of defining questions and answers: the *qandaset* element. This element is intended for Q&A sections of books or FAQs. So it is not suitable for assignments by itself (just for open answer questions without

automatic correction). If learning material were being developed, it would be a good feature the possibility of defining other question types in order to be used in assignments and that allow automatic correction.

The FREMA model [13] is a reference model for student assessment scenarios that captures key concepts and related specifications in the e-assessment environment. The questions types defined by this model are the followings:

- **Constrained:** question whose response is constrained to a space of solutions. That is, the student does not need to write any words just selects an option from a set. Examples of this type of question are multiple choice questions (MCQ), multiple response questions (MRQ), true/false (TF) and matching.
- **Constructed:** question whose response is open, that is, it is constructed by the student. Examples of this type of questions are short-answer questions, essay, fill in the blanks (FIB), crossword and so on.
- **Miscellaneous/mixed:** questions that do not fit in the previous categories, like practice sessions, simulations, etc.

Taking into account this classification, the questions supported by DocBook are constructed ones. So it would be nice to have constrained response questions also available because they can be used to assess other learning outcomes and they allow automatic correction. Miscellaneous questions would be also appropriate and its inclusion will be developed in future versions of this tool.

The DocBook markup for this behaviour is the use of the attribute role to indicate the question type attached to a qandaentry entity. The new questions types that the defined markup allows are multiple choice questions (role="MCQ"), multiple response questions (role="MRQ") and true/false (role="TF"). The correct response indication has been achieved using the condition attribute in the answer entity that corresponds to the correct response with the value correct (condition="correct").

Another feature included related to the assessment is the possibility of indicating incompatible questions. That feature is necessary when the question wording is created in several ways (for example, to use it in assessment, problems or examples). In the metadata of the question (qandaentry/blockinfo/abstract) a paragraph (para) element is included with the role value of incompat (role="incompat"). The content of this paragraph will be a reference to the question ID that is incompatible with the current question.

On the other hand, the assessment technique using questions is a successful and well-known strategy. It has been applied to assess the knowledge of student (summative assessment), but also to acquire knowledge during the learning process (formative assessment). Student assessment using questions is usually carried out at class, as a learning activity belonging to a course. This assessment type can be performed using intelligent tutors due to the advantages (effectiveness and utility) that they offer as explained in [14].

There are a variety of strategies related to the hint provision. The subjacent concept behind the hint provision is that the student can ask for help in case he or she were not able to solve a problem. Using hint, the student will have more possibilities to get to the problem solution following a reasoning process (discovering during learning process).

For all these reasons a hint system have been included in the question definition. This information is attached to the metadata of the question (qandaentry/blockinfo/abstract) in a paragraph (para) whose role attribute is declared as hint (role="hint"). The content of this paragraph will be the representation of the hint that can be textual, an image, or whatever the author want to include.

In this way, it has been determined as convenient the inclusion of information about the time that a question is supposed to be solved. That feature provides the author with the possibility of estimating that time that a question is going to take to the student. Using a simple processing, the estimated time for complete exams can be obtained.

The duration of a question is marked in DocBook in the metadata section of the question (qandaentry/blockinfo/abstract) in a paragraph (para) whose role attribute is declared as duration (role="duration"). The content of this paragraph will be the estimated duration of the question expressed in minutes.

D. Learning pills

Flashcard [15] is a set of cards bearing information, as words or numbers, on either or both sides, used in classroom drills or in private study. One writes a question on a card and an answer overleaf. Flashcards are widely used as a learning drill to aid memorization by way of spaced repetition.

A widely used method to efficiently use flashcards was proposed by the German scientific Sebastian Leitner in the 1970s. In his method, known as the Leitner system, flashcards are sorted into groups according to how well you know each one in the Leitner's deck. This is how it works: you try to recall the solution written on a flashcard. If you succeed, you send the card to the next group. But if you fail, you send it back to the end of the current group. Each succeeding group has a longer period of time before you are required to revisit the cards.

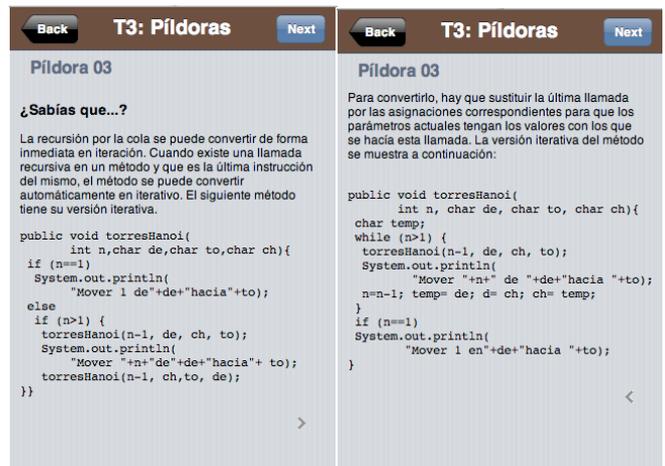


Figure 2. Learning pill example (front and back)

Leitner's flashcards is what is called learning pills in this article. But, instead of referring to physical cards we are referring to learning pills as electronic material (see Figure 2). They are constructed using HTML and JavaScript in order to achieve the effects of flipping the card and deck of cards management.

Regarding to the DocBook markup, block and inline elements are used like para or phrase but using the attribute role to indicate the learning pill question (role="learnpill_question") or the learning pill answer (role="learnpill_answer"). In order to identify the question and answer that compose the same learning pill, the attribute condition has been used to indicate the learning pill name (ID cannot be used because it has to be unique in the document).

E. Other features

One of the materials that teachers provide to course students is a problem set. This type of material is very useful to students in order to prepare assignments and to learn practical part of the course. The problem set is composed by open questions that the teacher obtains from the learning material (assessment and learning pills).

The proposed DocBook markup for this type of content is simple: the remap attribute is used to indicate questions (or learning pill questions) that belong to the set using a special value (remap="question_set"). It is important to point out that the selected questions will compose a set of open response questions. So, the selected answer will have to maintain their meaning without including responses.

Another type of markup defined for learning material in the macro-document is the indication of appearance of certain content in the student lesson. This exclusion can be marked by the author because some part of the content can be intended only for the teacher, like comments and estimated time for explanations, assessment question for the final assignment and so on.

This markup has been defined using the attribute condition of the child elements of the current section or chapter. The value for this attribute is no_show (condition="no_show") in case the author do not want this element to appear in the student lesson. If this attribute has a value in the selected element, it is possible to indicate several values of this attribute separated by commas (i.e. condition="learnpill_example,no_show").

F. DocBook markup summary and complete example

As a summary of the previous defined markups has been defined in a table (see TABLE I.) for quick reference.

TABLE I. QUICK REFERENCE MARKUP

Feature	DocBook reference	
	Meaning	Markup
objectives	Learning objectives attached to a chapter or section.	section/sectioninfo/abstract/para role="objective"

Feature	DocBook reference	
	Meaning	Markup
cross-references	Identification of elements that can be referenced all across the document.	xref endterm="refered_ID" xi:include xpointer="path"
new question types	Definition of new question types.	qandaentry role="MCQ/MRQ/TF" ID="question_name"
hints	Hints attached to a question.	qandaentry/blockinfo/abstract/para role="hint"
duration	Duration of the question.	qandaentry/blockinfo/abstract/para role="duration"
incompatibility	Incompatibility between questions indication. This value can be multiple.	qandaentry/blockinfo/abstract/para role="incompat"
correctness	Correct response indication.	answer condition="correct"
learning pills	Learning pills definition.	role="learnpill_answer" role="learnpill_question" ID="learnpill_name"
not show	Content not showed to the student.	condition="no_show"
question set	Question set indication.	answer/learnpill_answer remap="question_set"

G. XSLT processing and ADA integration

As stated before, XSLT technology has been used for post-processing the macro-document. A series of templates (one for each learning material type) have been developed using XSLT. In this development, considered learning material types have been: units of learning (student lessons), assessments, learning pills and problem sets, so a XSLT for each of them has been developed. The process is detailed graphically in Figure 3.

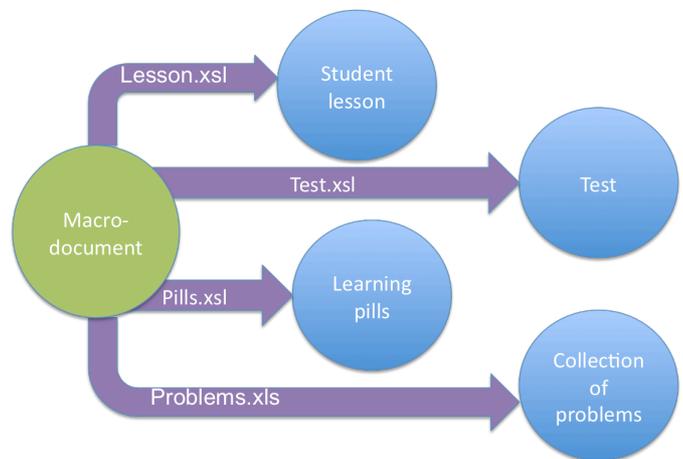


Figure 3. XSLT processing

Finally, integration in ADA production system has been achieved. A properties file (Properties.txt) is used in order to indicate the processing parameters used for the XSLT transformation. But in this case, the process of XSLT transformation has to be performed several times (one for each material type), what is not concerned by the tool. For solving

this problem a module programmed using Apache Ant [16] has been developed specifically for macro-document processing.

H. Complete example

A chunk of a complete example of a macro-document is also provided in Figure 4. In this example, some of the explained features are shown. At section level a global learning objective is indicated using metadata markup (*sectioninfo/abstract*) and the attribute role with value *objective*. That exemplifies its simple usage for the material author. After that, some chunks of text have been marked with an ID (*phrase, programlisting*) so they can be referenced from anywhere in the document through *xref* tag and *xi:include* if some content should be included. This referencing is also exemplified in some *answers* of a *qandaset*. This same question is defined as a multiple choice question (MCQ). In the metadata of this question have been also included incompatible questions, duration and hints using the suitable roles. Finally, a learning pill answer is presented that reuses some text of the student lesson. Thus, flexibility and reusability features of the defined markup are shown.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE section PUBLIC "-//OASIS//DTD DocBook XML V4.5//EN"
"http://www.oasis-open.org/docbook/xml/4.5/docbookx.dtd">
<section>
<sectioninfo>
<date>2009-08-21</date>

<author>
<firstname>Israel</firstname>

<surname>Gutiérrez</surname>
</author>

<abstract>
<para role="objective">Aprender los tipos de recursión lineal</para>
</abstract>
</sectioninfo>

<title>Tipos de recursión lineal</title>

<para>Un método recursivo lineal puede serlo por la cola o no por la
cola.</para>

<para>El siguiente método es <phrase id="no_cola_n" role="id">recursivo no
por la cola</phrase>:
<programlisting id="fac" role="id">
long fac (int n){
if(n<=1) return 1;
else return n*fac(n-1);
}
</programlisting>
</para>
<qandaset>
<qandadiv>
<qandaentry id="q_no_cola" role="mcq">
<blockinfo>
<abstract>
<para role="incompat">q_cola</para>
<para role="incompat">q_cola_no_cola</para>
<para role="hint"><xref endterm="cola" linkend="cola" /></para>
<para role="duration">1</para>
</abstract>
</blockinfo>

<question remap="question_set">
<para>¿De qué tipo es este método? <programlisting><xref
endterm="fac" linkend="fac" /></programlisting></para>
</question>

```

```
<answer condition="correct">
<para><xref endterm="no_cola_n" linkend="no_cola_n" /></para>
</answer>

<answer>
<para><xref endterm="cola_n" linkend="cola_n" /></para>
</answer>

<answer>
<para>En cascada</para>
</answer>

<answer>
<para>Ninguna de las anteriores</para>
</answer>
</qandaentry>
</qandadiv>
</qandaset>

<para condition="cola" role="learnpill_answer">El siguiente método es
<phrase id="cola_n" role="id">recursivo por la cola </phrase>:
<programlisting id="fact" role="id">
long fact (int n,m){
if(n<=1) return m; else
return fact(n-1,n*m);
}
</programlisting>
</para>

```

Figure 4. Macro-document example

The macro-document processing is carried out in the production phase. An example of XSLT processing for learning pills extraction is presented in Figure 5. This piece of code searches all entities in the macro-document that have the role of *learnpill_question* or *learnpill_answer* and put together the ones with the same learning pill name (using the attribute condition). Besides, it includes some DocBook code in order to create the output file with a defined syntax used in learning pills production.

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="xml" indent="yes" />
<xsl:template match="/">
<section condition="learnpills">
<qandaset>
<xsl:for-each select="//*[@role='learnpill_question']">
<qandaentry>
<question class="question">
<title>
<phrase lang="es">¿SABÍAS QUE...?</phrase>
<phrase lang="en">¿DID YOU KNOW THAT...?</phrase>
</title>
<para lang="es">
<xsl:copy-of select="child::node()" />
</para>
</question>
<answer>
<para lang="es">
<xsl:variable name="learnpillname" select="@condition" />
<xsl:for-each select="//*[@role='learnpill_answer']">
<xsl:if test="//*[@condition=$learnpillname]">
<xsl:copy-of select="child::node()" />
</xsl:if>
</xsl:for-each>
</para>
</answer>
</qandaentry>
</xsl:for-each>
</qandaset>
</section>
</xsl:template>
</xsl:stylesheet>

```

Figure 5. XSLT example for learning pills

The output file obtained after the XSLT processing is shown in Figure 6. This output is a DocBook format document

used for learning pills declaration. After a usual DocBook processing it can be converted in the final HTML final that shows the learning pills deck, like the one shown in Figure 2.

```
<?xml version="1.0" encoding="utf-8"?>
<section condition="learnpills">
  <qandaset>
    <qandaentry>
      <question class="question">
        <title>
          <phrase lang="es">¿SABÍAS QUE...?</phrase>
          <phrase lang="en">¿DID YOU KNOW THAT...?</phrase>
        </title>
        <para lang="es">En un método recursivo por la cola, en cada rama del condicional la última operación (la más externa) que se ejecuta es la llamada recursiva.</para>
      </question>
      <answer>
        <para lang="es">El siguiente método es
          <phrase id="cola_n" role="id">recursivo por la cola</phrase>:
          <phrase id="fact" role="id">
            <programlisting>
              long fact (int n,m){
                if(n<=1) return m; else
                return fact(n-1,n*m);
              }
            </programlisting>
          </phrase>
        </para>
      </answer>
    </qandaentry>
  </qandaset>
</section>
```

Figure 6. Learning pills output document

IV. CONCLUSIONS AND FUTURE WORK

A markup using DocBook has been defined in order to solve the main problems of material authors: material variety and continuous update. The solution developed also includes tools for learning outcomes and objectives management and evidences automatic generation.

In order to test the validity of this development it has been integrated into the tool of material production ADA. In the near future it will be used to create material for some subjects in the Carlos III University in the second semester of the current course (2009-2010).

As future works for this development, the learning outcomes management is in a very preliminary stage because it is only defined at section level (learning objectives). The learning outcomes management definition at course level will be the next step. Besides, the automatic creation of evidences as a proof of the section objectives and then the course level learning outcomes. In this way, it would be necessary the introduction of new types of assessment types like practice seasons or simulations. It would be also a great feature the possibility of creating adaptive questions using hints.

Finally, an evolution of the markup is proposed in order to define high-level concepts and to automate more widely the creation of material. This markup should be defined out of the DocBook schema and should be composed by high-level concepts like definition, concept, taxonomy, etc. In this way, some material like assessments or learning pills can be automated.

ACKNOWLEDGMENT

This work has been partially funded by Learn3 project, “Plan Nacional de I+D+I” TIN2008-05163/TSI and the Best Practice Network iCoper “Interoperable Content for Performance in a Competency-driven Society” of the eContentPlus Programme of the European Commission (Grant No. ECP-2007-EDU-417007).

REFERENCES

- [1] DocBook, <http://www.docbook.org> [access June 2009]
- [2] OASIS, <http://www.oasis-open.org/> [access June 2009]
- [3] Molloy, D., 2003, Single Source Interactive and Printed Content Publishing Using the DocBook XML Standard (Second International Conference on Multimedia and ICTs in Education m-ICTE. Badajoz, Spain)
- [4] Baudry, A., Bungenstock, M., Mertsching, B., 2004, Reusing Document Formats for Modular Course Development (Word Conference on Educational Multimedia, Hypermedia & Telecommunications, EDMEDIA), pp. 535-537 1. Birbeck, M. et al., 2001, Professional
- [5] Martínez-Ortiz, I., Moreno-Ger, P., Sierra, J.L., Fernández-Manjón, B., 2006. Using DocBook and XML Technologies to Create Adaptive Learning Content in Technical Domains. International Journal of Computer Science and Applications, 3(2), 91-108.
- [6] Knuth, Donald E. “Literate Programming”, 1984.
- [7] ADA FAQ, <http://www.it.uc3m.es/abel/ADA/doc/FAQ.html> [access August 2009]
- [8] DocBook Technical Committee Document Repository at OASIS, <http://www.oasis-open.org/docbook/> [access June 2009]
- [9] Walsh N. and Muellner L., “DocBook: The Definitive Guide”, <http://www.docbook.org/tdg/en/html/docbook.html> [access August 2009]
- [10] XSL Transformations, <http://www.w3.org/TR/xslt> [access August 2009]
- [11] XPath, <http://www.w3.org/TR/xpath> [access August 2009]
- [12] ICOPER project website. <http://www.icoper.org> [access June 2009]
- [13] FREMA E-learning framework reference model for assessment. <http://www.frema.ecs.soton.ac.uk/> [access June 2009]
- [14] Muñoz Merino P.J., “Teoría de modelado de e-learning y aplicación a un sistema de pistas adaptativo en tutoría inteligente utilizando técnicas de web semántica”, PhD thesis, May 2009.
- [15] Flashcard in Wikipedia, <http://en.wikipedia.org/wiki/Flashcard> [access September 2009]
- [16] Apache Ant, <http://ant.apache.org/> [access September 2009]

