*International Journal of Innovative Computing, Information and Control*
Volume **8**, Number **3(B)**, March **2012**

ICIC International ©2012 ISSN 1349-4198
pp. **2065–2088**

# E-LEARNING CONTENTS AUTOMATIC CONVERSION

Jose-Maria Gutierrez-Martinez, Antonio Garcia-Cabot, Eva Garcia
Luis de-Marcos, Roberto Barchino, Jose-Antonio Gutierrez-de-Mesa
Jose-Javier Martinez, Salvador Oton, Jose-Ramon Hilera
and Juan-Manuel de-Blas

Computer Science Department
Technical School of Computer Science Engineering
University of Alcalá
28871, Alcalá de Henares, Madrid, Spain
{ josem.gutierrez; a.garciac; eva.garcial; luis.demarcos; roberto.barchino }@uah.es
{ jantonio.gutierrez; josej.martinez; salvador.oton; jose.hilera; juan.mbq }@uah.es

Received December 2010; revised April 2011

Abstract. *In this paper, an automatic transformation tool between different content packaging specifications is proposed. The aim of this tool is to serve as a gateway for independent, heterogeneous Learning Management Systems, thus achieving interoperability. The tool is designed to support all existing content packaging specifications, but in this paper we focus on two of the most important: IMS and SCORM. The paper also describes in detail the conversion process for these two specifications, by depicting all the aspects of the tool.*

**Keywords:** Transformation, IMS, Content packaging, SCORM

## 1. Introduction.

Information technologies have widely spread in our society and its influence in the learning process is very important at this moment. This influence has led to the rise of new specific techniques for teaching and learning, called e-learning [1]. The technological key in e-learning systems is Internet, and also how it can be integrated with other more common technologies, like using SMS (Sort Message Service) messages to mobile phones with data relative to the educative process [2]. These technologies are used to perform most of the activities of the learning process as proposed in the Learning Technology System Architecture (LTSA) from the Learning Technology Standards Committee of IEEE [3]. Below it is showed in a diagram (Figure 1) with main elements, relations and actions of the architecture.

This architecture defines the basic entities which are found in e-learning systems: learners, teachers, learning contents and assessments.

In the whole process there is an element always present, the content. These contents need to be maintained, delivered, fixed and could receive many other operations along their use through e-learning systems. As many of these operations can be simplified or aided by software tools, it is absolutely necessary to use a standard format in the representation of the contents. These learning contents are known as “learning objects”, which can be defined as “minimal units in which training materials can be organized, in order to ease contents’ management: creation, indexation, storage, delivery, use, reuse, evaluation and training improvement” [4]. And also they can be later assembled and/or aggregated in order to create greater units of instructions (lessons, courses, etc.).

The European Committee for Standardization in the Learning Technologies Standard Observatory [5] presents the most important specifications (Table 1) for content aggregation in e-learning system.

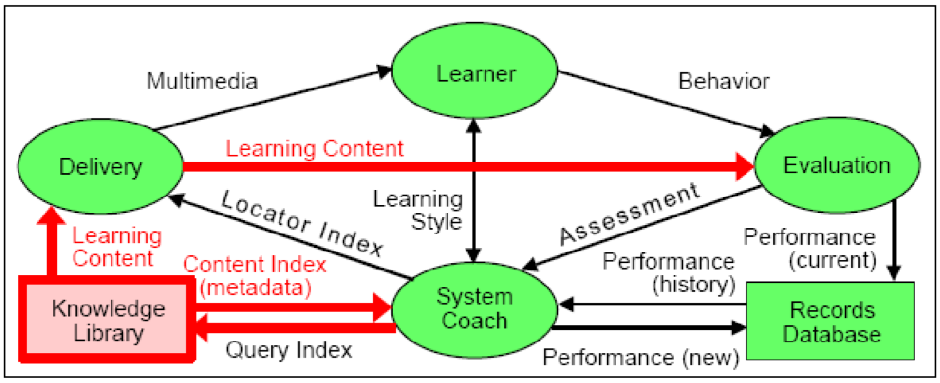


Figure 1. Learning technology system architecture

Table 1. Specifications in content aggregation in e-learning

| | |
|---|---|
| *ADL SCORM Content Aggregation Model* | *http://www.adlnet.org* |
| *IMS Content Packaging* | *http://www.imsglobal.org/content/packaging* |
| *AICC Course Structure* | *http://www.aicc.org/docs/tech/cmi001v4.pdf* |
| *AICC Packaging* | *http://aicc.org/docs/tech/cmi012v1.pdf* |
| *IEEE P1484.6 Course Sequencing* | *http://ieeeltsc.org/* |

IMS [6] and SCORM [7] specifications share many of their characteristics but also have some key differences in the representation of the structure of the learning content.

Other important issue to be considered is the extension of these systems and the huge number of organizations that use them. This situation leads to the fact that the same contents are created, maintained, modified and so on [8]. These are usually not identical copies, but the contents are the same and aim to the same learning objective. So it would be useful to share the work among the organization to avoid repeating the work over and over again. This would save time and resources and would lead to high quality contents. The main obstacle to get this is the, previously presented, existence of different types of models for contents.

From all of this, we can infer the utility of automatic conversion between formats. This conversion will allow the interaction between learning platforms at a high level of automation. The transformation is planned to be used as a part, the most relevant part, of a system of intelligent agents designed to allow the Learning Management System (LMS) interoperation with others LMS systems and with others systems of a learning content, e.g., repositories of learning objects [9]. The network of agents and its interaction with LMS and Learning Contents Managements Systems (LCMS) can be designed and developed using existing interfaces [10] and the systems of agents can be developed using tools and frameworks [11,12]. So, one of the main remaining problem is the automatic conversion of contents. We focus on this problem. We also consider that this interoperation would benefit ubiquity, allowing the use of e-learning tools in mobile devices (m-learning), using a wide range of representations, like 3D models [13].

## 2. Related Work.

The reusability of learning contents always has been an interesting topic in learning research [14]. The result of this are some works with comparative analyses between different learning content models [15], these analyses show the similarities and differences of some learning content models such as 'Learnativity content model', 'CISCO

RLO/RIO model' and 'SCORM content aggregation model'. This can be useful when you want to transform from a learning content model to another.

More specifically concerning to conversion of learning content models, there are some researches in developing an ontology for the conversion between learning contents [16], this ontology provides an explicit definition of the LO (Learning Object) content structure, formally specifying both LO component types and relationships between those components. Other researches are focused in the transformation of documents, e.g., transformation of a UML (Unified Modeling Language) diagrams to a IMS-learning design manifest [17]. This is possible because the diagrams are specified with XML and using XSLT sheets they can be converted in other specifications. This can be applied to conversion of learning content models because the models are specified using markup languages such as XML.

We propose an alternative model to try to overcome these drawbacks. We first analyze the features of the different standards but we also propose a method for the automatic conversion of contents that comply with different standards. Our approach is demonstrated by implementing an application that performs the conversion and then by checking its results. Our model does not consider any intermediate format, language or transformation. In our understanding, standards of content packaging are designed to be processed by systems and not by humans. A review of the format of the different specifications discloses it. So we think that automatic transformation tools are the only element that is really required to convert into different formats. Visual languages may be useful for content creation but, in our opinion, they are not so useful for transformation.

## 3. Automatic Conversion Process.

The main activity in the interaction process between platforms is the transformation from one content model to another. As this interaction could be very complex, and could imply many issues, we focused on the conversion process and the two main specifications (IMS and SCORM).

It is necessary a methodology for evaluating the complexity of the conversion and to ensure its feasibility, we perform these main tasks:

- Content model structure analysis: firstly it is necessary to find the common and not common points between the specifications. With the results of this analysis, the transformations required are discovered. Also, the planning for the next tasks was established, in order to set the appropriate contents to be transformed and the steps needed to do so.
- Sample content is selected and it is packaged using IMS and SCORM specifications. This task produced the starting material and the goal to be reached. With these elements the transformation can be done and can be verified for correction.
- Developing a prototype for transforming from one specification to other one. The transformation prototype was created using the Java language, and XML (Extensible Markup Language) [18].
- Finally a validation is needed, which ensures that the content modeled using IMS and automatically transformed to SCORM (using the prototype of task 3) is equivalent, not necessarily equal, to the content modeled in SCORM in the task 2 and equivalent to the original IMS formatted content.

### 3.1. Content model structure.

Both content models selected share the same components and differ in the structure of those components. The components and the comparison between their formats are:

- The main component of both specifications is a compressed ZIP file which comprises the rest of the files. The name of the file and the internal structure are not specified. In this component, both formats are the same.

- The second component is the manifest file, present in both specifications. The file is called manifest, and its name must be "imsmanifest.xml" because it was first used in IMS and subsequently SCORM used the same name and the general structure, adding elements. This component will require processing to add elements while transforming from IMS to SCORM and removing them in the opposite direction.
- The next component to consider contains multiple elements. These elements are the XML (more specifically XML Schema) files needed to validate automatically the format of the manifest file. These files are not the same in both specifications as SCORM includes all IMS files and some specific of SCORM. For our purpose, it is possible to add and remove the additional SCORM files or maintain always all files as IMS will not be affected by the presence of unused files and these files are lightweighted.
- The last component considered is the learning content. The format of this content must be web compatible and no other limits are set. As this is the only limitation in both specifications, the format could be the same and there is no need to do any transformation. But, in addition to the format requirements, SCORM adds one requirement to the web contents to be used in its packages. They must contain some ECMAScript code (formerly known as JavaScript). This code can be added automatically when converting from IMS to SCORM, and detected and deactivated when converting to IMS from SCORM.

### 3.2. Conversion.

As obtained from the model specifications, the conversion needs to act on three components of the content package. This section summarizes the changes in both directions. The conversion from IMS format to SCORM format implies the following steps:

1. Modification of manifest to include SCORM required elements and sequencing, this will be described in the following section.
2. Inclusion of SCORM XML Schema files to validate the modified manifest.
3. Modification of each learning content element to include ECMAScript code.

On the other hand, the conversion from SCORM format to IMS format implies the following steps:

1. Modification of manifest to eliminate SCORM required elements and sequencing. These elements are not IMS compatible.
2. Removal of SCORM XML Schema files.
3. Modification of each learning content element to disable ECMAScript code.

## 4. Sample Content and Its Transformation.

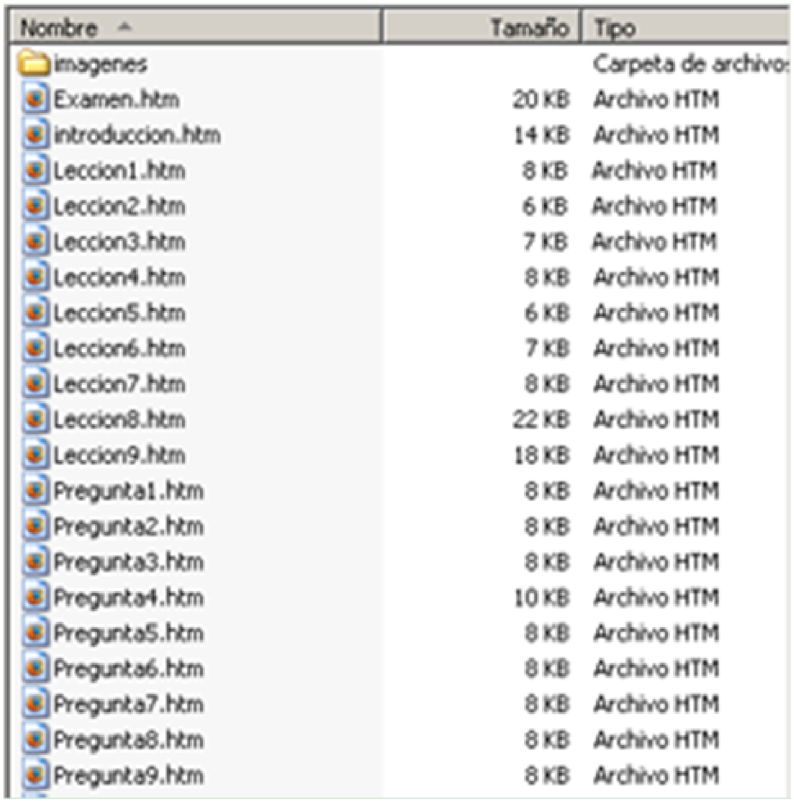
The sample content selected is an Adobe Photoshop course (Figure 2) created using HTML (HyperText Markup Language) pages and simple media such as images. The list of contents of the course is shown in Figure 2.

The pages of the course can be classified in four classes: the main page (introduction), section pages (the main content), reviewing pages and abstract page.

Figure 3 shows one content page.

### 4.1. Generation of the sample contents.

In order to model contents using the actual IMS and SCORM standards, existing tools are going to be used, specifically the Reload (Figure 4) editor and browser [19]. We carefully looked for tools that were both easy to use and regularly updated. So in this way, Reload is a good tool for this purpose, it supports IMS and SCORM content packaging as well as the generation of a simple IMS manifest including only metadata, but not the structure of contents. This structure then



| Nombre | Tamaño | Tipo |
|---|---|---|
| imagenes | | Carpeta de archivos |
| Examen.htm | 20 KB | Archivo HTM |
| introduccion.htm | 14 KB | Archivo HTM |
| Leccion1.htm | 8 KB | Archivo HTM |
| Leccion2.htm | 6 KB | Archivo HTM |
| Leccion3.htm | 7 KB | Archivo HTM |
| Leccion4.htm | 8 KB | Archivo HTM |
| Leccion5.htm | 6 KB | Archivo HTM |
| Leccion6.htm | 7 KB | Archivo HTM |
| Leccion7.htm | 8 KB | Archivo HTM |
| Leccion8.htm | 22 KB | Archivo HTM |
| Leccion9.htm | 18 KB | Archivo HTM |
| Pregunta1.htm | 8 KB | Archivo HTM |
| Pregunta2.htm | 8 KB | Archivo HTM |
| Pregunta3.htm | 8 KB | Archivo HTM |
| Pregunta4.htm | 10 KB | Archivo HTM |
| Pregunta5.htm | 8 KB | Archivo HTM |
| Pregunta6.htm | 8 KB | Archivo HTM |
| Pregunta7.htm | 8 KB | Archivo HTM |
| Pregunta8.htm | 8 KB | Archivo HTM |
| Pregunta9.htm | 8 KB | Archivo HTM |

FIGURE 2. Course contents list

has to be specified in another way, such as HTML links, to create a complete IMS or SCORM package.

Three different versions of the contents are created. One of them will be IMS compliant and another one will be SCORM compliant. The third one will be a plain metadata-only IMS package. The main objective is to create three packages containing the textual contents in the root folder, as well as a folder with images. We will have three folders (Figure 5) in this structure: one for IMS CP (Content Packaging), another one for the IMS manifest-only file, and the last one for the SCORM CP (Content Packaging). This structure is represented in Figure 5.

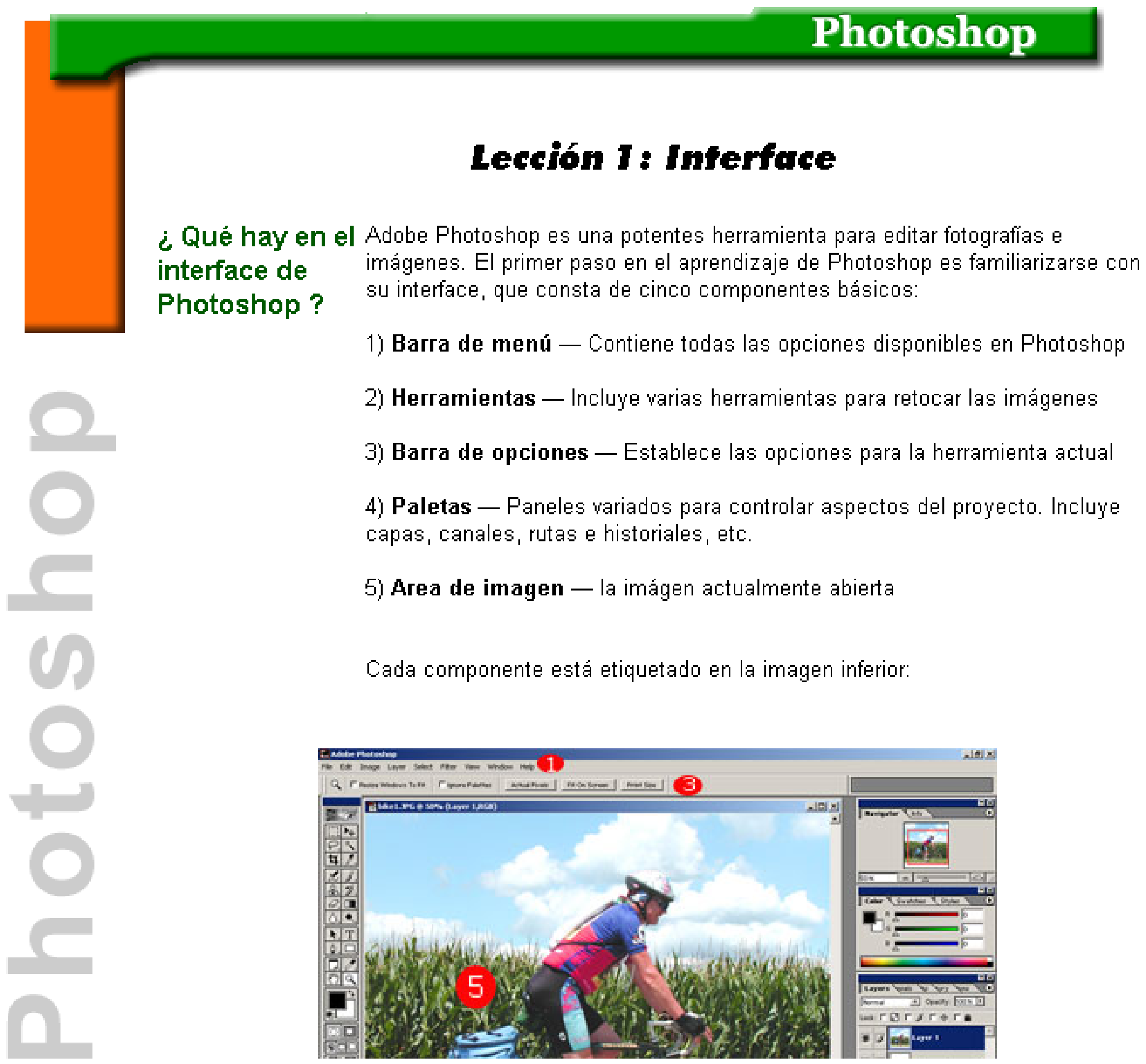
Once these operations are completed, we can browse the folders and the collection of files containing the packages structure information and the metadata.

The first column of Figure 6 shows the structure created for content where only the IMS metadata associated to a context is created. Only one file is added, with a fixed name, "imsmanifest.xml". This file meets the IMS requirements, and is written in the XML dialect created by IMS to model metadata.

The central column represents the files generated for an IMS CP package. Because of the package structure, the imsmanifest.xml file references the "xsd" files that establish the XML dialect. In addition to this, the file contains more information than the previous case.

Last but not least important, on the third column presents the files used in a SCORM SP 1.2 package. These files are the same used in the IMS CP, but here there are also additional xsd files to define the extra structure of the SCORM manifest. It is worth remembering that the manifest file is still named "imsmanifest.xml" in all the three cases.

Once this step is finished, it can be checked that the differences are minimal. In IMS CP there are three new files (with respect of IMS metadata only) called 'ims_xml.xsd',



**Lección 1 : Interface**

**¿ Qué hay en el interface de Photoshop ?**

Adobe Photoshop es una potentes herramienta para editar fotografías e imágenes. El primer paso en el aprendizaje de Photoshop es familiarizarse con su interface, que consta de cinco componentes básicos:

1) **Barra de menú** — Contiene todas las opciones disponibles en Photoshop

2) **Herramientas** — Incluye varias herramientas para retocar las imágenes

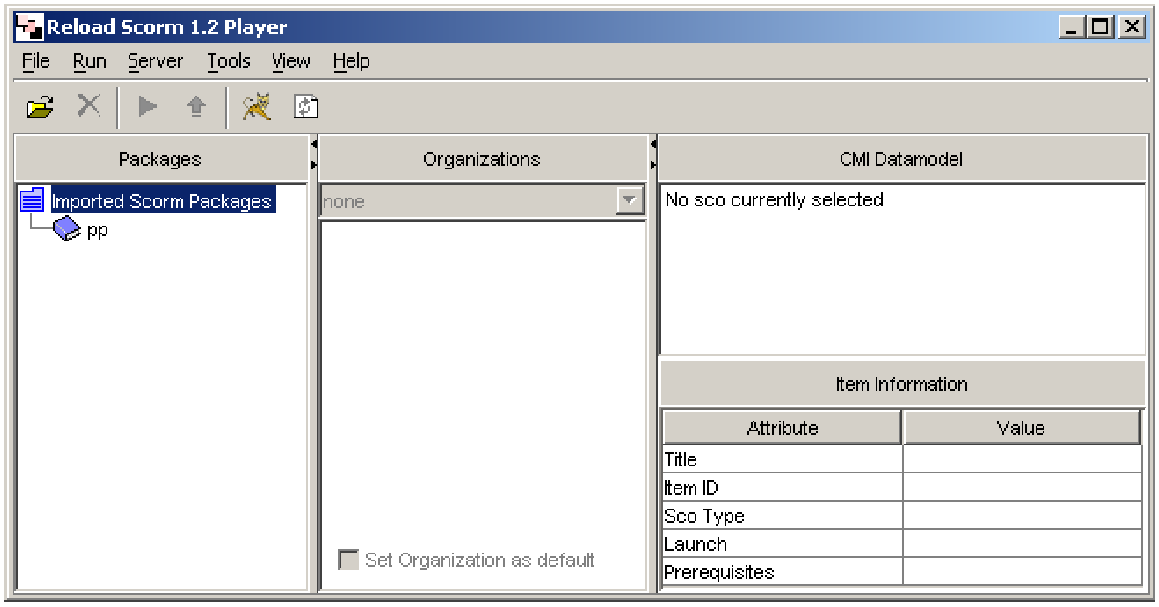
3) **Barra de opciones** — Establece las opciones para la herramienta actual

4) **Paletas** — Paneles variados para controlar aspectos del proyecto. Incluye capas, canales, rutas e historiales, etc.

5) **Area de imagen** — la imágen actualmente abierta

Cada componente está etiquetado en la imagen inferior:

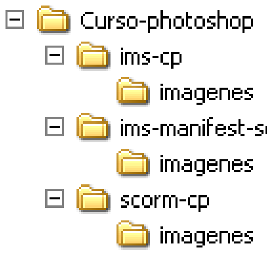
Figure 3. Contents layout (in Spanish)



No sco currently selected

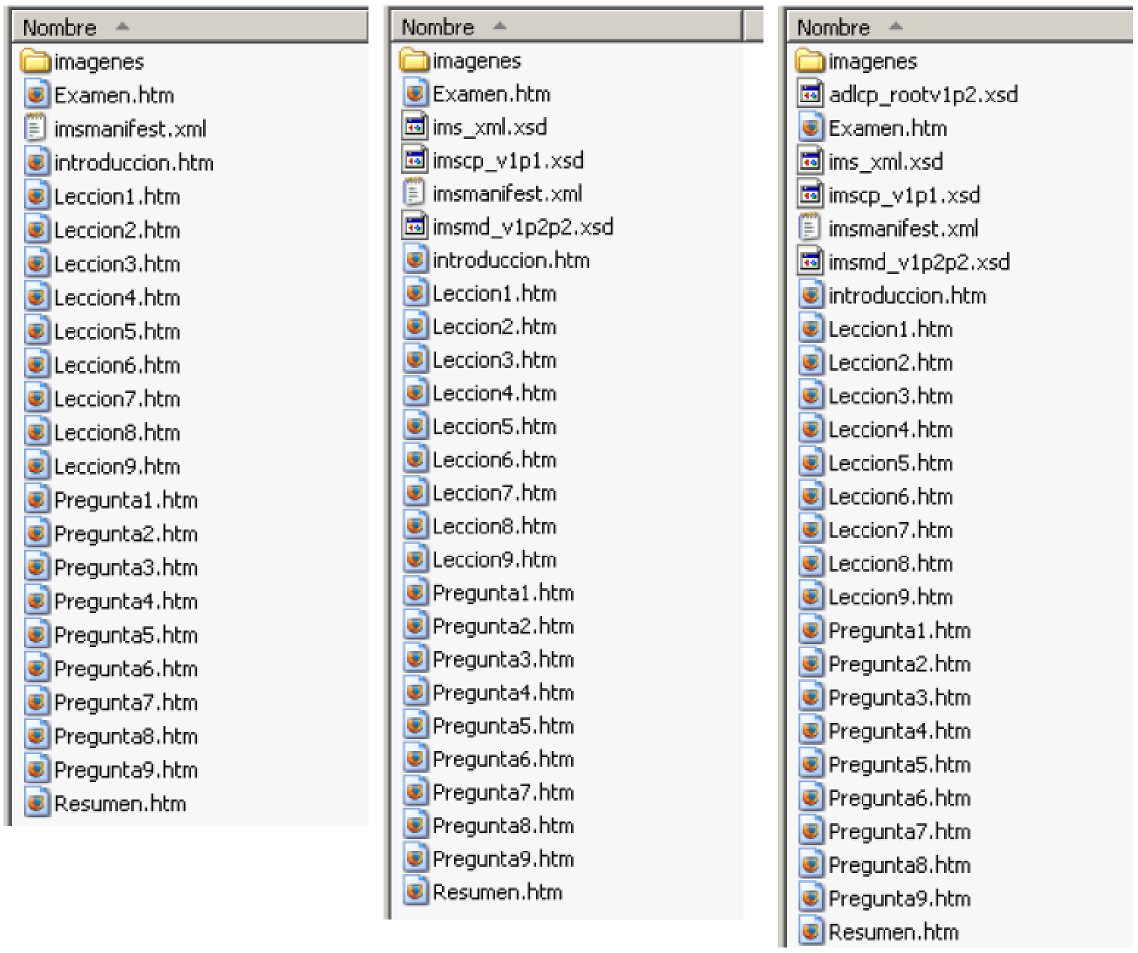
| Attribute | Value |
|---|---|
| Title | |
| Item ID | |
| Sco Type | |
| Launch | |
| Prerequisites | |

Figure 4. Reload SCORM player tool



```
⊟ Curso-photoshop
    ⊟ ims-cp
        imagenes
    ⊟ ims-manifest-solo
        imagenes
    ⊟ scorm-cp
        imagenes
```

FIGURE 5. Destination



```
Nombre                 Nombre                 Nombre
imagenes               imagenes               imagenes
Examen.htm             Examen.htm             adlcp_rootv1p2.xsd
imsmanifest.xml        ims_xml.xsd            Examen.htm
introduccion.htm       imscp_v1p1.xsd         ims_xml.xsd
Leccion1.htm           imsmanifest.xml        imscp_v1p1.xsd
Leccion2.htm           imsmd_v1p2p2.xsd       imsmanifest.xml
Leccion3.htm           introduccion.htm       imsmd_v1p2p2.xsd
Leccion4.htm           Leccion1.htm           introduccion.htm
Leccion5.htm           Leccion2.htm           Leccion1.htm
Leccion6.htm           Leccion3.htm           Leccion2.htm
Leccion7.htm           Leccion4.htm           Leccion3.htm
Leccion8.htm           Leccion5.htm           Leccion4.htm
Leccion9.htm           Leccion6.htm           Leccion5.htm
Pregunta1.htm          Leccion7.htm           Leccion6.htm
Pregunta2.htm          Leccion8.htm           Leccion7.htm
Pregunta3.htm          Leccion9.htm           Leccion8.htm
Pregunta4.htm          Pregunta1.htm          Leccion9.htm
Pregunta5.htm          Pregunta2.htm          Pregunta1.htm
Pregunta6.htm          Pregunta3.htm          Pregunta2.htm
Pregunta7.htm          Pregunta4.htm          Pregunta3.htm
Pregunta8.htm          Pregunta5.htm          Pregunta4.htm
Pregunta9.htm          Pregunta6.htm          Pregunta5.htm
Resumen.htm            Pregunta7.htm          Pregunta6.htm
                       Pregunta8.htm          Pregunta7.htm
                       Pregunta9.htm          Pregunta8.htm
                       Resumen.htm            Pregunta9.htm
                                              Resumen.htm
```

FIGURE 6. Structure created (from left to right: IMS metadata only, IMS CP and SCORM CP)

'imscp_v1p1.xsd' and 'imsmd_v1p2p2.xsd'. The different between IMS CP and SCORM CP is only one file called 'adlcp_rootv1p2.xsd'. All these files are schemas which are used to validate the manifest of each specification. Therefore, to convert from a specification to another one it is necessary adding or removing these files.

Because of this, we will focus now on the manifest, and the first step is to include the most relevant parts of the three "imsmanifest.xml" files, which were generated by the tool for each of the three cases.

Table 2 presents the file that was created using only IMS metadata.

The following was prepared for a full IMS learning content package. Please note that the following code (Table 3) is just a fragment, due to the extreme length of the file.

The following file (Table 4) was generated for a full SCORM learning content package.

Table 2. Fragment of IMS metadata only

```
<?xml version="1.0" encoding="UTF-8"?>
<lom xmlns="http://www.imsglobal.org/xsd/imsmd_v1p2"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.imsglobal.org/xsd/imsmd_v1p2 imsmd_v1p2p2.xsd">
  <general>
    <identifier>Photoshop 1</identifier>
    <title>
      <langstring xml:lang="en">curso de photoshop</langstring>
    </title>
    <language>es</language>
    <structure>
      <source>
        <langstring xml:lang="en">LOMv1.0</langstring>
      </source>
      <value>
        <langstring xml:lang="x-none">Linear</langstring>
      </value>
    </structure>
    <aggregationlevel>
      <source>
        <langstring xml:lang="en">LOMv1.0</langstring>
      </source>
      <value>
        <langstring xml:lang="x-none">2</langstring>
      </value>
    </aggregationlevel>
  </general>
  <lifecycle>
    <version>
      <langstring xml:lang="en">1.1</langstring>
    </version>
    <status>
      <source>
        <langstring xml:lang="en">LOMv1.0</langstring>
      </source>
      <value>
        <langstring xml:lang="x-none">Draft</langstring>
      </value>
    </status>
    <contribute>
      <role>
        <source>
          <langstring xml:lang="en">LOMv1.0</langstring>
        </source>
        <value>
          <langstring xml:lang="x-none">Author</langstring>
        </value>
      </role>
    </contribute>
  </lifecycle>
  <educational>
    <interactivitytype>
      <source>
        <langstring xml:lang="en">LOMv1.0</langstring>
      </source>
      <value>
        <langstring xml:lang="x-none">Expositive</langstring>
      </value>
    </interactivitytype>
    <learningresourcetype>
      <source>
        <langstring xml:lang="en">LOMv1.0</langstring>
      </source>
      <value>
        <langstring xml:lang="x-none">Narrative Text</langstring>
      </value>
    </learningresourcetype>
    <interactivitylevel>
      <source>
        <langstring xml:lang="en">LOMv1.0</langstring>
      </source>
      <value>
        <langstring xml:lang="x-none">very low</langstring>
      </value>
    </interactivitylevel>
    <semanticdensity>
      <source>
        <langstring xml:lang="en">LOMv1.0</langstring>
      </source>
      <value>
        <langstring xml:lang="x-none">medium</langstring>
      </value>
    </semanticdensity>
    <intendedenduserrole>
      <source>
        <langstring xml:lang="en">LOMv1.0</langstring>
      </source>
      <value>
        <langstring xml:lang="x-none">Learner</langstring>
      </value>
    </intendedenduserrole>
    <context>
      <source>
        <langstring xml:lang="en">LOMv1.0</langstring>
      </source>
      <value>
        <langstring xml:lang="x-none">Professional Formation</langstring>
      </value>
    </context>
    <typicalagerange>
      <langstring xml:lang="en">20-40</langstring>
    </typicalagerange>
    <difficulty>
      <source>
        <langstring xml:lang="en">LOMv1.0</langstring>
      </source>
      <value>
        <langstring xml:lang="x-none">medium</langstring>
      </value>
    </difficulty>
    <typicallearningtime>
      <datetime>20 h</datetime>
    </typicallearningtime>
    <language>en</language>
  </educational>
  <rights>
    <cost>
      <source>
        <langstring xml:lang="en">LOMv1.0</langstring>
      </source>
      <value>
        <langstring xml:lang="x-none">no</langstring>
      </value>
    </cost>
    <copyrightandotherrestrictions>
      <source>
        <langstring xml:lang="en">LOMv1.0</langstring>
      </source>
      <value>
        <langstring xml:lang="x-none">no</langstring>
      </value>
    </copyrightandotherrestrictions>
  </rights>
</lom>
```

The analysis of the three generated manifests led us to select the IMS and the SCORM CPs, discarding only the IMS metadata. We decided to exclude the latter because it allows only the specification of general metadata on learning content without including information about the resources included in the package.

Paying more attention to those two remaining models, we can conclude that the difference focuses more on the way the information is structured rather than the information itself. It is observed that SCORM CP has a tag called '⟨metadata⟩' with data about the organization, in this case ADL SCORM. Also, SCORM CP has an additional attribute in

Table 3. Sample of IMS content package

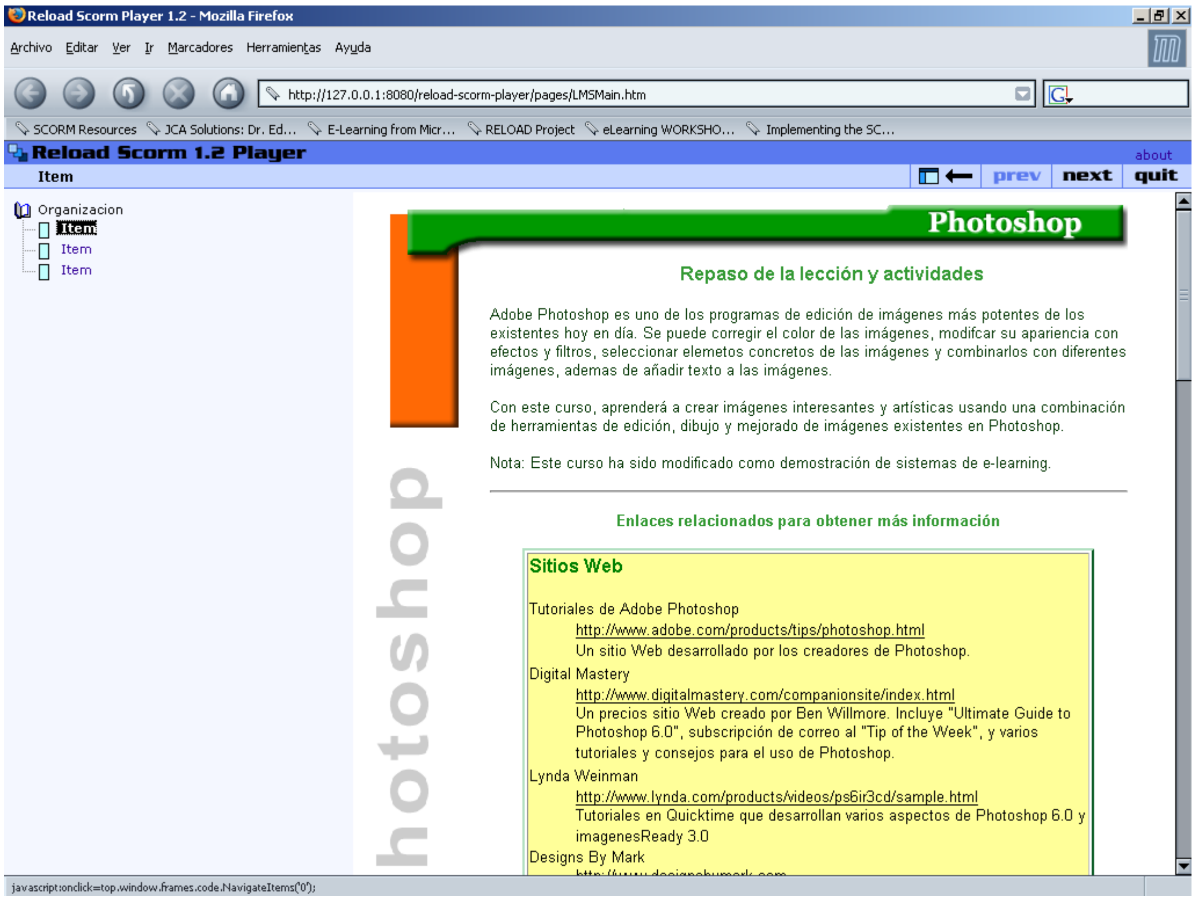
```
<?xml version="1.0" encoding="UTF-8"?>
<manifest xmlns="http://www.imsglobal.org/xsd/imscp_v1p1"
xmlns:imsmd="http://www.imsglobal.org/xsd/imsmd_v1p2"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
identifier="MANIFEST-18C19C64-78B3-C4F6-6822-829D982651E4"
xsi:schemaLocation="http://www.imsglobal.org/xsd/imscp_v1p1 imscp_v1p1.xsd
http://www.imsglobal.org/xsd/imsmd_v1p2 imsmd_v1p2p2.xsd">
  <metadata/>
  <organizations default="ORG-Secuencial-Primera">
    <organization identifier="ORG-Secuencial-Primera" structure="hierarchical">
      <title>Organizacion 1</title>
      <item identifier="ITEM-EE74C4CE-6A31-DCF8-F350-9FFE10BF984C"
identifierref="RES-869ACE59-2B8A-03D6-68EA-F270B98088C4" isvisible="true">
        <title>introducci&#195;³n</title>
        <metadata/>
      </item>
      <item identifier="ITEM-9E2A47F0-7D4A-E494-5014-48C11118D57D" isvisible="true">
        <title>lecci&#195;³n 1</title>
      </item>
      <metadata/>
    </organization>
    <organization identifier="ORG-Secuencial-Segunda" structure="hierarchical">
      <title>Organizacion 2</title>
    </organization>
  </organizations>
  <resources>
    <resource identifier="RES-869ACE59-2B8A-03D6-68EA-F270B98088C4" type="webcontent"
href="introduccion.htm">
      <file href="introduccion.htm"/>
    </resource>
    <resource identifier="RES-F67E05D0-8429-C548-7D5D-23A059707089" type="webcontent"
href="Leccion1.htm">
      <file href="Leccion1.htm"/>
      <file href="imagenes/EndOfLesson.gif"/>
      <file href="imagenes/headerside.gif"/>
      <file href="imagenes/headertop.gif"/>
      <file href="imagenes/interfacesmall.jpg"/>
      <file href="imagenes/LessonTitle1.gif"/>
      <file href="imagenes/pssidebar.gif"/>
    </resource>
    <resource identifier="RES-0E372E2C-190B-D439-8B91-91FDB3A3A5A3" type="webcontent"
href="Leccion2.htm">
      <file href="Leccion2.htm"/>
    </resource>
    <resource identifier="RES-B033AA3A-8B3C-E76B-459B-FBCE46E65671" type="webcontent"
href="Leccion3.htm">
      <file href="Leccion3.htm"/>
    </resource>
  </resources>
</manifest>
```

the tag 'resource' called 'adlcp:scormtype'. It defines the resource SCORM type, shall be set to "sco" or "asset". The reason is because the course that is being modeled exhibits compatible characteristics with the two models. We should keep in mind that SCORM has more possibilities, and it also includes the behavior modeling interaction between the IMS and the content. As a result, the transformation will be carried out as a XSLT XML type transformation. This will be performed with a Java application using the DOM and SAX libraries.

The next step required is the content validation against a tool capable of displaying contents. Microsoft LRM is capable of displaying SCORM and IMS contents. However, in the practice, the tool only works with SCORM CP. Also the Reload SCORM Player can display SCORM content and is known to work correctly with SCORM and IMS contents. In conclusion, for our purposes we use the Reload tool to validate correct contents packaging.

Table 4. Sample of SCORM content package

```xml
<?xml version="1.0" encoding="UTF-8"?>
<manifest xmlns="http://www.imsglobal.org/xsd/imscp_v1p1"
xmlns:imsmd="http://www.imsglobal.org/xsd/imsmd_v1p2"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:adlcp="http://www.adlnet.org/xsd/adlcp_rootv1p2"
identifier="MANIFEST-92B01B35-5474-C681-1E34-EB745957A784"
xsi:schemaLocation="http://www.imsglobal.org/xsd/imscp_v1p1 imscp_v1p1.xsd
http://www.imsglobal.org/xsd/imsmd_v1p2 imsmd_v1p2p2.xsd
http://www.adlnet.org/xsd/adlcp_rootv1p2 adlcp_rootv1p2.xsd">
  <metadata>
    <schema>ADL SCORM</schema>
    <schemaversion>1.2</schemaversion>
    <adlcp:location>adlcp_rootv1p2.xsd</adlcp:location>
  </metadata>
  <organizations default="ORG-62F5E237-DAC6-1EFA-A570-5EE14D9DFB9A">
    <organization identifier="ORG-62F5E237-DAC6-1EFA-A570-5EE4DDFB9A"
structure="hierarchical">
      <title>Organizacion</title>
      <item identifier="ITEM-09512994-B0DC-5FD6-0AFD-C287E47BA11"
identifierref="RES-F65CE6C7-5263-51C1-33C4-7F0733E54582" isvisible="true">
        <title>Item</title>
      </item>
      <item identifier="ITEM-825AF3F4-F58E-7AA2-C722-6983D171CA0C"
identifierref="RES-A0F0ACD6-830D-1954-70AB-3C977B15B55F" isvisible="true">
        <title>Item</title>
      </item>
      <item identifier="ITEM-3D58D2F3-A2F8-41B0-369D-EF0C9D3D34E6"
identifierref="RES-F174A160-0DF6-9ECD-8CDD-B9793F3F46ED" isvisible="true">
        <title>Item</title>
      </item>
      <metadata>
        <schema>ADL SCORM</schema>
        <schemaversion>1.2</schemaversion>
      </metadata>
    </organization>
    <organization identifier="ORG-ECA38232-D105-D0B8-08C1-5C0E21349A"
structure="hierarchical">
      <title>Organization</title>
    </organization>
  </organizations>
  <resources>
    <resource type="webcontent" adlcp:scormtype="asset"
identifier="RES-F65CE6C7-5263-51C1-33C4-7F0733E54582" href="introduccion.htm">
      <metadata/>
      <file href="introduccion.htm">
        <metadata/>
      </file>
    </resource>
    <resource type="webcontent" adlcp:scormtype="asset"
identifier="RES-A0F0ACD6-830D-1954-70AB-3C977B15B55F" href="Leccion1.htm">
      <file href="Leccion1.htm"/>
      <dependency/>
    </resource>
    <resource type="webcontent" adlcp:scormtype="asset"
identifier="RES-C686BE03-12B9-0DDA-C70B-061A638D0961" href="Leccion2.htm">
      <file href="Leccion2.htm"/>
    </resource>
    <resource type="webcontent" adlcp:scormtype="asset"
identifier="RES-F174A160-0DF6-9ECD-8CDD-B9793F3F46ED" href="Leccion3.htm">
      <file href="Leccion3.htm"/>
    </resource>
    <resource type="webcontent" adlcp:scormtype="asset"
identifier="RES-C04BFA9B-425B-14C1-BBC9-29AB5EFD5BE3" href="Leccion4.htm">
      <file href="Leccion4.htm"/>
    </resource>
    <resource type="webcontent" adlcp:scormtype="asset"
identifier="RES-37AAA952-ABBE-5421-20A8-80DEE5DFCDB2" href="Leccion5.htm">
      <file href="Leccion5.htm"/>
    </resource>
    <resource type="webcontent" adlcp:scormtype="asset"
identifier="RES-90EF4696-8647-1884-D391-737C1C6F16A4" href="Leccion6.htm">
      <file href="Leccion6.htm"/>
    </resource>
    <resource type="webcontent" adlcp:scormtype="asset"
identifier="RES-39120EA1-AD6A-E9E7-D609-75548053F96C" href="Leccion7.htm">
      <file href="Leccion7.htm"/>
    </resource>
    <resource type="webcontent" adlcp:scormtype="asset"
identifier="RES-745C8824-4684-7067-7192-BD5471CED99C" href="Leccion8.htm">
      <file href="Leccion8.htm"/>
    </resource>
    <resource type="webcontent" adlcp:scormtype="asset"
identifier="RES-5F583AD8-1141-8F97-A699-57A91B1B2252" href="Leccion9.htm">
      <file href="Leccion9.htm"/>
    </resource>
  </resources>
</manifest>
```


Reload Scorm Player 1.2 - Mozilla Firefox
Archivo Editar Ver Ir Marcadores Herramientas Ayuda
http://127.0.0.1:8080/reload-scorm-player/pages/LMSMain.htm
SCORM Resources
JCA Solutions: Dr. Ed...
E-Learning from Micr...
RELOAD Project
eLearning WORKSHO...
Implementing the SC...
Reload Scorm 1.2 Player
about
Item
prev
next
quit
Organizacion
Item
Item
Item
Photoshop
hotoshop
javascript:onclick=top.window.frames.code.NavigateItems('0');

Repaso de la lección y actividades

Adobe Photoshop es uno de los programas de edición de imágenes más potentes de los existentes hoy en día. Se puede corregir el color de las imágenes, modifcar su apariencia con efectos y filtros, seleccionar elemetos concretos de las imágenes y combinarlos con diferentes imágenes, ademas de añadir texto a las imágenes.

Con este curso, aprenderá a crear imágenes interesantes y artísticas usando una combinación de herramientas de edición, dibujo y mejorado de imágenes existentes en Photoshop.

Nota: Este curso ha sido modificado como demostración de sistemas de e-learning.

Enlaces relacionados para obtener más información

Sitios Web

Tutoriales de Adobe Photoshop
http://www.adobe.com/products/tips/photoshop.html
Un sitio Web desarrollado por los creadores de Photoshop.
Digital Mastery
http://www.digitalmastery.com/companionsite/index.html
Un precios sitio Web creado por Ben Willmore. Incluye "Ultimate Guide to Photoshop 6.0", subscripción de correo al "Tip of the Week", y varios tutoriales y consejos para el uso de Photoshop.
Lynda Weinman
http://www.lynda.com/products/videos/ps6ir3cd/sample.html
Tutoriales en Quicktime que desarrollan varios aspectos de Photoshop 6.0 y imagenesReady 3.0
Designs By Mark

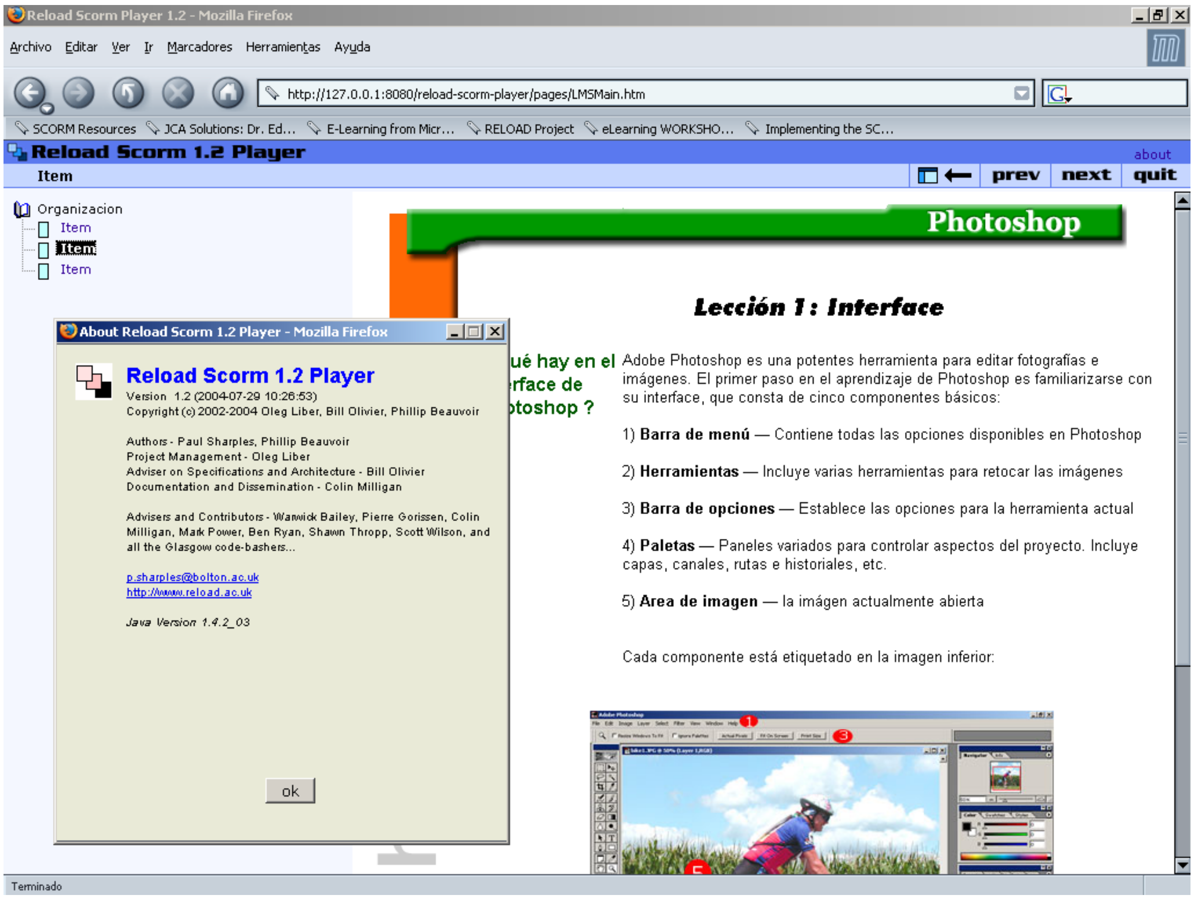
FIGURE 7. First content element

In Figure 7, it is depicted an Adobe Photoshop course by showing its first page. The content is shown running in a Web browser, and there are also some controls that allow navigation through the course in the LMS. This part of the LMS is included in the client browser.

The navigation bar in the LMS client part offers the possibility to step to the next content, or the previous one, exit the LMS, show or hide the navigation tree. In our example, there are three elements from the ten available for that course. All the elements are equally marked as ‘item’. If the course has multiple organizations, and one of them has some contents only, like in this case, the rest of the elements will not be taken into account, and will not appear on the tree.

The last issue to highlight is the available options in the top bar, which complement the navigation window, and are useful when courses have subchapters and complex structures.

In Figure 8, we can see the “about” screen, accessible through the related element on the LMS navigation bar. This window is used to check the product version, as well as internal details like the building date.


Reload Scorm Player 1.2 - Mozilla Firefox
Archivo Editar Ver Ir Marcadores Herramientas Ayuda
http://127.0.0.1:8080/reload-scorm-player/pages/LMSMain.htm
SCORM Resources JCA Solutions: Dr. Ed... E-Learning from Micr... RELOAD Project eLearning WORKSHO... Implementing the SC...
Reload Scorm 1.2 Player
about
Item
prev next quit
Organizacion
Item
Item
Item
Photoshop
ué hay en el
rface de
otoshop ?
ok
Terminado

About Reload Scorm 1.2 Player - Mozilla Firefox

Reload Scorm 1.2 Player
Version 1.2 (2004-07-29 10:26:53)
Copyright (c) 2002-2004 Oleg Liber, Bill Olivier, Phillip Beauvoir

Authors - Paul Sharples, Phillip Beauvoir
Project Management - Oleg Liber
Adviser on Specifications and Architecture - Bill Olivier
Documentation and Dissemination - Colin Milligan

Advisers and Contributors - Warwick Bailey, Pierre Gorissen, Colin Milligan, Mark Power, Ben Ryan, Shawn Thropp, Scott Wilson, and all the Glasgow code-bashers...

p.sharples@bolton.ac.uk
http://www.reload.ac.uk

*Java Version 1.4.2_03*

***Lección 1: Interface***

Adobe Photoshop es una potentes herramienta para editar fotografías e imágenes. El primer paso en el aprendizaje de Photoshop es familiarizarse con su interface, que consta de cinco componentes básicos:

1) **Barra de menú** — Contiene todas las opciones disponibles en Photoshop

2) **Herramientas** — Incluye varias herramientas para retocar las imágenes

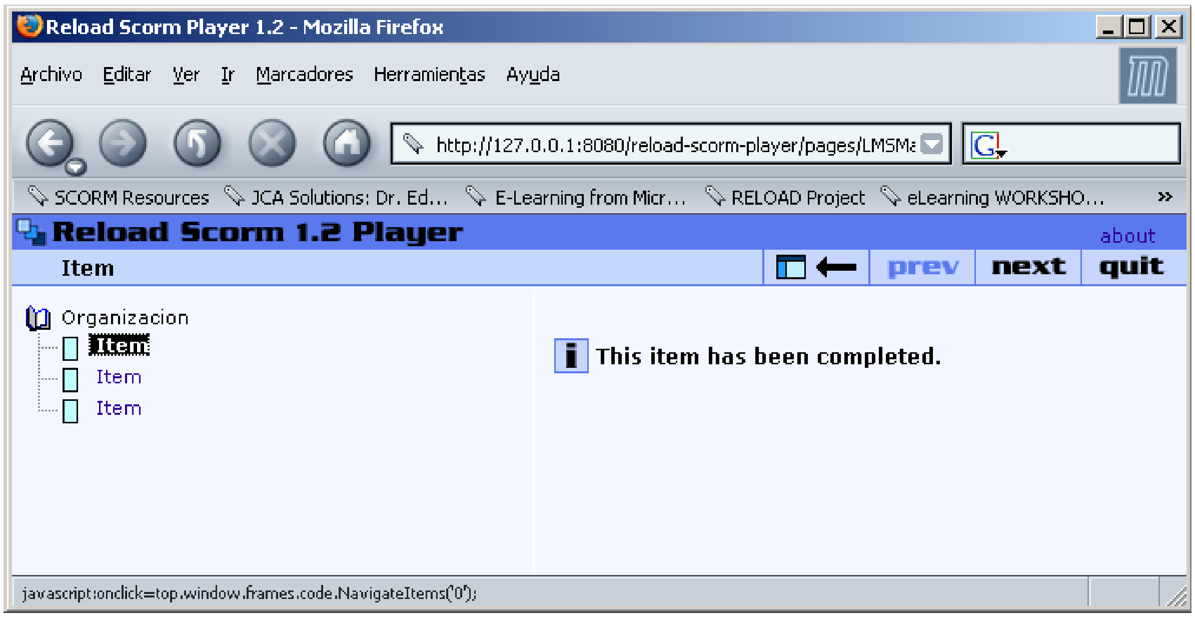
3) **Barra de opciones** — Establece las opciones para la herramienta actual

4) **Paletas** — Paneles variados para controlar aspectos del proyecto. Incluye capas, canales, rutas e historiales, etc.

5) **Area de imagen** — la imágen actualmente abierta

Cada componente está etiquetado en la imagen inferior:

Figure 8. Second element content, and “about” window


Reload Scorm Player 1.2 - Mozilla Firefox
Archivo Editar Ver Ir Marcadores Herramientas Ayuda
http://127.0.0.1:8080/reload-scorm-player/pages/LMSMa
SCORM Resources JCA Solutions: Dr. Ed... E-Learning from Micr... RELOAD Project eLearning WORKSHO...
Reload Scorm 1.2 Player
about
Item
prev next quit
Organizacion
Item
Item
Item

This item has been completed.

javascript:onclick=top.window.frames.code.NavigateItems('0');

Figure 9. Trying to access a forbidden content

Once an item is completed, it cannot be reviewed again. Doing so would imply repeating that element in the course structure, which is not allowed in the selected content organization.

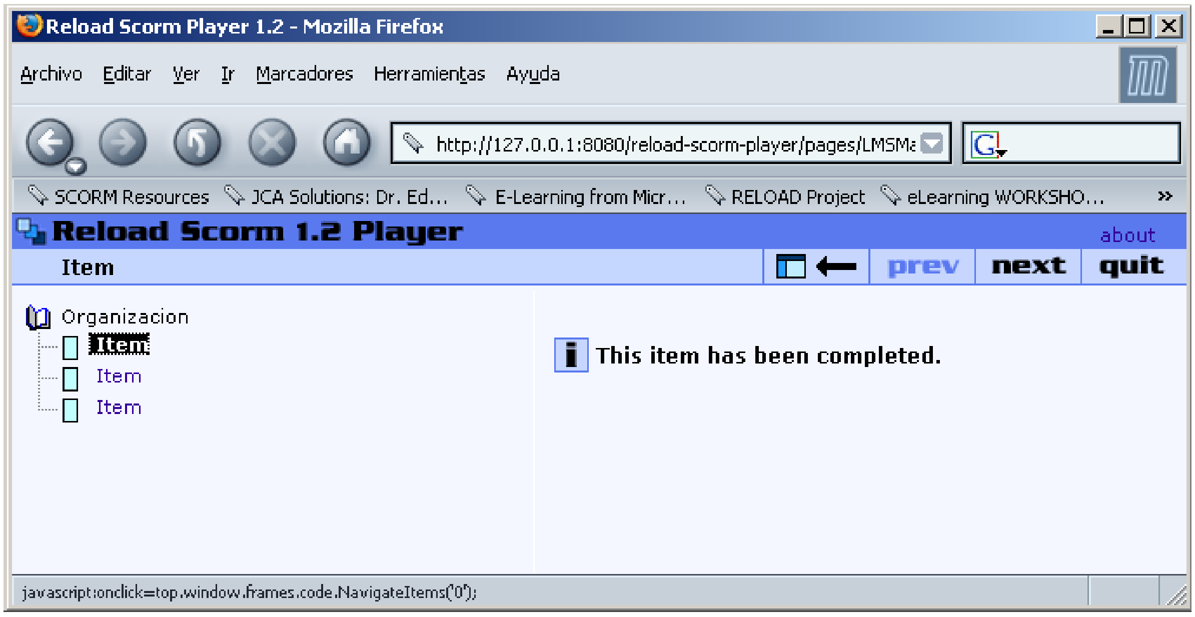
To avoid this situation, and thus preventing the students from going back, the LMS server is aware of the content status, because it maintains a connection with the client. This way, the server can deny the item access, and shows a screen to the user with more information.

What we have seen is only a fraction of the possible control over the contents and the training process. This control is richer in features than the current alternatives, even more if we realize that all control is over standard contents, compatible with all LMS that support SCORM.

In Figure 9 the information message is shown. This message is generated by the LMS server. This message can also be generated because the established navigation rules are very strict, setting up the need of sequentially accessing the contents without repeating and without going back.

This possibility can be useful in some cases, and undesirable in others. It has been included for testing purposes as part of the available options when designing contents.

Finally, when the course is completed, the LMS will show a screen to inform the user. Achieving this type of detection at the end of the course, and also keeping a track of the learner learning status is one of the inherent advantages of using SCORM. Figure 10 shows this screen with a summary of a finished course.



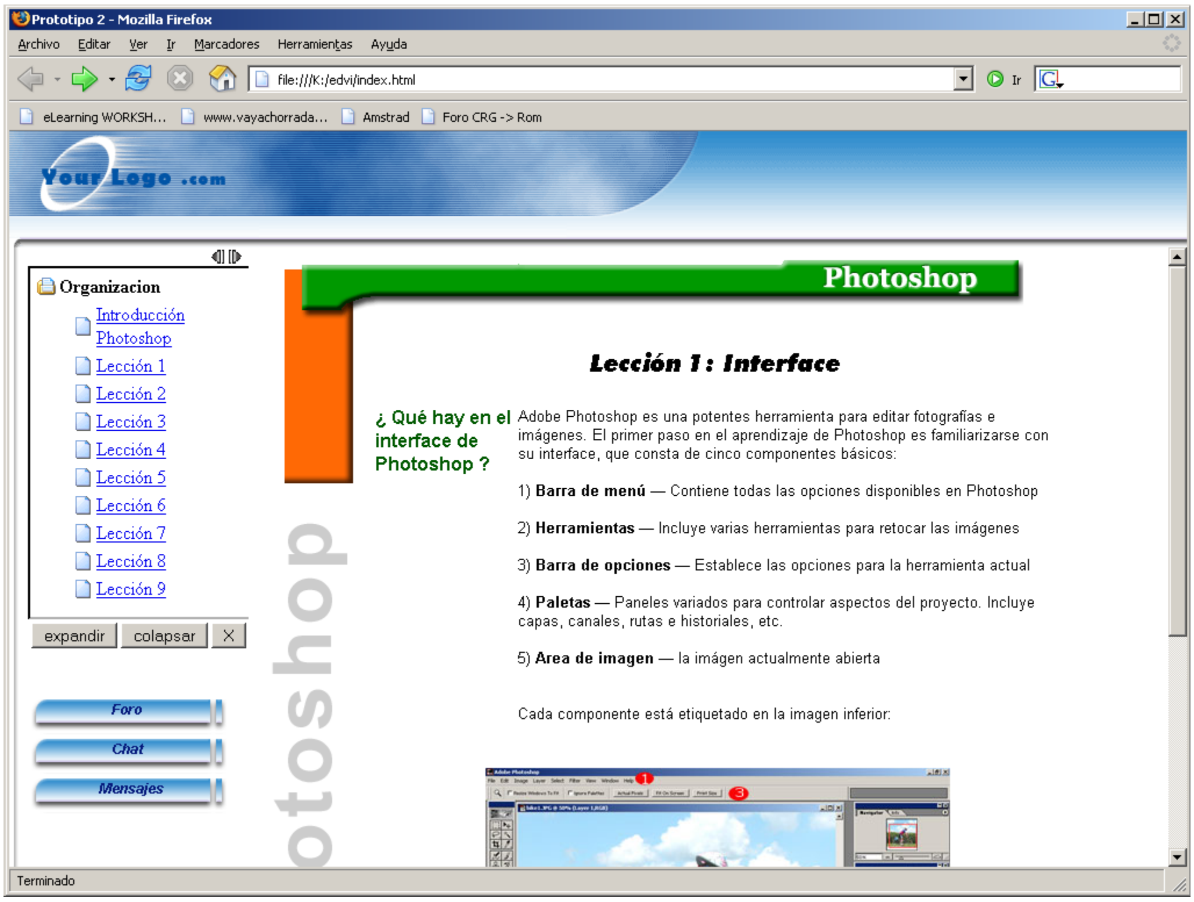
This item has been completed.

javascript:onclick=top.window.frames.code.NavigateItems('0');

FIGURE 10. Summary of a finished course

As last evidence, shown in Figure 11, we present a working prototype of the EDVI LMS tool developed in our Computer Science Department [20-22]. As it can be seen, the structure of the tool is similar, although has its own characteristics, and a higher level of complexity and adaptability than the LRM tool.

At this point, we have already selected the chosen content. It has been packaged using three different tested options, and the two most interesting have been selected. Finally, the packages have been validated by loading them in the available tools. In the next section, we will discuss the issues related to transformation.



**Lección 1 : Interface**

¿ Qué hay en el interface de Photoshop ?

Adobe Photoshop es una potentes herramienta para editar fotografías e imágenes. El primer paso en el aprendizaje de Photoshop es familiarizarse con su interface, que consta de cinco componentes básicos:

1) **Barra de menú** — Contiene todas las opciones disponibles en Photoshop

2) **Herramientas** — Incluye varias herramientas para retocar las imágenes

3) **Barra de opciones** — Establece las opciones para la herramienta actual

4) **Paletas** — Paneles variados para controlar aspectos del proyecto. Incluye capas, canales, rutas e historiales, etc.

5) **Area de imagen** — la imágen actualmente abierta

Cada componente está etiquetado en la imagen inferior:

FIGURE 11. EDVI visualization tool

4.2. **Transformation.** Once we have the contents to transform, they will be displayed in the source and destination formats, which will be destination and source in the second transformation. These contents are formatted using the IMS Content Packaging and SCORM. The transformation will be done following the below steps (Table 5).

TABLE 5. Steps for transformation contents

| | |
|---|---|
| *1.* | *Analyze differences between specifications (IMS and SCORM)* |
| | *a) Observing the folders* |
| | *b) Looking for differences in the files* |
| | *c) Analyzing the metadata of manifest file* |
| *2.* | *Copy the files needed into the new specification* |
| *3.* | *Add/Remove the new attributes to the new manifest file* |

With these contents, the first task is to establish the differences between the two packaging methods, with the enough detail to perform the transformation. This analysis is performed in three levels:

- The first level is about folders. No differences can be found in this level. The folder structure is the same in both cases, and also, a root directory containing the packaging key files can be found. The files related with the content are stored in the packaging root folder or inside subfolders.
- The next level of analysis is related to key files. In this case, it is easy to find the first difference. On the one hand, the manifest file and the XML schemas files are

needed to validate the manifest depending on the packaging version used. On the other hand, using SCORM, there are the same files, plus the SCORM schema files that validate SCORM metadata and other data types.

- Finally, at file level, the analysis focuses on the manifest file. There are some remarkable differences (Table 6 and Table 7). The most important is the inclusion of SCORM metadata like the following.

Table 6. Metadata of SCORM content package

```
<metadata>
  <schema>ADLSCORM</schema>
  <schemaversion>1.2</schemaversion>
  <adlcp:location>adlcp_rootv1p2.xsd</adlcp:location>
</metadata>
```

Another important difference is the existence of XML attributes and items defined by the SCORM guide, besides the related to IMS. An example of these attributes could already be found in the previous code, and also in the following.

Table 7. Resource of SCORM content package

```
<resource type="webcontent" adlcp:scormtype="asset"
identifier="RES-F174A160-0DF6-9ECD-8CDD-B9793F3F46ED"
href="Leccion3.htm">
```

It is clear enough that in order to perform the transformation, although it will not be necessary to modify the folder structure, it will be necessary, though, to add the schema files when transforming to SCORM, and to delete them when transforming to IMS. Finally, it will be necessary to modify the manifest file structure in order to add SCORM metadata in one way, and to delete them when doing the opposite operation.

How can this operation be performed? By using a simple program, all the needed files can be copied. But the real difficulty comes with the part of adding metadata and attributes.

To add attributes, it would be necessary to handle the manifest documents in a smart way, not only doing simple transformations, but handling the information at semantic level.

However, the specifications are XML-based. One of the most important characteristics of XML in this field is the content and structure separation. This feature also allows any XML document to capture the semantics. Thanks to this, it is easy to create a program that accepts an XML document as an input, in this case the IMS or SCORM manifest file, and generate as an output a memory representation of this document, keeping intact the original semantic representation. This representation is part of one of the XML outlying specifications designed with the name of DOM (Document Object Model). The existence of DOM was prior to the release of XML, and it was firstly used to model HTML documents on the web browsers. Later in a new version, DOM was adapted to represent XML documents.

In DOM, each element is an object stored in memory integrated in a tree. The way each element of the tree (children, neighbors, ancestors, or even a random element) is accessed is defined (and standardized) by its name.

As a result of this, it is easy to establish a sequence of steps to perform the transformation from the source document to the destination one. It will be necessary to load into memory the SCORM manifest and locate and delete the SCORM "metadata" specific nodes. All the SCORM-only attributes will be deleted as well.

Table 8. Sample code of transformation

```
try {
SAXBuilder builder = new SAXBuilder("org.apache.xerces.parsers.SAXParser", true);

    //VALIDATE IMSMANIFEST.XML
    builder.setFeature("http://apache.org/xml/features/validation/schema", true);

builder.setProperty(http://apache.org/xml/properties/schema/external-noNamespaceSchemaLocat
ion ,"adlcp_v1p2.xsd");

    //Use the Xerces parser and we dont want to valídate the doc

    Document doc=builder.build("imsmanifest.xml");
    //Building the tree in memory from the file

    Element raiz=doc.getRootElement();
    //The root element

    //Read the namespaces
    Namespace xmlns = raiz.getNamespace();
    List nSpaces = raiz.getAdditionalNamespaces();
    Namespace imsmd = (Namespace) nSpaces.get(0);
    Namespace xsi = (Namespace) nSpaces.get(1);
    Namespace adlcp = (Namespace) nSpaces.get(2);

    System.out.println("- "+raiz.getName());

    Element eOrganizations=(Element) raiz.getChild("organizations",xmlns); //<organization>
    //  Element eResources=(Element) raiz.getChild("resources",xmlns);//<resource>

    List lOrganization=eOrganizations.getChildren("organization",xmlns);

    Iterator iOrganization=lOrganization.iterator();
    //   Iterator recorreRecursos=recursos.iterator();

    while(iOrganization.hasNext())
       {//Loop of level 1 of the tree
        Element eOrganization=(Element) iOrganization.next();//organization element
        Element eTitulo1=(Element) eOrganization.getChild("title",xmlns);
        String identificador1=eOrganization.getAttributeValue("identifier",xmlns);
        List lItem1=eOrganization.getChildren("item",xmlns);
        Iterator iItem1=lItem1.iterator();

        System.out.println(" |");
        System.out.println("  - "+eTitulo1.getText());

        while(iItem1.hasNext())
              {//Loop of the level 2
               Element eItem1=(Element) iItem1.next();//item element (2nd level of the tree)
               Element eTitulo2=(Element) eItem1.getChild("title",xmlns);
               String identificador2=eItem1.getAttributeValue("identifier",xmlns);
               List lItem2=eItem1.getChildren("item",xmlns);
               Iterator iItem2=lItem2.iterator();

               System.out.println(" | |");
               System.out.println(" |  -- "+eTitulo2.getText());

               while(iItem2.hasNext())
                  {//Loop of the level 3

                   Element eItem2=(Element) iItem2.next();//item element (3rd level of the tree)
                   Element eTitulo3=(Element) eItem2.getChild("title",xmlns);
                   String identificador3=eItem2.getAttributeValue("identifier",xmlns);

                   System.out.println(" | |  |");
                   System.out.println(" | |   --- "+eTitulo3.getText());
                  }
              }
       }
   } catch (Exception e)
         {
          e.printStackTrace();
         }
```

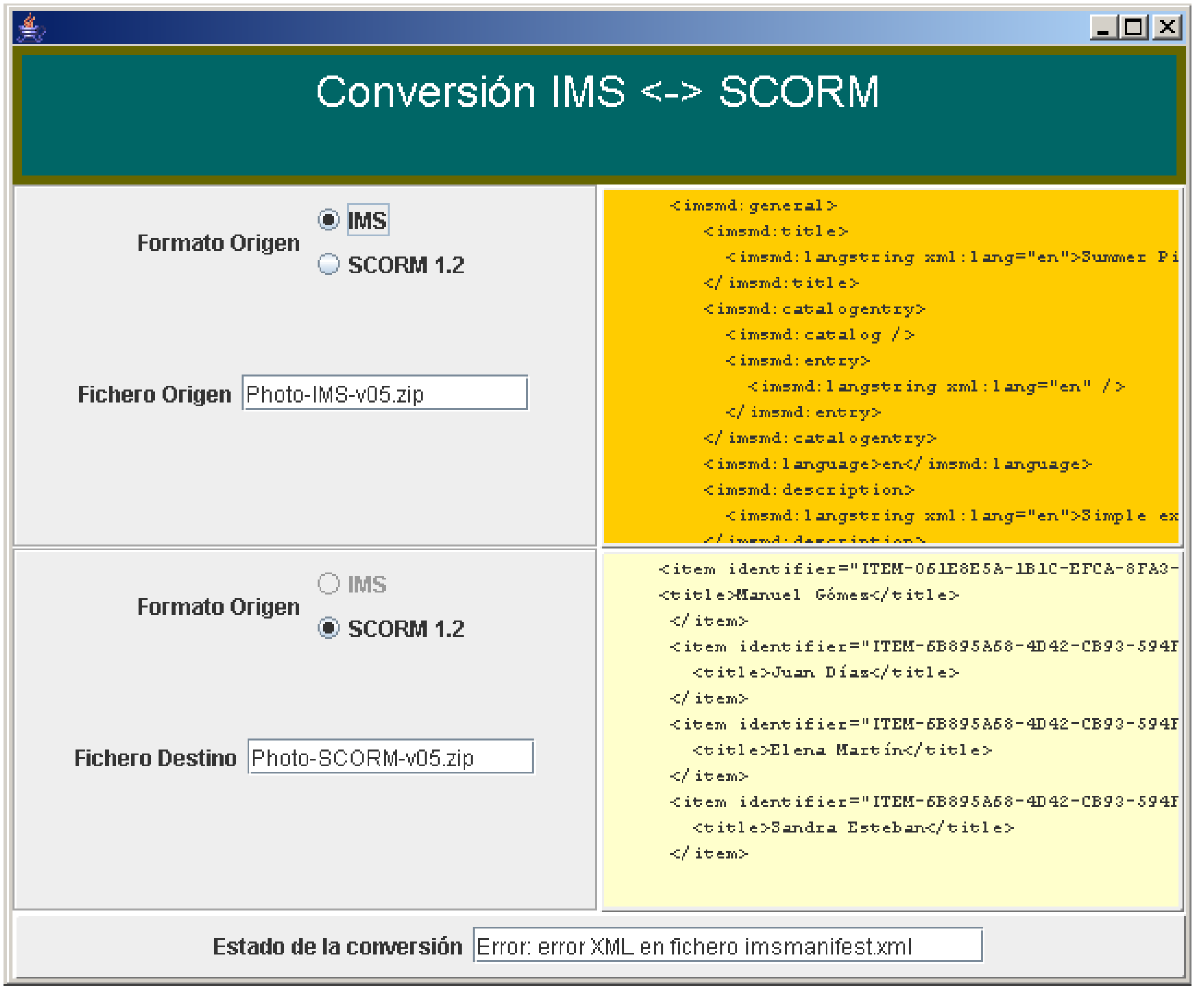
On the other hand, when transforming to SCORM, it will be necessary to locate the metadata in order to add the SCORM content, as well as the resources and attributes.

The platform chosen to perform the transformation will be Java.

The basic implementation for the document reader module, which prepares a document for a later processing, is shown in Table 8.

The above code in Table 8 is a part of the completed code developed for the transformation. In this sample is shown the reader of the file. Later it would process the changes in the file and the following it would add the changes to the manifest objective. Finally, the files are encapsulated in a 'zip' file.

The developed application enables to specify, using simple controls, the input transformation file, the output file, and the contents used in both files (Figure 12). The application has an interface like the following:


Conversión IMS <-> SCORM

Formato Origen [✓ IMS] SCORM 1.2

Fichero Origen: Photo-IMS-v05.zip

```
      <imsmd:general>
         <imsmd:title>
           <imsmd:langstring xml:lang="en">Summer Pi
         </imsmd:title>
         <imsmd:catalogentry>
           <imsmd:catalog />
           <imsmd:entry>
             <imsmd:langstring xml:lang="en" />
           </imsmd:entry>
         </imsmd:catalogentry>
         <imsmd:language>en</imsmd:language>
         <imsmd:description>
           <imsmd:langstring xml:lang="en">Simple ex
```

Formato Origen IMS [✓ SCORM 1.2]

Fichero Destino: Photo-SCORM-v05.zip

```
    <item identifier="ITEM-061E8E5A-1B1C-EFCA-8FA3-
    <title>Manuel Gómez</title>
     </item>
     <item identifier="ITEM-6B895A68-4D42-CB93-594F
       <title>Juan Díaz</title>
     </item>
     <item identifier="ITEM-6B895A68-4D42-CB93-594F
       <title>Elena Martín</title>
     </item>
     <item identifier="ITEM-6B895A68-4D42-CB93-594F
       <title>Sandra Esteban</title>
     </item>
```

Estado de la conversión: Error: error XML en fichero imsmanifest.xml

FIGURE 12. Transformation application (in Spanish)

In Figure 12, it can be observed that the graphical interface can be divided in six main areas. On the top, information about the name and the purpose of the application is displayed. The bottom is related with the status of the operations. On the center, there is a grid composed of four boxes, which can be divided in two types: transformation specification and transformation contents. On the specification ones (on the left), there is a place to specify the transformation source and destination. This information includes the file name and the format. When a format is selected on the source box, the other format is automatically selected on the destination box. Meanwhile, the boxes on the right will be displayed in every moment, and for information purposes, the processed information from and to the source and destination files.

The possible transformation states (Figure 13) are the following:

- Transformation not initiated: the source file is missing.
- Transformation not initiated: the destination file is missing.
- Transformation in progress.
- Transformation ended successfully.
- Error "__".

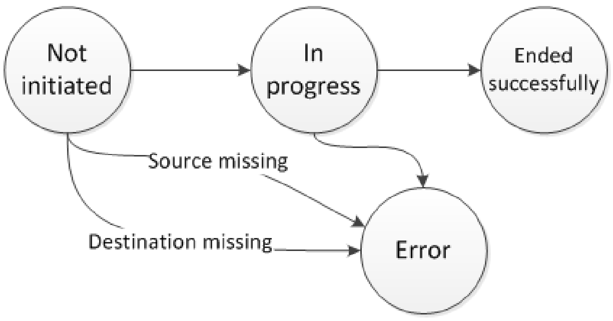
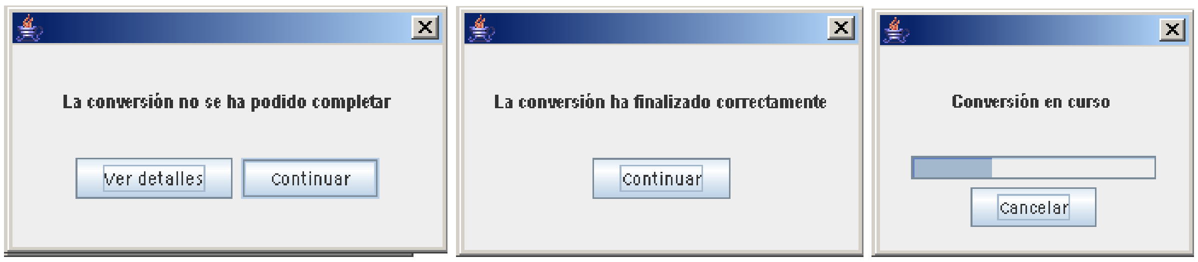


FIGURE 13. States diagram of transformation process

Transformation starts on a 'Not initiated' state. If the source or the destination files are missing then an error is produced. If the files are present the transformation moves to the state 'In progress' where the transformation is performed. If there is any error during the transformation then a transition to the Error state occur informing about the nature of the fault. Otherwise the transformation ends successfully. If there is any error during the process, an error message will be generated depending on the case scenario.

The application shows, status and information dialog windows depending on the operations. These windows also show information related to the obtained results. For instance, the window displayed after a successful transformation is one of these windows (Figure 14).


La conversión no se ha podido completar

La conversión ha finalizado correctamente

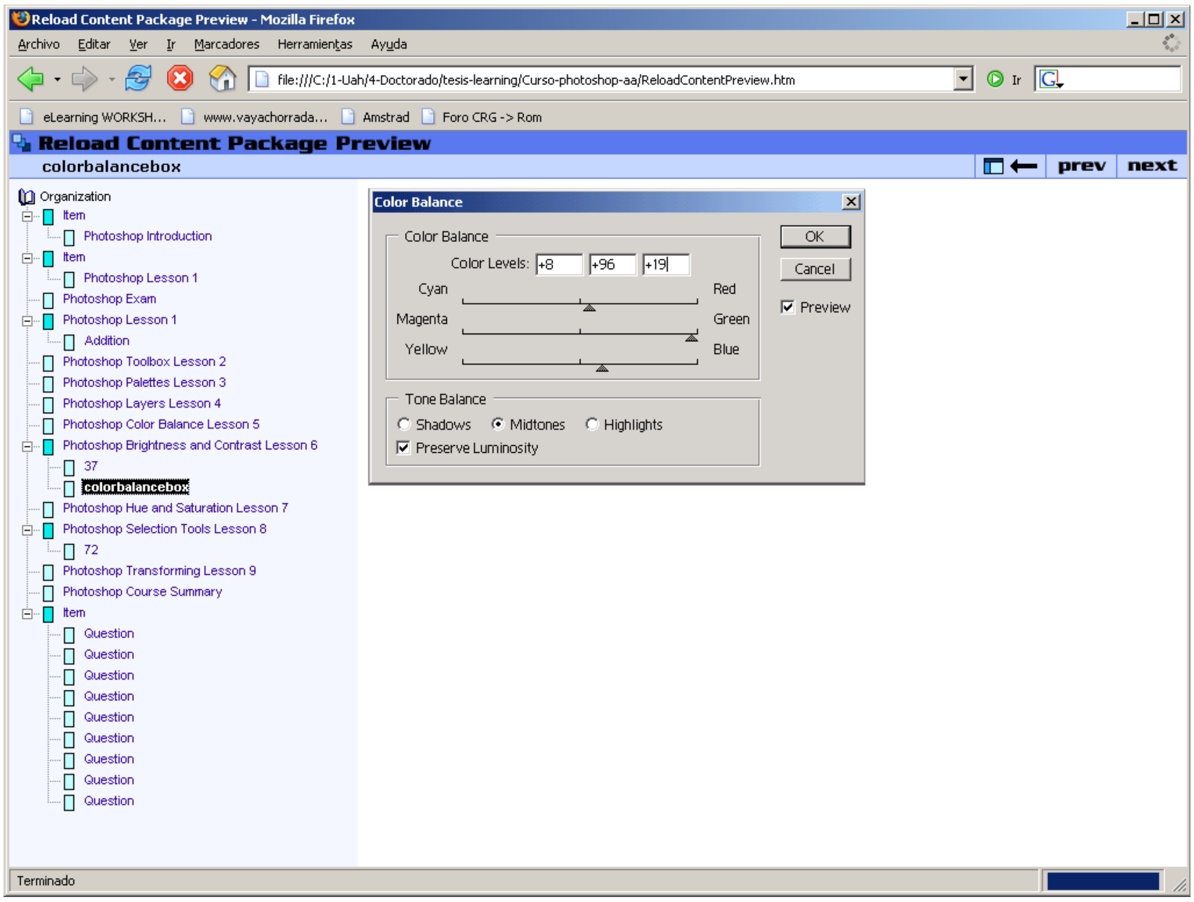
Conversión en curso

FIGURE 14. Application information dialog windows (in Spanish)

4.3. **Validation.** The validation method to use depends on the type and characteristics of the application. In this case, for the transformation application, there are two possible methods. The first one is a detailed comparison of the obtained files, and the second one is a comparison of the results using content tools. The first method would completely ensure similarity between both results. However, it would have been almost impossible to achieve a transformation like this due to partial incompatibilities present in the related standards. Small changes in the internal identifiers or partial losses during the transformation of an attribute would now allow achieving strictly exact results. The second comparison method, conceptually on a higher level, applied to the results of content tools, will led us to perfectly valid results, and would indicate us that even the contents are not exactly the same, they are indeed equivalent from the point of view of they are being used on a learning management system.

Because of this, the high level comparison model was chosen, where the tests will be made up of a transformation and an equivalence check by running the contents on tools compatible with the standard used on each content. The purpose of this method is to demonstrate the impossibility to find differences, whatever standard is being used, from the point of view of the contents used, and when multiple versions have been obtained from the automatic transformation program.

Once the test method is established, the next step is to choose the tool. For this test, the tool selected is Reload (Figure 15), which offers support for IMS and SCORM packets. This tool has been proved easy to operate, and useful on all the development stages in our work. As just one tool is used to show the contents, the results will be more transparent, because the only possible changes would be in the contents section.



file:///C:/1-Uah/4-Doctorado/tesis-learning/Curso-photoshop-aa/ReloadContentPreview.htm

Reload Content Package Preview

colorbalancebox

Organization
- Item
  - Photoshop Introduction
- Item
  - Photoshop Lesson 1
- Photoshop Exam
- Photoshop Lesson 1
  - Addition
- Photoshop Toolbox Lesson 2
- Photoshop Palettes Lesson 3
- Photoshop Layers Lesson 4
- Photoshop Color Balance Lesson 5
- Photoshop Brightness and Contrast Lesson 6
  - 37
  - colorbalancebox
- Photoshop Hue and Saturation Lesson 7
- Photoshop Selection Tools Lesson 8
  - 72
- Photoshop Transforming Lesson 9
- Photoshop Course Summary
- Item
  - Question
  - Question
  - Question
  - Question
  - Question
  - Question
  - Question
  - Question
  - Question

Color Balance
Color Levels: +8 +96 +19
Cyan — Red
Magenta — Green
Yellow — Blue
[✓ Preview]

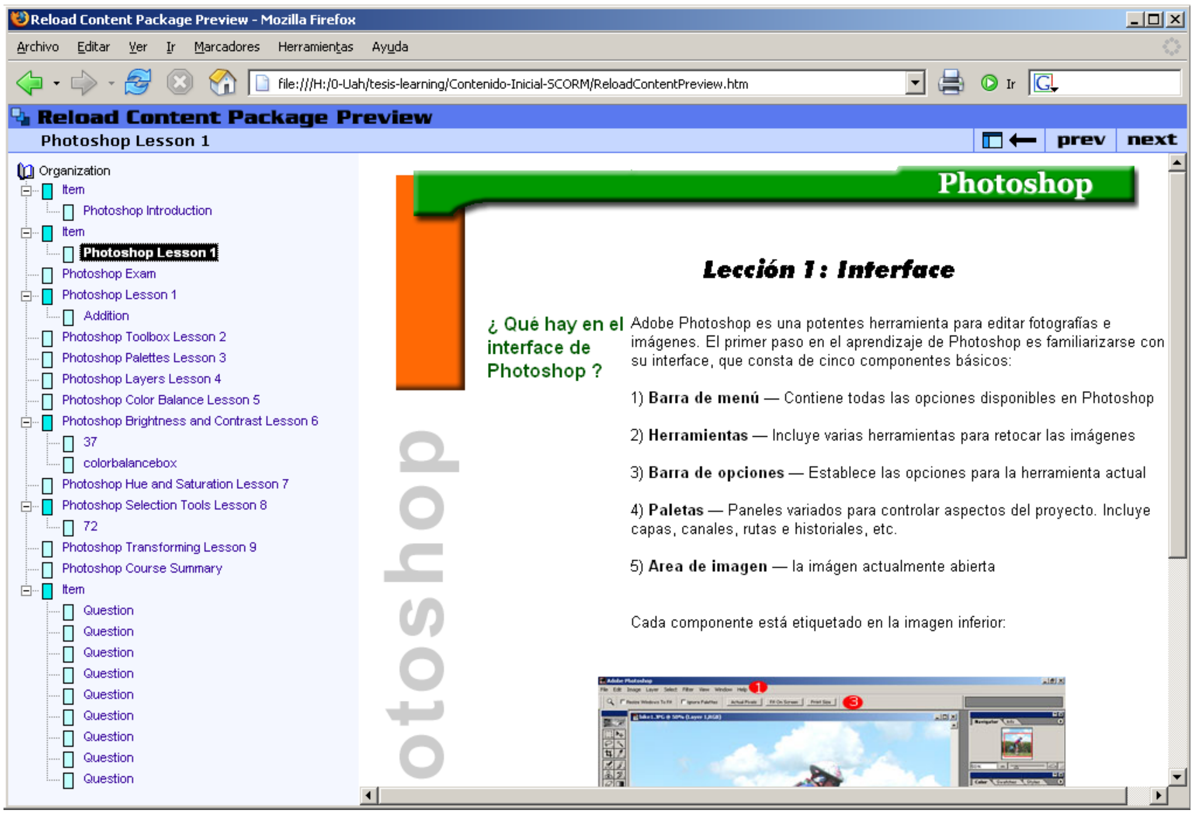
Tone Balance
Shadows [✓ Midtones] Highlights
[✓ Preserve Luminosity]

FIGURE 15. Tool interface used on demo

As it can be seen in Figure 15, the tool is really simple, and besides of including the course content, it almost has no external additional information elements. Of course, it also has basic navigation controls. These characteristics, specially its limitations, make the tool unusable regularly, but they make it an excellent tool for testing and verifying contents, which is obviously what we want.

So using this tool, the initial course and the result of the transformation were loaded. In addition to this, a cyclic transformation was also tested. This method consists of transforming the already transformed content, going back to the original format.

In the following image, the first lesson is presented for a transformed IMS course (Figure 16), as well as a self-evaluation question. In the first place, the first content lesson is represented, in this case, using IMS.


Reload Content Package Preview - Mozilla Firefox
Archivo Editar Ver Ir Marcadores Herramientas Ayuda
Ir
Photoshop
otoshop

file:///H:/0-Uah/tesis-learning/Contenido-Inicial-SCORM/ReloadContentPreview.htm

Reload Content Package Preview

Photoshop Lesson 1 prev next

Organization
- Item
  - Photoshop Introduction
- Item
  - Photoshop Lesson 1
- Photoshop Exam
- Photoshop Lesson 1
  - Addition
- Photoshop Toolbox Lesson 2
- Photoshop Palettes Lesson 3
- Photoshop Layers Lesson 4
- Photoshop Color Balance Lesson 5
- Photoshop Brightness and Contrast Lesson 6
  - 37
  - colorbalancebox
- Photoshop Hue and Saturation Lesson 7
- Photoshop Selection Tools Lesson 8
  - 72
- Photoshop Transforming Lesson 9
- Photoshop Course Summary
- Item
  - Question
  - Question
  - Question
  - Question
  - Question
  - Question
  - Question
  - Question
  - Question

**Lección 1: Interface**

¿ Qué hay en el interface de Photoshop ?

Adobe Photoshop es una potentes herramienta para editar fotografías e imágenes. El primer paso en el aprendizaje de Photoshop es familiarizarse con su interface, que consta de cinco componentes básicos:

1) **Barra de menú** — Contiene todas las opciones disponibles en Photoshop

2) **Herramientas** — Incluye varias herramientas para retocar las imágenes

3) **Barra de opciones** — Establece las opciones para la herramienta actual

4) **Paletas** — Paneles variados para controlar aspectos del proyecto. Incluye capas, canales, rutas e historiales, etc.

5) **Area de imagen** — la imágen actualmente abierta

Cada componente está etiquetado en la imagen inferior:

FIGURE 16. First lesson of a transformed IMS course

The content shown in Figure 16 is the IMS version. It is stored in a folder, which is composed of the contents, the libraries, and programs that IMS must load to implement the tracking. These libraries will allow the platform to know the student's work evolution. This code, which is mainly composed by ECMAScript and HTML, is saved in a different folder checking duplications file names.

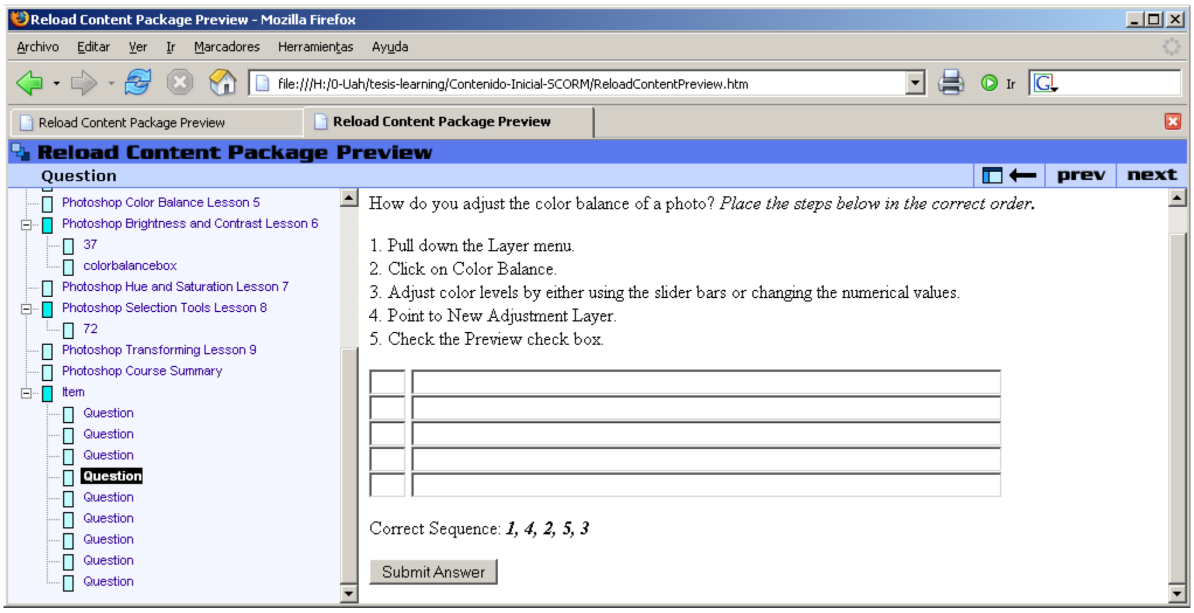
It is observed that the first lesson in IMS transformed course is similar to the SCORM non-transformed course.

As the represented contents can include non-static element types, we also include a contents transformation of a questionnaire (Figure 17). The next image show the initial (SCORM) of a question. It is important to ensure the compatibility of questions' transformation since questionnaires are common on many e-learning contents. This will enable the integration of our process on any initiative that includes them. Questionnaires are becoming increasingly important since they are susceptible of adaptation and personalization applying different artificial intelligence techniques (like in [23]).

The content files of the two versions, saved in the previous mentioned folders, can be measured by their respective, and different, sizes. In this case, these sizes are very similar, which ensure us the similarity between the original and the generated version.

The values are:

Initial-Content-SCORM folder. Total size: 762,129 bytes.


Reload Content Package Preview - Mozilla Firefox

Archivo Editar Ver Ir Marcadores Herramientas Ayuda

file:///H:/0-Uah/tesis-learning/Contenido-Inicial-SCORM/ReloadContentPreview.htm

Reload Content Package Preview | Reload Content Package Preview

Reload Content Package Preview

Question | prev | next

- Photoshop Color Balance Lesson 5
- Photoshop Brightness and Contrast Lesson 6
  - 37
  - colorbalancebox
- Photoshop Hue and Saturation Lesson 7
- Photoshop Selection Tools Lesson 8
  - 72
- Photoshop Transforming Lesson 9
- Photoshop Course Summary
- Item
  - Question
  - Question
  - Question
  - Question
  - Question
  - Question
  - Question
  - Question
  - Question

How do you adjust the color balance of a photo? *Place the steps below in the correct order.*

1. Pull down the Layer menu.
2. Click on Color Balance.
3. Adjust color levels by either using the slider bars or changing the numerical values.
4. Point to New Adjustment Layer.
5. Check the Preview check box.

Correct Sequence: ***1, 4, 2, 5, 3***

Submit Answer

FIGURE 17. Self-evaluation SCORM version question

Final-Content-IMS folder. Total size 765,043 bytes.

These tests, whose results are presented on this paper, allow us to state that the transformation produces a packet that is not internally the same, but it creates an equivalent version, indistinguishable when used on a learning management tool. This result validates our main goal of obtaining interoperability between heterogeneous systems by using agents, automatic and independent applications.

## 5. Conclusions and Future Work.

Content interoperability is important because it ensures that learning contents can be used on different learning platforms. There are different specifications that define the structure that learning contents must have in order to be interoperable. Thus, a higher degree of interoperability can be achieved if automatic tools for transforming contents between different specifications. We have analyzed two packaging specifications and defined the methods for transforming contents compliant with one specification to the other one. We have developed the transforming agent and tested it in a sample scenario. Based on this work, it can be concluded that the transformation between different learning content models is possible. It provides interoperability with LMS systems using different content packaging formats, it ensures the reusability between platforms.

Some deficiencies have been detected in this system such as (1) the only use of formats of SCORM and IMS and (2) not use of an efficient method of transformation (e.g., XSLT sheets). Taking into account this, the future work it would be to extend the number of available formats and to allow the use of modern formats such as IMS Common Cartridge. Also, it is proposed as future work the use of an efficient transformation method as XSLT sheets, this standard would allow to develop easily the use of other learning content models. We have also focused on the transformation process and we have not investigated how transformation should be integrated within the teaching-learning process. It is also important to define how it will be integrated into the different systems involved in the process, notably the LMS and other aiding systems [24], and how the design perspective of these systems could be affected [25].

**Acknowledgment.** This research is co-funded by: (1) the FPI research staff education program of the "Junta de Comunidades de Castilla-La Mancha" and (2) the Government of Madrid Autonomous Community and University of Alcalá (grant CCG10-UAH/TIC-5915). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

## REFERENCES

[1] R. Zemsky and W. F. Massy, Thwarted innovation: What happened to e-learning and why, *The Learning Alliance at the University of Pennsylvania*, 2004.

[2] R. Barchino, J. M. Gutiérrez, S. Otón and L. Jiménez, Experiences in applying mobile technologies in an e-learning environment, *International Journal of Engineering Education*, vol.23, no.3, pp.454-459, 2007.

[3] Learning Technology System Architecture – LTSA Home Page, *Current LTSA Specification*, http://edutool.com/ltsa/, 2006.

[4] D. A. Wiley, Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy, in *The Instructional Use of Learning Objects*, D. A. Wiley (ed.), 2000.

[5] European Committee for Standarization, Information Society Standarization System, *Learning Technologies Standards Observatory*, http://www.cenorm.be/cenorm/bussinessdomains/isss/index.asp, 2006.

[6] IEEE, Learning Technology Standards Committee, *Learning Object Metadata*, 1484.12.1. 2002.

[7] ADL, *Shareable Content Object Reference Model. The SCORM 2004 Content Aggregation Model*, Advanced Distributed Learning Initiative, vol.2005, 2004.

[8] S. Otón, A. Ortiz and J. R. Hilera, *Sistema de Reutilización de Objetos de Aprendizaje*, VIII congreso Iberoamericano de Informática Educativa.

[9] S. Otón, A. Ortiz, J. R. Hilera, R. Barchino, J. M. Gutierrez, J. J. Martínez, J. A. Gutiérrez, L. de Marcos and M. L. Jiménez, Service oriented architecture for the implementation of distributed repositories of learning objects, *International Journal of Innovative Computing, Information and Control*, vol.6, no.3(A), pp.843-854, 2010.

[10] P. Vandepitte, L. Van Rentergem, E. Duval, S. Ternier and F. Neven, Bridging an LCMS and an LMS: A blackboard building block for the ARIADNE knowledge pool system, *Proc. of World Conference on Educational Multimedia, Hypermedia and Telecommunications*, pp.423-424, 2003.

[11] E. W. C. Leung and Q. Li, Agent-based approach to e-learning: An architectural framework, *The Human Society and the Internet Internet-related Socio-Economic Issues*, pp.341-353, 2001.

[12] C.-M. Wang, M.-C. Chiang, C.-S. Yang and C.-W. Tsai, Aided contents supporting service for e-learning systems, *International Journal of Innovative Computing, Information and Control*, vol.7, no.7(A), pp.4005-4026, 2011.

[13] J. M. Gutiérrez, S. Otón, L. Jiménez and R. Barchino, M-learning enhancement using 3D worlds, *International Journal of Engineering Education*, vol.24, no.1, pp.56-61, 2008.

[14] C. Gunn, S. Woodgate and O. Winnie, Repurposing learning objects: A sustainable alternative? *Research in Learning Technology*, vol.13, no.3, pp.189-200, 2005.

[15] K. Verbert and E. Duval, Towards a global component architecture for learning objects: A comparative analysis of learning object content models, *Proc. of the ED-MEDIA World Conference on Educational Multimedia, Hypermedia and Telecommunications*, pp.202-208, 2004.

[16] K. Verbert, D. Gašević, J. Jovanović and E. Duval, Ontology-based learning content repurposing, *Special Interest Tracks and Posters of the 14th International Conference on World Wide Web*, Chiba, Japan, 2005.

[17] J. R. Hilera, L. de-Marcos, J. M. Gutiérrez, S. Otón, S. Barchino, E. García, J. A. Gutiérrez, J. J. Martínez and A. García, Transforming UML-activity diagrams into IMS-learning design manifest using XMI and XSL, *Proc. of the IADIS International Conference E-Learning*, pp.422-424, 2011.

[18] E. R. Harold, *Processing XML with Java: A Guide to SAX, DOM, JDOM, JAXP, and TrAX*, Addison-Wesley, Boston, 2003.

[19] C. D. Milligan, P. Beauvoir and P. Sharples, The reload learning design tools, *Journal of Interactive Media in Education*, 2005.

[20] R. Barchino, J. M. Gutiérrez and S. Otón, An example of learning management system, in *IADIS Virtual Multi Conference on Computer Science and Information Systems*, P. Isaías, M. Baptista and A. Palma (eds.), IADIS Press, 2005.

[21] J. M. Gutiérrez, L. Jimenez, R. Barchino and J. A. Gutiérrez, Un sistema de apoyo a la enseñanza presencial basado en Internet, *Revista Iberoamericana de Sistemas, Cibernética e Informática*, vol.2, no.2, 2004.

[22] L. de-Marcos, J.-J. Martinez, J.-A. Gutierrez, R. Barchino, J.-R. Hilera, S. Oton and J.-M. Gutierrez, Genetic algorithms for courseware engineering, *International Journal of Innovative Computing, Information and Control*, vol.7, no.7(A), pp.3981-4004, 2011.

[23] P. J. Munoz-Merino, C. D. Kloos and M. Munoz-Organero, Deciding on different hinting techniques in assessments for intelligent tutoring systems, *International Journal of Innovative Computing, Information and Control*, vol.7, no.2, pp.841-858, 2011.

[24] C.-M. Wang, M.-C. Chiang, C.-S. Yang and C.-W. Tsai, Aided contents supporting service for e-learning systems, *International Journal of Innovative Computing, Information and Control*, vol.7, no.7(A), pp.4005-4026, 2011.

[25] H.-Y. Kao, M.-C. Liu, C.-L. Huang and Y.-C. Chang, Evaluation and classification of e-learning systems from design perspectives, *ICIC Express Letters, Part B: Applications*, vol.2, no.4, pp.873-878, 2011.