Chapter 9

UNIVERSALIZING CHASQUI REPOSITORIES WITH A FLEXIBLE IMPORTATION / EXPORTATION SYSTEM

José Luis Sierra, Alfredo Fernández-Valmayor
Facultad de Informática. Universidad Complutense de Madrid

Abstract: Chasqui defines an architecture for repositories of learning objects in specialized domains that has been refined during the virtualization of two academic museums at Complutense University of Madrid (Spain). Learning objects in Chasqui follow the *virtual object* model, a specific model that has arisen and evolved during the two virtualization experiences cited. In this paper we describe how the Chasqui architecture can be extended with a flexible importation / exportation system that lets Chasqui repositories store learning material developed with other application profiles as virtual objects. This system makes the practical interoperability between Chasqui repositories and other third-party e-learning platforms and authoring tools possible. We also illustrate this feature with several case-studies.

Key words: Domain-Specific Learning Objects, Repositories, Interoperability, Virtual Objects, Domain-specific Descriptive Markup Languages, IMS Content Packaging, IMS Learning Design

1. INTRODUCTION

The dissemination of cultural heritage is a key application area for information and communication technologies (Adison, 2000; Gladney, 2006; Walczak et al., 2006). By digitalizing the materials contained in the museums and other sorts of cultural and research archives, and by giving access to these digitalized materials by means of usable (frequently web-based) user interfaces, it is possible to make an invaluable cultural patrimony, which would otherwise be relegated to the elite, accessible to the general public. This is especially true for the academic museums and archives that can be found in many academic and research centers.

We have realized this fact during the virtualization of two academic museums at Complutense University of Madrid (Spain): the *Antonio Ballesteros* Museum of Archeology and Ethnology, located at the Department of American History II, and the *José García Santesmases Museum* of the History of Computing, located at the Computer Science School (Navarro et al., 2005). In addition, we have realized the benefits of structuring the digitalized materials as reusable learning objects (Polsani, 2003; Wiley, 2000) in order to enhance the educational value of these materials. As described in (Sierra et al., 2006c) the adoption of the learning object paradigm enables many interesting educational experiences based on the resulting repositories, facilitates problem-based learning strategies (Hmelo-Silver, 2004) and encourages the active involvement of learners in the production of new knowledge by assembling simpler learning objects to yield more complex ones. This fact is leading us to apply similar approaches to more abstract learning domains, like the domain of *language processors* and the domain of *computational linguistics*.

As result of the aforementioned virtualization experiences we have formulated an architecture for the production and maintenance of repositories of learning objects called *Chasqui*[1] (Navarro et al., 2005; Sierra et al. 2006c). As described in (Sierra et al., 2006c), the use of a specific model of learning object is one of the main features of the architecture. Learning objects in *Chasqui* are called *virtual objects*, and in their more basic form they strongly resemble the real objects and dossiers present in the virtualized museums and archives. This model has greatly improved the usability and the acceptance of the resulting applications by the final users who populate, update and consult the repositories (researchers, teachers, and students experts in the domain, but not necessarily experts in computer science). Nevertheless, the use of a specific model of learning object hinders interoperability between *Chasqui* repositories and other e-learning systems (e.g. authoring tools, players and other e-learning platforms). In this paper we describe how these drawbacks can be overcome by adding a flexible importation and exportation system on top of the Chasqui architecture.

The structure of the paper is as follows. In section 2 we summarize the main details of the *Chasqui* repositories. In section 3 we describe the envisioned importation / exportation system. Section 4 exemplifies its applicability in several scenarios. Finally, section 5 gives the conclusions and some lines of future work.

[1] *Chasqui* is *Messenger* in *Quechua*, the language spoken in the Inca Empire.

2. CHASQUI REPOSITORIES

In this section we survey the main features of the *Chasqui* repositories: the virtual object model (point 2.1) and the *Chasqui* architecture (point 2.2).

2.1 Virtual Objects

A virtual object (VO) structures the information that represents, or is closely related to, one specific object (physical or conceptual). This information includes a set of attribute-value pairs with *data* about the object that are considered relevant to its scientific study (e.g. *word length* and *bus bandwidth* of a computer). It also includes another set of attribute-value pairs containing the *metadata* of the object, which represent features useful for the description and classification of the object from a pedagogical point of view (e.g. the *author* of the virtual object, its *version number* and its *classification* in one or several taxonomies). Metadata are taken from the IEEE LTSC Learning Object Metadata (LOM) (IEEE, 2002). Finally, it includes the set of *resources:* digital files that result from the virtualization process, together with another set of attribute-value pairs describing these resources and their relation to the object.

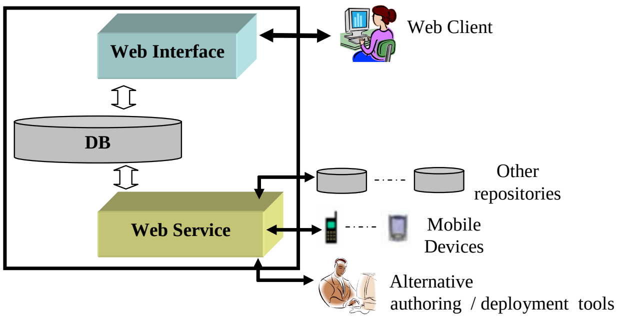

Fig. 1. *Chasqui* architecture.

Resources are further categorized into three different types. *Own resources* are a set of multimedia archives (e.g. a set of photographs of a computer or a video illustrating its operation). *Foreign* resources are references to resources owned by other VOs (e.g. the documents related to the research and design processes of a type of computer will be owned by a conceptual VO but referenced by all the VO describing computers of that type). Finally, *VO resources* are references to other VOs (e.g. VOs for the different components of a computer). Foreign and VO resources allows for the representation of *ad hoc* networks of VOs, where VO resources are used for constructing more complex VOs by aggregating simpler ones, and foreign resources establish other basic relationships between VOs. Besides, VOs can be classified in several taxonomies, by using the LOM `classification` element in the metadata. These taxonomies are not necessarily pre-established, but can be dynamically generated as more and more VOs are added to the repositories.

2.2 The Chasqui Architecture

The architecture for *Chasqui* repositories is outlined in Fig. 1. According to this architecture, the repository is supported by a relational database. The architecture also includes a domain-specific web-based authoring and deployment tool, which is employed by users to populate, update and browse the repository. Finally, it also adds a web service interface (Cerami, 2002).

The web service interface is very valuable for facilitating the connection with third-party systems (e.g. other *Chasqui* repositories, mobile devices providing alternative ways of accessing the repository, or other authoring and deployment tools). The next section also details how this interface can be used to feature the *Chasqui* repository with sophisticated importation and exportation capabilities, therefore increasing the interoperability spectrum.

3. ADDING A FLEXIBLE IMPORTATION / EXPORTATION SYSTEM TO CHASQUI REPOSITORIES

Chasqui already incorporates some basic importation and exportation facilities of VOs encoded in accordance with the IMS Content Packaging specification (IMS CP) (IMS, 2004), which are accessible by using both the web interface and the web service. Data are encoded in a XML file that follows a specific XML-based markup language, and which is included as a resource of the resulting package. In Fig. 2a we show an example of such a file for a VO in the *José García Santesmases Museum* (original tags are in Spanish; we have translated them to English). Metadata are included in the global metadata section of the IMS manifest. In this manifest, resources are listed in the resource part, and they are organized in a single organization as a sequence of items. In Fig. 2b we show an example corresponding to the VO mentioned before. When packed to be exported, VO's foreign resources are converted into resources owned by the object. Finally, all the VOs connected to the one packaged are included as subfolders in the package, and VO resources are encoded by referring to the corresponding IMS manifests. Therefore, and despite of the fact that IMS CP is being used as an interchange format, the encoding conventions followed in *Chasqui*, which are a consequence of the specific nature of the VO model, only allow for direct interoperability with *Chasqui*-aware systems, as well as direct exportation to generic IMS CP tools, e.g. IMS packagers like Reload editor (Reload, 2006). More complex importation and exportation processes require the use of the appropriate *mappings*, which must translate between the VO model and other leaning objects models. In this section we propose an extension of the VO model that facilitates this incorporation (point 3.1), we identify the operational support required (point 3.2), and we depict how the resulting importation / exportation system can be plugged into the architecture by using the web service interface (point 3.3).

(a)

```
<attributes>
  <attribute>
    <name>Nombre</name>
    <value>Analizador
           diferencial analógico</value>
  </attribute>
  <attribute>
    <name>Conservación</name>
    <value>Buena</value>
  </attribute>
  ...
  <attribute>
    <name>Alto</name>
    <value>180</value>
    <unit>cm</unit>
  </attribute>
  ...
  <attribute>
    <name>Generación</name>
    <value>Primera (válvulas
                   de vacío)</value>
  </attribute>
</attributes>
```

(b)

```
<manifest ...
          identifier="VO01">
  <metadata>
    <lom:classification>
      <lom:taxonpath>
        <lom:taxon>
          <lom:entry>
            <lom:langstring>
               Catálogo de Piezas
             </lom:langstring>
          </lom:entry>
        <lom:taxon>
          <lom:entry>
           <lom:langstring>
              Computador Analógico
            </lom:langstring>

        <!—- continue -->
```

```
        <!—- cont. of the manifest -->

           </lom:entry>
         </lom:taxon>
        </lom:taxon>
       </lom:taxonpath>
     </lom:classification>
      ... <!--other metadata -->
  </metadata>
  <organizations>
     <organization identifier="default">
       <item identifier="r1"/>
       <item identifier="r2">
         <title>Explicación del analizador
                diferencial electrónico
         </title>
        </item>
        <item identifier="r3">
          <title>Biografía de José
                 García Santesmases
           </title>
        </item>
     </organization>
  </organizations>
  <resources>
    <resource identifier="r1"
              type="image/jpg">
       <file href="MIGS-0001-pl.jpg"/>
    </resource>
    <resource identifier="r2"
              type="application/pdf">
       <file href="pieza1recurso2.pdf"/>
    </resource>
    <resource identifier="r3"
              type="application/pdf">
       <file href="pieza1recurso3.html"/>
    </resource>
    <resource identifier="r4"
              type="text/xml">
       <file href="datos.xml"/>
    </resource>
  </resources>
</manifest>
```

Fig. 2. (a) Example of XML document with the data of a VO; (b) Fragment of the IMS manifest for this VO.

3.1 Evolution of the Virtual Object Model

As recognized in (Sierra et al., 2006c), adding new features to *Chasqui* usually implies an evolution of the VO model. This is also true when adding new flexible importation and exportation capabilities. For this purpose, two new features are required:

- *Composite resources*. Composite resources are sets of digital files that make up an inseparable entity. They will usually be hidden to final users. Notice that composite resources can be directly represented as IMS CP resources, since they directly support grouping a main file with a set of secondary files. Nevertheless, the aim of these resources in the VO model is completely different and it responds to an evolution of the model to satisfy a pragmatic need.
- *View resources*. Resources generated to facilitate browsing the content of composite resources as well as other housekeeping processes. These resources can not be edited, updated or deleted, in order to keep them consistent with the corresponding composite resources.

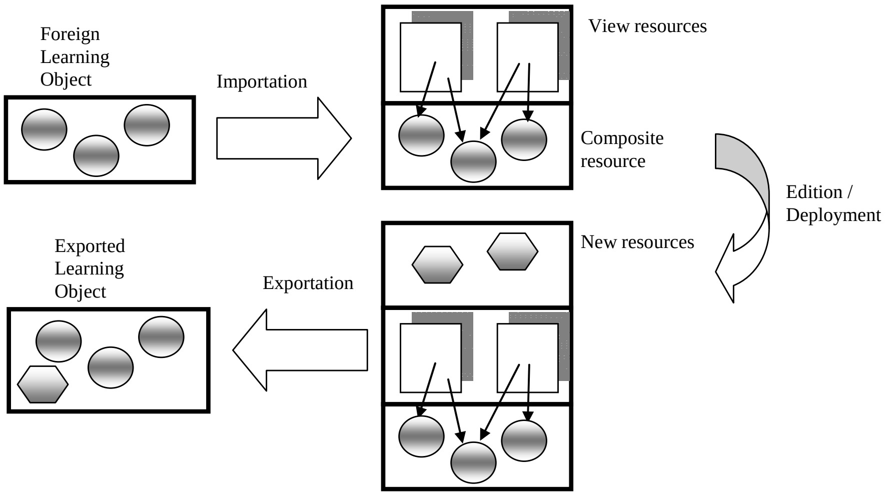

Fig. 3.Typical lifecycle of a foreign learning object inside a *Chasqui* repository.

Notice that these two extensions are very valuable in addressing the entire cycle of a foreign learning object inside a *Chasqui* repository. Indeed, when the object is imported, it will be stored as a composite resource in the resulting VO, and a suitable set of view resources will be produced. These resources will usually refer to the material comprising the imported object (e.g. if they are HTML pages, they will use the standard reference mechanisms of HTML for doing so). Users will be able to manipulate the resulting VO in the *Chasqui* repository by adding and maintaining new resources, as well as by updating the data and the metadata, although the composite and the view resources will remain unchanged. Finally, the object will be exportable, and the exportation process will be enabled by the original learning object and the new added resources (Fig. 3). Importation and exportation themselves will be instrumented using importation and exportation mappings, as described in the next point.

3.2 Importation and Exportation Mappings

The interoperability between *Chasqui* and another system dealing with a different application profile *P* is ruled with two different kinds of mappings:

- *Importation mappings*, which translate learning objects in *P* into VOs. This is usually comprised of an interpretation of the source learning objects as VOs composed of data, metatada and view resources. Besides, the original object will be stored as a composite resource.
- *Exportation mappings* translating VOs into *P* learning objects. In order to do so, the VO model needs to be represented in *P* terms. For this purpose the information stored in the composite resource, as well as in the updated data, metadata and the other normal (non view) resources of the VOs, can be used.

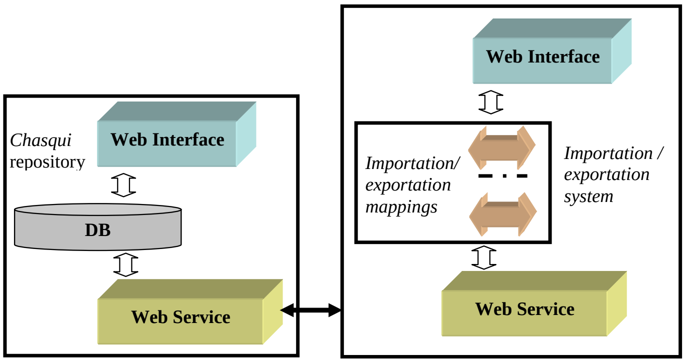

Fig. 4. Plugging the importation / exportation system into the *Chasqui* architecture

The implementation of importation / exportation mappings configures a flexible importation / exportation system that can enrich a *Chasqui* repository. Notice that it is possible to have several importation and exportation mappings to interoperate with the same application profile, each one supporting different interpretations. In this case, the user is in charge of choosing the appropriate mappings. The next section details how this system can be plugged into the overall architecture.

3.3 Evolution of the Chasqui Architecture

The *Chasqui* architecture can easily incorporate the importation / exportation system in a minimally invasive way, as depicted in Fig. 4:

- The internal organization of the repository only needs to be adapted to support composite and view resources. This extension can be accommodated in the current architecture in a straightforward way.
- Importation and exportation mappings are located in a different tier. They are made accessible with a new web application used to manage importation and exportation, as well as with a new web service interface, which acts as a common façade to the importation / exportation facilities. Besides, they are connected with the repository using the basic web service interface already available in the original architecture.

4. CASE STUDIES

This section outlines some examples that show how the importation / exportation system enhances *Chasqui* interoperability and contributes to the universalization of the *Chasqui* repositories. In point 4.1 we address the incorporation of further specialized VOs with some of their resources structured in terms of domain-specific descriptive markup languages. Point 4.2 outlines how to add support for the basic IMS CP specification. Point 4.3 addresses the support of IMS Learning Design (IMS LD) (IMS, 2003; Koper and Olivier, 2004).

4.1 Incorporating Domain-Specific Descriptive Markup Languages

XML-based domain-specific descriptive markup languages have been shown to be very valuable mechanisms in enhancing the production of VOs with document-oriented resources (Sierra et al., 2006a). Domain experts can indeed design their own markup languages, and they can use these languages to structure the documents. Then the marked documents are processed to produce presentations, which are included as resources in the VOs produced. With the inclusion of the importation / exportation system this situation, which until now has been manually managed using offline transformation support (e.g. an XSLT transformation tool), can be tackled in a general and systematic way.

Strictly speaking, the introduction of a specific markup language to structure a piece of information can be conceived as a specialization of the VO model to meet the needs of a particular expert or group of experts. This specialization comprises a set of document grammars (XML DTDs or Schemas), which can be used to author the values of the resources of the VOs. Besides, a set of transformations can be included to produce presentations of suitable XML documents. The specialized model is supported with suitable importation and exportation mappings:

- A typical importation mapping applies the transformations on the XML-structured resources to produce suitable (e.g. HTML based) presentations as view resources. Besides, the original XML documents, together with the document grammars and transformations, are stored in a composite resource. This importation behavior can be further specialized with other, more specific mappings, e.g. dealing with situations requiring more sophisticated transformation capabilities by using, for instance, advanced processing frameworks for domain-specific markup languages as that described in (Sierra et al., 2005).
- Typical exportation mappings recover the original XML documents from the corresponding composite resources. Besides, there are mappings that either preserve or ignore the newly added resources, as well as the updated data and metadata.

Notice that these mechanisms allow the editing of the corresponding XML documents by using a third-party tool like the described in (Kim, 2003), without needing to incorporate specific XML editing support into the basic authoring tool of the repository.

4.2 Supporting Generic IMS Content

Generic IMS packages can be readily incorporated into *Chasqui* as follows:

- A typical importation mapping generates a view resource with a DHTML page from the manifest's organizations, which can be used as a default user interface for browsing the resulting VO. The mapping also generates a default data file, as well as extracts the metadata from the global metadata of the package's manifest. The contents of the package itself are preserved in the corresponding composite resource. Other, more sophisticated mappings also try to extract and encode a VO from the package on the basis of the encoding conventions described in section 3.
- A typical and simple exportation mapping limits itself to recovering the original packages. Other mappings also add the new resources, as well as the data file as an additional resource, and consider the modifications performed on the metadata.

This mechanism is specially suited to enriching VOs with other IMS CP features (e.g. richer organizations and more localized metadata). These features can be authored by using external tools (e.g. Reload), and they can be exploited in third-party platforms.

4.3 Supporting IMS LD Units of Learning

IMS LD can be considered, to some extend, a specialization of IMS CP with a richer markup language for representing instructional designs. That way, the importation / exportation of IMS LD units of learning in *Chasqui* follows the same strategy as the importation / exportation of more generic IMS packages. In this case, the learning design itself is used to generate a more complete user interface for the VO as a DHTML page. Indeed, we envision a complementary web application / service that could be used to better simulate the *execution* of such units of learning.

As with IMS CP, the importation / exportation mechanisms allow the authoring of richer VOs. Now VOs will contain complex instructional designs encoded in IMS LD, they will be authored by using authoring tools oriented to IMS LD, e.g. Reload LD (Reload, 2006), and they will be deployed from *Chasqui* repositories to IMS LD-oriented players –e.g. CopperCore (Copercore, 2006). Still, they will preserve their specific nature as VOs able to be used in the context of the *Chasqui* system.

5. CONCLUSIONS AND FUTURE WORK

In this paper we have described how to extend *Chasqui* repositories with a flexible importation / exportation system that allows them to interoperate with different third-party platforms and tools dealing with different application profiles. On one hand, it lets *Chasqui* users take advantage of such external facilities to produce and maintain the VOs integrated in the system. On the other hand, it also lets them enrich VOs with additional features that can be exploited in the appropriate third-party platforms. Finally, such third-party systems can take advantage of the valuable material stored in the *Chasqui* repositories.

We have also carried out some preliminary experiments with the importation / exportation system. Currently we are starting a project oriented to implementing it using robust web-based technologies, and to integrating it in the two *Chasqui* repositories for the archeology and the computing museums. In this project we are also planning to deploy the material collected on a great variety of devices (e.g. mobile devices and interactive TV). In the near future we want to refine the architecture as a basic support for producing collections of reusable learning objects from pre-existing research and teaching material, and for integrating this material in the virtual campus at the Complutense University. We are also planning to further use the main principles behind our document-oriented paradigm for the production and maintenance of content-intensive applications (Sierra et al., 2006b) in order to maintain the exportation / importation system, as well as the rest of the *Chasqui* architecture.

6. ACKNOWLEDGEMENTS

This work has been partially supported by the Spanish Council of Education and Science (Projects TIN2004-08367-C02-02 and TIN2005-08788-C04-01).

7. REFERENCES

Adisson, A. C., 2000, Emerging Trends in Virtual Heritage. IEEE Multimedia 7(2), pp. 22-25.

Cerami, E. , 2002, Web Services Essentials. O'Reilly.

Coppercore Project, 2006, http://www.coppercore.org

Gladney, H. M., 2006, Principles for Digital Preservation. Communications of the ACM 49(2), pp. 111-116.

Hmelo-Silver, C., 2004, Problem-based learning: What and how do students learn? Educational Psychology Review, 16, pp. 235-266.

IEEE, 2002, Standard for Learning Object Metadata. IEEE Standard 1484.12.1-2002.

IMS, 2004, IMS Content Packaging Information Model Version 1.1.4 Final Specification. Retrieved June 8, 2006, from, http://www.imsglobal.org/ content/packaging.

IMS, 2003, *IMS Learning Design 1.0*. Retrieved June 8, 2006, from, http://www.imsglobal.org/learningdesign.

Kim, L. , 2003, The Official XMLSPY Handbook. Wiley Publishing.

Koper, R. and Olivier, B., 2004, Representing the Learning Design of Units of Learning, Educational Technology & Society, 7(3), pp. 97-111.

Navarro, A., Sierra, J. L., Fernández-Valmayor, A., Hernanz, H., 2005, From Chasqui to Chasqui II: an Evolution in the Conceptualization of Virtual Objects. Journal of Universal Computer Science 11(9). pp. 1518-1529.

Polsani, P.,2003, Use and Abuse of Reusable Learning Objects. Journal of Digital Information, 3(4).

Reload Project, 2006, website: http://www.reaload.ac.uk/

Sierra, J. L., Fernández-Valmayor, A, and Guinea, M., 2006a, Exploiting Author-Designed Domain-Specific Descriptive Markup Languages in the Production of Learning Content. 6th IEEE International Conference on Advanced Learning Technologies, ICALT 2006. July 5-7, Kerkrade, The Netherlands.

Sierra, J. L; Fernández-Valmayor, A; Fernández-Manjón, B., 2006b, A Document-Oriented Paradigm for the Construction of Content-Intensive Applications. The Computer Journal 49(5). pp. 562-584.

Sierra, J.L; Fernández-Valmayor, A; Guinea, M; Hernanz, H., 2006c, From Research Resources to Virtual Objects: Process model and Virtualization Experiences. Journal of Educational Technology & Society 9(3). pp. 56-68.

Sierra, J.L; Navarro, A; Fernández-Manjón, B; Fernández-Valmayor, A., 2005, Incremental Definition and Operationalization of Domain-Specific Markup Languages in ADDS. ACM SIGPLAN Notices, 40(12), pp. 28-37.

Walczak, K., Cellary, W., and White, M., 2006, Virtual Museum Exhibitions. IEEE Computer 39(3), pp. 93-95.

Wiley, D. A., 2000, Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy. In D. A. Wiley (Ed.), The Instructional Use of Learning Objects*: Online Version*. Retrieved May 23, 2006, from the World Wide Web: http://reusability.org/read/chapters/wiley.doc.