

Template proposal for the Theiere project

Introduction

One of the Theiere goals is the development of pieces of pedagogical tools available through the Internet. The main difficulty to achieve it is to co-ordinate a high number of sparse partners developing different educational resources on multiple fields. The current proposal intends to provide a common framework in order to structure and to organise these resources.

There has been a previous experience based on the Ineit-Mucon project. In this project have participated 47 partners. The main problems have been:

- the lack of co-ordination between the partners that has led to very different presentation and navigation models.
- the absence of normalised evaluation procedures.
- the difficulties to maintain/update the educational data.

However, the logical structure of the Ineit-Mucon didactic packages seemed to be adequate. Therefore, it has been adapted in the Theiere project. In order to take advantage of this logical structure and to overcome the previous problems, an approach based on XML has been proposed. This markup language has been used to represent the different didactic resources from the course structure to its contents, including the evaluation items. XML has been applied in several learning environments and standard definitions for educational metadata have been proposed. However, the current project requires a specific XML specification, given the next features.

Hierarchical structure

The first aspect is the hierarchical structure of the project. It is divided into 8 main groups: Computers and Information Systems, Communications, Electronics, Power Systems, Instrumentation and Control, Internet Services and Applications, Fundamental, and Virtual Labs. Each group has a lead site which co-ordinates the development of didactic resources performed by the group members. For example, the "Internet Services" group is driven by Vigo and Valencia lead sites. The remainder members are Valence, Cagliari, Lyon, Ilmenau, Nancy.

This hierarchical and distributed structure has been considered to divide the development of didactic resources into several levels. The first level is represented by the "package" entity and it is associated to each project group. For example, the XML document in Figure 1, represents the "Internet Services" package:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Package SYSTEM "Package.dtd">
<Package>
  <Description>This package is a new component of the Theiere project and it is divided into
  several parts.
  One is addressed to describe the main Internet services such as WWW, FTP, or e-mail...
</Description>
  <SubPackages Title="Internet Services" Author="jrial@ait.uvigo.es"
Resource="InternetServicesSubPackage.xml"/>
  <SubPackages Title="Internet Technologies" Author="fbuendia@disca.upv.es"
Resource="InternetTechnologiesSubPackage.xml"/>
  ...
</Package>
```

Figure 1.- Example of "package" entity.

The "Internet Services" package is based on a Document Definition Type (DTD) which is assigned in the <!DOCTYPE> entity reference. It defines the syntax for the remainder elements. Figure 2 shows the "Package.dtd".

```
<?xml version="1.0" encoding="UTF-8"?>
<!--DTD example for packages -->
<!ELEMENT Package (Description, SubPackages*)>
<!ELEMENT Description (#PCDATA)>
<!ELEMENT SubPackages EMPTY>
<!ATTLIST SubPackages
    Title CDATA #REQUIRED
    Author CDATA #IMPLIED
    Resource CDATA #IMPLIED
>
```

Figure 2.- Example of DTD.

The root element is <Package> and it contains a sequence of one <Description> element and 0 or more <Subpackage> elements. The <Description> elements contain PCDATA, that is, text information.

The <Subpackages> elements contain a list of attributes such as:

- *Title* which describes the name of the subpackage.
- *Author* which includes the e-mail address of the subpackage responsible. This information will be useful to contact with such person.
- *Resource* which contains the name of the XML document that describes the subpackage content. This level hierarchy contributes to the project modularity. In this way, a change in one level of the hierarchy does not affect to the other levels.

Once the DTD is defined, the generation of an associated XML document is straightforward. Some specialised XML editors (e.g. XML Spy ©) assist the author in this process. It can be also introduced using a plain text editor. It is particularly important to end the tags. In Figure 1, there have been used to different ways:

- An explicit end tag such as </Description> or </Package>.
- An implicit end tag such as <SubPackages Title=" ..." />

An important notice is that the corresponding lead site is in charge of introducing each XML package document..

The next level is represented by the "subpackage" entities and they have to be developed by the group members. An example is represented in Figure 3.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE SubPackage SYSTEM "SubPackage.dtd">
<SubPackage>
  <Units Title="WWW" Author="jrial@ait.uvigo.es" Resource="WWWUnit.xml"/>
  <Units Title="FTP" Author="jrial@ait.uvigo.es" Resource="FTPUnit.xml"/>
</SubPackage>
```

Figure 3.- Example of "subpackage" entity.

In this case, the <SubPackage> elements is composed by 1 or more <Units> elements. Each <Unit> element is also characterised with an attribute list (*Title, Author, Resource*). The *Resource* attribute addresses the Unit definition. Figure 4 shown an example of such XML "Unit" document.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Unit SYSTEM "Unit.dtd">
<Unit>
  <Objectives>Description of WWW fundamentals</Objectives>
  <Objectives>Main WWW applications</Objectives>
  <Contents Title="Basic concepts" Author="Felix Buendía" Resource="BasicConcepts.html"/>
  <Contents Title="Advanced Concepts" Author="Juan Carlos Burguillo"
Resource="AdvancedConcepts.xml"/>
  <References>Stallings, 2000</References>
</Unit>
```

Figure 4.- Example of "unit" entity.

The <Unit> element is composed by 0 or more <Objectives> elements, 1 or more <Contents> elements, and 0 or more <References> elements. The < Objectives > elements contain PCDATA, that is, text information. Each <Contents> element is also characterised with an attribute list (*Title, Author, Resource*). In this case, the resource attribute can be assigned to an HTML document (e.g. BasicConcepts.html) or a XML-based document if it has a given structure. Figure 5 shows an example of XML "Content" document. The "Content" definition provides a way to assign different interpretations of the same data. In this case, the "MemoryManagementAdvancedConcepts.txt" file is based on a paragraph format but it can be also processed using lower level units such as sentences or words.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Unit SYSTEM "Content.dtd">
<Content>
  <Label>About the basics of memory management in an operating system</Label>
  <Author>Félix Buendía García</Author>
  Id="GM-Intro"
  Category="browsing"
  Type="text"
  Units="paragraphs"
  Value="Memory ManagementAdvancedConcepts.txt"
</Content>
```

Figure 5.- Example of "Content" entity.

In a first attempt, the content resources are limited to HTML documents which can include embedded images, animations, applets, or other elements. In the future, several content types can be defined and managed according a given pattern. Once the "Units" and "Subpackages" have been implemented as XML documents by the group members, they have to send them to the corresponding lead site. The lead site will transmit the complete package information to the Technical Support Group. At this time, only an "English" version of the packages is planned to be developed. In the future, other languages can be used.

Package publication

The global Theiere package information can be displayed in a Web page such as the one shown in Figure 6.

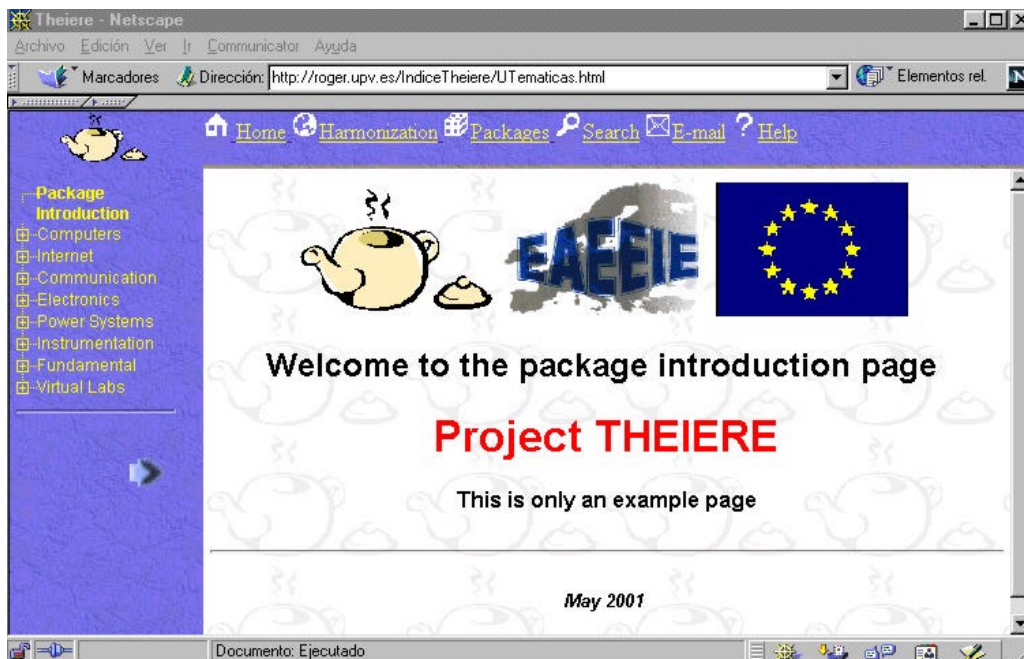


Figure 6.- Example of Web page.

It is only a prototype page but its main advantage is that it can be generated in a dynamic way from the XML specification. In fact, the current version is completely dynamic, that is every time a information item is pushed the server generates a new Web page with the corresponding structure items (left frame) and the content items (main frame). That feature is useful in adaptive environments but it burdens the performance. An "applet" version is being developed to process the XML specification in the client browser.

The current version provides some mechanisms to navigate through the Package items:

- Accessing directly to the structure items located at the left frame and used as an index.
- Using the navigation arrows: left and right (within a certain level) and up and down (to change to an upper or lower level).

The left frame only shows a maximum of two option levels. A reference to the previous levels is displayed in the top of the index.

The prototype Web page can be accessed at the next address:

<http://roger.disca.upv.es/IndiceTheiere/Principal.html>

The structure templates and an HTML version of this document can be obtained in the Harmonisation option. Other options such as Search, E-mail, or Help are not yet available.