CUSTOMDP PROJECT
THE FINAL DISSEMINATION EVENT
Athens October 30th and 31st 2003

The CustomDP system as a cross media publishing platform – learning objects, metadata and XML

The CustomDP system for managing learning content

Timo Järvinen
VTT Information Technology
P.O. Box 12041, FIN-02044 VTT, Finland
timo.h.jarvinen@vtt.fi

Abstract

This presentation gives an overview of the CustomDP system. CustomDP supports four kinds of learning material: SCOs (Sharable Content Objects), sharable resources, referenced content and aggregations. There are four user groups each with a different set of permissions: learners, content creators, validators and chief editors. Learners have the least permissions, mainly for searching and reading learning material. Metadata makes it possible for the users to find learning material they really need.

The CustomDP system has a typical three-tier application architecture. The system is be used with a web browser. The user interfaces are mostly HTML. The tool for creating aggregations is implemented as a Java applet. The HTML user interfaces are generated with Java servlets in co-operation with an open-source presentation framework, Barracuda. A relational database is the major data repository. However, images and XML files are stored in the file system. The system runs in two locations. Most of the servlets and the database run in servers located in Finland. Servlets for displaying content run in Germany.

1. Introduction

The purpose of this presentation is to give an overview of the CustomDP system.

2. Learning content types

The ideas related to learning content types are borrowed from the SCORM, (Sharable Content Object Reference Model) [ADL, 2001]. Metadata is used to describe the content, and most of the metadata is the same for all learning content types. CustomDP has four learning content types:

A *SCO (Sharable Content Object)* is content that is stored in the CustomDP database. Content has been created according to CustomDP SCO DTD, and it is stored in the XML format.

Referenced content is a reference to content that is been stored outside CustomDP. The reference might be URL or an ISBN number. The content to be referenced might be a web page, book.

A *Sharable resource* is a resource that can be used by zero or more SCOs. It is stored in the CustomDP database only once. SCOs using the sharable resource contain a reference to the single sharable resource. The content of a sharable resource is any file such as an image, video clip, text document, etc.

An *Aggregation* contains other aggregations, SCOs or referenced content. There are four levels in the aggregation hierarchy. A *course* contains one or more modules. A *module* contains one or more topics. A *topic* contains one or more SCOs or referenced content. Example of an aggregation:

(Course) Digital printing systems
(Module) Digital workflow
(Topic) From design to finished documents
(SCO) Case: Xerox Manual + Book factory
(Topic) Basic adjustments
(SCO) Tasks of RIP
(SCO) Imposition
(Module) Digital prepress
(Topic) Digital images
(SCO) Imaging Basics
(Topic) Digital Colour
(SCO) Colour Theory (V1)
(Referenced content) Glossary of Color Management Terms

3. User groups

The CustomDP system has four user groups – learners, content creators, validators, and chief editors – learners have the least permissions and the chief editors the most. Each user group has its own permissions plus all the permissions of the user groups above. The essential permissions are illustrated in Figure 1.

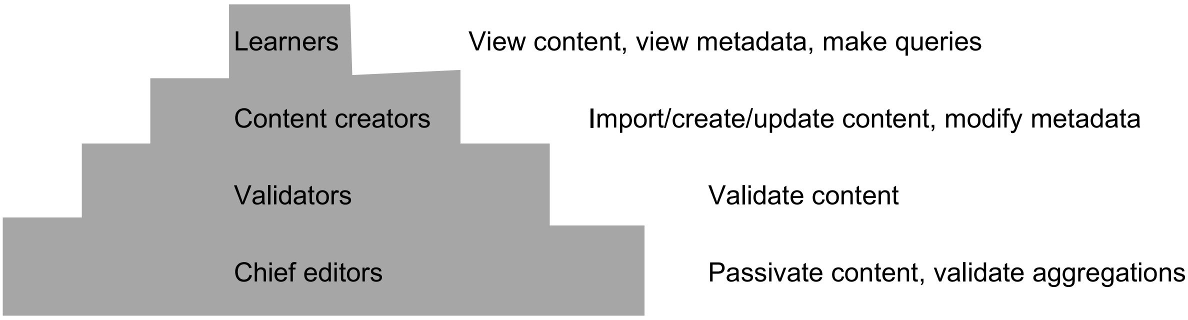
Learners
View content, view metadata, make queries
Content creators
Import/create/update content, modify metadata
Validators
Validate content
Chief editors
Passivate content, validate aggregations

Figure 1. The user groups and their main permissions in the CustomDP system, learners have the least permissions and chief editors the most.

4. Workflow

The system has workflow features related to content creation processes of SCOs, sharable resources and aggregations.

SCOs and sharable resources

1. Content is created outside the CustomDP system.
2. It is imported using the upload form. Metadata can be modified in this context.
3. As the content and metadata are valid, the SCO or sharable resource will be waiting for validation.
4. If the validator accepts the content and metadata, the SCO or sharable resource is published to all users. If the validator rejects the content and/or metadata, the SCO or sharable resource is passivated. It must be imported again, as the required corrections have been made.

Aggregations

1. Content and metadata are created using the Aggregation Editor Tool. As the aggregation is finished, the content creator has two options. First, he/she can publish the aggregation directly.

Second, he/she may request the chief editor to validate the aggregation. Only a course might be validated. Modules and topics can only be published directly.

The courses can be viewed by the learners also in draft state. This way it is possible to start using a course even though it is not completely ready.

The following step is needed only, if the content creator wants the course to be validated.

2. If the chief editor accepts the content and metadata, the course is published to all users. If the chief editor rejects the content and/or metadata, the course is moved back to draft status. The content creator can continue working it. Later, he/she may request validation again.

5. Metadata

Metadata is used in CustomDP to describe both learning content and users. Metadata enables searching learning content and users with various search criteria. Metadata is used to describe learning content in the bookmark lists, search result list and metadata page. It is used to describe users in the search result list and account info page. Users excluding learners can modify metadata using the web forms.

Learning content is described with five main metadata field groups:

Metadata field group	**Fields**
General	identifier, resource type (i.e. learning content type), main language, title, learning goals, subject, module, keywords, level, main target group, educational intention, see also (web address), see also (learning object identifier), see also (free description)
Life cycle	publication date, status, author, up-to-date until, availability, is based on
Educational	learning resource type, interactivity level, intended user role, learning contexts, typical learner age, difficulty, typical learning time, reading guide
Meta-metadata	metadata creator, metadata language
Resource-type specific	SCO and sharable resource: main format, new version of, translation of, translators, other contributors, copyright owner (person), copyright owner (company), system requirements Referenced content: main format, publication type, publisher, ISBN number, size, online location, reliability of content, suitability for training purposes Aggregation: new version of, translation of, taxonomy level, validation type

There is the following metadata of users:

- Username *
- Password *
- User group *
- Main subject *
- Last name *
- First name *
- E-mail *
- Preferred user interface language *
- Country of residence
- Gender
- Age
- School
- Course code
- Main activity
- Education (completed or current)
- Current field of employment
- Current employer
- Job title
- Work experience

Fields marked with * are mandatory:

6. Support for many languages

CustomDP currently supports six languages: Dutch, English, Finnish, German, Greek and Swedish. Thanks to the generic implementation, more languages with Latin alphabets could be quite easily supported. Other languages such as Greek require programming work. Support for many languages is implemented using three technologies:

1. The language versions of the static user interface texts are stored in *resource bundles*. ResourceBundle is a Java technology (see http://java.sun.com/j2se/1.4.2/docs/api/java/util/ResourceBundle.html). User interface texts are stored in simple text files with specific format and name.
2. The language versions of the metadata field names are stored in the relational database.
3. The language versions of some very commonly used terms, such as names of user groups, are hard-coded within the Java code.

The Greek language requires special treatment. The Greek characters are converted into Unicode-encoded characters before storing. For example, α (alpha) is converted into following character sequence: \u03b1. After retrieving the characters from the database or resource bundle, they are converted back to the Greek characters.

The following tools were used for conversion:

- The International Components for Unicode library was used for automatic conversion inside Java program (http://oss.software.ibm.com/icu4j/)
- The native2ascii tool was used in manual conversions (http://java.sun.com/j2se/1.4.2/docs/tooldocs/windows/native2ascii.html)

7. Implementation

The three-tier application architecture is illustrated in figure 2. Clients use the CustomDP system with web browsers. Most of the system features are implemented as HTML pages that are created dynamically using Java servlets. The Aggregation Editor Tool is a tool for creating courses and other content aggregations. It is implemented as a Java applet. It is also implemented as a Java application, because some firewalls prevented using the Java applet. To use the database, the Aggregation Editor Tool uses servlets that return serialized Java objects. All servlets use DP API written in Java to use the database. The purpose of the DB API is to hide the details of the database implementation from the servlets. CustomDP uses a relational database to store persistent data. Files such as SCO XML files and sharable resources are stored on the server computer file system. The system runs in two locations. Most of the servlets and the database run in servers located in Finland. Servlets for displaying content run in Germany.

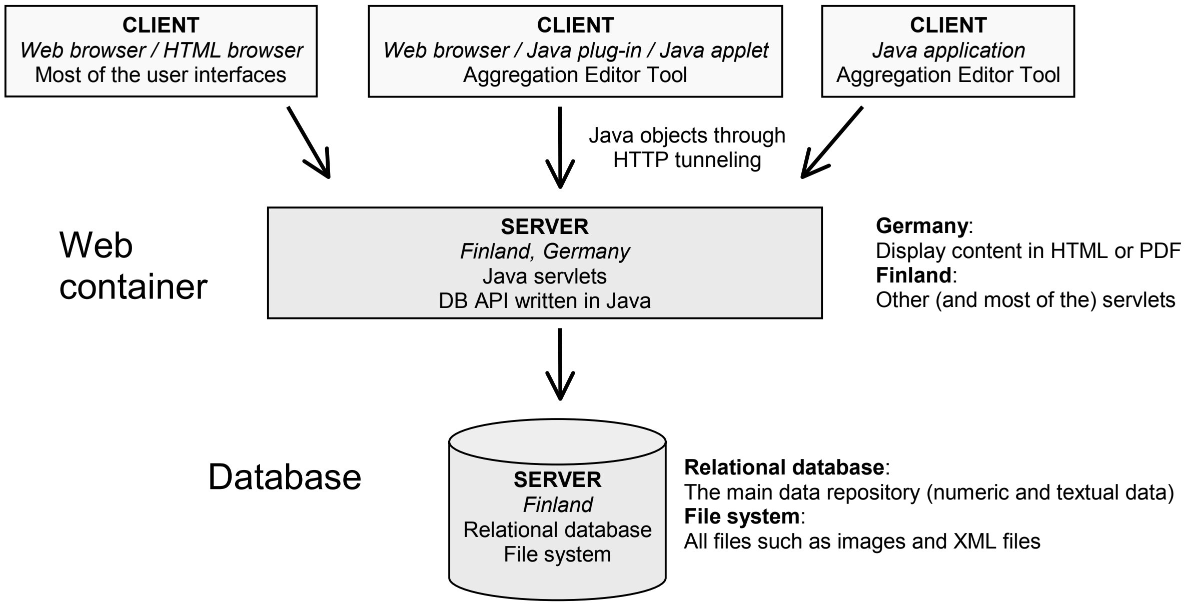

Figure 2. The three -tier architecture of the CustomDP system.

The following tools were used in implementation:

Server computers	Windows 2000, Windows NT
Programming language	Java 1.4
Web server	Apache HTTP server
Servlet container	Apache Tomcat (servlet spec 2.2)
Presentation framework	Barracuda (http://barracudamvc.org/)
Relational database	Informix Dynamic Server
JDBC driver	Informix (type 4, native protocol, pure Java)

References

ADL, 2001 Advanced Distributed Learning. Sharable Content Object Reference Model, Version 1.1. Dodds, P. & al. (ed) 223 pp. 2001. (http://www.adlnet.org/ADLDOCS/Document/SCORM1.1.pdf)