Distributed Content Management Framework for Digital Museum Exhibitions

Samuel Cruz-Lara, Bai-Hsuen Chen, Jen-Shin Hong

Distributed Content Management Framework for Digital Museum Exhibitions

Samuel CRUZ-LARA[1], Bai-Hsuen CHEN[2], Jen-Shin HONG[2]

[1]LORIA / INRIA Lorraine, FRANCE
Samuel.Cruz-Lara@loria.fr

[2]National Chi Nan University, TAIWAN
jshong@csie.ncnu.edu.tw

Abstract

A digital library focuses on conserving, cataloguing, accessing, and tracking the usage of digitized material. On the other hand, a digital museum, other than being a simple digital archive, indeed emphasizes more on providing users with highly educational and motivating exhibitions. Online exhibitions often consist of a variety of multimedia objects such as web pages, animation, and video clips. One can design different exhibitions about the same topic for children, experts, novices, high bandwidth users, low bandwidth users, all using the same set of digital artifacts. The difficulty here is that it is time consuming to produce illustrative and intriguing online exhibitions. To efficiently transform the organized media objects deposited in the digital archive into educational experiences, there is a need of a novel content management framework for organizing digital collections and for quickly selecting, integrating, and composing objects from the collection to produce exhibitions of different presentation styles.

Also, in order to retrieve the data distributed in various digital museums, we have to design a distributed software architecture through which it might be possible to access and to share multimedia resources, which would be spread among different servers. The general idea is that one should be able to bring together all existing multimedia resources, in order to provide any user with a global access to these. Obviously, there is a need for these resources to be created, and above all, maintained at a place where there is the competence to do so. But, there can also be specific constraints that can preclude some given resources to be deported to another site than the site that has originally created them.

Starting from 1996, Taiwan has initiated a multi-year digital museum project to digitally archive precious cultural collections. The National Chi Nan University (NCNU) has actively participated to the project and has successfully constructed the "Butterfly Digital Museum" [1], the "Lanyu Digital Museum" [2] and the "Ali-Mountain Digital Museum" (http://dlm.ncnu.edu.tw). Over the years, the "Language and Dialog" Project LORIA / INRIA Lorraine) has devoted much effort on SILFIDE[1] [3] and MLIS-ELAN[2] [4] projects. SILFIDE and MLIS-ELAN are both distributed language resources systems, offering access to existing linguistic resources to their potential users throughout Europe. In the framework of SILFIDE and MLIS-ELAN, all flowing data (i.e. requests, results, messages, etc.), as well as all

[1] Serveur Intéractif pour la Langue Française, son Identité sa Diffusion et son Etude.
[2] Multi Lingual Information Society – European Languages Activity Network.

information about users and user's working spaces, have been encoded using XML[3] (http://www.loria.fr/equipes/led).

The purpose of this project is to incorporate the NCNU and INRIA research effort for developing an XML-based distributed hypermedia digital museum content management framework. Design issues of the framework and prototype system implementation will be addressed in this paper.

1 Introduction.

A digital museum designer implements the virtual exhibitions using digital images, web pages, animation, video clips, and other multimedia gadgets [5]. Although there is no space and related limitations in the cyber world, there are other factors that a digital designer needs to consider. For instance, the bandwidth that is available to different users can vary tremendously. Attention span is another factor. While an adult may be willing to go through long explanations in text, a child may prefer graphic-intensive slideshows. Furthermore, the sophistication of users should also be considered. An expert and a novice apparently will expect different experiences when browsing through a digital museum. With the flexibility allowed to the digital form of media presentation, a digital museum should try to accommodate as many different user needs as possible. And a user, when browsing through an exhibition, can simply click and choose the style most suitable.

While most people might agree that such a multi-style exhibition scheme for different users is desirable, there are not so many such exhibitions on the Web today. The reason is that using current techniques, each style of the same exhibition needs to be constructed separately. For online exhibitions involving video, animation, or SMIL[4]-based shows, the realization of an exhibition scenario usually requires tedious multimedia composing, and is thereby very time-consuming. It is also quite cumbersome to modify the information content afterwards. If an online exhibition uses mainly HTML-based web pages, the resulting pages are relatively easy to compose, but they are also tight up with the visual art design and are thereby difficult to be modified by the content-provider alone. Furthermore, once the information content grows to a significant amount, the "hyperlinks" between associated exhibitions web pages are difficult to track and maintain.

The second problem with the handcrafted online exhibition approach is that the content can only be presented to the user in a fixed presentation style. Since the Web users may have different multimedia and bandwidth capacities, it is difficult to design a one-size-fit-all exhibition. In the literature, while large and coordinated efforts have focused on standardizing media formats, providing shared access to museum databases, and developing search mechanisms for data retrieval [6], little emphasis is placed on designing mechanisms for turning the digitized museum collection into educational experiences for users. Our goal is to design an optimal solution for building online digital museum exhibitions.

In this paper, we will present the design of an XSL[5]-based Multi-Presentation Content Management System (XMP-CMS) that provides a novel approach for organizing, integrating, and composing the digital museum collection into multi-style exhibitions to accommodate different user needs. Furthermore, for the content management framework to be "interoperable" among various distributed digital museum content servers, we also propose an "SOAP"-based framework for the user to access content deposited in various servers. In the

[3] http://www.w3.org/XML
[4] http://www.w3.org/AudioVideo
[5] http://www.w3.org/Style/XSL

following sections, we will introduce the design issues and the framework of the XMP-CMS. A tested system implemented will then be addressed to demonstrate the performance of the proposed framework. In the following paragraphs, the technical approaches and expected results will be addressed.

2 Design Issues of Digital Museum Exhibitions.

In this section we describe some design issues that our framework for constructing online exhibitions needs to address.

An exhibition can be regarded as a choreographed presentation of a certain topic using artifacts from a museum. We call the choreography the script of the exhibition. An online exhibition, from the user's viewpoint, can be treated as a sequence of web pages. (A webpage can, of course, contain complex objects such as a video clip.) We call each of these web pages an exhibition element. In our methodology, we address three issues that we consider central to producing flexible and easy to use exhibitions.

1. There should be an easy way to produce exhibition elements from the digital archive of the digital museum.
2. Once an exhibition is produced, there should be an easy way to create different presentation styles suitable for different users.
3. There should be an easy way to re-arrange an existing exhibition to accommodate different user need and to make new exhibitions from existing ones.

In the following we address our solution to these three issues.

The digital artifacts of the museum are stored in a multimedia database. Our approach treats each of them as a building block and provides a mechanism for drawing them easily to form XML documents. Each XML document is then an exhibition element.

For the second point, we note that an exhibition should accommodate users of different interests, backgrounds, sophistication, and computer/network capacity. Conventional online exhibitions often only provide a single presentation for all the users. If one must convey the same message to all the users, one is faced with the problem of choosing a single representational form that is deemed most suitable for the largest user population. The outcome is usually a compromise. This certainly is not an optimal solution.

In our approach we provide an easy way to generate multi-style exhibitions with the same exhibition elements. This means that the same exhibition may appear in different ways to different users, with different multimedia styles and levels of detail.

The deciding factors include:

1. The user's network bandwidth;
2. The user's web browser;
3. The user's profile.

For example, when interacting with school children with multimedia-capable computers and high bandwidth network, the system can provide high-resolution graphics and slide shows overlaid with narrative audio data. On the other extreme, for a domain expert keen on information content, the system may only provide static pages with loads of text information but little visual fanfare to reduce the data transmission. This is done by designing

different style sheets (XSL) for different purposes. Coupling the exhibition elements with a specific style sheet produces an exhibition for a user group. Note, for instance, that a style sheet for "lower bandwidth" consumers will automatically drop all the audio and video.

The third issue also has to do with different user needs. An exhibition designed according to a script may be suitable for adults but too long for children. The designer may want to re-organize the order of the exhibition elements in the sequence for different people. She may also want to combine several exhibitions and make a more extensive one. Our method provides an easy way to make such "film-editing".

It is also desirable to make an exhibition into a slide show, which automatically runs through all the exhibition elements. We call this *auto-navigation*, and is also provided in our method.

3 A Distributed Content Management Framework for Digital Museum.

In this section, we will first describe a Content Management System (CMS) [2] that has been used, in particular, in the framework of the NCNU's project "The Lanyu Digital Museum". Next, we will present a generalization of the CMS over a distributed network. The organization and implementation of this distributed network is based on LORIA / INRIA Lorraine's projects MLIS-ELAN and SILFIDE.

3.1 A Content Management Framework.

In this section we present in detail a Content Management System (CMS) framework for a digital museum exhibition. A CMS-based system takes a realistic approach to construct digital museum exhibitions based on XML and XSL techniques [7].

The exhibition scripts of the digital museum are differentiated to a sequence of "exhibition elements" based on an XML-based conceptual hypermedia document model. The conceptual representation of the exhibition elements is composed from several concrete digital objects managed by museum database. Using the XSL to present the XML documents, the system is able to select in a more or less automatic way the set of suitable presentations according to the user needs.

The main components of the XMP-CMS framework are the following:

- An XML-based conceptual document model, able to express a unique, formal representation of the content of a typical hypermedia document in digital museum exhibitions. The document content should include different types of information: text, data, table, graphics, images, hyperlinks, and so on.
- An authoring environment for editing the layouts and media presentation effects of XSL documents which provide different presentation styles for the XML documents of exhibition elements.
- An database management system providing operational environment for creating, viewing, editing, storing, and retrieving documents, through a set of tools for manipulating texts, table, graphic and images.
- A user-interface for selecting the suitable mappings between the contents and the presentations both during the initial document authoring activity, and during its successive reading.

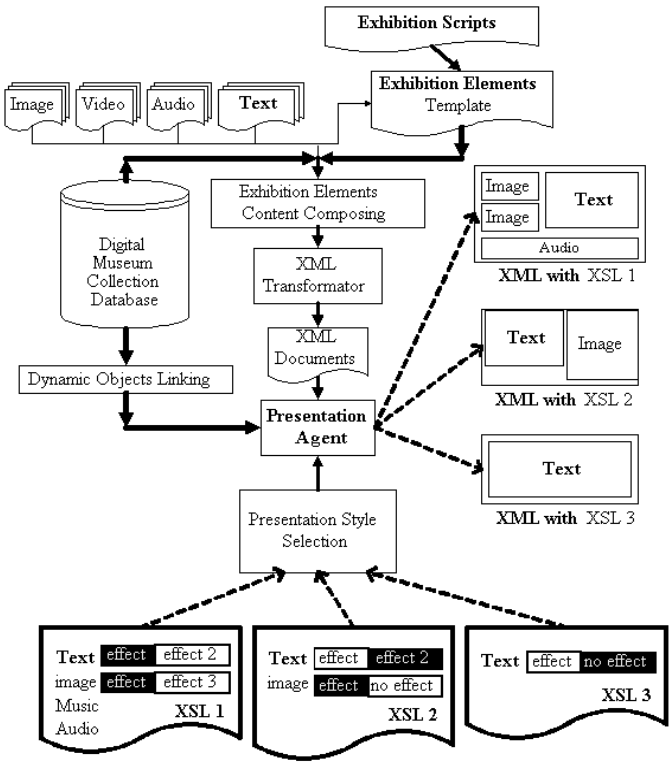

Figure. 1. The framework of the XSL-based Multi-Presentation Content Management System. An XML-based exhibition element can be presented in multi-styles using different XSL documents.

3.1.1 XML-based Hypermedia Document Model for Exhibition Elements.

Generally speaking, the common document component types [8] in a typical multimedia document could include the structural text, data series (e.g., spreadsheet-like tables), graphic representations of functions and tabular data, geometric drawing, animations, video, etc. Yet, in this stage, our XMP-CMS considers the following simplified hypermedia documents component types for the digital museum exhibition elements:

- Description text that are classified in three subclasses:
  1. The title;
  2. The main description of an exhibition element;
  3. The hyperlink address representing the ancestor nodes, descendant nodes, collateral nodes, and media objects.
- Images: including the information images and background images.
- Audios: including content narration of the exhibition element and background music.

3.1.2 XSL-based Multi-Style Presentation.

The eXtensible Style sheet Language (XSL) is used for versatile/convenient presentation of the exhibition element XML documents over WWW. The XSL documents specify the layout and presentation style of the title, main description text, image, audio, and hyperlinks in an exhibition element. Furthermore, to give the users a vivid and rich

presentation of the museum collections, numerous presentation and transition effects of the texts, images are incorporated into the XSL documents using sophisticated Java-Script technology.

In our design, a XSL document contains three major functional blocks:

1. XML Parser: to analyze and extract the tags in the XML documents representing exhibition elements;
2. Layout Modules: for organizing the layout of the textual/multimedia objects appears in an exhibition element;
3. Media Effect Modules: for incorporating various special effects for the media presentation using Java-Script techniques.

3.1.3 The "Lanyu Digital Museum".

Based on the aforementioned XSL-based Multi-Presentation Content Management System (XMP-CMS) framework, we have built a digital museum for the Lanyu Island and its inhabitants, the Yami people.

Based on the XMP-CMS framework proposed, we have implemented a Digital Museum Resource Management System for constructing the Lanyu Virtual Exhibition Hall. The exhibition scripts provided by the content expert, are first partitioned into a sequence of exhibition elements, and are organized using a relational database. The information contents in the database are also transformed into a group of XML documents representing the respective scenes in the exhibition scripts. The cross-linkages between the associated XML documents act as the dynamic hyperlinks in the resulting web pages presented to the users.

To present the XML documents in different fashions for users of different interests, backgrounds, sophistication, and computer/network capacity, several groups of XSL documents are used for multi-style presentation. Various web-based Java-scripting codes are inserted into the XSL documents to provide more intriguing multimedia presentation of the exhibitions elements.

3.1.3.1 Content Authoring Interface of the Lanyu Digital Museum.

The content authoring interfaces provide the content experts a user-friendly tool to input/organize the multimedia objects and the exhibition elements. Through the interface, the context experts first create the fundamental hierarchical structure of the exhibition based on the exhibition scripts designed. The textual content of each exhibition elements is then input through the content authoring interface using simple "fill-the-blank" method. In our system, the museum curators can easily compose the online exhibition without any technical backgrounds for the HTML pages composition. Figure 2 shows the user interface for the authoring of exhibition elements and hierarchical structure of exhibition scripts.

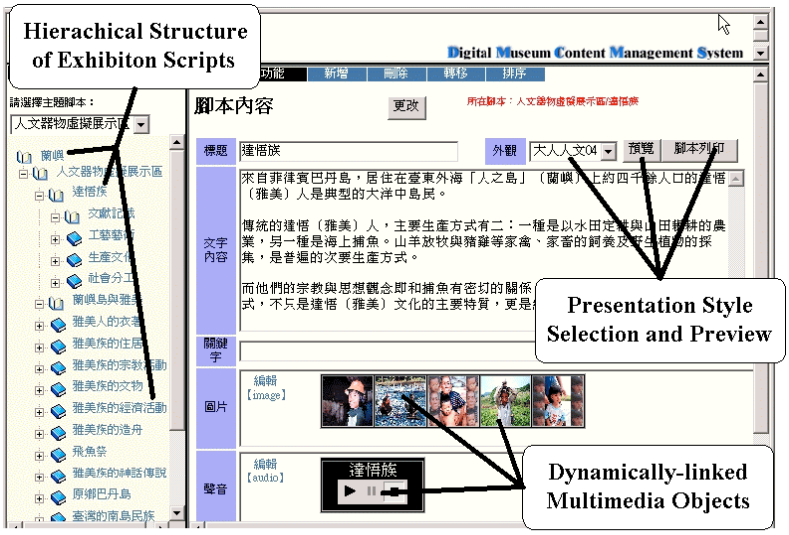

Figure. 2. The user interface for the authoring of exhibition elements and hierarchical structures of exhibition scripts.

In the XMP-CMS framework, all the texts, images, audio clips of the exhibition elements are deposited separately in the multimedia object database. The content expert can browse or search the multimedia database to locate specific media object to link to an exhibition element. The information contents of all exhibition elements are managed using a relational database. These information content are also translated into XML documents for XSL-based multi-style presentations. For each exhibition element, a keyword is provided for the purpose of fine-grained content retrieval of the museum exhibitions. Furthermore, the system provides easy-to-use "film-editing" functions for the content author to re-organize the hierarchical structures of the exhibition scripts for different user groups. Figure 3 shows the content authoring interface in the Lanyu Digital Museum.

3.2 A Distributed Content Management Framework.

Figure 3 shows the major components of the distributed digital museum resource management system. The digital museum management system supports the efficient execution of the essential administrative and documents organization/presentation related activities, and is characterized by the following features:

1. Input of multimedia digital objects, including the text, images, audio, video.
2. Organizing, composing, and integration of the XML-based multimedia exhibition element documents.
3. Management of the XSL documents for presentation.
4. Definition and management of user profiles.
5. Management of the document presentations according the user needs and profiles.

The proposed distributed framework has been tested in the MLIS-ELAN project [4]. One of objectives of MLIS-ELAN has been to define a general XML-based distributed software architecture through which it might be possible to access and to share resources, which would be spread among different servers [9]. There are different reasons why to go about in this fashion. First, it should be considered that there already exist several sites all around which actually act as brokers for different types of resources that are stocked in some

kind of databases. The general idea is that one should be able to bring together all the existing resources, in order to provide any user with a global access to these. In some cases this will lead to more coherence between these databases since redundancies might be detected or complementary resources put into correspondence. Second, the classical view of a centralized database containing all the information in a given domain is far from applicable to most kind of resources, because of their intrinsic diversity, for example. Also, there is a need for these resources to be created, and above all, maintained at a place where there is the competence to do so. Finally, there can be specific constraints that can preclude some given resources to be deported to another site than the site that has originally created them. For example, we can have "strong" conditions on the actual distribution of "electronic versions" of some resources. It is thus more sensible not to take the risk of hampering the agreement by overly spreading the corresponding contents. In the MLIS-ELAN proposed network, each resource is only accessible through specific queries that can thus be controlled as to their actual applicability.

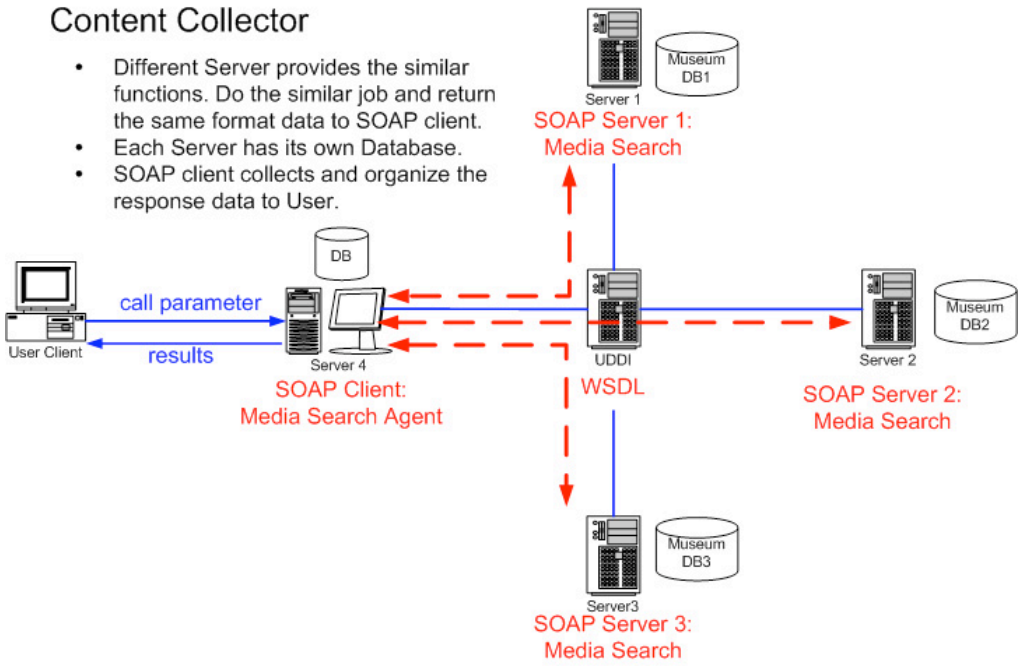

Figure. 3. A distributed digital museum resource management system.

Our main objective is to define a general distributed software architecture through which it might be possible to access and to share multimedia resources, which would be spread among different servers.

From the user's point of view, there should not be much change in the way the resources are to be accessed, which means that whether they are one or several servers should be rather transparent to him.

As we will see, adopting a distributed framework, as opposed to the classical view of a centralized database, induces several specific problems for which NCNU-LORIA collaboration is trying to provide some plausible answers. Among those, we will have to deal specially with the problem of broadcasting queries to different servers and conversely combining the corresponding result sets. As an example, statistics can only be dealt with in our distributed architecture if part of the computation is kept on the remote servers' sides and part is carried out locally (on the access server).

All tasks dealing with the user interface should be concentrated on the client side, while searching and other computationally intensive operations should be accomplished by the server.

Most of these ideas have been applied in MLIS-ELAN project ([4], [9]). The MLIS-ELAN project developed a general XML-based distributed software architecture for accessing and sharing natural language resources in an opened client/server environment. Natural language resources has been XML-encoded (by using the XML form of the "Text Encoding Initiative"[6] and CORBA[7] has been used to allow, in particular, communication between servers.

Based in our experience with MLIS-ELAN, we think that the multimedia resources network may have the following characteristics:

- each server is an autonomous unit containing its own data. For instance, all resources should be XML-encoded;
- each server act as a "broker" and transmit, if needed, the request to other servers in the network which are known to it. All flowing messages and requests between servers should also be XML-encoded;
- one server is accessible to registered users through a general purpose Java-compatible web browser. It would also be interesting to have an XSLT capable browser.

3.2.1 A hierarchy of users.

We should consider that there are several categories of users in the network:

1. Administrators: they organize LOCAL media on servers;
2. Designers: they design multimedia exhibitions;
3. End Users; they access exhibitions that have been built by designers.

3.2.1.1 Administrators.

Administrators are directly concerned with maintaining LOCAL servers. So each server in the network has an administrator who uses the Content Management System for:

- Multimedia Resources Database Feeding
- Multimedia Resources Integration
- Digitization Process
- Multimedia Resources Composition

Obviously, each server in the network may have several administrators. The important concept to keep in mind is that, there is a need for the resources belonging to each server in the network, to be created, and above all, maintained at a place where there is a competence to do so. An administrator can not manage any resource that may not be accessed locally in a server.

---

[6] http://www.tei-c.org
[7] http://www.corba.org

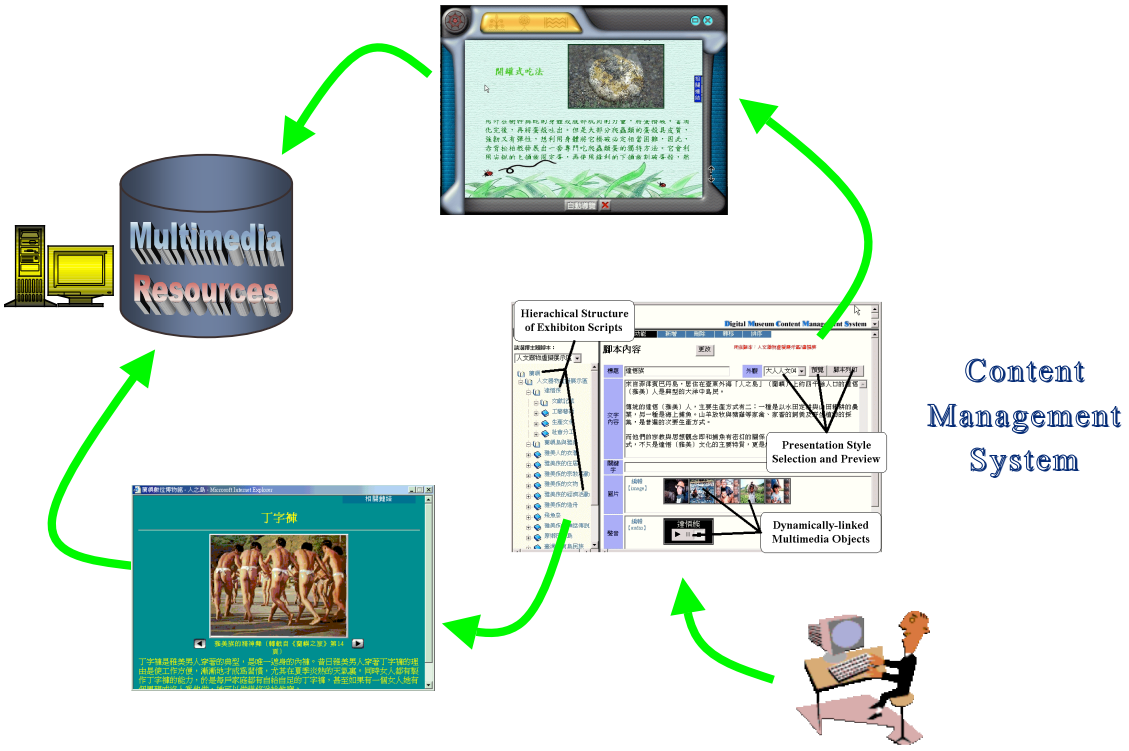

Figure 4. Administration of resources on a server.

3.2.1.2 Designers.

Designers are concerned with all servers in the network. They are able to access, obviously, all resources being stocked on the server that they have used to connect into the network, but they can also access any resource being stocked on any server belonging to the network.

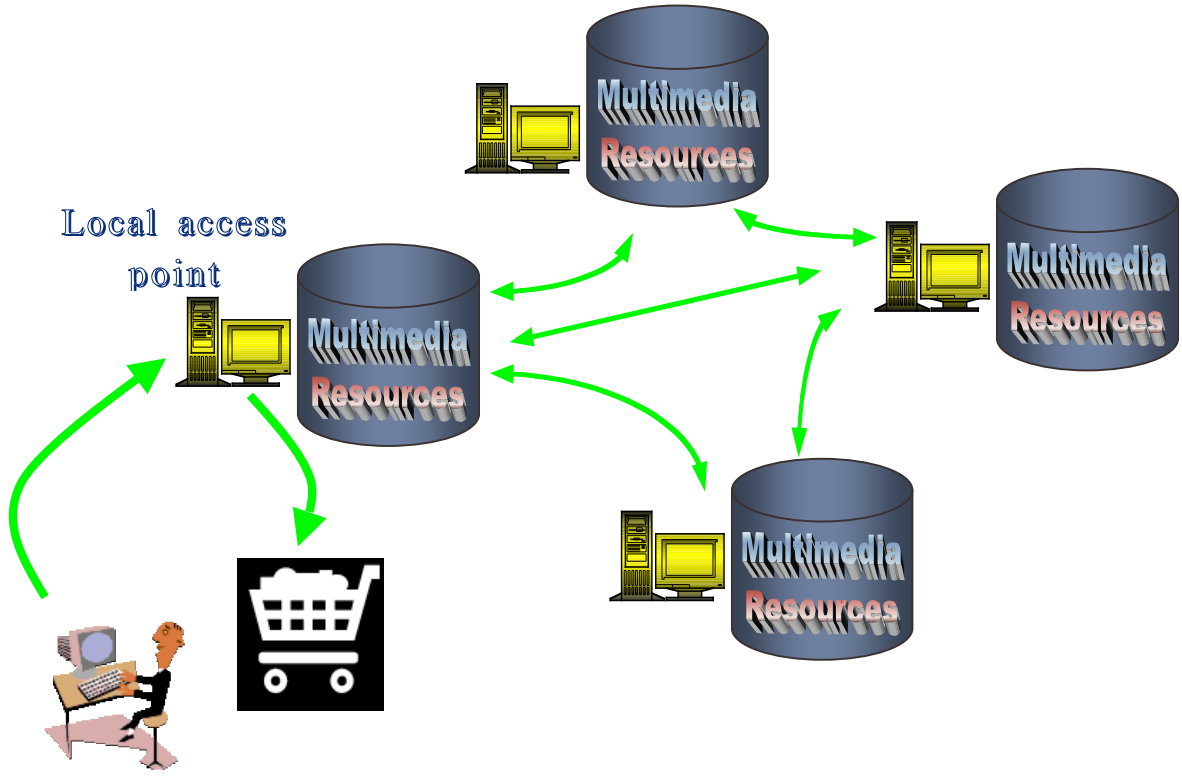

Figure 5. A designer filling his “shopping cart” with all desired resources.

Given the list of available servers - available through the local server - in the network, together with their respective server profiles, the designer will select those servers which may provide the proper resources or the proper services (tools) he wants to access or to use.

Through an iterative process of requests to the selected servers, the user will build up a virtual subset (i.e. by way of pointers to individual resources) upon which he will actually work. The construction of this virtual subset of resources is done by means of the "shopping cart paradigm", that is, while the designer choose all multimedia resources he want to work with, he stock these resources in a "shopping cart".

So, designers are concerned by:

- Building exhibitions by collecting multimedia resources allover the network
- Selection of the working servers
- Selection of multimedia resources ("Shopping Cart Paradigm")
- Manipulation of the selected resources
- "Local" access for all multimedia resources

3.2.1.3 End Users.

Finally, end users access to exhibitions that have been constructed by designers.

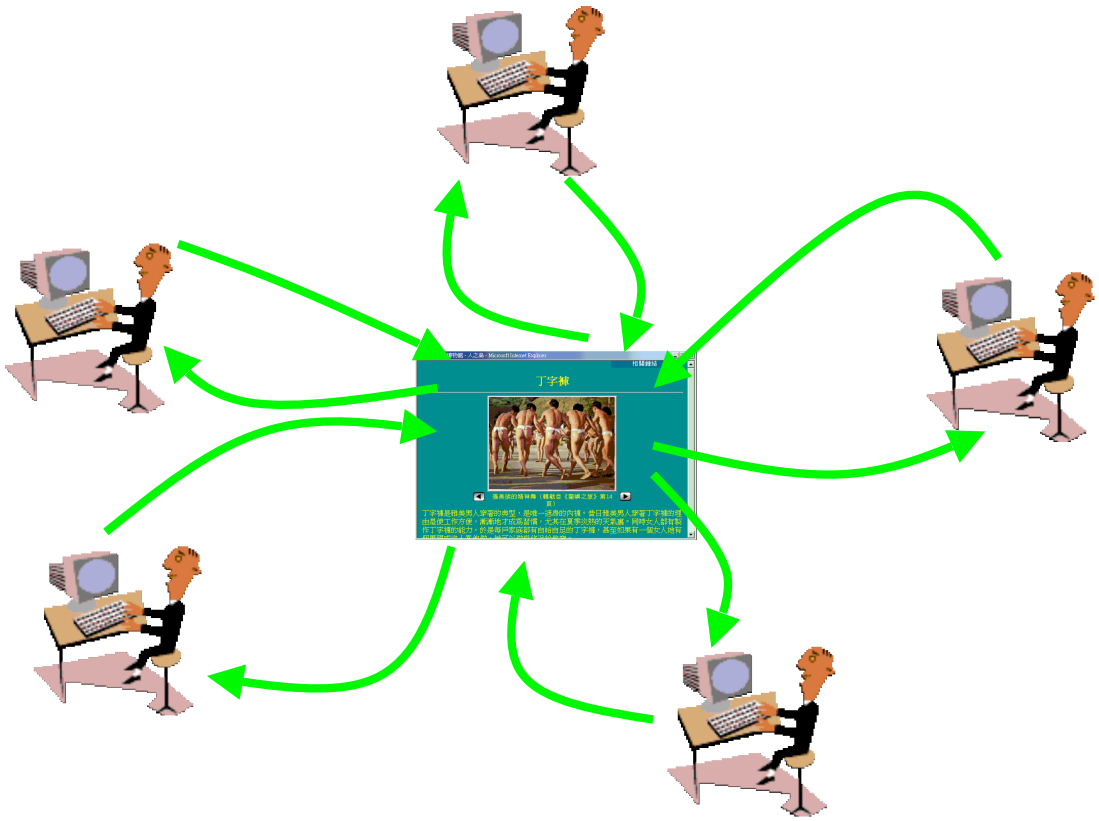

Figure 6. End Users accessing an exhibition.

3.2.2 A General Scenario for Designers.

Any authorized user[8] will be provided with an environment that will lead him along the following steps:

- Connection to a local server;
- User identification;
- Choice of working servers;
- Selection of a subset of resources.

3.2.2.1 User's side.

A user may interact with the network via a simple graphical user interface (in fact, only a general purpose Java-compatible web browser will be needed).

This interface is designed with a non-technical user in mind. However, technically advanced features should be available in an intuitive way. Although all system components will have their own interface, due to different functions, they must have the same "look and feel" (i.e. surface and behavior). This reduces the time the user needs to become acquainted with the network and contributes to the aspect of simplicity.

This user interface will be implemented as a client at the level of which little, not to say no, resource processing is to take place (notion of thin client). Basically, the interface will allow a user to make his different queries and will display result sets according to some specific style-sheets associated with these.

- Connection: To be in line with the idea of a decentralized network, we have considered that a given user should only have to be registered at one given site and that no central user database should have to be set up.

- Workspace: In order for a user to select the servers he wants to work with, he must interact with a "working space". The main purpose of the working space is to provide the user with a graphical interface that allows him to have access to the network and work on it. The working space will offer the following functions:

    - Selection of the working servers;
    - Selection of working resources;
    - Manipulation of the selected resources;
    - Definition of the user preferences.

- Working Servers Selection: Before any query session (resource selection), a user has the possibility to select among a set of online servers, the server(s) he wants to work with. At any moment of the process, the user can edit his list of working servers and modify it.

- Working Resources Selection: The principle is to restrict progressively the choice of resources, in such a way that at the end of the selection process, the user only keeps the subset he wants to work with. The user has the possibility to query either the whole network or his selected working servers. For that purpose, the user will have access to a friendly query interface for editing, modifying and sending his queries to his local server. Another more general way for the user to select his working resources is to browse

[8] In this section the word "user" must be considered as a synonymous of "designer".

through the whole set of referenced resources and to select the resources he is interesting to. Once a user has selected his working resources, he can keep them by using the "shopping basket" paradigm. Of course, one user can have several shopping baskets.

- Manipulation of the selected resources: As the main goal of this architecture is to give a user an access to a large set of resources, it is mostly important to provide a working space dedicated to the resources he wants to work with. The user may also use a set of tools allowing him to perform some kind of operations, queries for example, on the selected resources.

3.2.2.2 Server's side.

As we have said, each server is an autonomous unit containing its own data. We have also considered that a given user should only have to be registered at one given site. From the network point of view, this has two consequences:

1. in order to manage users and resources, the notion of "specialized servers"
2. in order to share all existing resources, the notion of "broker servers"

- Users and Resources: Each server should manage not only its own resources database, but also its own users database. The users database contains both, the general user identification information, and all information related to workspaces.

- Sharing Resources: Upon requests all servers act as "brokers", that is, in order to share all the existing resources, they transmit requests to others servers in the network. Each time a request is broadcast from an access server to a remote server, an identification tag (user id and authorization level) should be transmitted in order to evaluate the applicability of the request.

3.2.3 Simple Object Access Protocol (SOAP[9]).

SOAP is a protocol specification for invoking methods on servers, services, components and objects. SOAP codifies the existing practice of using XML and HTTP as a method invocation mechanism. The SOAP specification mandates a small number of HTTP headers that facilitate firewall/proxy filtering. The SOAP specification also mandates an XML vocabulary that is used for representing method parameters, return values, and exceptions.

SOAP provides a simple and lightweight mechanism for exchanging structured and typed information between peers in a decentralized, distributed environment using XML. SOAP does not itself define any application semantics such as a programming model or implementation specific semantics; rather it defines a simple mechanism for expressing application semantics by providing a modular packaging model and encoding mechanisms for encoding data within modules. This allows SOAP to be used in a large variety of systems ranging from messaging systems to RPC.

---

[9] http://www.w3.org/2000/xp/Group/

SOAP consists of three parts:

1. The SOAP envelope construct defines an overall framework for expressing what is in a message, who should deal with it, and whether it is optional or mandatory.
2. The SOAP encoding rules defines a serialization mechanism that can be used to exchange instances of application-defined data types.
3. The SOAP RPC representation defines a convention that can be used to represent remote procedure calls and responses.

Although these parts are described together as part of SOAP, they are functionally orthogonal. In particular, the envelope and the encoding rules are defined in different namespaces in order to promote simplicity through modularity.

In addition to the SOAP envelope, the SOAP encoding rules and the SOAP RPC conventions, this specification defines two protocol bindings that describe how a SOAP message can be carried in HTTP messages either with or without the HTTP Extension Framework.

As we have mentioned before, in MLIS-ELAN we have used CORBA to allow " low level" communication between servers, while XML has been used to encode all flowing information.

The advent of SOAP, will allow us to build a 100% XML-based distributed software architecture, that is, all flowing information in the network and all "low level" communications between servers will be encoded or defined by using XML[10].

4 Concluding Remarks and Future Work.

In this paper we have presented the design of an XSL-based Multi-Presentation Content Management System (XMP-CMS). The framework provides mechanism for quickly producing multi-style exhibitions from a digital collection. The effectiveness of the methodology is amply demonstrated in the Lanyu Digital Museum. Based on XMP-CMS, the Lanyu Digital Museum exhibition contains several novel features that seem superior to the conventional approaches:

1. It provides an easy way to compose artifacts extracted from the digital collection into exhibitions.
2. It provides an easy way to create different presentations of the same exhibition content that are catered to users with different needs.
3. It provides easy-to-use film-editing capability to re-arrange an exhibition and to produce new exhibitions from existing ones.
4. (It provides *auto-navigation* functions that automatically runs through all the exhibition elements.

Also, in order to retrieve the data distributed in various digital museums, we have talked about the design of a distributed software architecture through which it might be possible to access and to share multimedia resources, which would be spread among different servers. The general idea is that one should be able to bring together all existing multimedia resources, in order to provide any user with a global access to these. Obviously, there is a

[10] One should know that CORBA is a specification of services that are useful for building distributed applications while SOAP is simply a communication protocol that CORBA objects can use to communicate.

need for these resources to be created, and above all, maintained at a place where there is the competence to do so. But, there can also be specific constraints that can preclude some given resources to be deported to another site than the site that has originally created them.

Currently, we are trying to define the SOAP-based API and web services of the distributed framework. In the future we plan to define more robust conceptual document model for representing the exhibition element.

5 Acknowledgments.

The project is sponsored by the National Science Council of Republic of China, numbered NSC-89-2750-P-260-001. This project is also supported by INRIA, France.

6 References.

1. Jen-Shin Hong, Herng-Yow Chen and Jieh Hsiang; A digital museum of Taiwanese butterflies, in Proc. of the fifth ACM conference on ACM 2000 digital libraries, 2000, pp. 260 – 261.
2. Jen-Shin Hong, Bai-Hsuen Chen, Jieh Hsiang: XSL-based Content Management for Multi-presentation Digital Museum Exhibitions. European Conference on Digital Libraries, ECDL 2001: 378-389
3. P. Bonhomme, S. Cruz-Lara, and L. Romary. The SILFIDE Network : An Interactive Service for Using, Studying, Distributing and Sharing Natural Language Resources. SGML / XML '97. Washington D.C., USA. December 1997.
4. S. Cruz-Lara, L. Romary, Ch. de Saint-Rat and, P. Bonhomme. Multi-Lingual Information Society : The European Language Activity Network Project. Working Package 3: "Software and Networking". Deliverables Working Packages 3.1.1 "User Client" & 3.2.1 "Network Management Unit". February 1999
5. Rutledge L., et al., Practical application of existing hypermedia standards and tools, in Proc. Of the 3rd ACM Conference on Digital Libraries (Pittsburgh, June 23-26,1998), ACM Press, pp. 191-199.
6. Su-Shing Chen, Digital Libraries- The Life Cycle of Information, BE Publisher, Columbia, USA, 1998.
7. Royappa, Andrew V. Implementing catalog clearinghouses with XML and XSL. In Proceedings of the 1999 ACM symposium on Applied computing 1999, pp. 616 – 621.
8. Augusto Celentano, SilvanoPozzi, and Donato Toppeta. A Multiple Presentation Document Management System. In Proceedings of the 10th ACM annual international conference on Systems documentation 1992, pp. 63-71.
9. S. Cruz-Lara, P. Bonhomme, C. de Saint-Rat et L. Romary, A General XML-based distributed Software Architecture for Accessing and Sharing Resources. XML Finland'99, Helsinki, Finland, September 1999.