X-System: An Extensible Digital Library System for Flexible and Multi-Purpose Contents Management

Jian-hua Yeh, assistant professor
Department of Computer Science and Information Engineering,
Aletheia University, Taiwan
Chao-chen Chen, professor
Graduate Institute of Library and Information Studies,
National Taiwan Normal University, Taiwan
Shun-hong Sie, system engineer
Graduate Institute of Library and Information Studies,
National Taiwan Normal University, Taiwan
Cjien-cheng Liu
Graduate Institute of Library and Information Studies,
National Taiwan Normal University, Taiwan

I N T R O D U C T I O N

Advances in computer network and storage technologies have inspired the design of digital libraries in recent years. The emergence of digital libraries has introduced a number of important issues (Chen, Chen, Chen & Hsiang, 2002; Chen, Chien, Chen, 2001; Dempsey & Winkler, n.d.). One issue that has not attracted much attention but is essential to digital library development is the flexible design of digital library systems. After the construction of a digital library, it is natural for the digital library to push the content circulation and application as much as possible in order to show its maximum effect. Based on this point of view, it is important to design an architecture to support flexible contents management and fast service development. For flexible contents management aspect, it is known that if a closed data storage system is adopted, the circulation of digital library contents decreases. So the data model and storage with high portability is necessary for easy data access and manipulation. For fast service development aspect, it is more competitive for a system architecture with stackable service feature to develop digital library services according to different information needs. If a digital library system meets the requirements of both flexible contents management and fast service development, it is more likely to create many kinds of information service based on it. In this chapter, we introduce X-System (Yeh & Chen, 2003), a general digital library platform which is capable of handling large-scale digital contents with flexible, extensible management features.
DESIGN ISSUES OF DIGITAL LIBRARY SYSTEMS

As mentioned earlier, the most important features of a modern digital library system are flexibility and generality. Since 2003, the year of X-System’s announcement, several extensions and applications have been developed and deployed. The aim of our research team is to create a powerful digital library system which meets the following important design issues:

A. Fully-functional digital library system
   The design of X-System aims at handling multiple metadata formats which meets the needs of various digital content applications. For example, digital archive systems, knowledge management systems, and e-learning systems all require various kinds of metadata co-existed in a repository system. So the ability of handling various metadata formats becomes a fundamental need of digital library systems. This is also one of the basic features of function design in X-System.

B. Platform-neutral, fully portable system architecture
   The development of X-System is totally based on platform-independent technologies, including Java programming language, Java application servers, XML data presentation, XSL/XPath data transformation, and so on. Also we use a Java-based, native XML database server as the metadata storage, which means the whole system is fully portable. Currently the X-System has many deployment experiences, including Windows-based systems, Linux-based systems, and Solaris-based systems. Any platform which supports Java virtual machine will run our system.

C. Stackable information service modules
   In addition to the modular and object-oriented design concepts, the X-System introduces the layered function design concept, which makes services of X-System stackable. Any newly extended service may use basic services (we call them “upper-layer services”) provided by the system, and the extended services will also be usable by future extended services. The layered architecture makes X-System very extensible, and there will be several extension case studies of X-System demonstrated in the later section.

D. Portable data exchange
   The XML data flow between service modules is a basic design of the X-System. In the layered architecture of X-System, the data between services is XML-encapsulated, and the data rendering to end users is also XML-based, plus XSL transformation as the way of HTML rendering. This feature makes data exchange very easy to be customized for user needs. Also the extension
of any X-System service is easy since the XML-based data semantics is all a service programmer should know.

E. Universal data access

X-System is a web-based digital library system, so the data accessibility is a major concern of the system design. We introduce the handle system, which behaves similar to CNRI handle system (Sun, 1998), as the universal way of resource access. Any resource registered in X-System will be able to accessible globally through the build-in handle system. Since the handle system is able to register original and alternative resources, the accessibility of multiple versions of resource instances is thus increased in X-System.

The above issues are core design principles of X-System. Most of the XML-based digital libraries today are either using relational database solutions (such as Oracle, DB2, or SQL server) with XML middleware or just connect native XML storage as the second and third tier (Villanova & MuseiCapitolini.Net, n.d.). For (MyCoRe, n.d.), the three-tier module is design and made it easy to modify the middle tier, but the middle tier is still not in stackable style. In X-System, the native XML storage is act as backend storage and the XML data exchange is the standard way for passing data between system services in the middle-tier. Besides, we found that it is hard to find a digital library with stackable services currently. In the next section, the data model and system architecture of X-System is depicted.

**COMPARISON OF DIGITAL LIBRARY SYSTEMS**

In recent years, there are several widely used digital library systems which host valuable information for institutions. The advantages of these digital library systems are:

- Preservation: valuable and rare collections can be preserved and shared.
- Access: the holding of digital library systems can be access fast world wide through digital catalogues.
- Versioning: different versions of contents can be easily kept, including physical and digital versions.
- Search: searching in digital library systems is easy through indexing function. The search function can be further optimized based on the specific requirements of institutions.
- Extension: digital library systems can be extended to meet special purposes such as e-learning, knowledge organization, federated cataloguing, and so on.

Here we compare several popular digital library systems in respect to the design issues mentioned in previous section.
A. **DSpace**

DSpace is a complex, open source digital library system which primarily intended for academic use. This system was developed by MIT and HP (DSpace, n.d.). Since its complex nature, the customization of the system requires deep knowledge about the system environment. DSpace includes repository and complete user interface which makes it easy to deploy in the organization. It uses Handle system for unique identifier processing (URN), Dublin Core (Dublin Core Metadata Initiative, n.d.) for standard metadata format with export function. DSpace also support collection concept to group and store digital objects.

B. **EPrints**

EPrints is an open source software for building digital repositories, which initially specified for scientific publications (Open Access and Institutional Repositories with EPrints, n.d.). The metadata support of EPrints are internal, not public standards. EPrints supports multiple archives for one system, that is, capable of manipulating more than one archive at a time. Because pages of the user interface are generated statically, this system is not recommended for frequently data update operations.

C. **Greenstone**

Greenstone is an open source system which aims at creating and distributing digital collections (Greenstone, n.d.). Greenstone can be used to organizing and publishing collections on the Internet and removable media, which is a flexible function on distribution. On data exchange issue, Greenstone support OAI-PMH protocol, which makes metadata exchange easy. The metadata are stored using Dublin Core standard originally, but also support for defining proprietary metadata format. Greenstone supports collection concept to group and store digital objects like DSpace.

D. **Fedora**

Fedora (Flexible Extensible Digital Object Repository Architecture) was developed by Cornell University. The digital object processing of Fedora is compliant with the well known Kahn-Wilensky Framework (Kahn & Wilensky, 2006), which makes it a flexible digital library architecture. Fedora does not contain presentation tier, which lefts to the users to define their own user interface. Since the Fedora system is implemented in Java language, it can be run on many platforms. Fedora also designed a set of SOAP-based application interfaces. Like Greenstone, Fedora support OAI-PMH protocol which makes metadata exchange easy. The metadata of Fedora are stored using Dublin Core standard.
According to the description of the digital library systems above, the comparison chart, including our X-System, in respect to a flexible digital library system design issues is shown in the following table.

<table>
<thead>
<tr>
<th></th>
<th>DSpace</th>
<th>EPrints</th>
<th>Greenstone</th>
<th>Fedora</th>
<th>X-System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fully-functional, function extension</td>
<td></td>
<td></td>
<td></td>
<td>V</td>
<td>V</td>
</tr>
<tr>
<td>Platform-neutral, fully portable</td>
<td>V</td>
<td></td>
<td></td>
<td>V</td>
<td>V</td>
</tr>
<tr>
<td>Stackable information service</td>
<td></td>
<td></td>
<td></td>
<td>V</td>
<td>V</td>
</tr>
<tr>
<td>Portable data exchange</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>V</td>
</tr>
<tr>
<td>Universal data access</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>V</td>
</tr>
</tbody>
</table>

**Fig. 1. The service architecture of X-System**
In Fig. 1, the service architecture of X-System contains front-end service manager, back-end service manager, physical storage service manager, and handle service manager. These services provide main functions of X-System, as discussed below:

1. **Front-end Service Manager**

   The front-end service manager acts as a single service entry for all add-on modules to access X-System functions, so the front-end service manager is also called a front-end proxy. The design of the front-end proxy is to provide a single, integrated service interface for other modules or parties, thus increases the extensibility, scalability, and availability of the X-System. The module structure of front-end proxy consists of several core levels, which classifies the service level and functionality. The core levels are also creates the modular design rules: the higher core level functions must formed by the lower core level functions, if some function can not be formed by the lower level functions, then it should be stay in lower level, or it should be the most generic functions. Currently the front-end proxy contains two core levels, which is core level 0 and core level 1 functions, as shown in Fig. 2.

   ![Fig. 2. Core level functions of X-System](image)

2. **Back-end Service Manager**

   The back-end service manager is the communication gateway of front-end service manager and underlying database systems. The core level functions in the front-end service manager have corresponding service functions in back-end service manager. The functions in the front-end service manager only perform logical data manipulations, which are called object level manipulations. But the service functions in back-end service manager will interact with database system directly. This version of back-end service manager in X-System communicates with native XML databases, which is compatible with XML:DB API 1.0 (Staken, 2001). There are two reasons of this design: open standard compatibility and native XML processing. The XML DB API supports various XML processing standards, such as XPath (Clark &
DeRose, 1999), XQL/XQuery (Boag, 2004), XUpdate (Laux & Martin, n.d.), which are all W3C standards.

3. Physical Storage Service Manager

Unlike the back-end service manager, which focuses on metadata management, the physical storage service manager provides functions for physical digital material management. In X-System, the physical storage is organized by a set of mount points in file systems, that is, the management of physical digital materials is file based. The advantage of this design is simplicity and flexibility. The digital content imported into X-System will be no longer needed to change the physical form they stored, and the mount point can be changed easily when necessary. Another advantage of this design is easy to integrate other applications such as video-on-demand server for streaming media. Currently the physical digital material is saved by classifying the MIME (Freed & Borenstein, 1996) type.

4. Handle Service Manager

The digital library contents are mostly archive metadata and physical digital materials. Since there will be large amount of metadata and digital files, it is important to develop an appropriate identification scheme in the system, which guarantees the local and global uniqueness of digital content. In X-System, we introduce a handle service manager to support digital content identification covered both metadata and physical digital materials. The handle service manager operates similar to CNRI Handle System (Sun, 1998), and we are going to collaborate with CNRI handle system in the future version of X-System.

The Data Model

The data entities designed for X-System include document schema, metadata documents, user domains (or called organizations), user groups (or called projects), document collections, users, and universal identifiers. These data entities are processed as basic data units in the whole system, as described below:

1. Metadata documents

The metadata documents store in X-System is created according to the specification of document schema. In our system, all the metadata documents and other digital objects are regarded as important intellectual properties, so the metadata structure defined in X-System consists of several aspects of described targets, including subject description, rights description, and administrative description:

a. Subject/object metadata
The subject metadata contains semantic description of target object, such as background information, subject keywords, creator information, etc. Any attribute that is subjective explanation can be treated as subject metadata. The object metadata represents format description of digital objects, such as file format, digital scale, color information, etc.

b. Digital rights metadata
The digital rights metadata describe the rights management information of target object. DRM (Digital Rights Management) is getting more and more attention these years, so embedded rights management information in media file is an emerging trend of many digital technologies, such as computer media player, cell phones software, etc. In X-System, it is allowed to use custom designed rights management metadata formats, with the use of international standard is recommended. (XrML, ODRL, or other standards) The digital rights metadata in X-System is not only able to provide digital rights information of target object, but also capable of interchanging with other systems for interoperability.

c. Administrative metadata
The administrative metadata, like its name, is designed to record information for administration. This part of metadata aims at representing administrative information of target object such as data maintenance history, record creation time, language of description, and other remarks.

In Fig. 3, three aspects of metadata form a metadata document. The X-System supports definition of each aspect by assigning different XML schema, so a single metadata document is composed by three XML structures. Since the metadata document is a big XML structure, it is easy to manage these documents in the back-end storage of X-System, as it is native XML database in our design.
2. Document schema

As mentioned earlier, all metadata documents stored in X-System is XML-based with user-defined document formats. These formats are called document schema which are different from XML schema since the metadata document exists in X-System consists of three parts (subject/object metadata, digital rights metadata, and administrative metadata), so the document schema here is a combination of three XML schema, not a single one. However, the settings of each part are followed by the standard W3C specification. Other X-System specific configuration is recorded outside the document schema.

3. User domains, and user group, and individual users

The X-System is originally designed for maintenance across multiple user domains and groups. With this prerequisite, this system must support different organization to operate their digital content separately and will not interfere with each other. Currently, the X-System organizes domains, groups, and individual users in a hierarchical manner, as depicted in Fig. 4.

![Fig. 4. Hierarchical relationships of user domains(organization), user groups(projects), and individual users.](image)

4. Document collections

The document collection design in X-System is just like the shopping cart in an online store. Each collection belongs to a single user which is able to collect multiple metadata documents at a time.
EXTENSIONS AND APPLICATIONS

X-Learning: E-learning Content Management System

Nowadays, most electronic learning system did not integrate with digital archive system. Therefore, institution usually has to hold one digital archive contents management system and one electronic learning contents management system independently. Therefore, teachers or learners are not able to use the digital archive contents in electronic learning system seamlessly. As the concept of COLIS (Collaborative Online Learning and Information Services project) planned by Department of Education, Science and Training, learning functions should be integrated to digital library system. The COLIS system modules include: learning content management, content management, learning management, library e-services/e-reserve/e-journal, digital rights management, directory services, and registries (McLean, 2001). We thought to create a platform which can convert the metadata of digital archive contents into LOM and an authoring tool that can aggregate the assets to courses by teacher is very important after a large number of digital contents were produced by digital projects. This is the key motivation for us to develop the X-Learning platform. The X-Learning platform is capable of integrating the digital archive contents and e-learning objects.

The major function of X-Learning (Chen, Yeh & Sie, 2006) is to provide support for all e-learning related services (Macromedia, 2001; McLean, 2001; McLean, 2002; The MASIE Center e-Learning Consortium, 2002) including course and material management, course presentation, and so on. For the transformation from digital archive content to e-learning content, the X-Learning provides necessary functions, including schema mapping rule setup, SCORM-compatible (ADL, 2001; IMS Global Learning Consortium, 2001; IMS Global Learning Consortium, 2003) content packaging service, IEEE learning object metadata (LOM) (IEEE, 2002) transformation, and so on. X-
Learning is design to act as a SCORM-compatible learning content management system (LCMS), which is capable of packaging and previewing e-learning packages.

![Diagram of X-Learning system architecture]

**Fig. 6.** The service architecture of X-Learning over X-System

![Screenshots of X-Learning: schema mapping rule setup (left) and content packaging/manifestation utility (right)]

**Fig. 7.** The screenshots of X-Learning (Left part: schema mapping rule setup, right part: content packaging/manifestation utility)

**X-Ontology: Government Ontology Management System**

The number of web sites of Taiwan government grows fast in recent years, which made Taiwan a popular e-government country. Due to the quantity and the diversity involved in e-government presentations and operations, traditional approaches to web sites information management have been found to be rather inefficient in time and cost. Consequently, the necessity naturally arises regarding the trend to establish government knowledge management system, so as to speed up information lookups, sharing, and linkups. Moreover, knowledge management system would in turn enhance e-government effectiveness as it helps to store and transmit information, be it explicit or
implicit in nature. The first thing of knowledge management system is building up the government ontology and thesaurus. Upon the completion of the ontology and thesaurus needed, semantic searching can begin to function properly, that in turn would kick into place mechanisms required for effective information management.

The major component in X-Ontology (Chen, Yeh & Sie, 2005) is the ontology hierarchy maintenance service. The X-Ontology provides a tree-based hierarchy maintenance interface for user to manipulate a concept hierarchy at a time. In addition, associations between hierarchies (ontologies) can also be maintained in this interface. Fig. 5 shows the tree-based hierarchy maintenance applet. In the upper-right part of the applet shows the basic information form of the selected concept along with associations with other concept (intra-hierarchy or inter-hierarchy is possible). The lower-right part of the applet maintains the additional information, which comes from the thesaurus existed in the system.

The ontology presentation service in X-Ontology is quite straight-forward: the information contained in the concept is presented. Fig. 6 shows a tree-based web page for ontology presentation in X-Ontology. The user can interactively traverse the hierarchy, check the thesaurus information, and jump to other concept through the association created by ontology maintenance interface.

Fig. 8. The service architecture of X-Ontology over X-System
X-Library: Library Automation System

There are two major set of functions of X-Library: reader service module and cataloguing/circulation support. The X-Library provides user a simple interface to lookup bibliographic records, reserve books, organize personal favorites using shopping cart function, and so on.

The library cataloguing and circulation process supported by X-Library include various features based on the underlying X-System. Except for the bibliographic records maintenance supported by the base system, X-Library has developed library-specific functions that are comparative to a traditional library automation system. In reader service module, a user of X-Library is not only able to browse and search library catalogue but also able to make reservation on books interested. Another service module of X-Library operated by library staff contains cataloguing and circulation functions, these functions are similar to traditional library automation system.
Digital image and metadata are two critical functionality-related components in their applications in digital archiving and management. However, such applications have proven to be relatively difficult, since metadata and digital image are independent. In order to combine metadata in image, this extension work attempts to store and exchange image with related embedded metadata. This reference platform, Embedded Metadata Framework (EMF), are developed to meet the embedded metadata requirements for digital images.

Our design adopts the format developed by the Web-based information system to run the embedded metadata through Adobe XMP technology (Adobe, 2008) applying to digital preservation systems, including Embedded Metadata Framework definitions, digital preservation system architecture, and end-user operating design. The EMF architecture is created as a shared reference model for embedded standards of metadata in digital images; it is also intended to provide a frame in exchange for information among platforms. The data structure is based on XML/RDF and adopts Unicode (UTF-8) encoding while EMF cores are packed by Adobe XMP API to communicate with other programming languages for further developments. Fig. 12 depicts the structure of EMF framework.

**Fig. 11.** The screenshots of X-Library (Left part: reservation management, right part: bibliographic/MARC records maintenance)
Our implementation includes annotation works, editing metadata, EMF process, transmitting image files to database, and data display for displaying data:

- **Annotation Works**
  Metadata editing can be done either offline or online, then written into image files and saved in the system. It can also be developed into an independent editing platform to achieve distributed creation. With the above features, the editing tools can be customized and incorporated in accordance with the needs of the editing teams.

- **EMF Process**
  The EMF core includes operating process, API Services, database to save metadata and image files (digital objects). The API Service provides uploading, linking and access database, transmission of image files, producing resized images and rebuilding index for search engine. Operating process contains database input, embedded metadata access and built-in metadata in database.

- **Data Display**
  Data Display aims at displaying and controlling data. The system provides browsing and searching for records of digital archive. Through the detailed browsing, data can be downloaded and printed.

The three operating processes compose a digital preservation system and an online editing works system. The design allows the access of XMP when data is saved to the database which avoids building up files separately. The
system is able to download, browse files and retrieve image information, including related metadata, even in the offline status.

Fig. 13. The system architecture of EMF extension

The EMF metadata presentation in this extension contains file properties and metadata of digital images. In Fig. 14, the file properties on the left side of the display are automatically generated when the file is firstly created by digitalized information facilities. Metadata information on the right side of the display is manually established which allows amendment. It also support multi-language text input and display since this system is Unicode-based. Adobe XMP is served as the EMF core in the system and the definition of metadata structure can be customized. In Fig. 14, the sample used here is the IPTC metadata schema (IPTC, 2008). On exchange of the digital image, the related metadata will accompany the image so the users or other software are able to utilize the embedded metadata.
C O N C L U S I O N

More extensions of X-System are still under development and the application domains of X-System are still increasing. The next steps of our system aim at integrating popular access protocols such as OAI (Lagoze & Sompel, 2001), Z39.50 (Lynch, 1991), DOI (The International DOI Foundation, n.d.), and so forth. Although Z39.50 protocol has been integrated in X-Library system, we are planning to implement it as an access protocol support by the base X-System. Some ongoing extensions of X-System include media asset management (MAM), e-publishing, and virtual reality (VR) presentation integration. We are looking forward to the integration of multi-discipline domains and make X-System a more general and useful digital library system.

R E F E R E N C E S


Villanova University Digital Library. (n.d.). From http://digital.library.villanova.edu/