DLworm: a system for workflow and repository management for digital libraries

Ratna Sanyal, Dhwaj Raj, Divanshu Kumar Gupta, and Ajay Verma
Indian Institute of Information Technology, Allahabad,
Deoghat, Jhalwa, Allahabad – 211 012, Uttar Pradesh, India

World Digital Libraries 1(2): 121–136

Abstract

This paper describes the Digital Library Workflow and Repository Management system (DLworm) to manage the repositories and to automate the workflow for digital library at any institution/ organization. After a detailed and comprehensive analysis of the steps involved in maintaining a digital library system, a concrete model has been developed and tested. DLworm supports most of the functionalities required for such a large repository. DLworm is intended to formulate a fault tolerant system providing integrated environment for all users involved in the processes of maintaining and accessing a digital library. DLworm is a flexible and extensible system supporting phonetic matching for search queries. DLworm uses a three level hierarchy for maintaining the data in the repositories. We describe briefly the architecture that is robust and fulfils the demand for long term preservation of digital data. This article illustrates mechanisms for process management that makes DLworm highly efficient and user friendly.

Email rsanyal@iiita.ac.in; draj_b05@iiita.ac.in; dkgupta_b05@iiita.ac.in; averma_b05@iiita.ac.in

Introduction

As the roots of technology are deepening into the human race, there is a vast increase in usage, processing and accessing of digital data like text, images, moving images, datasets, and so on. Coordination of different tasks involved in collecting, processing, storing and disseminating this digital data, which could be of the order of Terabytes, is highly complex. Most of the processes involved in managing digital repositories are performed manually and this is very tedious work and results in high wastage of resources.

A digital library system (Chen, Kim, Nnadi, *et al.* 2003) can result in a high resource centre for all sections of society. Numerous rare and highly valuable manuscripts can be easily stored in digital format and made accessible to many individuals at the same time. Moreover, digital libraries can transcend the limitations of conventional libraries and individuals in remote areas can be provided with easy and fast access to the contents in an 'any-time any-place' fashion.

Manual management of a digital library involves interaction with various kinds of users. Multiple data addition centres add digital content to the repository. Maintaining consistency, duplicity check and quality control over this constantly increasing data are tremendously complex and error prone. Managing the organization of data manually may include multiple errors like repository failure, mismanagement and resource wastage. Data addition centres require data to be digitized which in turn involves numerous persons who are manually contributing by digitizing the data, checking its quality and adding to the resource centre. Moreover, since there can be many data contributors for a digital library, hence there exists a high probability that the same data can be received at the data addition centre multiple times. Digitizing again and again the same data results in increased human resource depletion and augmented expenditure of time and money.

After extensive analysis of the framework for Distributed Object services (Kahn, Wilensky 2006), which was being developed at CNRI (Corporation for National Research Initiatives) the authors gained knowledge on efficient use and maintenance of repositories in distributed environment. In this framework, the authors make use of handle system for distributed digital object services. They describe an infrastructure that can support digital library services and which is an open architecture. The system supports a high level of expandability and suppleness by providing the method to access and identify the digital objects in digital repositories which may be established in distributed environment. This architecture permits us to name and locate the digital objects. This architecture makes it easier for users to access digital data because the services provided by the system allow the naming authorities to generate identifier for objects. The basic services used by this architecture are as follows.

- Repository access protocol to provide access to an object and for depositing the object in distributed digital repositories.
- Handle server system which maps handles to network resources to provide the representation for accessing a digital object.
- Value added services to provide safety to the access system.

Our system is targeted at managing the workflow in a digital library and maintaining storage of digital data at any organization. It is designed to accept all kind of digitized and born-digital materials including text, images, audio, and so on. It creates an automated interaction model for various users involved in maintenance and storage of digital data. The system is designed to maintain digital

repositories in a distributed environment which can be implemented at any institution, organization or at a personal level. The system facilitates easy workflow management and interaction for the users maintaining and contributing to the digital repositories.

In the next section, related works, DSpace, OpenDLib and other implementations are presented. In the third section, we describe the functional overview of DLworm. We present the architectural overview of DLworm in the fourth section. Implementation and detailed design are presented in the fifth section. Conclusion and future steps are described in the last section.

Related work

DLworm inherits its primary strength from the detailed study of the digital library processes going on in the nodal centre at Indian Institute of Science, Bangalore and other mega centres of Universal Digital Library in India. Comprehensive study of the protocol followed at different centres provides us a detailed functional overview for workflow involved in digital library management. Study of presently available systems like DSpace (Tansley, Bass, Stuve, *et al.* 2003) and OpenDLib (Castelli and Pagano 2002) also enables us to model a robust system supporting a high level of expandability and flexibility.

DSpace

DSpace has been developed by Massachusetts Institute of Technology (MIT) in collaboration with Hewlett Packard Company (Smith, Barton, Bass, *et al.* 2003). This system is intended to serve the long term preservation and access issues. DSpace is built around the idea of organizational communities. This System uses Dublin Core metadata elements (Lagoze and Hunter 2001) for describing the digital data. This system has web based interface for user interaction. It has three types of users: web user, administrator, depositors and it also provides the service to appoint a special type of user, 'Gatekeeper', for reviewing and editing the data. This system provides different access privileges over the data determined by the community. DSpace is designed for UNIX platforms and it also uses other open source middleware and tools. It uses PostgreSQL, JAVA SERVLET engine, JENA (an RDF toolkit from HP Labs), OAICs open source OCLCat framework.[1]

DSpace is basically a three layered architecture for storage, business, and application. For better control it provides authorization so that depositing and reviewing is performed only by authorized personals. Browsing and retrieval can be made anonymously but for using other services the user should be authenticated. This system uses CNRI[2] Handles for creating and maintaining the globally unique identifiers. This system uses Lucene so it provides ability to inclemently add new indexed content without regenerating entire index.

OpenDLib

OpenDLib is the toolkit which is used to manage digital repository for a given user community. It includes an open amalgamation of services that can be distributed and replicated. This provides all the basic functions of digital libraries and it also manages the services dynamically and according to the access policy. It allows different types of data formats. Its orthogonal facility provides the service to define the virtual view on the data to different communities (Castelli, Pagano, Simi 2004). It also provides the service to manage new types of data which have no physical counterpart. It also enables the facility to maintain edition or version of different data by multiple formats of metadata.

[1] <http://www.oclc.org/research/software/oai/cat.htm>
[2] <http://www.cnri.reston.va.us/home/cstr/arch/k-w.html>

Other implementations

Another important achievement in the field of digital repositories is the Reference Model for an OAIS (Open Archival Information System)[3] created by Consultative Committee for Space Data Systems. The EPrints system developed at the University of Southampton[4] is also a similar type of system but it provides access only to data deposited by authors in document format while DLworm provides an environment to maintain repositories accepting any kind of digital data like text, images, multimedia format, and so on. New Zealand Digital Library has also developed a system for maintaining digital libraries called Greenstone software.[5]

- After the analysis of already existing similar systems described above, we have recognized the need of new requirements which are not present in these systems. Also from the experience of dealing with the digitization processes at Universal Digital Library, the automation of some new requirements is realized to be very necessary. The novelty of the system lies majorly in the following functionalities:
  - Quality checking module of the scanned material
  - Duplicity checking in the repository
  - Easy Web based interface
  - Highly adaptable browsing environment using phonetic matching etc
  - Three level hierarchical hashing and optimized indexing in archives

All the major functionalities provided by the DLworm will be discussed in the next section in detail.

[3] <http://public.ccsds.org/publications/archive/650x0b1.pdf>
[4] <http://www.eprints.org/software/>
[5] <http://www.greenstone.org/>

Functional overview of DLworm

DLworm provides a central service for community based digital libraries. It is designed to maintain digital repositories in a distributed environment which can be implemented at any institution, organization or at personal level. The product facilitates easy workflow management and interaction for the users maintaining and contributing towards the digital repositories.

The major functionalities provided by DLworm can be summarized as follows.

- Central database maintains the information, permissions and statistics for all users.
- DLworm provides a quality checking module for reviewing the submitted data.
- A three level hierarchy for data organization is defined.
- Dublin core specifications are used for metadata.
- At each level metadata/submitted data is checked for duplicity.
- It provides metadata based with phonetic matching query system to end users for searching into the data.
- DLworm provides facility for communication between users involved in the system.
- Interface for each user is web based.

The following sections discuss each of these functionalities in detail.

Users

There are four major categories of users in the DLworm. These are the administrator, scanners, quality checkers and the end users. Each category of user has specific roles, responsibilities and access rights. The view of the entire system, in terms of the functionalities available to each category of users, is as follows:

Administrator

Installation	• Define parameters for each slave (IP, directory, and so on.) • Create tables/databases • Password and login settings for admin, scanners and quality checkers
Permissions	• Can give permissions for documents (readable/ downloadable) • Can give permissions for users (comments/downloading)
Message passing	• Can send message to other users
Logging	• Maintains log in database (DB) of each user activity
Statistics	• Calculates salary, and like functions based on logs
View	• All the queries from DB can be made by the administrator according to the requirement

Scanner

Basic function	• Scanner converts the data in required digital format • Scanner can see logs and stats of his work
New document entry	• Upload metadata if entry already prepared (then use metadata extractor) else fill the metadata form (Dublin core) • Give directory location of data • All files are sent to master and stored in mounted drives of QC resource centre • In DB at master entry for each document is made
Uploading	• Metadata is checked for duplicity with DB • Each file's MD5 is stored in a DB

Quality Checker

Basic function	• List of all the non checked documents is displayed as a list • Log of activities of QC is maintained in DB
Review data	• QC can view the metadata and pages of the book • Each file's MD5 uniqueness is checked
Approve	• DB entry in 'checked ok' column for that document
Reject	• Re-uploading request goes to the same scanner who uploaded the document • Files deleted from the FS

Web-User

Queries	• Alphabetical indexing • Can search for a document based on different parameters • Read the document (if permitted) • Download the document (if permitted <document + user>) • Read comments
Registers	• Fill the form • Request goes to admin (admin gives all kinds of permissions)
Make comments	• Write comments for a document if permitted by admin

The functionalities of each user described in the tables above, are summarized in a use case diagram depicted in Figure 1.

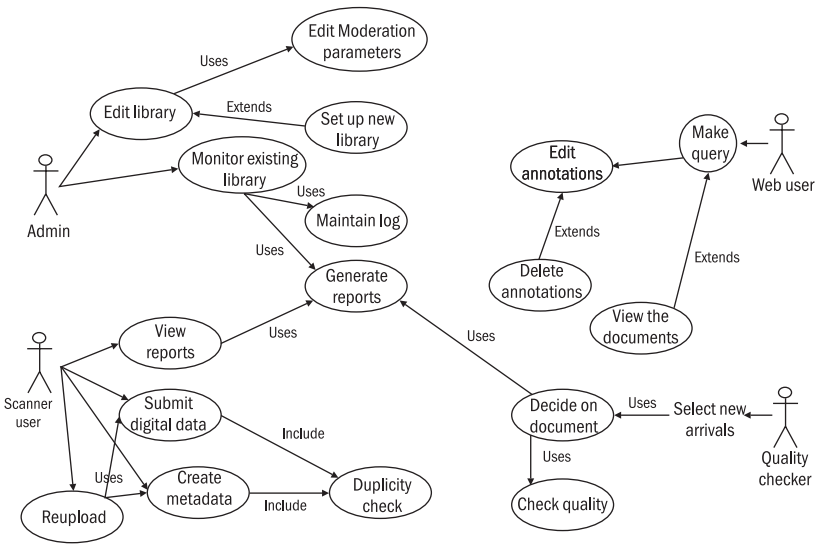

Figure 1 Use case diagram for DLworm

Data model

The data organization model for DLworm is shown in Figure 2. The data model of DLworm clearly indicates the basic data organization used by the system. Each class of user submits their contribution towards digital library and DLworm manages the workflow among all users.

At storage a three level hierarchy is used. At first the entry in database is made about files, path, author, title, and so on. Then at the second level, metadata is associated with each item. At the third level, original files are kept in the same format in which those were submitted. By keeping the files in same format it is easier to perform content based search or other operations on data. Replication of all data ensures reliability and high availability.

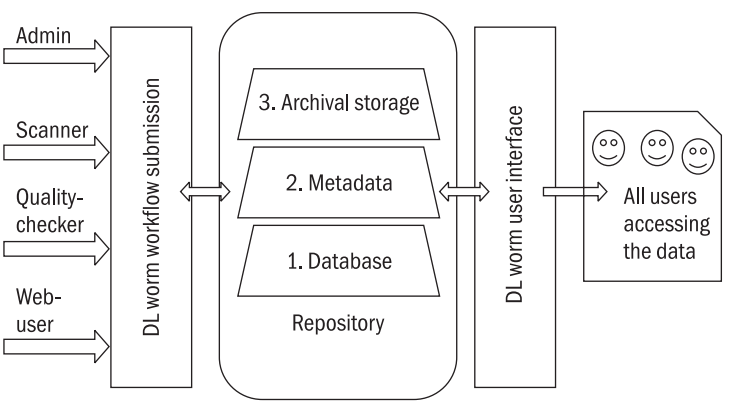

Figure 2 Data organization model for DLworm

In addition to the above, DLworm also maintains a temporary backup system. This is a temporary database that gets populated whenever there is a new digital resource created by the scanners and which has not yet been checked for quality. The items in this temporary database are allotted to the quality checkers either by the administrator or by DLworm. In case the content passes the quality check, then it is sent for permanent storage and deleted from the temporary storage. On the other hand, if the quality checker observes some flaws in the submitted content, then the content is sent back to the scanner and is again deleted from the temporary backup. In case of born-digital contents, the data is directly sent to the metadata entry process.

Quality checking

Each data item submitted should be reviewed in terms of quality of data. In case of born-digital data, this process is not required. DLworm provides a semiautomatic mechanism for checking the quality of data. DLworm uses image processing (Bertrand, Giuseppe 2004) and OCR tools to check the quality of submitted images and we also define a class of users called quality checkers. The quality checkers review each data item submitted by the scanners and they can use the automatic tools provided by DLworm or they can do this task manually depending upon the complexity of task. Entry of data items in the final repository depends on the decision of acceptance by the quality checkers.

In addition to the above semi-automatic process for quality checking, each item in the repository also undergoes a duplicity check. DLworm checks for duplicity of data both at metadata and content levels. At metadata level main fields associated with submitted item are checked with metadata from the repository. This check is performed before the digitization of the data. If all fields for newly submitted data are same as that of some already existing metadata, then the user is notified about it.

A mistake in submitting files of a data item in the name of some other item may happen. Hence, content level duplicity is checked using message digest 5 encryption code (Forman, Eshghi, Chiocchetti 2005). It is assumed that MD5 for each file is unique and if a file is submitted with same MD5 as that of an already existing file in repository then the user is notified about it. These methods of checking duplicity enable us to properly utilize the available resources by minimizing the wastage.

With advances in image processing techniques, it is entirely possible that some of the tasks that are currently being performed manually will be performed automatically in future. This implies that we may have to add new modules corresponding to these emerging techniques. The design of DLworm takes this fact into account and provides mechanisms for easy plug-ability of new modules or modification of existing modules.

Metadata

Metadata is a structured data providing the information about some other data item. Metadata provide the facility to understand the characteristics and usage of data. The elements of metadata for each data item and born-digital contents depend on the type of data and context of use. DLworm attaches metadata file with each data item/content submitted. Each metadata files is stored within the archives and also the entry for each element is maintained in a relational database. Although the metadata is maintained according to Dublin core specifications but the DLworm metadata specification is dynamic and it can be changed by the administrator according to the requirement.

Some of major elements for metadata are:[6]

- Title
- Subject and Keyword
- Description
- Resource Type
- Source
- Relation
- Coverage
- Creator
- Publisher
- Contributor
- Rights management
- Date
- Format
- Resource Identifier
- Language
- Audience
- Provenance
- Rights Holder

In addition, DLworm also maintains some extra elements in the metadata that help in the management of the repository. These include –

- Number of pages
- Date of digitization
- Identification of storage device where content is available including path (if any)
- Identification of scanner, quality checker and scanning centre
- Viewing statistics of the content

Each Dublin Core element is defined using a set of ten attributes from the ISO/IEC 11179 (ISO11179) standard for the description of data elements. These include Name, Identifier, Version, Registration Authority, Language, Definition, Obligation, Datatype, Maximum Occurrence and Comment.

Phonetic query system

Since the data items available in a digital library may be in different natural languages, so DLworm provides natural language tools operable on digital data, for example, phonetic search, summarization, and so on. The phonetic search is required because the end user may not be aware of the exact spelling used in the creation of metadata for a particular word. Thus, the query system should try to find all items that match the query terms phonetically. Hence phonetic query system is designed for DLworm which phonetically matches the entries of database or metadata with query text of the user. For the phonetic query system we integrated the Soundex algorithm (Odell and Russell, 1918, 1922) and Levenshtein edit distance algorithms (Levenshtein 1966) and adjusted them according to Indian pronunciation system. The Editex algorithm was designed by Zobel and Dart (Zobel and Dart, 1996). It is an enhancement of the Levenshtein edit distance algorithm. We have modified Soundex algorithm and Editex algorithm. These modified algorithms are described in details in our earlier work (Raj and Sanyal, 2008). After applying both these algorithm in this work, we integrate the results of both algorithms and find the matched index for two strings. A threshold is defined and the system displays all those digital contents whose score is above the threshold for a given query string. So this system provides the strength to the searching and browsing facilities.

DLworm interface

DLworm provides web based interface for all users. It uses JAVA Applets/JSP Pages based interface for administrator, quality-checker and scanner. Whereas for web-user a JSP based interface is used. The DLworm interface provides facilities of message passing and viewing statistics of users. Usage of JSP facilitates faster HTTP communication and hence helps Web-user in better and faster results.

[6] <http://dublincore.org/documents/2006/12/18/dces/>

Fault-tolerance

Fault-tolerance is an important requirement for any Digital Library System. As it is known that digital data may include rare manuscripts and historic items, hence extensive care should be taken not to lose these data in case of failure of any node. Moreover digitization of data involves a high cost associated with it in terms of human resource, time and money. Hence one of our primary objectives should be to prevent the system from any kind of failure or loss of data. Even after taking all the possible precautions, the computer systems are always prone to failure. So, a recovery system is also required to deal with such failures. DLworm implements fault tolerance and recovery at each level of data organization.

To implement fault tolerance at Master server and Global Web server a backup of data of each Server is replicated on another and the services are transferred to other system in case of failure of any Server. For Quality Checking resource centre a temporary backup system is maintained just for replication objective. Any data added to this resource centre is also copied onto the temporary backup system and it is removed from temporary system once the data is uploaded onto the repository systems. To take care of failure of any archival storage node each data added onto the repository is replicated at two different nodes in repository system.

The architectural overview of DLworm and details of each process are described in the next section.

Architectural overview of DLworm

We have premeditated the architecture of DLworm such that it can also provide access to distributed digital repositories in a way almost similar as described by the framework for distributed digital objects (Kahn, Wilensky 2006) developed at CNRI. DLworm makes use of Data Transporters service at service layer in the architecture to provide access to digital data stored in digital repositories. Data Transporters service of DLworm uses a general item naming system based on metadata fields for digital data and incorporating it with network protocols like NFS and SCP to provide a fast and efficient approach for data access and depositing tasks.

DLworm incorporates many underlying open source third party systems like

- MySQL: an open source relational Database
- Apache Server

DLworm can also support any alternatives for these open source systems. For reflection MySQL can be replaced by any relational database. The architecture of DLworm (technical perspective) is shown in Figure 3. DLworm can be organized in the following three basic architectural layers.

- Archival Storage
- Functional Toolkit
- Service Layer

Figure 4 presents the functionalities of architecture drive and the workflow of a digital library. This becomes clear from the Figure 5, which presents the activity diagram of each actor.

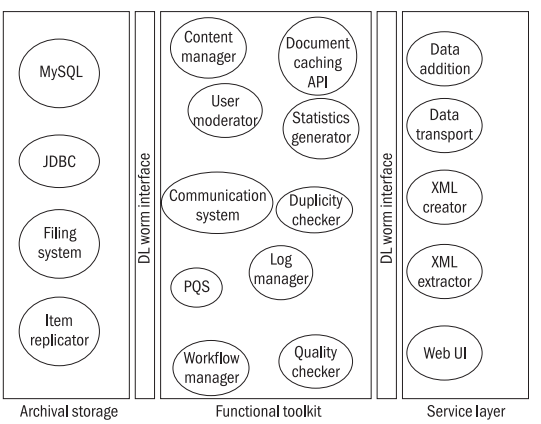

Figure 3 Architecture of DLworm (technical perspective)

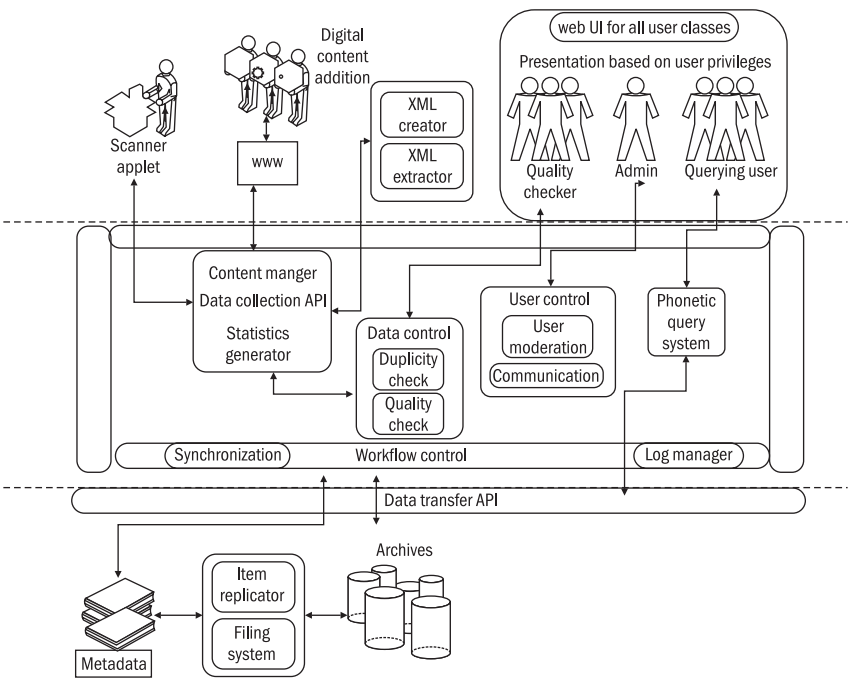

Figure 4 Workflow of a digital library swayed by DLworm services

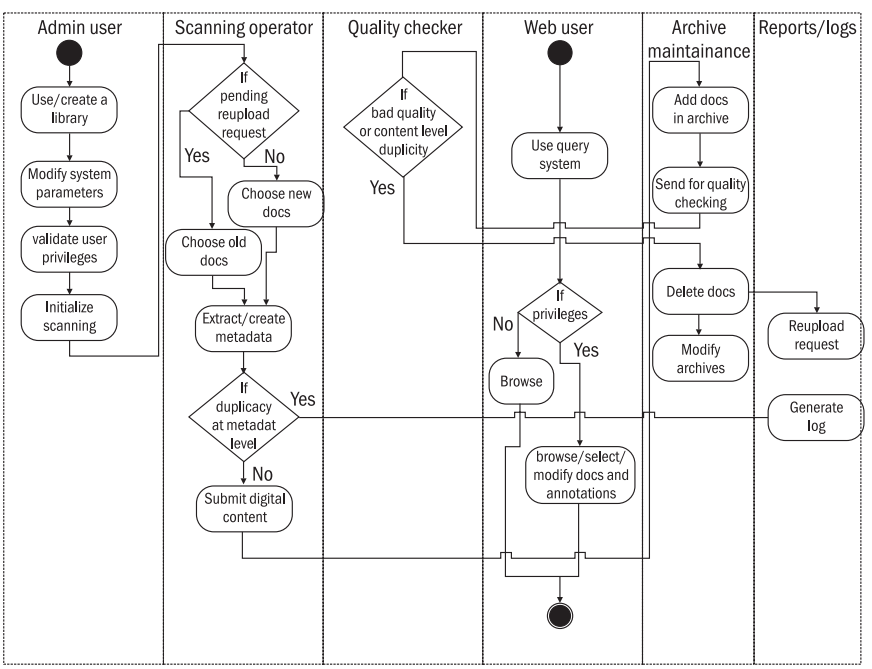

Figure 5 Activity Diagram for DLworm system

Archival storage

Archival Storage Layer provides the manipulation and storage of digital data indexed in the memory disks contributing to a repository. It involves usage of JDBC for database access and SCP protocol for the exchange of digital data within different nodes. The number of storage nodes is scalable because more number of nodes can be added to the repository at any time using the NFS protocol. The archival storage layer composed of following components.

- Relational database service provided by MySQL.
- JDBC connectivity driver to connect the database with Java applications.
- DLworm provides filing system service which performs operations of file transfer and manipulations on them.
- Item Replicator is responsible for providing two different memory locations for an item based on parameters like free space on any repository node or network load. This enables to retain data even in case of failure of any node.

Functional Toolkit

Functional Toolkit of DLworm performs the basic operations of Data retrieving and manipulation. Various functions of Functional Toolkit involve operations like—

- Content manager is responsible for establishing connections for data transfer from repositories to end users.
- DLworm uses document caching API to perform the end user caching of the most probable data to be accessed by the user after accessing the current one. It provides fast browsing experience to the users.
- Different users can be provided with different level of privileges. These authorization issues can be handled by Users moderator service.
- All users are provided with functionalities through which they can check the activity statistics of themselves or of other users depending upon the access privileges.
- Different users can communicate with other users by message passing module. It also provides automatic mail generation depending upon subscription of services.
- To use the available resources effectively we restrain from maintaining duplicate data in the repository. Before adding any new data to the repository the duplicity checker checks it with already existing data in the repository.
- Every event or any change is logged into the database by log manager. It enables us track the changes in the system.
- Such situations may arise when a user searches for a data item but the given query text does not match exactly with database or metadata entry because of different pronunciation system. Hence we use Phonetic Query System (PQS) to strengthen the data searching.
- Different actions performed and information-flow among different users is coordinated by workflow manager.
- Quality checker module provides semi-automatic mechanism to accept only quality data.

Service Layer

Service Layer is the interaction layer for various Web-users. This layer involves the components of DLworm which provide an interface to users. Different Components of the Service Layer are PQS, XML extraction tool, XML generation tool, Web Interface using JSP Pages and Java Applets, and so on. Service layer involves the following components.

- DLworm provides web based user interface for all users. DLworm uses JAVA applets, JSP, PHP and HTML for providing user interface.
- Each digital data is associated with an XML file. This XML file can be created using XML creator or already existing XML files

can be uploaded by data addition centre by using XML extractor tool.
- Data Transporters performs functionality of uploading and downloading data from different nodes in the repository.
- Some users, who are domain experts for a data item, can be given privilege to add some summary or comments associated with that data item. These services are handled by data addition tool.

Each Layer invokes the layer below it and the access is strictly sequential, that is, Service Layer cannot directly invoke Archival Storage Layer. The clearly identified boundaries enable the system to be modular where each module is separately replaceable and modifiable. This modularity enables high level of Plug-in ability and expandability of DLworm.

The implementation and detailed design is described in the next section.

Implementation and detailed design

DLworm has its primary strength in its implementation and detailed design. The design that will be described soon in the article enables DLworm to handle large amount of data redundantly and maintaining fault tolerance at

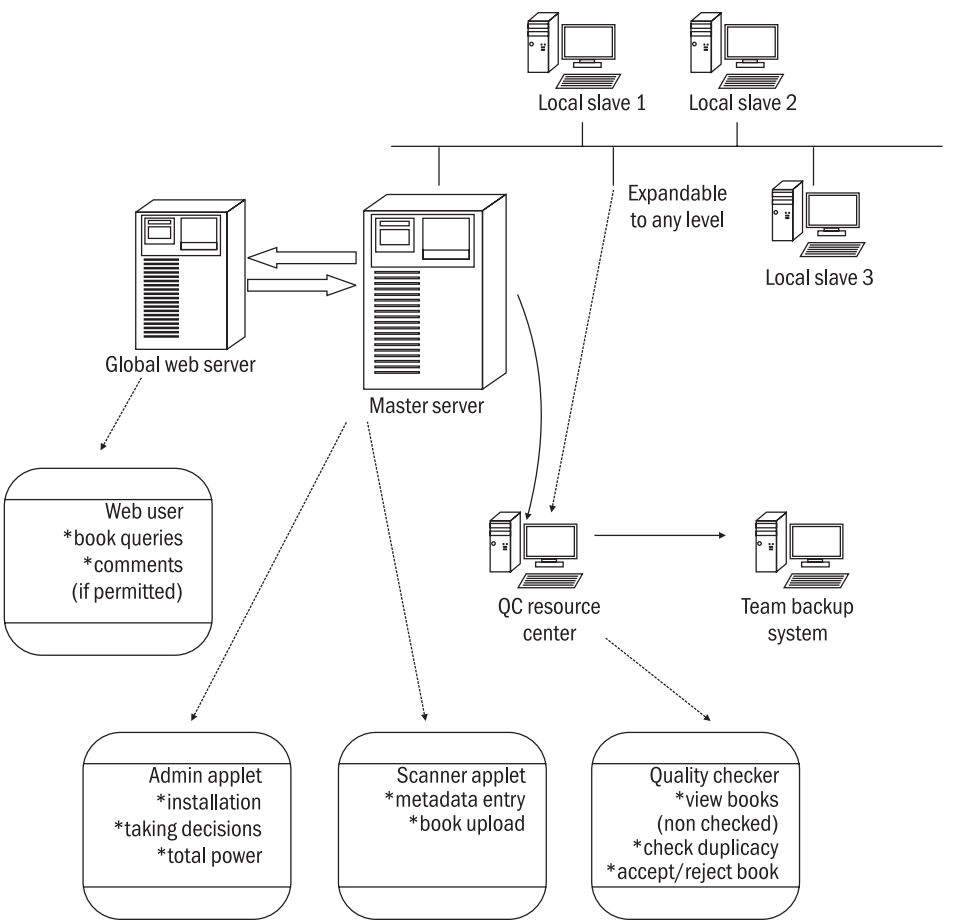

Figure 6 Deployment strategy for DLworm

all levels of data handling and storage. Figure 6 shows the deployment strategy of the DLworm system. We can divide the whole design into following four basic sections.

- DLworm Master Server
- Web-User Server
- QC Resource Centre
- Repository Systems

The basic functionalities of each section can be listed as

- DLworm Master Server
- Web-user Server
- QC resource centre
- Repository Systems

DLworm Master Server

This server has primary responsibilities of maintaining the database and manipulating connections to the Repository Systems. The memory drives of Repository Systems are accessed using NFS protocol that enables Master Server to mount the hard drives. The database stores the XML information of all the documents which helps in retrieving the location of the document queried. Master Server holds the database for authorization of all the users connected and maintains logs of all the actions performed by different users.

Web-user Server

This section is responsible for providing and manipulating connections to different Web-users, that is, the users who connect through Internet and query for any Digital Data. The server is provided by a Global Access so that it accepts connections from Web-users. On accepting a connection the query result is provided by performing PQS, that is, Phonetic Query Search on the Database of the Master Server. This server also maintains the events performed by the Web-users and supports Moderation of the Users. Different Web-users can be provided with different level of Document Access depending upon the moderation privileges provided by the System Administrator. Web-user Server maintains all the JSP pages and databases necessary for the Web-user. The JSP pages act as the interface for the Web-user through which any user can search and view the desired manuscript/ documents.

QC resource centre

Quality checking resource centre has the basic responsibility of data inclusion in the Library System. Data is collected from all data addition centres and Quality Checking procedure is performed before adding the data to the primary repository of the Library System. It also checks for the duplicity of the submitted data at content level. Each document added to the primary repository is replicated at two different locations in distributed primary repository which enables document retention even in case of failure of some nodes of the repository. This centre also runs a Database to log the activities of different data providers and Quality Checkers.

Repository Systems

These are the different nodes in distributed environment which archive the digital data of the Library System. The memory drives of these nodes are accessed by the Master Serves using NFS protocol.

Conclusion and future improvements

DLworm has been developed as a modular system where new functionalities have been added in stepwise fashion taking advantage of the object oriented design of the system. The underlying code of DLworm has been developed in JAVA. JAVA Applets running on Unix-like Systems such as Linux. DLworm fulfils most of the functionalities required for Digital Library System. Moreover, the use of Java and an Applet based interface allows the system to be platform independent.

DLworm provides a powerful system for building large digital libraries. One can, of course, add new functionalities that will make the system more user friendly. Keeping in view the recent development of OCR technologies, we expect that the full text of the digital content will become available. Thus, some of the improvements that are already in channel involve the automatic document summarizer (Sanyal, 2006; Tejaswi, Parai, Borah, *et al.* 2008), coreference resolution (Vishnuprasad and Sanyal, 2007) and some more additional modules for automatic quality checking. A document summarizer provides the summary of the data which might be helpful to provide an essence of the document. Coreference resolution will improve the quality of information retrieval from full text searches. Improvements in quality checking technologies give the strength to the repository and removes strain from the users associated with quality checking.

The success of DLworm achieved till now boosts our objective to contribute in the field of document preservation and workflow management and in the field of free access of educational material.

Acknowledgements

The authors are acknowledging the financial support to Ministry of Communication and Information Technology, New Delhi, India.

References

Bertrand Le S and Giuseppe A. 2004
Image recognition for digital libraries, pp. 91–98
[Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval]
New York: ACM

Castelli D and Pagano P. 2002
OpenDLib: A Digital Library Service System, pp. 292–308
[Proceedings of the 6th European Conference on Research and Advanced Technology for Digital Libraries]
London: Springer-Verlag

Castelli D, Pagano P, and Simi M. 2004
eLibrary and ARTE: two opendlib digital libraries, pp. 413
[Proceedings of the 4th ACM/IEEE-CS joint conference on Digital Libraries]

Chen X, Kim D, Nnadi N. Shah H, Shrivastava P, Bieber M, Im I, Yi-Fang Wu. 2003
Digital library service integration, pp. 384
[Proceedings of the 3rd ACM/IEEE-CS joint conference on Digital Libraries]
Washington, DC, USA: IEEE Computer Society

Forman G, Eshghi K, and Chiocchetti S. 2005
Finding similar files in large document repositories. pp. 394–400
[Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining]
New York, NY, USA: ACM

Kahn R and Wilensky R. 2006
A Framework for Distributed Digital Object Services
Intl. J. Digital Libraries, **6**(2):115–123

Lagoze C and Hunter J. 2001
The ABC Ontology and Model. pp. 160–176
[Proceedings of the International Conference on Dublin Core and Metadata Applications]
National Institute of Informatics, Tokyo

Levenshtein V.I. 1966
Binary codes capable of correcting deletions, insertions, and reversals.
Soviet Physics Doklady **10**: 707–710

Odell M K and Russel R C.1918,1922
US Patents 1261167 (1918), 1435663 (1922)

Raj D and Sanyal R. 2008
Indexing heritage documents in encyclopaedia using optimized Named Entity Recognition techniques
The 14th International Conference on Virtual Systems and Multimedia, VSMM 2008
Limassol, Cyprus, 20–25 October 2008

Sanyal R. 2006
Multilingual document summarization for digital libraries
Invited to talk at the International Conference on Digital Libraries, 5–8 December 2006, New Delhi, India

Smith M, Barton M, Bass M, Branschofsky M, McClellan G, Stuve D, Tansley R, Walker J H. 2003
***DSpace: An* OpenSource Dynamic Digital Repository**
D-Lib Magazine **9**(1) Details available at <http://www.dlib.org/dlib/january03/smith/01smith.html>

Tansley R, Bass M, Stuve D, Branschofsky M, Chudnov D, McClellan G, Smith M. 2003
The DSpace institutional digital repository system: current functionality. pp. 87–89
Proceedings of the 3rd ACM/IEEE-CS joint Conference on Digital Libraries
Washington, DC: IEEE Computer Society

Tejaswi T, Parai G K, Borah P K, Shah S, Sanyal S. 2008
SCORES: Summarizer using Combination and Reduction of Extracted Sequences
Accepted for publication in *International Journal of Reasoning-based Intelligent System*, Science Direct, Elsevier

Vishnuprasad T M and Sanyal R. 2007
Coreference resolution using hybrid approach
UDL publication: The International Conference on Digital Libraries, 2–4 November 2007, Pittsburg

Zobel J and Dart P. 1996
Phonetic string matching: lessons from information retrieval. pp. 166–173
Proceedings of the Eighteenth ACM SIGIR International Conference on Research and Development in Information Retrieval