Chapter 5

Development of BURA: Software and Standards

“Technology is dominated by two types of people: those who understand what they do not manage and those who manage what they do not understand.”

Archibald Putt
5.0 Introduction
5.1 Design of Software Framework
5.2 Design of Multilingual Interface
5.3 Design of Submission Interface
5.4 Development of Subject Access Control
5.5 Development of Federated Search System
5.6 Incorporation of User Interaction Tools
5.7 Back up and Restoration Process
5.8 Development of ISO image
5.0 Introduction

This chapter aims to develop a software framework to support Web-enabled and Unicode-compliant IDR system for the University of Burdwan based on open standards and open source software (OSS) architecture. It has already been decided that the system should use open standard as far as feasible and should have ability to integrate with the other interoperable OAR system in order to harvest resources. This chapter describes how to make Bengali/Hindi language based user interfaces for storing, processing and retrieving Bengali/Hindi language based resources. This software framework allows searching and browsing of Bengali/Hindi languages based resources and renders services based on Bengali/Hindi languages. As it is not possible to produce here all the programmes in textual form due to space limitation, the programmes are given in the accompanying CDROM with proper indexes. This chapter is concerned with the development of librarian interface as well as user interface of the software framework and is available in Bengali script (along with English language interfaces). This is the management layer (librarian interface) of the model (BURA) through which librarians can perform various professional and administrative operations such as organization of resources under community and sub-community, managing of user registration, E-People selection and privilege control, management of interaction tools, import and export of resources etc. The success of user interfaces of the model (BURA) largely depends on this administrative (librarian) interface. The chapter is also concerned with the designing and development of a metadata harvesting framework using PKP harvester that will harvest metadata from different OAI-PMH compliant repositories and thereby ensures a federated search interface for end users. Several parameters have been framed for selecting repositories from OpenDOAR database and a comparative study of different harvesting tools has been conducted for selecting the most comprehensive harvester. This chapter reports applications of controlled vocabulary device to browse and search specific subject in BURA software framework. The subject access system of BURA also supports indexing, searching and browsing of resources in Bengali. This chapter also shows integration of selected social networking software like Blog, RSS in BURA software framework. Finally a ready-to-use product has been developed to help fellow professionals in implementing IDR in their respective institute without reconfiguring the system.

5.1 Design of Software Framework

The open source software of the Cluster - 1 (Basic Cluster) must be installed and linked properly before installation of software of other Clusters. Installation of Apache, MySQL, PostgreSQL, JAVA, PERL, and PHP is quite easy and straightforward in Linux operating system but their seamless linking is bit tricky. This
seamless linking of software in Basic Cluster (Cluster I) is must for developing the environment where all the software in different Clusters including community communication tools can interact with each other impeccably. So customization in DSpace (IDR Cluster) is required for the following reasons –

- Multilingual information processing and retrieval;
- Translation in Indic script base languages (here Bengali script);
- Incorporation of social networking software;
- Designing and development of metadata harvesting framework; and
- Incorporation of metadata schemas for managing special type of documents (like ETD, learning objects etc.).

The development process of IDR Cluster in BURA software framework involves following major steps –

**Step 1:** Integration of Basic Cluster with IDR Cluster in the software framework so that administrator can use and access IDR Cluster as a separate module in Basic Cluster;

**Step 2:** Preparation and management of community, sub-community and collection of BURA resources as prescribed in section 4.2.1.2 of chapter 4 along with their Bengali language equivalent;

**Step 3:** Developing controlled vocabulary device in Bengali in order to support indexing, browsing and searching specific subject categories;

**Step 4:** Designing and development of a metadata harvesting framework in IDR Cluster of the Model (BURA) using selected harvester (as proposed in section 4.2.2.4.3 of chapter 4);

**Step 5:** Enhancement of IDR Cluster by integrating selected social networking software like Blog, RSS in BURA software framework;

**Step 6:** Development of maintenance, backup and restoration procedures; and

**Step 7:** Preparation of ISO image file for porting BURA framework to help installing in other libraries.

DSpace has been selected for building IDR Cluster of BURA (see section 4.3 of chapter 4). DSpace itself is not Unicode-compliant but all supporting software necessary for running DSpace support universal character set. Therefore, the first task is to make DSpace Unicode-compliant for supporting Bengali script (both interface and processing of records). This research work deals with this programme level task.
without changing the original source code of DSpace. It develops two Unicode-compliant themes – one for librarian interface and another for user interface. Librarian or the person responsible for managing IDR Cluster can select any of these themes by setting language preferences in browser (e.g. Mozilla Firefox). This is the management layer of BURA through which repository manager can perform various professional and administrative operations such as organization of resources under different community, sub-community and collection, management of community communication and interaction tools like RSS, Blog. However, it is possible to go back to default themes of DSpace at any time. It includes all the modules necessary for repository management activities as well as administrative operation.

a) Linking of Apache with PERL and PHP

The configuration file of Apache (httpd.conf) is modified to link PERL programming environment with Apache server. It is required to develop application environment for additional software in basic IDR framework such as Blog software and harvesting software.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP_HOST</td>
<td>localhost</td>
</tr>
<tr>
<td>HTTP_USER_AGENT</td>
<td>Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.2.20) Gecko/20110805 Ubuntu/10.04 (lucid) Firefox/3.6.20</td>
</tr>
<tr>
<td>HTTP_ACCEPT</td>
<td>text/html,application/xhtml+xml,application/xml;q=0.9,<em>/</em>;q=0.8</td>
</tr>
<tr>
<td>HTTP_ACCEPT_LANGUAGE</td>
<td>en;es;q=0.8,en;q=0.5,en-us;q=0.3</td>
</tr>
<tr>
<td>HTTP_ACCEPT_ENCODING</td>
<td>gzip, deflate</td>
</tr>
<tr>
<td>HTTP_ACCEPT_CHARSET</td>
<td>ISO-8859-1;utf-8;q=0.7;q=0.7</td>
</tr>
<tr>
<td>HTTP_KEEP_ALIVE</td>
<td>115</td>
</tr>
<tr>
<td>HTTP_CONNECTION</td>
<td>keep-alive</td>
</tr>
<tr>
<td>PATH</td>
<td>/usr/local/bin:/usr/bin:/bin</td>
</tr>
<tr>
<td>SERVER_SIGNATURE</td>
<td>&lt;address&gt;Apache/2.2.14 (Ubuntu) Server at localhost Port 80&lt;/address&gt;</td>
</tr>
<tr>
<td>SERVER_SOFTWARE</td>
<td>Apache/2.2.14 (Ubuntu)</td>
</tr>
<tr>
<td>SERVER_NAME</td>
<td>localhost</td>
</tr>
<tr>
<td>SERVER_ADDR</td>
<td>::1</td>
</tr>
<tr>
<td>SERVER_PORT</td>
<td>80</td>
</tr>
<tr>
<td>REMOTE_ADDR</td>
<td>::1</td>
</tr>
<tr>
<td>DOCUMENT_ROOT</td>
<td>/var/www</td>
</tr>
<tr>
<td>SERVER_ADMIN</td>
<td>webmaster@localhost</td>
</tr>
<tr>
<td>SCRIPT_FILENAME</td>
<td>/var/www/bijan/test.php</td>
</tr>
<tr>
<td>REMOTE_PORT</td>
<td>51938</td>
</tr>
<tr>
<td>GATEWAY_INTERFACE</td>
<td>CGI/1.1</td>
</tr>
<tr>
<td>SERVER_PROTOCOL</td>
<td>HTTP/1.1</td>
</tr>
</tbody>
</table>

Fig. 5.1: Apache and PERL Linking in Basic Cluster
Similarly, the file (httpd.conf) can be modified to link PHP programming environment with Apache server.

![PHP Version 5.3.2-1ubuntu4.9](image)

**Fig. 5.1.1: Apache and PHP Linking in Basic Cluster**

### b) Linking of MySQL with PERL and PHP

Interaction of PHP and PERL with MySQL is necessary to retrieve data from databases (Blog database and Harvester database). Generally, both PHP and PERL language pack includes linking instructions in the configuration file by default. PHP does it through php.ini file and PERL sets the link through DBI module.
(DBD::mysql). The testing of link between PHP and MySQL is shown through two screen snapshots (Fig. 5.1.2 & Fig. 5.1.3).

<table>
<thead>
<tr>
<th>Directive</th>
<th>Local Value</th>
<th>Master Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>mysql.allow.local.infile</td>
<td>On</td>
<td>On</td>
</tr>
<tr>
<td>mysql.allow.persistent</td>
<td>On</td>
<td>On</td>
</tr>
<tr>
<td>mysql.connect_timeout</td>
<td>60</td>
<td>60</td>
</tr>
<tr>
<td>mysql.default_host</td>
<td>no value</td>
<td>no value</td>
</tr>
<tr>
<td>mysql.default_password</td>
<td>no value</td>
<td>no value</td>
</tr>
<tr>
<td>mysql.default_port</td>
<td>no value</td>
<td>no value</td>
</tr>
<tr>
<td>mysql.default_socket</td>
<td>/var/run/mysql/mysql.sock</td>
<td>/var/run/mysql/mysql.sock</td>
</tr>
<tr>
<td>mysql.default_user</td>
<td>no value</td>
<td>no value</td>
</tr>
<tr>
<td>mysql.max_links</td>
<td>Unlimited</td>
<td>Unlimited</td>
</tr>
<tr>
<td>mysql.max_persistent</td>
<td>Unlimited</td>
<td>Unlimited</td>
</tr>
<tr>
<td>mysql.trace_mode</td>
<td>Off</td>
<td>Off</td>
</tr>
</tbody>
</table>

Fig. 5.1.2: Linking of MySQL with PERL and PHP for DSpace

After successful integration of MySQL and PHP, the web access shows the result:

Fig. 5.1.3: Connection setting of MySQL in Basic Cluster
5.2 Design of Multilingual Interface

Contents creation in local languages is one of the most vital issues in any digital library system. It is necessary for promoting internet use, empowerment of the masses and to preserve cultural heritage and social identity. Therefore, the model BURA, recommends Bengali/Hindi language based interfaces (vide section 4.2.1.8 & 4.2.1.14 of chapter 4) and contents development in Bengali/Hindi languages in a multilingual environment. The development process of Multilingual Cluster (Cluster III) in BURA software framework involves following major steps –

Step 1: Development of Multilingual Cluster for BURA i.e. performing necessary programme level modifications of DSpace to make it Unicode compatible;

Step 2: Development of Multilingual Cluster in the software framework so that librarian can access and use Bengali script based interface for managerial activities;

Step 3: Development of Multilingual Cluster in the software framework so that end users can access Bengali script based interface for retrieval activities like browsing, simple searching and advanced searching;

Step 4: Designing and developing communities, sub-communities, collections, preparation of categories and sub-categories of BURA resources and other administrative works (as prescribed in section 4.2.1.2 of chapter 4) along with their Bengali language equivalent;

Step 5: Development of user registration form in Bengali script, password authentication mechanism and associated help file to ease the job of user registration (vide section 6.3 of chapter 6); and

Step 6: Developing controlled vocabulary device in Bengali in order to support searching and browsing specific subject category in Bengali language.

5.2.1 Multilingual Data Entry

Data entry activity in this BURA software framework is thus quite an easy job as it is supported by a number of data standardization tools. However, multilingual data entry works by the application of tools from Multilingual Cluster (Cluster III) of software framework requires special mention. It shows and explains use of Unicode-
compliant tools in entering Bengali script based records into the IDR cluster of the software framework. The virtual keyboard (Avro) of the framework can be invoked and to Bengali script entry mode simply by pressing F12 function key or from system tray icon of Avro (Khan, 2006) in Windows client. System manager can display of the Bengali script keyboard easily from system tray icon to enter Bengali script based data into the framework. At any point of time system manager can go to the default English language keyboard mode by pressing F12 function key again or by closing the Avro top bar in Windows client. The next step in developing application environment is to install and configure open source software of Cluster III (Multilingual Cluster) i.e. Avro virtual keyboard, USP10.DLL rendering engine and Open type fonts for Bengali script for Windows client. Presently rendering of Indic scripts in Linux (Fig. 5.2.0) is available readily for all Indic scripts (for example Ubuntu LTS 12.04 allows selection of any Indic script for the purpose of native script, interface and data entry works).

Entering Bengali script based data into Linux server through Windows client may also be done by submitters and end users. The Avro keyboard supports three mode of typing – direct typing by using layout (5 layouts are available), phonetic typing and click and type facility i.e. typing through mouse. Direct typing can be invoked either by pressing the F12 function key or by clicking the system tray icon of Avro keyboard. The following screen snapshots show the layout of Avro virtual keyboard (Fig. 5.2.1); click and type interface of the Avro virtual keyboard (Fig. 5.2.1.1) and layout Google Online Translator keyboard (Fig. 5.2.1.2). Avro adopted ‘UniBijoy
"Keyboard Layout", which is actually a modified version of ‘Bijoy Keyboard Layout’. There are some major differences of key locations between these two layouts and moreover, ‘UniBijoy Keyboard Layout’ is phonetic whereas ‘Bijoy Keyboard Layout’ is symbolic in architecture.

Fig. 5.2.1: Layout of Avro Virtual Keyboard

Fig. 5.2.1.1: Click on Type Interface of Avro Virtual Keyboard
5.2.2 Multilingual Features

Multilingual support is a relatively new phenomenon in digital library environment and one of the goals that BURA has been designed to pursue is to allow multilingual support in the easiest possible way. So the system is expected to be Unicode compliant and fully multilingual in nature in order to support processing and retrieving different regional languages. This section presents a method of information harvesting and consolidation to support the multilingual information requirements for cross-language information retrieval within digital repository systems. So, it is vital to implement a multilingual interface (e.g. User-interface in Bengali & Hindi) for a digital knowledge repository (BURA) to increase the world-wide potential for access to public funded research output.

DSpace is an excellent institutional repository system for archiving digital objects in a variety of formats. It, as open source sophisticated institutional repository software, can be customized extensively to develop Indic script based IDR on different subjects. It supports multiple languages in the user interface, in the metadata fields and in the digital contents. The Unicode Standard is the universal character-encoding scheme for written characters and text. It defines a consistent way of encoding multilingual text that enables the exchange of text data internationally and creates the foundation for global software. In the beginning Unicode was a simple, fixed-width 16 bit encoding. Over the time, Unicode changed this fixed-width encoding style and presently allows three different forms of encoding to meet different requirements.

- UTF-8 attempts to allow legacy systems to use Unicode by coding the characters in the ASCII character set with only eight bits, and encoding characters that are not in the ASCII character set with 16 bits. This is commonly used for Web pages.
• UTF-16 is supplementary characters outside the basic multilingual plane. It encodes most of the world’s major languages in a fixed 16-bit character representation (2 bytes). This is the most common implementation.
• UTF-32 is an actually UCS 4, given a new name. It uses four bytes (32 bits) to encode all possible characters (rarely used).

5.2.3 Bengali Script based User Interface

Developing a user interface for IDR is not as easy as users are not geographically bounded. User interface of an IDR system demands more functionalities because such a system needs to support not only information retrieval but also submission, community communication and interaction and other administrative operations such as community creation, E-person management etc. In India, another requirement for user interface is its availability in local language. Therefore design of a Bengali language based interface (as proposed in section 4.2.1.8 & 4.2.1.14 of chapter 4) by following established design standards and incorporating easy-to-use search engine that has different levels of search sophistication can contribute greatly in success of BURA. This layout is designed in view of barriers like lack of contents in local language, a low rate of literacy and lack of necessary ICT skills among general users. The interface has the following characteristics -

• Provides a friendly and interactive interface to enable users to navigate effortlessly through the IDR system;
• Supports universal character set or in other words helps in contents development and contents access in Unicode-compliant Bengali script;
• Allows integrated browsing and searching in multilingual environment and dissemination of multiple media such as text, audio, video objects etc;
• Provides to access community communication tools (like E-mail, Instant messenger) and community interaction tools (like Blogs, Discussion forum);
• Allows to create Community and Sub-community, E-People, user registration;
• Allows to organize resources under community and sub-community and by using standard subject access system;
• Provides access and submission of dissertations, theses and other educational objects; and
• Allows easy submission of objects of different forms and formats by authorized users.

5.2.4 Bengali Script based Digital Repository through DSpace

Bengali script based IDR system on DSpace has been developed by the application of different open source software for the University of Burdwan. As PostGreSQL (the backend RDBMS for DSpace) supports UTF-8, the Bengali language based records can be retrieved and exhibited directly by using suitable GUI interface of PostGreSQL. Design of Bengali script based user interface, searching and browsing
mechanisms in DSpace requires three steps to be performed in logical sequence and is demonstrated here through three screen snapshots (Fig. 5.2.4 through Fig. 5.2.4.4).

**Step 1**

The first logical step of designing Indic script based user interface is making UTF-8 as default character encoding scheme or native character set for DSpace database in PostGreSQL RDBMS (Fig. 5.2.4). This step maintains a link between BURA software framework and the database used by the system.

![Fig. 5.2.4: DSpace database with UTF-8 as native character set](image)

**Step 2**

Making necessary changes and translations of messages (from English to Bengali) in Messeage.properties file (available in the location - /webapps/dspace/WEB-INF/classes/Messages.properties) is the next logical step to make the default interface in Bengali script. The following figure (5.2.4.1) shows the English language messages for Submission Aspect section of DSpace.
Chapter 5: Development of BURA

This default message property file has been translated in Bengali by using carefully translated message headings through the use of Unicode-compliant text editor. The figure (5.2.4.2) below shows the Bengali equivalents of English language messages for Submission Aspects section of DSpace.

Fig. 5.2.4.1: Original Message File in English (Submission Aspect)

Fig. 5.2.4.2: Modified Message File in Bengali (Submission Aspect)
Step 3

Making necessary changes in server.xml file (Configuration file of Tomcat) is required to ensure UTF-8 enabled transactions between Tomcat server and DSpace database. The first figure (5.2.4.3) shows default server.xml file and the second figure (5.2.4.4) shows the modified server.xml to support Unicode based database transactions.

As discussed earlier (in section 4.2.1.8 of chapter 4), this research work recommends design and development of user interfaces both in Bengali and Hindi languages. Here steps are demonstrated only for making interface into Bengali language. In the same fashion, default English interface can be modified into Hindi language.
5.3 Design of Submission Interface

BURA holds varieties of textual and non-textual resources in different forms and formats (as suggested in section 4.2.1.3 of chapter 4). So it is essential for BURA to have standard metadata schema(s) for describing various items. BURA has decided to use Dublin Core (DC) and qualified DC as generic metadata schema for describing wide range of textual resources and suggests community/or domain-specific metadata elements for those objects where no suitable element or element refinement exists (as recommended in section 4.2.1.7 of chapter 4). In case of non-textual objects, BURA recommends respective domain-specific schemas for special objects like ETD (ETD-MS), Learning Objects (IEEE-LOM), Journal articles (Qualified DCMES) (as suggested in section 4.2.2 of chapter 4) respectively.

5.4 Development of Subject Access Control

This research work has incorporated DDC (22\textsuperscript{nd}) in BURA software framework and modified it in Bengali. It has been restricted up to the third (3\textsuperscript{rd}) Summary in view of the limited manpower and time. But it is a team work and can further be extended to incorporate the entire DDC structure by following the SKOS based method developed by this research work. The steps of making DDC (in English and Bengali) at the time of indexing as well as searching is discussed below with different screen snapshots (Fig. 5.4 through 5.4.3).

Step 1

Making necessary changes in \textit{DSpace.cfg} file is the first logical step in integrating DDC (22\textsuperscript{nd} edition – upto 3\textsuperscript{rd} summary) in software framework. This step provides link (requires to display subject access system in submitter, editor and end user interfaces) between digital archive and available controlled vocabularies.

```plaintext
### Controlled Vocabulary Settings ####

# Enable or disable the controlled vocabulary add-on
# Warning: this feature is not compatible with WAI (it requires javascript to function)
# webui.controlledvocabulary.enable = true
```

Fig. 5.4: Script to Link to Control Vocabulary Device (DDC in DSpace)

Step 2

Next logical step is to link the target controlled vocabulary with submitter and editor interfaces through necessary modifications in \textit{input.xml} file. The modified \textit{input.xml} file of the system is displayed below:
Step 3

This research work attempts to integrate DDC 22nd edition (upto 3rd summary) in both English and Bengali scripts. English language subject divisions and subdivisions are based on DDC 22nd edition and Bengali language equivalents are based on the available translation work (Saha, 2008). The structure of the SKOS-enabled XML-formatted file displaying the hierarchy Social sc. >> Education >> Subject categories under ‘Education’ is given here in English.

Fig. 5.4.1: Modified Input.xml File

Fig. 5.4.2: Original HTML File of DDC (3rd Summary)
Similarly, hierarchy Social sc. >> Education >> Subject categories under ‘Education’ is also displayed for Bengali script.

![Fig. 5.4.3: Modified HTML File in Bengali (DDC 3rd Summary)](image)

The novelty of this SKOS-enabled vocabulary control system is that, submitter can add standard subject term(s) from DDC. It (DDC) has been incorporated in BURA software framework and will appear during submission and searching. At the time of submission of items to BURA, submitter has to select at least one term (one in English and another in Bengali) from each category, otherwise the system will show the caution message (Fig. 5.4.5) and submitter will not be allowed to go to the next step. The selection of subject term(s) from DDC for both categories (English and Bengali) is shown here in two different screen snapshots (Fig. 5.4.4 & Fig. 5.4.6).

![Fig. 5.4.4: Selection of Subject Categories from DDC (Category 1 – English)](image)
The next subsequent window (Fig. 5.4.7) displays the selected subjects in each category (English and Bengali) from DDC.
After successful submission, reviewer/or metadata editor can further edit metadata of the submitted item(s) provided by the submitter at the time of submission and can add a new metadata for both categories (English & Bengali) from DDC (Fig. 5.4.8).
A click on ‘Edit Metadata’ option (Fig. 5.4.8) will lead to following facility:

![Fig. 5.4.9: Display of Results of Subject Categories (English & Bengali)](image)

### 5.5 Development of Federated Search System

In section 4.2.1.7 of chapter 4, this research study proposed a metadata harvesting framework that will facilitate metadata extraction from OAI-PMH compliant repositories. The purpose of this federated search interface is to build an easy-to-use search service through automatic record harvesting for providing access to databases of worldwide open access repositories on different disciplines of interest.

The prototype harvesting framework for BURA is based on open source software (OSS) and open standards. This framework, like many other systems is based on LAMP (Linux – Apache – MySQL - PHP) architecture and all the necessary dependent tools must be activated and configured properly before installation of PKP (vide section 4.2.2.4.3 of chapter 4) harvester in Ubuntu. The figure (5.5) shows the harvester setting options in `dspace.cfg` file under the section with the following heading: ### Harvester settings
As a whole, the use of OSS in developing prototype software framework depends on structured strategies, planning and methodology. The steps related to the process of designing harvesting prototype framework for the model BURA may be divided into four broad groups. These are as follows-

A. Installation related Functions

This harvesting framework requires different OSS that need to be installed are mentioned below -

- Operating system (OS): Ubuntu Version
- Database: MySQL Version
- Web server: Apache Version
- Programming language environment: PHP Version
- Harvester: PKP Harvester Version 2.0.1

B. Software (LAMP) related Activities

This group is mainly concerned with the development of the software framework. For this purpose, all those software are required to be customized and linked with each other properly. PKP harvester 2.X is based on AMP (Apache – MySQL – PHP) architecture. So the proposed framework requires installing (Apache, MySQL and PHP) those software before installation of PKP harvester. Although there is no hard and fast rule, the order of installation sequence is as follows:
Apache (The Apache httpd server is powerful, flexible, HTTP/1.1 compliant open source Web server):

- Installation of Apache;
- Testing of Apache; and
- Apache Configuration and Control.

MySQL (MySQL, is the most powerful open source SQL database in open domain, is developed, distributed and supported by MySQL AB):

- Installation of MySQL;
- Initialization of MYSQL Server;
- Creation of database, user and manage permission;
- Testing of AMP Links through scripts;
- Testing PHP - Apache Link; and
- Testing PHP - MYSQL Link.

PHP (PHP is an open source server side scripting language):

- Installation of PHP; and
- Configuration of PHP

C. Harvester related Functions

This part deals with selection, installation and configuration of harvesting tool. This group includes two major activities:

- Installation of PKP harvester (as proposed in section 4.2.2.4.3 of chapter 4)

The installation of PKP harvester is quite easy. It requires two sets of information –

- Login name and password (for administrator); and
- Database details (name of the MYSQL database, user of database and password of the database user).

- Configuration of PKP harvester

The configuration processes are divided into three main groups:

- Site management (configuration of site-specific details, language, crosswalk, plugins and reading tools);
- Archives (addition or creation of archives; editing or managing created archives); and
- Other administrative functions (layout, interface customization etc.).
D. Repository related Activities

The most important task of the repository administrator is to set up archive(s) for metadata harvesting. This part is concerned with the selection of repositories against predefined parameters (as proposed in section 4.2.2.4.2 of chapter 4) and collection of essential attributes for harvesting mentioned below –

- Name of the repository;
- Description of the repository;
- Base URL of the repository; and
- OAI-PMH URL (resource URL) of the repository.

This research study has selected repositories registered in OpenDoar database. The system (BURA) has the capability to include new OAI-PMH compatible OARs and to update records for existing OARs already in the system for the purpose of developing a comprehensive single window local search services for OARs.

5.5.1 Organization and Management of Repositories Harvested

This research study has selected sixty (60) OAI-PMH compliant repositories from OpenDOAR database against selected parameters (as suggested in section 4.2.2.4.2 of chapter 4). Any one can become a member of the system and can avail all the facilities provided by the system. User registration process includes ‘username’ and ‘password’ (Fig. 5.5.1) and the ‘Forgot your password?’ link (Fig. 5.5.2) allows users to set a new password.

![Fig. 5.5.1: User Registration to Global Repositories System](image)
The organization and management of repositories is vital task and system manager can do only after duly log in as administrator to BURA (Fig. 5.5.3).
The window (Fig. 5.5.4) allows administrator to perform various administrative operations (e.g. add, edit, manage etc.) in system level and is shown through different screen snapshots (Fig. 5.5.5 through Fig. 5.5.8).

A click on ‘Site Settings’ option (Fig. 5.4.4) displays another window (Fig. 5.5.5) to configure the administrative interface of the harvesting framework.
Designing Institutional Digital Repository for the University of Burdwan: A FLOSS Based Prototype

Fig. 5.5.5: Site Management of Global Repositories System
Any time, addition, deletion, edition of repository(ies) is possible and button ‘Add Archive’ (Fig. 5.5.4) can be clicked to add a new repository to the system (Fig. 5.5.6).

Managing archives is a vital task and can be done by clicking on ‘Manage Archives’ button (Fig. 5.5.4). A click on this button displays another window (Fig. 5.5.7) having three options and repository manager can pick up any buttons to perform the specific operation.
The moment administrator clicks on ‘Edit’ option (Fig. 5.5.7), the system displays the selected archive (here E-LIS repository) and allows administrator making necessary changes (Fig. 5.5.8).
This software framework allows metadata to be harvested and extracted globally and can be done anytime by clicking on ‘Fetch archive metadata’ button (Fig. 5.5.6). Administrator can harvest all metadata or can restrict it by selecting different pre-defined categories (e.g. subject, type etc.) set by the repositories (Fig. 5.5.9). The system displays total number of metadata harvested and even allows updating metadata records by pressing ‘Update Metadata Index’ button (Fig. 5.5.9).

![Fig. 5.5.9: Fetch and Update Archive Metadata](image)

Administrator can sort/or arrange repository(ies) in different orders (e.g. title, author, repository type ) as specified by the repositories (Fig. 5.5.10).
Any time, a new sorting order can be established by clicking on ‘Create Sort Order’ option (Fig. 5.5.10). The system manager can pick up any fields (Fig. 5.5.11) using drop down menu for creating a new order.
Even this existing sorting order can be modified and a fresh order can be re-established again by selecting different pre-defined fields as specified by the system (Fig. 5.5.12).

**Fig. 5.5.12: Edition of Sorting Order of Repositories**

### 5.6 Incorporation of User Interaction Tools

Community communication software (known as social software) allows digital interaction among community members through different user interaction tools such as RSS, Blog Discussion forum etc. BURA is fully compliant with these tools and can be incorporated in the software framework. These two social networking tools are developed with administrative interface of BURA and finally linked with BURA user interface.

#### A. RSS

RSS allows users to easily stay informed by retrieving the latest contents from the sites (BURA) without visiting it. The following are the steps to enable the use of RSS feeds in BURA. The following are the steps mentioned below:
Step 1: Administrator need to go the files (dspace-source/config/dspace.cfg) and has to perform the following steps:

- Edit the file [dspace-source] /config/dspace.cfg;
- Set webui.feed.enable = true; and
- Start Tomcat to get the result or for the changes to take effect.

All RSS feed options are available in dspace.cfg file under the section with the following heading: <p class="Example">#### Syndication Feed Settings ######

To enable RSS feeds for new/updated items in DSpace installation, Administrators need to set the following variable to -

true
webui.feed.enable = false

---

**Fig. 5.6: Script to Link to RSS Feed in User Interface**

### B. Blog

There are many open source software tools in the category of social networking software such as WordPress, Wblogs (Blog software). Selection of required Blog software for BURA software framework is absolutely based on the download statistics of Sourceforge (Sourceforge, 2012). It is the world’s largest repository in the domain of open source software. As per this statistics the most popular Blog software, WordPress has maximum number of downloads and installations. It has been integrated with BURA software framework. The first figure (5.6.1) is meant for repository manager to configure the WordPress and the second figure (5.6.2) displays configuration window of WordPress (User interface of Blog service is shown in section 6.6 of chapter 6).
Fig. 5.6.1: Log in as Administrator to Blog of BURA

Fig. 5.6.2: Modified Interface of Blog of BURA
5.7 Backup and Restoration Process

The software framework of BURA is complex in nature as it includes seamless agglomeration of different domain-specific open standard software or software groups. Keeping in the requirements of the authority, this research work includes an accompanying CDROM containing all the ingredient software and BURA framework programmes. Regular backup of BURA resources in suitable media is necessary so that all processed data and records can be recovered to the point of system failure. BURA resources are stored in the IDR Cluster. The process of backup and restoration is demonstrated by using pgAdmin III (a third party open source tool for managing PostGreSQL DBMS) through three screen snapshots (Fig. 5.7 through Fig. 5.7.2).

![Backup and Restoration (step 1)](image)

Fig. 5.7: Backup and Restoration (step 1)

The next logical step of this backup and restoration process is selecting the proper location where resources in the database of the system are to be stored (Fig. 5.7.1).
Any time resources stored in the database can be restored by clicking on ‘Restore’ button (Fig. 5.7) and administrator can do it by selecting the file where resources were stored (Fig. 5.7.2).
5.8 Development of ISO image

This research promises to deliver a ready to use off-the-shelf products so that other universities can apply the results of this research in an easy way. The needs for developing such products are: (i) ready to use Bengali script based interfaces developed for administrator, submitters, editors and end users; (ii) seamless export of SKOS-enabled DDC (22nd edition up to the 3rd summary) as controlled vocabulary to ensure standardization of subject authority file; (iii) allowing users of this software framework to implement integrated use of federated searching, community Blog, Discussion Forum and RSS based alerting services.

The basic purpose of this ready-to-use product is to help fellow professionals in implementing IDR in their respective institute based on the model BURA. This product, when installed, incorporates all the customizations and integration done by this research work without reconfiguring the system. This provides an opportunity to implement the software products as-it-is- basis without requiring technical skills and competencies. This research work also plans to release the product as open source utility through sourceforge.net.

The development of ISO file for BURA software framework is based on two sets of utilities -i) Remastersys GUI and ii) Remastersys CUI. The first set of utility generates raw ISO file and the second set of utility is responsible for fine tuning the ISO file in terms of file size control, name control, security control etc. The mechanisms for developing this product are as follows:

A Checking of the BURA software framework to ensure all software features, utility software and add-ons are working in desired ways;

B Generation of raw ISO image through remastersys; and

C Fine tuning of ISO image through remastersys command.

B. Generation of raw ISO image through remastersys

In this stage, first work is to invoke remastersys (Fig. 5.8).
Remastersys allows generating of the ISO image of the software framework through the option “Distribution” (second in sequence). But before utilizing this facility, development process needs to set configuration. The appropriate configuration setting required for exporting BURA framework as ISO image is given in Fig. 5.8.1.

![Fig. 5.8: Log in to Remastersys](image1)

![Fig. 5.8.1: Modification in Remastersys](image2)
C Fine tuning of ISO image through remastersys command

It is the final process of creating ISO image. After setting configuration in right manner, the process of developing ISO image may be started (Fig. 5.8.2 & Fig. 5.8.3).

![Remastersys](image1)

Fig. 5.8.1: Process of developing ISO image (step 1)

![Remastersys](image2)

Fig. 5.8.1: Process of developing ISO image (step 2)