DAR: A Digital Assets Repository for Library Collections – An Extended Overview

Iman Saleh[†] Noha Adly[†*] Magdy Nagi[†*]

[†]Bibliotheca Alexandrina
El Shatby 21526
Alexandria, Egypt
{iman.saleh, noha.adly,
magdy.nagi} @bibalex.org

[*]Computer and Systems
Engineering Department
Alexandria University
Alexandria, Egypt

ABSTRACT

The Digital Assets Repository (DAR) is a system developed at the Bibliotheca Alexandrina, the Library of Alexandria, to create and maintain the digital library collections. The system introduces a data model capable of associating the metadata of different types of resources with the content such that searching and retrieval can be done efficiently. The system automates the digitization process of library collections as well as the preservation and archiving of the digitized output and provides public access to the collection through browsing and searching capabilities. The goal of this project is building a digital resources repository by supporting the creation, use, and preservation of varieties of digital resources as well as the development of management tools. These tools help the library to preserve, manage and share digital assets. The system is based on evolving standards for easy integration with web-based interoperable digital libraries.

1. INTRODUCTION

The advent of digital technology and high speed networks are leading to widespread changes in services offered by libraries. Based on their experiences on the Internet, users now expect the same easy and seamless access to the library information resources. Librarians are recognizing the importance of assets management to deliver comprehensive information solutions. The heightened user expectations combined with the growth of collections based on digital content makes it increasingly important for all libraries to find efficient tools to manage their digital contents and enable instant access to their digital assets. Also, there is an increasing need to deploy tools to facilitate the digitization and the long-term preservation of all materials.

The Digital Assets Repository (DAR) of Bibliotheca Alexandrina (BA) acts as a repository for all types of digital material and provides public access to the digitized collections through web-based search and browsing facilities. DAR is also concerned with the digitization of material already available in the library or acquired from other research-related institutions. A digitization laboratory was built for this purpose at the Bibliotheca Alexandrina. The lab is equipped with the state of the art technologies for digitizing different types of material including slides in multi formats, negatives, books, manuscripts, pictures and maps, audio and video.

The goal of this project is building a digital resources repository by supporting the creation, use, and preservation of varieties of digital resources as well as the development of management tools. These tools help the library to preserve, manage and share digital assets.

One of the main issues in designing digital libraries is the association of the digital content with its metadata such that indexing, browsing, searching and retrieval can be done efficiently. The metadata attached to the objects is also used by the system to provide guided search to the user by displaying related objects e.g. objects that have

same creator or that fall under the same subject heading. This linkage between objects provides the user with a rich search experience and facilitates the exploration of the repository contents based on the user interests.

DAR is built for a library institution and therefore the collection that it should accommodate includes a wide variety of materials such as books, images, audio and video. Driven by the desire to describe digital objects that include books as well as images and multimedia, DAR adopted a data model able to describe objects in either MARC 21 [1] standard, which is designed for textual material or VRA core [2] which is becoming the most widely used format for describing images and multimedia. Further, the system is augmented with a web interface used by the catalogers that enables them to either retrieve metadata from current systems such as the Integrated Library System (ILS) or imaging systems or to enter manually new metadata that would be MARC or VRA compliant.

Another major objective of DAR is the automation of the digitization workflow and its integration with the repository. Most existing digital libraries systems assume material to be *born digital* or has been already digitized and produced into digital form. From the experience of BA as a partner in the Million Book Project [3], it was revealed that producing digital objects from their original form is not a trivial task and span much more than just scanning. A complete workflow for the digitization is necessary that should be tightly coupled with the repository system. It is desirable to automate the digitization workflow as much as possible, unifying the folder structure, file naming convention, keeping track of digital material and detecting digitization errors at early stages such that human intervention is minimized in the whole process in order to avoid human errors.

Another objective while implementing the system is the ease of integration with web-based repositories. Hence, the system is based on standards like the VRA Core Categories for describing digital objects, XML, MARC and DC for metadata presentation, OAI-PMH for content dissemination, and Web services for exposing the repository APIs.

In summary, the following goals were driving us while designing and implementing DAR:

- Integrating the actual content and metadata of varieties of objects types included in different library catalogs into one homogeneous repository.
- The automation of the digitization process such that human intervention is minimized and the outputs are integrated within the repository system.
- The preservation and archiving of digital media produced by the digital lab or acquired by the library in digital format.
- Enhancing the interoperability and seamless access to the library digital assets.

The rest of this paper is organized as follows. Section 2 presents some of the related work. Section 3 gives an overview of the system architecture. Sections 4 and 5 present the two main modules; the Digital Assets Keeper and the Digital Assets Factory, respectively. Section 6 presents the tools provided by the system. Section 7 concludes the paper and presents proposed directions for future work.

2. RELATED WORK

There is an increasing number of digital solutions motivated by the increase in the need of preserving and maintaining digital assets.

EPrints [4,5] is a digital repository for educational material developed at University of Southampton, UK. EPrints is an open source software and is intended for use by universities and research institutions. The system was created to enable the authors *self archiving* their work. A registered user can submit a document to the EPrint archive, the document is described using a super-set of the BibTeX fields. A submitted document is indexed for searching and positioned within a subject hierarchy defined in the system. EPrints makes the metadata available for harvesting by the OAI-PMH interface.

Dspace [6], developed jointly by MIT Libraries and Hewlett-Packard, is another repository system for handling educational material and addressing the long-term preservation of *digitally born* assets. DSpace depends mainly

on Dublin Core records to describe an item. The system supports submission, searching, browsing and retrieval. Similarly to EPrints, the system defines a workflow for the submission; the user submits an item with its metadata, a *reviewer* accepts or rejects the submission, an *approver* accepts or rejects the submission and can edit the metadata, finally an *editor* can edit the metadata but may not reject the submission, when done, the item is committed to the archive. Dspace implements the OAI-PMH protocol [7] for metadata harvesting.

Greenstone [8,9] is an open source software that provides out-of-the-box solution for the creation and publishing digital material. The system provides easy-to-use interface to define collections of digital objects, the metadata used to describe items within the collection and how items are displayed. Also, the user defines what types of searches are available. According to these configurations, new collections are created and indexes are built for browsing and searching. Once a collection is created and configured, documents can be added, either imported from external sources or manually selected by the user and added to the collection. Greenstone supports different document formats such as HTML, PDF, DJVU and Microsoft Word files. OpenDlib [10] proposes a similar system that aims at providing expandable and searchable system through customizable services.

Commercial library solutions and document management software are used by some libraries and institutions to manage their digital assets. However, most of these systems fail to address interoperability, extendibility and integration with other tools and services in the library due to their proprietary nature.

Contrary to other systems that only manage digital objects or are dedicated to educational material, DAR incorporates in one repository all types of material and formats belonging to the library collections, either born digital or digitized through the system. The DAR data model is capable of describing different metadata sets required by the heterogeneous nature of the collection while still complying with existing and evolving standards. Also, DAR integrates the digitization and OCR process with the digital repository and introduces as much automation as possible to minimize the human intervention in the process. As far as we know, this is an exclusive feature of DAR.

3. SYSTEM ARCHITECTURE

The architecture of DAR is depicted in Figure 1.The system core consists of two fundamental modules:

- The Digital Assets Factory (DAF) which is responsible for the automation of the digitization workflow, and
- The Digital Assets Keeper (DAK) which acts as a repository for digital assets either produced by the DAF or directly introduced into the repository.

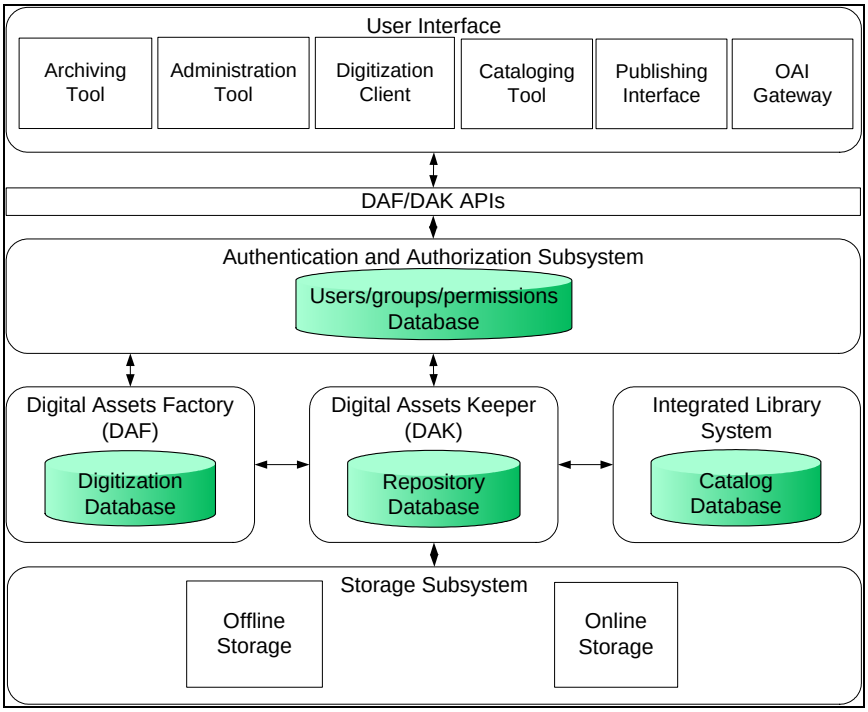

Figure 1. Architecture of DAR

Both systems interact with the digital objects storage system. The storage system is used to store digital files either for online access and publishing purposes, or offline for long-term preservation. The system contains a set of user interfaces that interact with the system components through APIs. The user interfaces provide tools for the automation of the digitization process, the system parameterization, metadata entry, searching and browsing the repository content, and tools for the interoperability with other repositories. An authentication and authorization system controls the access to the repository contents and functionalities based on the user identity. The repository is integrated with the Integrated Library System (ILS). Plug-in modules control the metadata exchange between the repository database and the ILS database. In the following sections, each of the system components will be described in details along with the functionalities it offers.

The system is implemented in C# using the Microsoft .Net technology. The web-based components are implemented as ASPX pages running on Microsoft IIS web server. The repository APIs are implemented as Web services. SQL sever database is used as the main repository database. The repository is integrated with the Virtua ILS [11] which uses Oracle database on UNIX platform.

4. DIGITAL ASSETS KEEPER – DAK

The DAK acts as a repository for digital material either produced by the digital lab or introduced directly in a digital format. All metadata related to a digital object is stored in the DAK repository database. Data describing the digital files are automatically extracted from the files, descriptive metadata is either manually assigned to the digital object by the librarians, or retrieved from the library catalog or any external source of metadata. Other metadata such as archiving and publishing metadata are fed from the DAF module.

4.1 Data Model

One of the challenges faced by DAR is to derive a data model capable of describing all types of library assets including books, maps, slides, posters, videos and sound recordings. For this purpose, two existing standard for data representation have been studied, namely MARC 21 [1] and VRA Core Categories [2].

While the MARC standard is widely used as a data interchange standard for bibliographic data, it is designed mainly for textual materials. Therefore, MARC is seen by the visual resources community as overly elaborate and complex in ways that provide no benefit to visual resources collections, while at the same time lacking or obscuring some concepts which are important to them. On the other hand, VRA core is designed specifically for works of art and architecture that the library is likely to include in its multimedia collections. The VRA Core Categories capture sufficient information to support a wide variety of sophisticated queries. One of the imaging systems based on the VRA is the Luna Insight [12] which is a commercial imaging software that is widely used by many libraries, universities and museums as a repository for visual assets. The advantages and attraction of the VRA are discussed in [13]. The data model used by DAR is inspired by the one proposed by the VRA. However, the VRA categories have been extended to accommodate for bibliographic data supported by the MARC standard. This resulted in a data model capable of describing both visual and textual materials in one homogeneous model that is, at the same time, compliant with both standards. The data model is depicted in Figure 2.

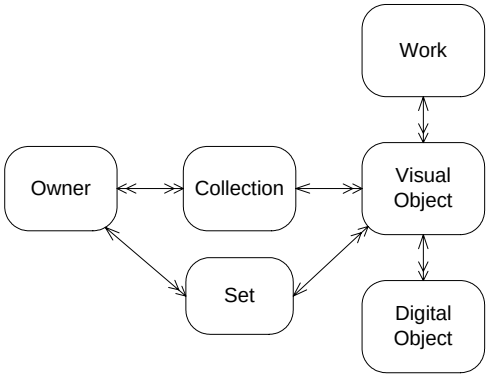

Figure 2: DAR Basic Data Model

DAR represents a digital object by a *Work* entity related to one or more *Visual Object* entities. This is inspired by the VRA *Work* and *Image* entities. The *Image* is called *a Visual Object* in DAR since the system has extended the model to accommodate for all types of assets including books, images, audio and video materials.

The *Work* refers to a physical entity; it might be a performance, composition, literary work, sculpture, event, or even a building, while the *Visual Object* refers to a visual representation of a *Work*. It can exist in photomechanical, photographic and digital formats. In a typical visual resources collection, a *Visual Object* is a reproduction of the *Work* that is owned by the cataloging institution and is typically a slide, photograph, or digital file. A *Visual Object* exists in one or more digital forms denoted as *Digital Objects*. A *Digital Object* might be a JPG file presenting a scanned slide, an Image-On-Text PDF for an OCR-ed book or an audio or video file.

A Visual Object has one *Owner*. The *Owner* is typically an institution, a department or a person. *Visual Objects* related to an *Owner* are grouped into *Sets*. The *Set* represents a physical grouping of *Visual Objects*; this grouping is established at the digitization phase. On the other hand, the *Collection* represents a descriptive grouping of *Visual Objects* based on a common criteria. Table 1 shows examples of values for each of the described objects.

Table 1. Example of digital objects

Object	Value
Collection	Million Book Project, OACIS Collection
Set	Box of 100 slides donated to the library
Owner	Bibliotheca Alexandrina, Yale University
Work	The building of Bibliotheca Alexandrina, The new year concert
Visual Object	A slide for a panoramic view of the library building, a video taken in the new year concert
Digital Object	A JPG file produced by scanning a slide , a PDF file produced by scanning and OCR-ing a book

4.2 Metadata

Within the DAR data model, the system holds six categories of metadata describing assets and its digital reproductions:

4.2.1 Descriptive Metadata

This includes metadata common to all types of Works and Visual Objects, examples of these are:

- Type, for a *Work* object, the type could be a painting, map, statue, coin, photograph, event or building. For a *Visual Object*, the type could be a slide, photograph, image, video, audio or book
- Title
- Creator(s), a creator could be the author, publisher, architect or the work engraver.
- Date(s), a date could represent the date of the creation, design, beginning, completion, alteration or restoration
- Keywords
- Description
- Dimensions, the dimensions of an object can be expressed in terms of its height, width, area, scale or other depending on the type of object being described
- Location, this includes both a physical location where the object is stored or displayed, and the geographic location of the object

Other metadata that is specific to a *Work* type include fields like the ISBN, language and publisher in the case of books, the technique and material in the case of a work of art. For textual materials like books, the metadata describing the material is entered in the text language, unless a translation is available. For other types of materials, the data is entered in both Arabic and English to extend the search capabilities on the material in both languages.

4.2.2 Digital Content Metadata

This includes metadata describing a Digital Object. DAR supports a variety of digital objects' formats including JPG, TIFF, JPG 2000, PDF, DJVU, OCR Text and others. Metadata such as image resolution, dimensions, profile, or a video duration are extracted from digital files automatically and stored in DAK. It should be noted that the design is flexible in a way that it allows new formats to be introduced into the system and appropriate tools to be integrated to deal with the new file formats.

4.2.3 Archiving Metadata

This includes metadata about the archiving location of a *Digital Object* file. The object is archived on offline storage media such as CDs or tapes. The archiving metadata consists of the media unique identifier and keeps track of file versions by linking the newer version archiving location to the older one. The archiving metadata can also be attached to the *Visual Object*, denoting the physical location where the object can be found in the owning institution.

4.2.4 Publishing Metadata

Encoded objects for publishing are stored on online storage. The publishing metadata includes the path of the published *Digital Object* on the server, the date of publishing, duration of publishing as well as the category of targeted users e.g. students, researchers, etc. The file path is associated with each *Digital Object* whereas other fields are stored on the *Collection* level.

4.2.5 Access Right Metadata

Copyright restrictions on the repository contents are forced by defining access rights attached to each object. This consists of a copyright statement linked to the *Visual Object* and displayed with the object. Also, an access right

level is used by the system to indicate whether a *Visual Object* and its related *Digital Objects* are free of copyright restrictions or not. This level is used by the publishing interface to determine how objects are displayed. The current implementation supports four levels of access rights as follows:

- *Level 1* – viewing metadata only: this level corresponds to an object with copyright restrictions where the actual content cannot be displayed
- *Level 2* – viewing metadata and thumbnail: this level could be used for objects representing images which have copyright on the high resolution version of images.
- *Level 3* – viewing metadata and excerpt of the con-tents: this is typically attached to composite material, for example a book under copyright where only the abstract or some selected pages of the digitized book can be displayed, or a video where a snapshot of a certain duration can be viewed.
- *Level 4* – viewing metadata and full access: this level is attached to objects that are completely free of copyright and full content can be displayed

More levels of restrictions may be added in the future to accommodate other types of material.

4.2.6 Authentication and Authorization Metadata

DAR users are identified by a username and a password. Further, user groups are defined where a user can belong to one or more group. Permissions are given to each user or group, which are checked before accessing an application and/or digital object. User and group rights can be specified on the *Visual Object* level or, more practically, on the *Collection* level.

Table 2 shows a typical set of metadata used to describe a photo of the Statue of Liberty. It should be noted that the database design of DAR is flexible in a way that it allows generic user-defined fields to be added by the institution deploying DAR according to its specific needs

Table 2. Typical Set of Metadata Describing a Photo of the Statue of Liberty

Work	Title	Statue of Liberty
	Type	Sculpture
	Subject	Statue of Liberty National Monument (New York, N.Y.) - National monuments
	Description	It's been over a hundred years since the Statue of Liberty found her home in the harbor of New York and it has become an important part of American culture…
	Keywords	Statue - Sculpture - Harbor
	Date/Type	1884/ Creation
	Material	Copper
	Dimension Type/Value	Height/46 m
	Geographic Location	New York, USA
	Creator Name/ Nationality	Auguste Bartholdi/ French
Visual Object	Title	Statue of Liberty at dawn
	Type	Photograph
	Subject	*see work*
	Date/Type	1984/ Creation
	Creator	Abell, Sam
	Creator Nationality	American
	Description	This picture was taken in the 100 annual anniversary of the statue of the Statue of

		Liberty		
	Technique	Digital Photographing		
	Keywords	celebration - fireworks		
	Arabic Keywords	احتفال - ألعاب نارية		
	Dimension Type/Value	Width/ 1652 Pixels		
	Dimension Type/Value	Height/ 2342 Pixels		
	Style	Color - Horizontal		
	Copyright Level	4		
	Copyright Notice	(c) 2004 Bibliotheca Alexandrina		
	Source	Photography Unit		
	Physical Location	Photography Unit Archive at BA		
	Geographical Location	EGYPT		
	Audience	Public		
Digital Objects	Format	JPEG	JPEG	TIFF
	Width	520	174	520
	Height	785	262	785
	Resolution	720	200	1800
	Path	http://sedcw2a-h2001/Medium/0025.jpg	http://sedcw2a-h2001/Low/0025.jpg	-
	Color	Y	Y	Y
	Profile	NTSC1953.icc	NTSC1953.icc	NTSC1953.icc
	Size	1.0 MB	0.5 MB	7.5 MB
	Archiving Type/Barcode	-	-	CD/ B000055

DAK APIs

The DAK interacts with other system components through a set of APIs. Examples of the DAK APIs include:

- Insert new *Work*, *Visual Object* and *Digital Object*
- Import and export an object metadata in MARC and DC in XML format.
- Extract metadata from files' headers
- Add and edit group permissions
- List all digital object versions

5. DIGITAL ASSETS FACTORY – DAF

The DAF governs the digitization process of the library collection at the digital lab. DAF realizes one of the main goals of DAR which is the automation of the digitization process. This supports the digitization of library assets including textual material, slides, maps and others. It provides the digital lab operators with tools for entering a digitization job metadata, keeping track of digitization status, applying validation tests on digitized material, recording productions, archiving the digitized material for long term preservation and retrieving the archived material when needed. The system supports different workflows for different types of material.

After initiating a new job, the asset passes by the general phases depicted in Figure 3:

- Scanning the material.
- Processing the scanned files to enhance the quality.
- Perform Optical Character Recognition (OCR) on the textual material.
- Encoding the digitized material by generating a version suitable for publishing. The current version publishes the textual material in DJVU and PDF formats, different JPG resolutions for images, slides and maps and different qualities for audio and video.
- Archiving the output of each step of the digitization. Two offline backups are taken for a file, one on CD and the other on tape. Encoded versions are moved to on online storage for publishing.

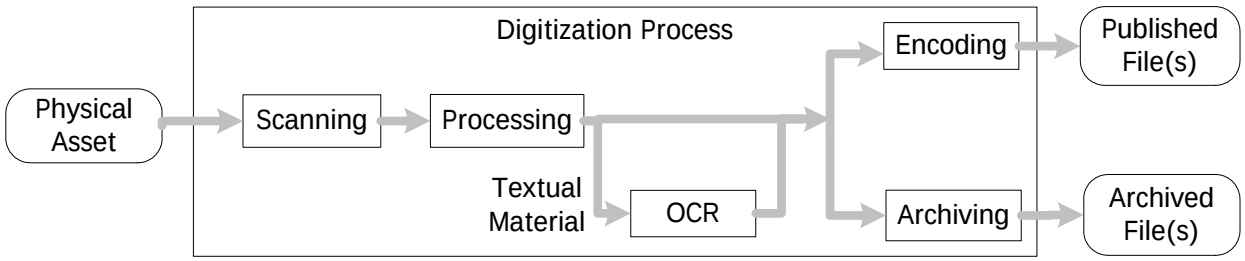

Figure 3. Digitization Phases

The files and folders produced by each phase are stored in separate queues on a central storage server. A job folder resides in one of the four main queues: *scanned*, *processed*, *OCRed* and *ready for archiving* queue. The digital lab operator withdraws jobs from the queues, performs the job and places the output in the next queue in the process. Alternatively, files can be introduced directly into any of the queues, for example an image that is already provided in digital form is placed directly into the *processing queue*.

Table 3 shows a one year digitization statistics at the BA digital lab since the system deployment in March 2004.

Table 3. Digital Lab Production Statistics

	Arabic	**Latin**
Scanned pages	4,591,463	730,141
Processed pages	4,585,833	730,141
OCRed pages	1,148,465	693,978
Scanned Slides	12,013	
Archived Data on CDs/Tapes	480 GB	

The main goals of DAF are:

- To provide a database system to keep track of the digitization process through the scanning, processing, OCR-ing, archiving and publishing.
- To keep track of digitized materials; unifying the naming conventions and exhaustively checking the produced folders and files for consistency.
- To provide timely reports to various levels of management describing the workflow on a daily, weekly or longer basis and to allow online queries about the current status of a certain asset at the digital lab.
- To apply necessary encodings on the scanned materials to be suitable for electronic publishing.
- To manage the archiving and retrieval of the digitized material.

Digitization Metadata

For objects that are digitized using the DAF applications, the digitization metadata is gathered during the different digitization stages, examples of these are:

- Scanning date(s)

- Scanning operator(s)
- Processing date(s)
- Processing operator(s)
- OCR font data
- Accuracy achieved by the OCR before and after learning

DAF APIs

Similarly to the DAK, The DAF interacts with other system components through a set of APIs. Examples of the DAF APIs include:

- Initiate a new job
- Assign a job to a user
- Record the completion of a job
- Get the job digitization status
- Bind a digitized material to an archiving media

6. TOOLS

The DAR system deals with three types of users; digitization operators, librarians - which are divided into catalogers and reviewers - and the end users. Each type of users is provided with tools to make use of the system functionalities. We present these tools in the following sub-sections.

6.1 Administration Tool

The *Administration Tool* is one of the DAF Web-based tools used by the operator in the digital lab. The tool is used to initiate a new job by entering minimal descriptive metadata for the material to be digitized. If the material is cataloged in the library catalog, the ILS id - a book barcode, for example - is used to retrieve the metadata from the library catalog. This id is also used to link the record in DAR to the one in the library catalog for future synchronization. If the material is not previously cataloged, the operator enters the minimal metadata that can be deduced from the physical item in hand, namely the title and the author name in case of books, the set name and the owner name in case of other types of materials. The tool uses this metadata to derive a unique folder name for the scanned files. In case of Arabic materials, a transliteration module uses a mapping table to derive a Latin folder name from the Arabic title and author name. In general, the main objective is to generate a folder name that is human readable to facilitate the digitization process, and more important, to guarantee the uniqueness of this name.

The tool is also used for the system parameterization and to generate reports on production rates and jobs in different digitization queues in the lab as shown in Figure 4.

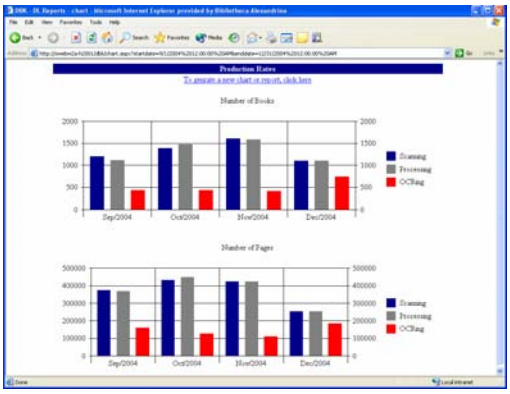

Figure 4. Administration Tool - Reporting Page

6.2 Digitization Client – DLClient

The *DLClient* is a DAF Windows-based application used by the operator in the digital lab. The tool creates structured folders for new digitization jobs, handles the movement of folders and files between different queues in the lab, checks folders and files consistency and updates the job status in the database.

After the completion of each digitization phase, the *DLClient* tool is used to perform the following:

- Validate the files; generate errors and/or warnings if any inconsistencies are detected.
- Update the job status in the database by setting the username for the operator who performed the job, the job completion date and the number of produced files.
- Move the folders and files to the queue of the next digitization phase on a storage server. To preserve files consistency, a secure file movement procedure is applied; a file is copied from the source to destination, then the size of the destination file is compared against the source, finally the source is deleted. Before moving any folder, a lock is acquired on the folder and sub files to avoid concurrent access to the folder while moving.

The *DLClient* is used by the operator through the three main digitization phases; scanning, processing and OCR.

Scanning

Physical assets submitted to the lab for digitization are placed in a *scanning queue*. The operator retrieves a job from the queue and uses the *DLClient* to create the folder structure where scanned files are to be stored. Mainly, a digitization folder contains three subfolders for three types of files: the original scanned files, the processed files and the encoded output. The encoded output, the folder structure and the scanning resolution varies according to the material type; text, image, audio or video. Table 4 shows the different folder structures, file formats and scanning resolutions used by the current digitization workflow for some types of scanned materials. When the scanning is done, the *DLClient* places the produced files in the *processing queue*.

Table 4. Digitization Settings

	B\W Books	Colored Books	Colored Slides
Scanning Resolution	300	300	1800
Scanning Output	Tiff CCITT-4 Compression	JP2	JP2
Processing Output	Tiff CCITT-4 Compression	JPG	JPG
Encoded Output	Searchable PDF and DJVU	Searchable PDF and DJVU	Three JPG resolutions
Folder Structure	Book Name\ OTIFF\ PTIFF\ TXT\	Book Name\ OJP2\ PJPG\ TXT\	Set Name\ OJP2\ PJPG\ JPG\

Processing

The operators use the *DLClient* to retrieve a job from the *processing queue*. A combination of manual and automated image processing tools is used to enhance the quality of the scanned images. The process mainly involves the removal of noise, the reduction of the size of the file, removing any extra white spaces and margins, and curvature correction. For images, the processing phase includes generation of different image resolutions suitable for web publishing. After the job is done, the *DLClient* places the job at the *OCR queue* for textual material and directly to the *archiving queue* for other types of material.

Optical Character Recognition

Using the *DLClient*, a processed textual material is retrieved from the *processing queue* to be OCR-ed extracting text from the scanned images. OCR is used to enable full text searching, not to create textual equivalents of the original. Currently, the system supports Latin OCR and Arabic OCR.

- Latin OCR

The system uses Fine Reader 6.0 from ABBYY [14] to perform optical character recognition on Latin books. The processed images of the book are fed in batch to the OCR engine and, upon completion of the recognition work; the native format generated by the Fine Reader format is stored as '.frf'.

- Arabic OCR

Given the peculiarities of Arabic fonts and the characteristics of the Arabic language, recognizing Arabic characters is a difficult job. The Arabic OCR runs in two stages; first the text is enhanced to give better OCR quality. This includes some processing procedure on the text like closing some letters, sticking letters or sticking the dots, eroding the image to remove excess thickness, smoothing the image in case of toothed text, etc. The above procedures are done using ScanFix software [15].

After the text enhancement, Sakhr Automatic Reader [16] software is used for the recognition stage. To enhance the recognition quality of Arabic text, BA has built a library of fonts using learning samples taken from different books. Before starting the recognition, the OCR operator matches the book font with the nearest font library. This font is used as a starting font and the operator modifies it by applying a learning phase for the OCR engine using two pages of the book. Based on the starting font and the learning, a new font is produced by the engine that is customized on the text being recognized. The new created font is used to recognize the whole text and is saved within the book folder.

Reprocessing

The system supports a special workflow for reprocessing a digitized material. Reprocessing may be needed to enhance the OCR quality, to apply new image processing procedure or simply to generate new publishing format of the digitized material. Reprocessing begins by searching and retrieving the files to be reprocessed from the archive. The files are then placed in the appropriate digitization queue according to the type of reprocessing required. The reprocessed files go through the normal digitization steps described before until they reach the archiving phase. Only altered files are re-archived, changes in files are detected using checksums that are calculated before and after the reprocessing. The archiving information of a new file version is recorded in the repository database and a link is made to the parent version archiving location so that file versions may be tracked in the database from the most recent to the base version.

6.3 Archiving Tool

In the current version, a digital object is represented by one or more files with different formats and/or resolutions, these files are stored for online access on RAID storage system or on offline storage for long-term preservation. Typically, the preserved material is the scanned originals and the processed version with high resolution. Lower versions derived for publishing purposes are saved on online storage for ease of access; this includes low resolution JPG, PDF, and DJVU. Files stored offline are archived on two medias; CDs and tapes. Unique labels are generated, printed and attached to the media for future retrieval. The system keeps track of different versions of a file by linking a newer version to its older one. More sophisticated content versioning and object representation is to be applied in future versions of DAR, this could build on the architecture proposed by the Fedora repository [17].

The *Archiving Tool* is one of the DAF Windows-based applications used by the lab operators and offers the following functionalities to support the archiving and retrieval of data:

- Checking files and folders consistency.
- Preparing the folders for archiving by compressing the subfolders and files, grouping them into bundles that fit into the media capacity (CD or tape), generating the media label, printing the label and relating the archived folders to the media in the database.
- Generates XML file for an archived folder, this file contains all metadata related to the corresponding record in the database in XML format. The XML file is archived with the folder.
- The tool generates checksums for the archived files to detect changes in case of downloading and reprocessing a file.

- A search facility enables the user to retrieve an archived folder by locating the folder, uncompressing the subfolders and files and copying the uncompressed files and folders to a destination specified by the user.
- Managing the space on the storage server hard drives, the tool generates warnings when storage level exceeds a predefined value for each drive.
- The tool updates the DAK database by recording the archiving information related to a digital file.

6.4 Encoding Tool

In the encoding step, a final product is generated for publishing. For images, slides and maps, different JPG resolutions are generated. Mainly, three resolutions are generated representing the image thumbnail, medium size and large size. For audio and video, different qualities are generated to accommodate for different network connections' speed.

For textual material like books, special developed tools are used to generate the image-on-text equivalent of the text; this is done on an encoding server built on Linux platform. The Encoding Server encodes digital books into light-weight image-on-text documents in DjVu and PDF. Support for DjVu is built around DjVu Libre, an open source implementation of a DjVu environment, or, alternatively, Document Express, LizardTech's commercial DjVu product. Support for PDF is implemented based on iText, an open source API for composing and manipulating PDF documents. The Encoding Server supports multilingual content through integration with Sakhr Automatic Reader [16], an OCR suite that works with Arabic, Persian, and 18 Latin languages. Further, the Encoding Server allows for the integration of any OCR engine through writing OCR converters, which transforms the native OCR format into a common OCR format that the Encoding Server is capable of processing along with page images in TIFF or JFIF format to compose image-on-text documents. Two PDF files are generated for a book; a high resolution version (150 dpi) for internal publishing within BA Intranet and a low resolution version (72 dpi) for Internet publishing.

A generated file is copied to a publishing server, the encoding tool updates the DAK database by inserting the corresponding *Digital Object* record. The record is populated with metadata extracted from the digital files and with the publishing information; publishing server and URL.

6.5 Cataloging Tool

The *Cataloging Tool* is a Web-based application used by the librarian to add and edit metadata in the DAK subsystem. Using the Cataloging Tool, the librarian enriches the digital repository records – created in the digitization phase - by adding metadata. The librarian can also create new records for digital objects and upload their corresponding files. The repository is preloaded with controlled vocabularies lists, for example; the Library of Congress Subject Headings list for Latin subject headings and a local Arabic subject headings list *QRMAK* for Arabic subjects.

The following functions are offered by the tool:

- A set of configurable templates tailored to specific types of material such as books, maps and slides.
- Importing metadata from external sources such as the library catalog. This is achieved through integration modules that will be discussed in more details in section 6.7.
- Automatic extraction of digital content metadata. For example, extracting the height, width and resolution from a JPG file.
- Batch uploading of files. Through this utility, the cataloger can upload a batch of files, and define minimal metadata to be linked to the uploaded files. The system automatically creates *Visual Objects* and *Digital Objects* records and populates them with the provided metadata.
- Reviewing tools; a reviewer is provided with tools to view the recently catalogued records and the updates in any controlled list of values.

6.6 Publishing Interface

The *Publishing Interface* is a Web-based interface related to the DAK that provides access to the repository of digital objects through search and browsing facilities. Through one homogenous interface, the user is offered a very rich search and viewing experience on different types of materials.

The repository Publishing Interface offers the following functions:

- Browse the repository contents by *Collection*, *Work* Type, *Visual Object* Type, Subject, Creator and Title.
- Search the content by an indexed metadata field; Creator, Title, Subject, Keywords,…
- For textual material, a search in the full text can be conducted. The user can choose whether exact or morphological matching is applied. The search can be applied to a group of books for example, or within the book text.
- For images, different levels of zooming are available beside the thumbnail view for a set of images.
- Display brief record information.
- Display full record information with links to the digital objects.
- Display the records in MARC, DC or XML formats.
- Hyperlinked data fields that can invoke searches e.g. by keywords, subjects and creator.

The advanced search page is depicted in Figure 5.

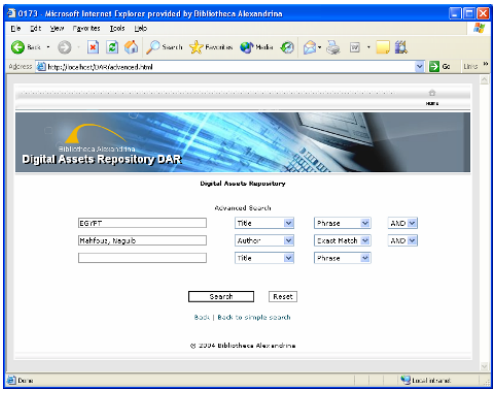

Figure 5: DAR Advanced Search

The *Publishing Interface* is based on ASPX technology. All interactions with the repository are implemented as Web services. This makes the changes and improvements in the interface separated from the logic coded in the Web Services.

Parameters are passed between pages as request variables. Page content is displayed based on the passed parameters. For example, a page displaying the full object record takes as parameters the object Id. The page then passes the Id to a web service that returns the full object record. This achieves modularity and reusability of pages within the interface and facilitates the formation of hyperlinked data fields that invokes new pages.

6.7 Integration with the ILS

DAR can be easily integrated and synchronized with external sources – e.g. bibliographic catalog, external repository, imaging systems - by implementing appropriate integration modules. An integration module is a plug-in component designed to export records from DAR to an external repository, or to import records from an external repository into DAR, or both. This is invoked by a cataloger when adding new objects to the repository, or periodically to synchronize the repository with external sources of metadata. The need for this synchronization

is rising from the fact that a library is likely to have its main catalog and/or multimedia system before deploying DAR. This module eliminates the need for duplication of data entry.

The integration module is fully configured based on the following:

1. A record unique identification: This identifier is used as a link between the record in DAR and the one in the external repository.
2. Metadata mapping table: The mapping table defines how data fields are mapped from DAR to the external repository and vice versa.
3. Data model mapping scheme: This mapping scheme defines how the concepts of *Work* and *Visual Object* are mapped to the external repository and alternatively how *Work* and *Visual Object* records are extracted from the external schema.
4. Synchronization schedule: This schedule defines how often the two repositories are synchronized. The synchronization process considers only the newly created and modified records.

In the current version, a module is implemented for integration with the Virtua ILS [11] which is deployed at BA. The bibliographic records in the library catalog are cataloged using MARC 21. Records are encoded in ISO2709 format and stored in the database as Blobs. Each record has a unique ILS identifier, a creation date and modifications dates.

The integration module between DAR and the library catalog is configured as follows:

1. The record unique ILS id in the library catalog is used to link synchronized records. This id is provided either by the digitization operator or by the cataloger.
2. A mapping table is used to define how MARC tags and subfields are mapped to DAR data fields and vice versa. The mapping draws on the one proposed by the VRA [2].
3. A *Work* and its related *Visual Object* are mapped into two MARC records. The relationship between the two records is expressed using MARC linking tags. The object type is encoded in the fixed-length tags defining the record general information.
4. A daily scheduled process synchronizes the DAR database with the library catalog based on the record ILS id. A special MARC parsing module is implemented to extract data from the ISO2709 format.

6.8 Authentication and Authorization

The Authentication and Authorization subsystem provides basic authentication and enforces an access control policy based on user identity. In DAR, a *User* is a member of one or more *Group*. Each *Group* is assigned access permissions on the repository contents and functionalities. A basic username and password scheme is used to identify the user. Anonymous access to the repository is also allowed, the access right of an anonymous user are defined by the permissions assigned to the group *Guest*. The following grouping of users is used by the current version to define a user role and permissions.

- Digital lab operators: The members of these groups are authorized to use all DAF applications. After a basic authentication, the DAF applications are impersonated with a special user that has access to the digitization queues, can move files and folders across different queues and perform read and update operations
- Catalogers: The members of this group are authorized to enter and edit metadata through the cataloging tool. Some actions are forbidden for this group like editing a list of controlled values. Also, a cataloger cannot alter a *Digital Object*.
- Reviewers: The members of this group are authorized to use all functionalities provided by the cataloging tool.
- Guest: Any anonymous access to the repository is given the permissions assigned to this group. This group has access only to the publishing interface.

Permissions assigned to each of these groups can take one of the following values:

- **Read**: viewing an object metadata. This is typically assigned to all groups.
- **Add**: adding new objects. This permission is assigned to the catalogers, reviewers and the lab operators.
- **Edit**: Modifying an object metadata. This permission is assigned to the catalogers and reviewers.
- **Delete**: deletion of an object from the repository. This permission is assigned to the reviewers.

This simple authorization scheme will be augmented in future versions to accommodate for special groups' permissions, for example, new groups may be needed to represent special communities representing another library, organization or web site that have subscriptions to access the repository contents. The permissions of such groups can be determined based on a predefined agreement.

6.9 OAI Gateway

The Open Archive Initiative [7] has developed the OAI-PMH protocol for metadata harvesting. This allows sites and software systems to retrieve metadata from several repositories to provide access to information from large number of repositories that are collated in a central catalog. DAR OAI Gateway provides access to the repository contents across such organization's architecture.

The Gateway receives XML requests and translates them to the equivalent database queries. When the request result sets are retrieved, the gateway translates them into XML and responds to the requesting application.

The gateway implements the six types of requests required for OAI-PMH compliance:

- Identify
- ListMetadataFormats
- ListSets
- ListIdentifiers
- ListRecords
- GetRecord

7. CONCLUSIONS AND FUTURE WORK

We have presented in this paper the DAR system implemented at the Bibliotheca Alexandrina. The system acts as a repository for digital assets owned by the library and associates the metadata with the content to provide efficient search and retrieval. DAR supports different digital formats and incorporates in one integrated system the digitization, OCR, preservation and dissemination of material. DAR provides a flexible platform for any library or institution to build its own digital assets repository and integrate it with its ILS or other sources of metadata. The system addresses the main challenges faced by digital repositories; digitization workflows, preservation of digital material and content dissemination.

The presented DAF subsystem has been fully implemented and deployed since March 2004. The DAK subsystem is in its beta version with the first version deployment planned on May 2005. This version will include features presented in this paper.

Future enhancements include:

- Building a more sophisticated security system based as on existing and emerging standards that are appropriate for the web services environment.
- Designing and implementing a more sophisticated and efficient content versioning support.
- Implementing a generic digital assets viewer. The viewer should support different file formats (PDF, DJVU, Images, Video and Audio).

- Joining the Open Source community by making the system source code publicly available and using free-of-charge development tools and database engine.
- Providing query translation tools to enable cross-language information retrieval.
- Using XML format for encoding objects metadata. This will facilitate exchange of objects among repositories. However performance issues must be carefully considered. A number of native XML databases and Object Oriented databases are being evaluated for this purpose.

8. ACKNOWLEDGMENTS

We would like to thank Shady Elbassuoni for his contribution in the design and implementation of the DAF. Also, we would like to thank Mohamed Ramadan, Youssef Eldakar and Khalid Elgazzar for their valuable input related to the digitization process.

9. REFERENCES

[1] MARC 21 Standard. http://www.loc.gov/marc/

[2] VRA Core Categories, Version 3.0. http://www.vraweb.org/vracore3.htm

[3] R. Reddy and G. StClair: The Million Book Digital Library Project. Available at http://www.rr.cs.cmu.edu/mbdl.htm, December 2001.

[4] GNU EPrints. http://software.eprints.org/

[5] L. Carr, G. Wills, G. Power, C. Bailey, W. Hall and S. Grange: Extending the Role of the Digital Library: Computer Support for Creating Articles, in *Proceedings of Hypertext 2004* (Santa Cruz, California, August 2004).

[6] R. Tansley, M. Bass, D. Stuve, M. Branschofsky, D. Chudnov, G. McClellan and M. Smith: The DSpace Institutional Digital Repository System: Current Functionality, in *Proceedings of JCDL '03* (Houston, Texas, May 2003).

[7] The Open Archives Initiatives. http://www.openarchives.org/

[8] D. Bainbridge, J. Thompson and I. H. Witten: Assembling and Enriching Digital Library Collections, in *Proceedings of JCDL '03* (Houston, Texas, May 2003).

[9] I. H. Witten, S. J. Boddie, D. Bainbridge and R. J. McNab: Greenstone: a comprehensive open-source digital library software system, in *Proceedings of the fifth ACM conference on Digital libraries* (June 2000).

[10] D. Castelli and P. Pagano: A System for Building Expandable Digital Libraries, in *Proceedings of JCDL '03* (Houston, Texas, May 2003).

[11] Virtua Integrated Library System. http://www.vtls.com/

[12] Luna Imaging Software. http://www.luna-imaging.com/

[13] P. Caplan: International Metadata Initiatives: Lessons in Bibliographic Control. Available at http://www.loc.gov/catdir/bibcontrol/caplan_paper.html, January 2001.

[14] ABBYY Fine Reader OCR software. http://www.abbyy.com/

[15] ScanFix Image Processing Software. http://www.capitol-image.com/scanfix.htm

[16] Sakhr Automatic Reader OCR software. http://www.sakhr.com/

[17] S. Payette and C. Lagoze: Flexible and Extensible Digital Object and Repository Architecture, in *Proceedings of ECDL'98* (Greece, September, 1998).