Project no.507618

DELOS

A Network of Excellence on Digital Libraries

Instrument: Network of Excellence

Thematic Priority: IST-2002-2.3.1.12
Technology-enhanced Learning and Access to Cultural Heritage

Deliverable D1.4.1
Current Digital Library Systems: User Requirements vs Provided Functionality

Due date of deliverable: 31st December 2005
Actual submission date: 16th March 2006

Start Date of Project: 01 January 2004
Duration: 48 Months

Organisation Name of Lead Contractor for this Deliverable: ISTI - CNR

Final-V2

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)

Dissemination Level: [PU (Public)]

Current Digital Library Systems:

User Requirements vs Provided Functionality

Authors

L. Candela [1], D. Castelli [1], N. Fuhr [2], Y. Ioannidis [3], C.-P. Klas [2], P. Pagano [1], S. Ross [4], C. Saidis [3], H.-J. Schek [5], H. Schuldt [6], M. Springmann [5]

[1] Institute of Information Sciente and Technologies (ISTI)
Italian Narional Research Council (CNR)
Pisa, Italy

[2] Information Systems, Department of Informatics
Universität Duisburg-Essen
Duisburg, Germany

[3] Department of Informatics and Telecommunications
University of Athens
Athens, Greece

[4] Humanities Advanced Technology and Information Institute (HATII)
University of Glasgow
Glasgow, United Kingdom

[5] Information & Software Engineering
University for Health Sciences, Medical Informatics and Technology (UMIT)
Austria

[6] Database and Information Systems Group
University of Basel,
Switzerland

Table of Contents

Executive Summary

DELOS recognizes the need for a Reference Model for Digital Library Management Systems providing a formal and conceptual framework describing the characteristics of this class of information system. A Task-group under the Digital Library Architecture Cluster has begun constructing such a reference architecture for digital libraries. As part of the process of constructing this architecture the five core members analysed a suite of existing digital library systems in order to highlight commonalities. As we are very well aware that existing systems only marginally satisfy the needs of the potential digital library application areas, as a second step we considered the requirements collected in the framework of on-going projects that aim at building the digital libraries of the future. These two activities enabled us to sketch a broad overview of the concepts of digital library systems that we will use to refine our working model. Readers will find this report useful as it indicates how existing implementations have influenced our thinking.

1 Introduction

This review examines a selection of the current generation of software that implements the Digital Library (DL) functionality. By looking at digital library functionality we aim to inform research to design a reference model for digital libraries. In the first instance the process of developing a digital library reference model will increase debate about what constitutes the fundamental concepts, building blocks, and processes underlying digital libraries is essential. Establishing a benchmark for digital libraries which will arise from the adoption on a large scale of the reference model will have an analogous impact in the digital library arena that widespread adoption of the proposals such as those underlying the world-wide web have had. Any casual glance at the information systems which claim to be digital libraries suggest that we are operating at the information landscape of equivalent of the phylum rather than the species level; a reference model will help us to resolve this problem.

As Ross explained in a study of the context of the development of National Library of New Zealand [38],

> A digital library is the infrastructure, policies and procedures, and organisational, political and economic mechanisms necessary to enable access to and preservation of digital content. … There are numerous digital library experiments both within the academic sphere and within national, regional, and university libraries.[1] Some are services provided through many libraries, others subscription services[2], and still others are the digital resource face of traditional libraries. They all vary in character and type of content, with some being homogeneous collections on particular topics or media to others being heterogeneous entities. What is lacking though is general agreement as to what a digital library is.

Construction and take-up of digital libraries is hampered by the lack mechanism for measuring the technical qualities of different Digital Library implementations. Task 1.4 of the DELOS Network of Excellence in Digital Libraries aims to address this problem by establishing a digital library reference model. The construction of any reference model needs to engage with the design of current systems and appreciate the expectations of users.

DELOS is an integrated activity on focused digital libraries. The development of a reference model benefits from our ability to build on the work of the colleagues working on digital library research challenges from digital library architectures to evaluation. In conducting this background study we took the advice of our colleagues in other clusters to ensure that we built most effectively on work that was already completed. This was especially the case in the area of the expectations of users—we looked to the work of our colleagues in the User Interfaces and Visualisation (WP4) for guidance on user aspirations. In discussion with colleagues within DELOS we also considered how we might review the DL landscape in an effective way that would provide us with a characterisation of current models and approaches.

---

[1] Academic led projects include: The Open Video Digital Library (OVDL) http://www.open-video.org [30], the Alexandria Digital Library Project, http://www.alexandria.ucsb.edu/; Variation2: Indiana University Digital Music Library Project, http://dml.indiana.edu/. International projects include the Networked European Deposit Library (NEDLIB) project, http://www.konbib.nl/nedlib/. Among the national initiatives see the National Library of Canada Electronic Collection, http://collection.nlc-bnc.ca/e-coll-e/index-e.htm which already digitally holds more 'than 9,894 titles and more than 40,000 serial issues published by both the commercial publishing sector and the government publishing sector'.

[2] For example, IEEE Computer Society Digital Library, http://www.computer.org/publications/dlib/

As we explain in section 2 our theoretical analysis has led us to focus on five core concepts in thinking about DLs: architecture, information space, functionality, users, and quality of service. Within the typology of digital libraries we have identified four core classes of digital libraries: repository models, bespoke digital libraries, digital library systems, and emerging digital libraries systems—especially service oriented, peer-to-peer, and grid oriented architectures.

There are numerous digital library implementations that have been excluded. Some have been excluded because they are constructed by augmenting existing digital library systems and SCHOLNET (A Digital Library Testbed to support Networked Scholarly Communities) [39] is a good example this sub-class. While not including SCHOLNET in our discussions we have used the evidence that it provides about the needs of users including their need for support for hypermedia annotation, cross-language search and retrieval, and personalised distribution in developing our understanding of user requirements and categorisation templates. Gateway and portal services, such as those typified by RENARDUS [37] which focuses on providing a 'trusted source of selected, high quality Internet resources for those teaching, learning and researching in higher education in Europe', have been excluded from the discussion here as well.

Experimental implementations that are designed to demonstrate how users might work in the digital library environment have also helped us to validate the requirements of users and have enabled us to anticipate the ways that digital libraries may be used in the future. These have been reviewed elsewhere. The Fifth Framework funded project Collaboratory for Annotation, Indexing and Retrieval of Digitised Historical Archive Material, more commonly known as COLLATE [10] focused on creating a 'content-centric, user-driven information system for the management of surrogates of fragile historic multimedia objects. As a distributed Web-based multimedia repository, it function as a "collaboratory" supporting distributed user groups by dedicated knowledge management facilities like content-based access, comparison and in-depth indexing/annotation of digitised sources' [47]. Collate provides a platform to support collaborative scholarship and its concepts and services of the kinds explored by COLLATE which could be laid on top of or developed as a service of digital libraries more generally.

It may seem odd that the growth of entities tagged as digital libraries is so rapid in the face of a lack of a widely accepted definition and reference model. Currently the information landscape is a more closely paralleled by the western frontier of the United States in the 19th century than to the orderly landscape of libraries and archives in the second half of the 20th. The developments of the past ten years have laid out the options for digital library models and indicated the kinds of expectations that users have for digital libraries. Analysing current implementations and examining user expectations which themselves are richer because they reflect experiences as well as aspirations provides a foundation for a reference model that will create comparability and consistency across digital libraries.

As digital libraries lie at the heart of the information landscape of the 21st century how they are constructed and managed is of concern to public and private sector institutions. The Reference Model that DELOS Cluster 1 is creating will itself be a crucial element in the sustainability of digital libraries because it will provide a foundation for the audit and certification of repositories which in the longer term will be central to allowing service providers to produce adequate economic models and business cases for digital libraries.

2 Characteristics

The framework for this survey reflects the needs of work done so far towards the production of a first draft of the digital library (DL) reference model. In the model we distinguish between the following concepts:

- *DL*—which we define as the entity perceived by the end-user;
- *DL system*—which covers the software system that provides the digital library functionality on a set of information objects; and,
- *DL Management System* (DLMS), a term used to encompass the software system in charge of creating and managing DLs.

In the rest of this report we focus our attention on the aspects that characterise the last two of these concepts[3] since our goal is to appreciate the capabilities of the existing systems more than the DLs managed by them.

The main concepts that characterise a DL system according to the reference model are those illustrated in Figure 1.

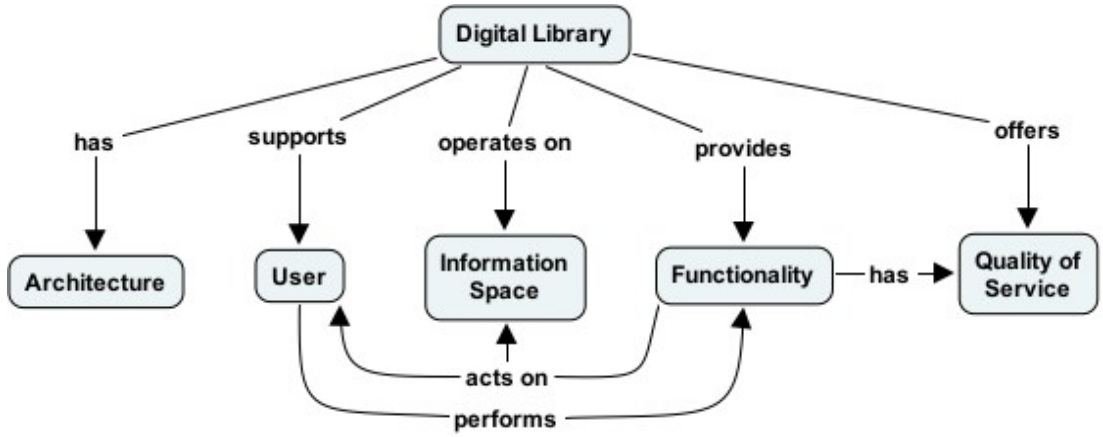

Figure 1. The main Concepts of a DL System

These five concepts, information space, user, functionality, quality of service and architecture, provide the framework for our consideration of other systems and lie at the heart of the conceptual reference model.

2.1 Information Space

The *Information Space* concept models the content that the digital library offers to its users. The information space comprises a set of *information objects* organised into *collections*. For "collections" in this context we mean a mechanism that groups a set of information objects collected either by enumeration or by specifying a set of characterization criteria. Collections can be organised hierarchically to model their subset relationships.

Information objects conform to an *information object model* that defines the set of possible objects that can be managed by the DL system. This model can represent aspects of a single object, like *versions*, *manifestations*, and related *annotations*, as a single entity or as the

[3] For simplicity of presentation, in the report we will generically refer to DL system to mean both DL system and DLMS when no confusion arises.

composition of different entities, possibly related among them. Objects are usually associated with a number of *metadata*. A system can support one or more types of metadata that differ in relation to their purpose and context of use. For example, a system can support: *descriptive metadata*, which describe the content of an information object; *structural metadata*, which represent the logical or physical relationships between information objects and their parts; *preservation metadata*, that represent the data structures and the systems needed for the preservation of the information objects over the time, and *administrative metadata*, which provide information about the creation of the objects and the constraints that governs their use, such as copyrights, use restrictions and license arrangements.

2.2 User

The *User* concept models the actors entitled to interact with the DL system. These are consumers and producers of the DL content and/or managers of both the content and of the DL functionality, including librarians, system administrators. The DL system can associate with each user a *profile* that contains both identification and behavioural information. Moreover, it can differentiate between different user *roles* each of which can be characterised by a set of access rights specified by *policies.* A DL system can also maintain a notion of *user group* which include, for example, users sharing a common objective.

2.3 Functionality

The Functionality concept models all the functions that can be activated by any of the DL system users. Logically, the basic functions that the DL system offers to its end-users[4] can be organised in four categories: Access, Submission, Management, and Personalisation.

The Access class comprises functions for supporting the consumers of the information objects in discovering, accessing, or transforming these objects. The discovery can be provided through different type of search functions. Typical functions are *monolingual search*, which measures the similarity between the query and the information object manifestation or its metadata; and *cross-language search*, which differ from the previous one since it measures the similarity without being influenced by the language of the query condition. Queries and similarity measures can act not only on texts but also on a large variety of other media, like image-, sound-, or 3D-objects. Variations of the search functions are: *relevance feedback search*, which taking into account the user judgement provides a customizable measurement of the similarity; and *browse*, which delivers access to ordered lists of object references built on selected metadata attributes.

Once discovered, the information object is consumed by means of a *visualize* function that produces a human understandable perception of an information object manifestation. To improve its usefulness before being visualised the information object can also be transformed through a *translate* function that alters its original manifestation changing the language, the format, and/or the presentation format.[5]

[4] For DL end-users we mean those users that submit and consume the DL information objects and the librarians that manage the content of the information space.

[5] Consideration is being given to whether in circumstances the visualize function may be replaced by a process function—e.g. where materials are discovered and submitted for reprocessing without visualisation.

The Submission category groups the functions that populate the information space. It includes the functionality to *submit* or to *update* an information object; to *annotate*; or to express a judgement by means of the *review* function.

The DL Management category brings together the functions that support the administration of both the information space and the users. These functions can be activated by the DL librarians. Typical management functions on the information space are: *publish*[6] and *withdraw* information objects or collections; *describe* them with appropriate metadata; *update* existing objects and collections in order to make them available with new formats; *disseminate*, i.e. notify the users whose subscription requests are correlated with the object descriptions; and, finally, *preserve* the digital object through the activation of appropriate mechanisms or processes. The *registration of new users*, their *role management* and the *policies management* are examples of the most common user management functions.

Personalize covers end-users customization functions. These can operate on the information space organization, as the *collection management* functions that enable the dynamic creation of new collections meeting the user interests, and on the user profile, as the *subscription* function that allows the users to explicitly define their topics of interest.

Those illustrated above are the main functionalities that can be provided by a DL system to its users. In analysing existing systems, we are also interested in understanding which functionality they offer for the DL management[7]. This functionality may comprise functions for:

- the *configuration of the information space*, such as the specification of the document format or the selection of the information sources to harvest from providers;
- the *configuration of user aspects*, including the specification of user profile formats, the assignment of roles to end-users, and the assignment of users to groups;
- the *specification of policies*, such as the definition of policies controlling the access to the information objects and services;
- the *customisation of specific services*, such as the user interface, the query language, and the browsable fields; and,
- the *definition of the log level* which allows the DL administrator to maintain the history of the activity performed by the users for a number of purposes, such as accounting, statistical analysis, diagnostic inspection, monitoring.

In addition to these capabilities, which mostly concerns the definition of a configuration that satisfies the end-user needs, the DL system can also provide functionality for the *configuration, deployment, and monitoring of the software components* that implement the system functionality.

2.4 Quality of Service

The Quality of Service concept encapsulates the capability of the system to satisfy the expectations of the different users that operate with the system. The degree of satisfaction is measured by the specific metrics associated with the quality parameters. Some of these

[6] Note that an information object submitted by the information producer is not always necessarily automatically published in the DL, it can be subjected to a review process and be published by the administrator only when the review process has been completed successfully.

[7] By analysing this functionality we are looking at the existing system as DLMSs.

parameters concern the DL as a whole, like the cost paid to access its content, whereas others may either apply to specific elements of the library, like the recall of a collection, or to the single functionality element, like the performance of the search function. The most required qualities of a DL are:

- *security*, the ability of the system to protect against threats such as illegal use or malicious attack;
- *economics*, the terms and conditions associated with a system capability;
- *availability*, the probability that a functionality responds to a consumer request;
- *reliability*, the likelihood that an element will behave in the way it is expected to do; and,
- *performance*, how well a function performs its task or activity. A particular kind of performance is the *response time* which captures the time spent from the formulation of the request to the reception of the response.

2.5 Architecture

The Architecture concept describes the architectural aspect of a DL. There are three main elements that characterize a DL system independently from its particular instantiations:

- its *software components*, the software modules that offer a well defined functionality, are autonomously configurable and deployable on one or more hosting nodes;
- its *application framework*, the set of libraries or subsystems that provide standard functionalities to the software components; and,
- its *constraints on the components distribution*, the constraints on the allocation of the components to the hosting nodes.

The discussion of DL systems in subsequent sections will be done within the conceptual framework of these five core concepts.

3 User Requirements

The DELOS Cluster on User Interfaces and Visualisation (WP4) conducted a questionnaire-based survey to develop an appreciation of the expectations that users have of digital libraries [13]. The main results of this study complement conclusions of other narrower studies (e.g. Bricks User Survey) and provide a validation for the reference models user-centric view. The study analysed the five classes of functionality: functions for locating information, functions for presenting resources, functions for personalization of content and services, facilities for communicating and collaborating with other DL users, and other common DL functions (such as Social navigation support, Multilingual support, Personal annotation, notification/alerting services, Glossaries, Thesaurus, and Dictionaries, Printing / Print preview facilities, and Downloading / uploading facilities).

3.1 Functionality

3.1.1 Integration of knowledge

The DL functions at the provider site identified by users as of highest priority include: Organizing resources;

- Archiving resources;
- Storing metadata about resources (creator, content, technical requirements, etc.);
- Locating resources;
- Creating cross reference links between similar resources, and
- Storing metadata about resources were classified homophonous by stakeholders as of highest importance.

Figure 2 provides an overview of common DL functions (content management) at the stakeholder site and their importance to stakeholders. The DL stakeholder functional requirements centre around two large functional areas: content management and membership management.

3.1.2 Access to knowledge

Among the functions for locating information higher significance ratings were allocated to those associated with *Search* (e.g., keywords search, parametric search). Moderate importance ratings were allocated to Index facilities and to Navigation related functions (e.g., browsing predefined catalogues) and lower importance ratings were given to “See also” items (e.g., similar to the one at hand) and to functions for Filtering search/browsing results (e.g., according to personal profile(s)), see Figure 2.

3.1.3 Administration of content

In the area of content administration the user survey found only moderate interest in: modifying existing classification schemes, retrieving services usage statistics, updating end-users on new/refined contents. Checking for inconsistencies appears to be of high importance to stakeholders. Stakeholders also expressed their wish/need for ‘access to control policies’ and for retrieval of DL usage statistical data.

3.1.4 Tool creation and management

Among the capabilities for personalisation of content and services: higher importance ratings were given to functions supporting the *Presentation* of contents according to profiles, and to *Bookmarks* facility (i.e., Favourites); moderate importance ratings were allocated to the provision of *Suggestions* for contents based on user profile, and to services offered for *Profile definition* (e.g., professional interests, personal interests); and lower importance ratings were allocated to the provision of *Suggestions* for discussion with other library members with similar interest profiles, and to *History facility*.

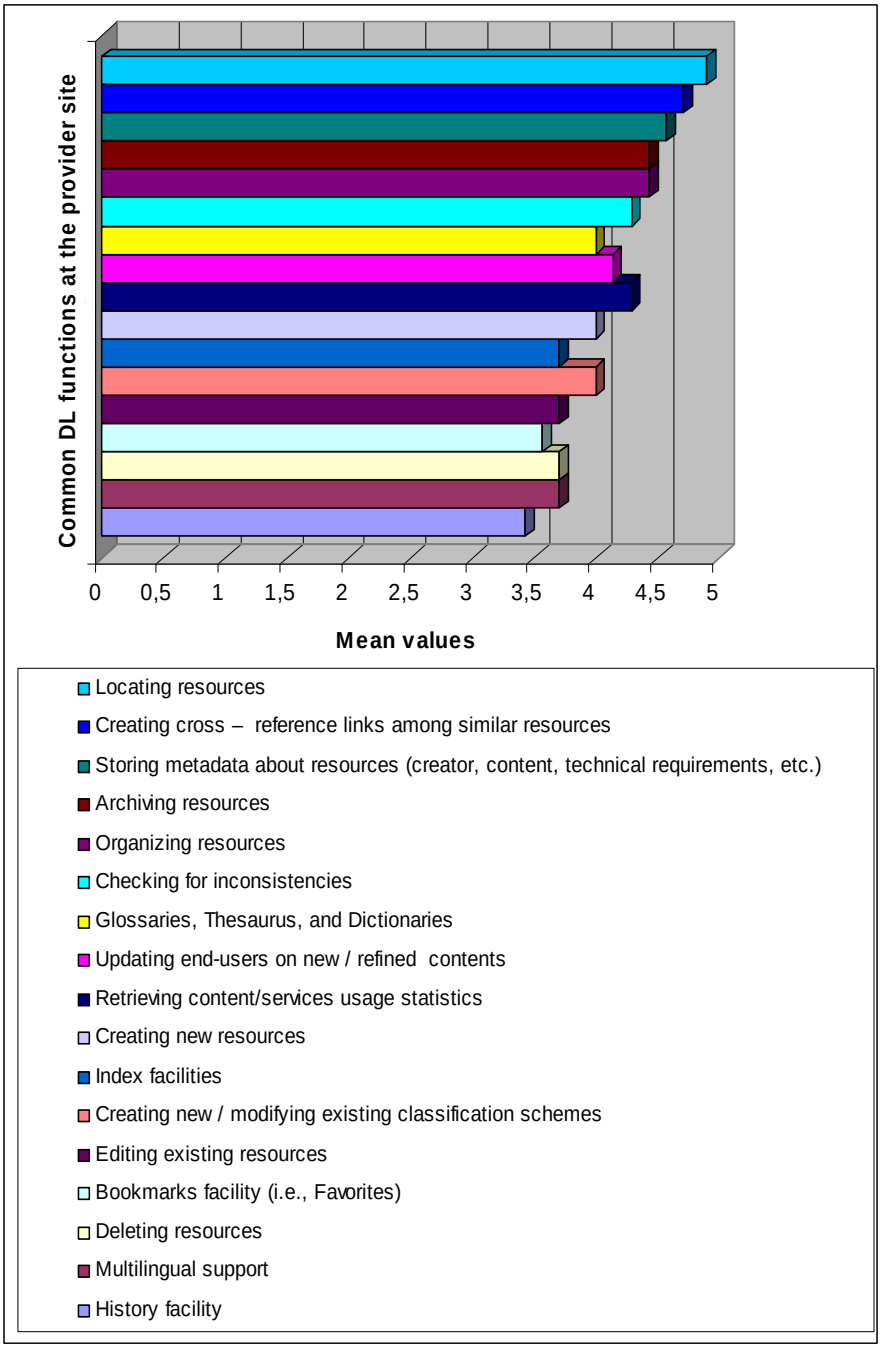

Figure 2. Common DL functions at the provider site and their importance to DL stakeholders

3.1.5 Services – Interfaces for access to integrated knowledge

Higher importance ratings were allocated to Short description/Previews, to Author(s)/editor(s), and to Title; moderate importance ratings were allocated to Popularity (e.g., number of visits), to Insertion /modification date, and to Related items; and lower importance ratings were allocated to Users' Ratings and to Users' discussions and reviews. An overview of the ratings is presented in Figure 3. As can be seen, the most important fields are the "title" and "author".

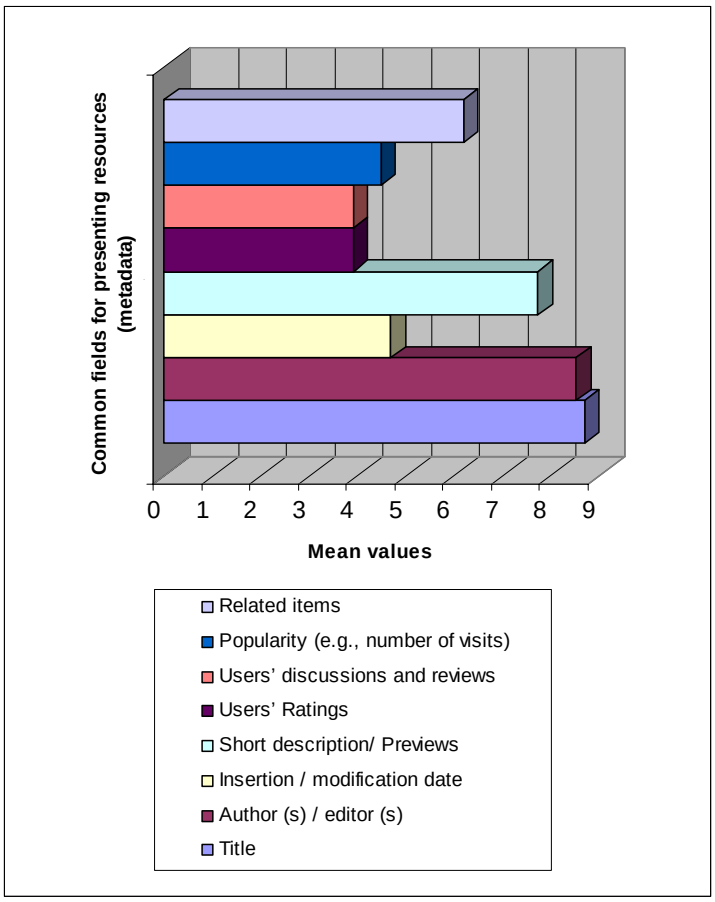

Figure 3. Common fields for presenting DL resources, and their importance to DL users.

3.1.6 Services – Interfaces for sharing/integration of knowledge

Among the functions for communicating and collaborating with other DL users: high importance ratings given for shared annotation facilities (e.g., peer reviews), and to e-mail services; moderate importance ratings were allocated to Message Boards services; and lower importance ratings were allocated to video conferencing and chat. Regarding communication, email is the most frequent way required for communicating, while video conferencing appears to be in little demand. Furthermore, among all facilities for collaboration, "track changes' facilities received the lowest scores.

3.1.7 Other DL functionalities

High importance ratings were allocated to Printing/Print preview facilities, and to Downloading/uploading facilities; moderate importance ratings were allocated to Personal

annotation, and Notification (Alerting) services; and lower importance ratings were allocated to Social navigation support (e.g., through users' rating of content) and to Multilingual support.

3.2 Ranking of DL Services/Functionality – User Perspective

Figure 4 shows a comprehensive list of high importance requirements for the DL user. Not surprisingly by far the most important functions are Search and Performance:

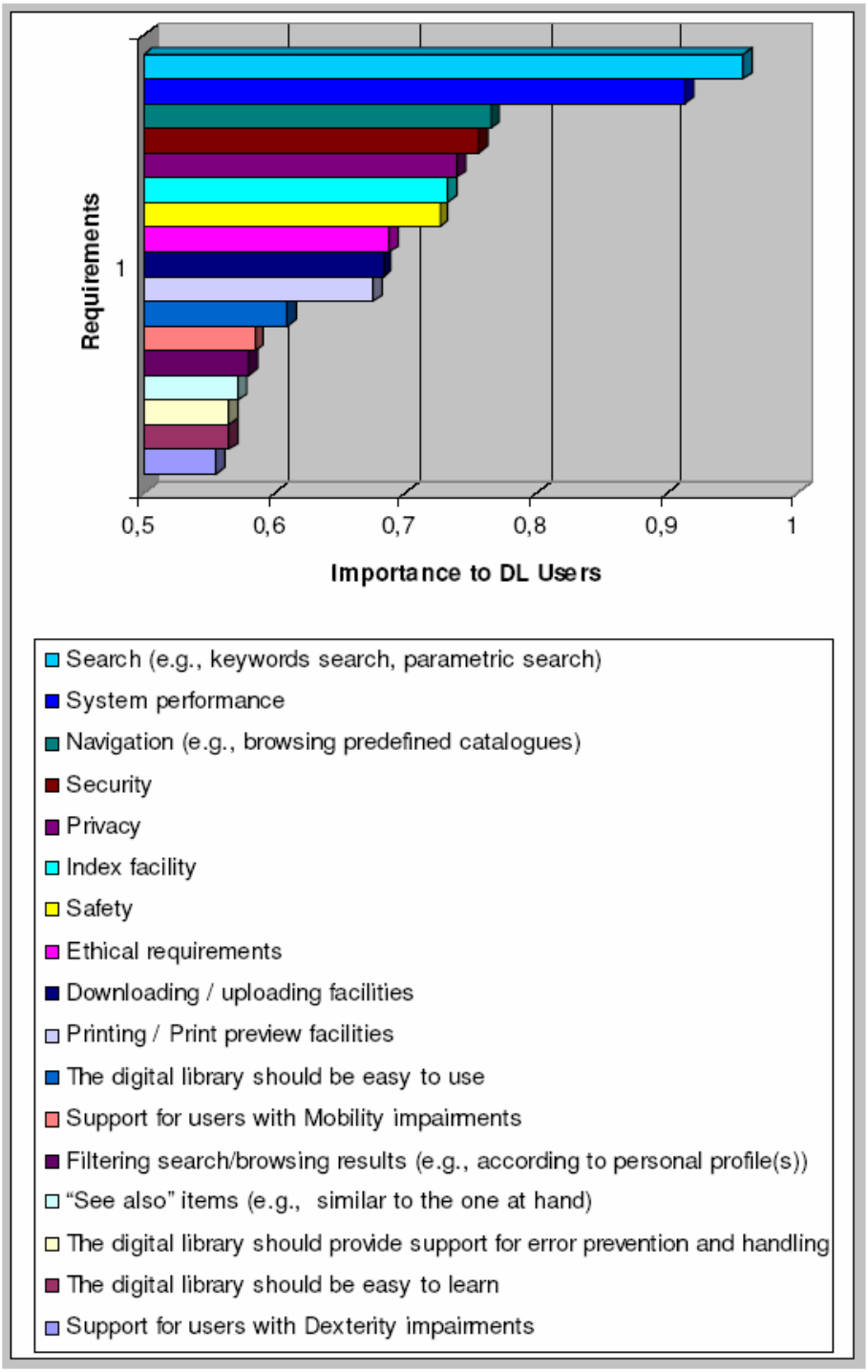

Figure 4. Highly important Requirements for DL Users

4 Repository Applications

Repository components and capabilities lie at the core of digital library systems. The core nature and containable functionality of repository components has become manifest in a number of repository systems. Anderson and Heery's examination of repository models provides an excellent overview of the characteristics of repositories, although it does not delve into any repositories very deeply [1]. Here we have chosen three repository applications, DSpace, Fedora, and aDORe, as foils for our discussion.

4.1 DSpace

DSpace [48][49] is an open source digital repository software system for research institutions. It has been developed jointly by the MIT Libraries and Hewlett-Packard Labs, and it is available under the BSD open source license for research institutions to run as-is, or to modify and extend as needed. It enables organizations to:

- Capture and describe digital material using a submission workflow module, or a variety of programmatic ingest options
- Distribute an organization's digital assets over the web through a search and retrieval system
- Preserve digital assets over the long term.

4.1.1 DSpace Information Space

The way data is organized in DSpace is intended to reflect the structure of the organization using the DSpace system. Each DSpace site is divided into communities; these typically correspond to a laboratory, research center or department. Starting from DSpace version 1.2, these communities can be organized into a hierarchy (Figure 5).

Communities contain collections, which are groupings of related content. A collection may appear in more than one community. Each collection is composed of items, which are the basic archival elements of the archive. Each item is owned by one collection. Additionally, an item may appear in additional collections; however every item has one and only one owning collection. Items are further subdivided into named bundles of bitstreams. Bitstreams are, as the name suggests, streams of bits, usually ordinary computer files. Bitstreams that are somehow closely related, for example HTML files and images that compose a single HTML document, are organized into bundles.

Each bitstream is associated with one Bitstream Format. Because preservation services may be an important aspect of the DSpace service, it is important to capture the specific formats of files that users submit. In DSpace, a bitstream format is a unique and consistent way to refer to a particular file format. An integral part of a bitstream format is an either implicit or explicit notion of how material in that format can be interpreted. For example, the interpretation for bitstreams encoded in the JPEG standard for still image compression is defined explicitly in the Standard ISO/IEC 10918-1. The interpretation of bitstreams in Microsoft Word 2000 format is defined implicitly, through reference to the Microsoft Word 2000 application. Bitstream formats can be more specific than MIME types or file suffixes. For example, application/ms-word and .doc span multiple versions of the Microsoft Word application, each of which produces bitstreams with presumably different characteristics. Each bitstream format additionally has a support level, indicating how well the hosting institution is likely to be able to preserve content in the format in the future. There are three possible support levels that bitstream formats maybe assigned by the hosting institution: (i)

Supported - the format is recognized, and the hosting institution is confident it can make bitstreams of this format useable in the future, using whatever combination of techniques (such as migration, emulation, etc.) is appropriate given the context of need; (ii) Known- the format is recognized, and the hosting institution promises to preserve the bitstream as-is, and allows it to be retrieved. The hosting institution will attempt to obtain enough information to enable the format to be upgraded to the "supported" level; (iii) Unsupported - the format is unrecognized, but the hosting institution undertakes to preserve the bitstream as-is and allows it to be retrieved.

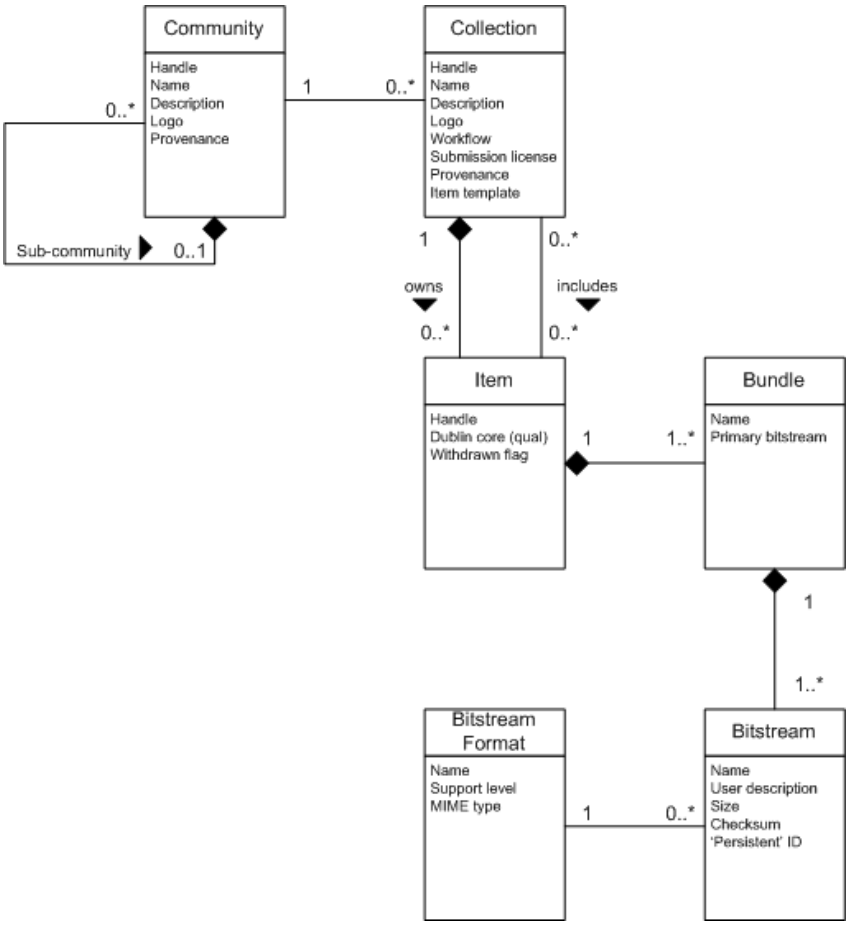

Figure 5. DSpace Data Model

Each item has one qualified Dublin Core metadata record. Other metadata might be stored in an item as a serialized bitstream, but DSpace stores Dublin Core for every item for interoperability and ease of discovery. The Dublin Core may be entered by end-users as they submit content, or it might be derived from other metadata as part of an ingest process.
Items can be removed from DSpace in one of two ways. They may be with- drawn, which means they remain in the archive but are completely hidden from view. In this case, if an end-user attempts to access the withdrawn item, they are presented with a "tombstone", that indicates the item has been removed. For whatever reason, an item may also be expunged if necessary, in which case all traces of it are removed from the archive.

4.1.2 User

Even if DSpace's features for information objects discovery and retrieval can be used anonymously, users must be authenticated to perform actions such as submission, email

notification or administration. The system supports groups in order to ease their administration. For each DSpace user, described as e-people, the system stores an: e-mail address, first and last name, a password, a list of collections she/he wishes to be notified of new items. E-people authenticate with username/password pairs, X509 certificates, or LDAP.

4.1.3 Functionality

DSpace provides basic functionality for managing users and group of users, collections and items. In particular, it maintains the user's basic information such as that described above in order to personalise the systems functionality, for example, submission information. Particular emphasis is placed on the management of items and in particular in the publishing phase. DSpace provides an easy and customisable way for items to be brought into the archive. All the DSpace items are held in one of three spaces: items being assembled, pending submissions, and archived items. The 'items being assembled' are those in the early stages of ingest and are awaiting the addition of metadata and may be missing components. The submission pending items have been presented to the collection's submission approval process. Finally, the archived items have been approved for entry into the collection to which they was submitted. Only authorized users can initiate submissions to a DSpace collection.Each collection can have its own approval process, which specifies the individuals who can participate in the process and at what level.

4.1.4 DSpace Architecture

The DSpace system is organized into three layers, each of which consists of a number of components, as presented in Figure 6.
The storage layer is responsible for physical storage of metadata and content. The business logic layer deals with managing the content of the archive, users of the archive, also called e-people, authorization, and workflow. The application layer contains components that communicate with the world outside of the individual DSpace installation, for example the Web user interface and the Open Archives Initiative protocol for metadata harvesting service.

Each layer only invokes the layer below it: the application layer may not use the storage layer directly, for example. Each component in the storage and business logic layers has a defined public API. The union of the APIs of those components are referred to as the Storage API (in the case of the storage layer) and the DSpace Public API (in the case of the business logic layer). These APIs are in-process Java classes, objects and methods.

It is important to note that each layer is trusted. Although the logic for authorizing actions is in the business logic layer, the system relies on individual applications in the application layer to correctly and securely authenticate e-people. If a "hostile" or insecure application were allowed to invoke the Public API directly, it could very easily perform actions as any e-person in the system.

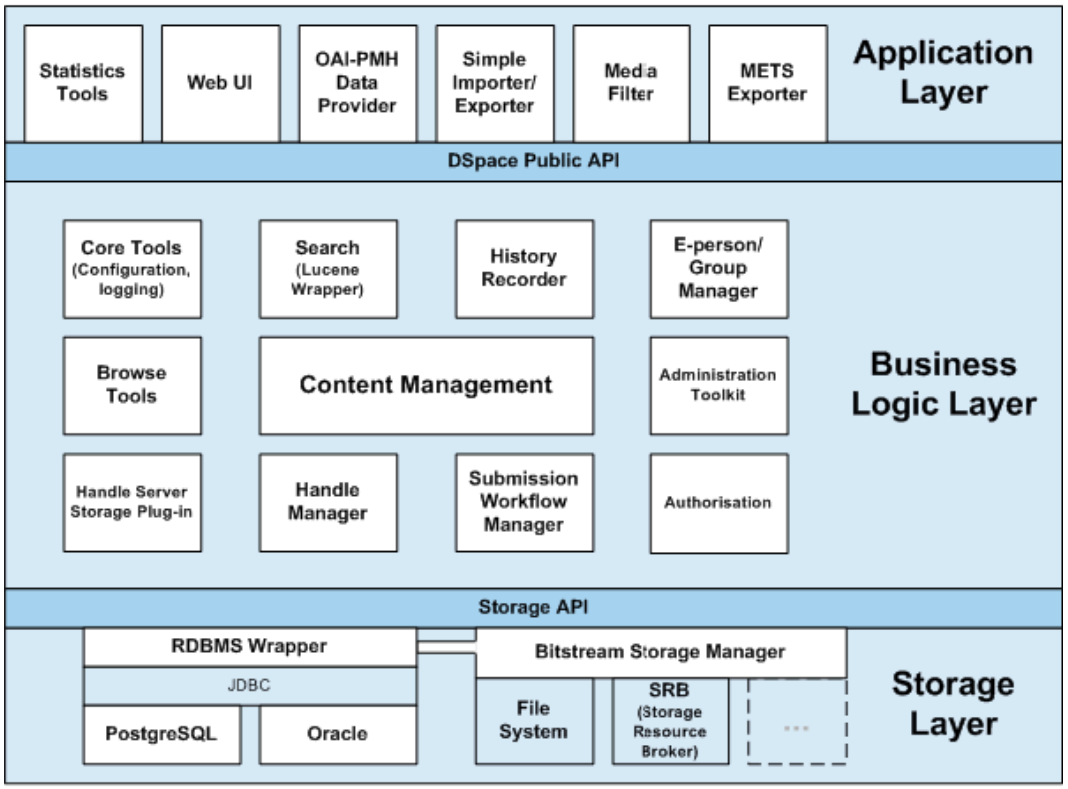

Figure 6. DSpace Architecture

The reason for this design choice is that authentication methods will vary widely between different applications, so it makes sense to leave the logic and responsibility for that in these applications.
The source code is organized to cohere very strictly to this three-layer architecture. Also, only methods in a component's public API are given the public access level. This means that the Java compiler helps ensure that the source code conforms to the architecture.

DSpace uses a relational database to store all information about the organization of content, metadata about the content, information about e-people and authorization, and the state of currently running workflows. The DSpace system also uses the relational database in order to maintain indices that users can browse. Most of the functionality that DSpace uses can be offered by any standard SQL database that supports transactions. Presently, the browse indices use some features specific to PostgreSQL and Oracle, so some modification to the code would be needed before DSpace would function fully with an alternative database back-end. DSpace offers two means for storing content. The first is in the file system on the server. The second is using SRB (Storage Resource Broker). Both are achieved using a simple, lightweight API. SRB is purely an option but may be used in lieu of the server's file system or in addition to the file system. Without going into a full description, SRB is a very robust, sophisticated storage manager that offers storage and straightforward means to replicate (in simple terms, backup) the content on other local or remote storage resources.

The Search functionality is realized exploiting the Apache Lucene capabilities. However, DSpace presents some limitations. First of all, it creates an IndexReader for each query, which isn't the most efficient use of resources: a wildcard query will open many filehandles generating a really heavy load. Since Lucene is thread-safe, DSpace is evaluating a better

future implementation that will rely on a single Lucene IndexReader shared by all queries, that is invalidated and re-opened when the index changes. The second limitation is represented by the fact that the API does not include relevance scores (Lucene generates them, but DSpace ignores them) and abstractions for more advanced search concepts such as boolean queries.

4.2 Fedora

Flexible Extensible Digital Object and Repository Architecture (Fedora) [17] supports interoperability and extensibility of digital library systems and institutional repositories. The Digital Library Research Group at Cornell University originally developed the Flexible Extensible Digital Object Repository Architecture (Fedora) under a National Science Foundation Grant. The transition of Fedora from a research prototype to production repository software began when the University of Virginia Library, seeking a solution for managing increasingly complex digital content, experimented with the Fedora architecture [28]. The experimentation proved successful, providing the basis for subsequent funding from the Andrew W. Mellon Foundation to Cornell and Virginia to jointly develop Fedora and make it available as open source software to libraries, museums, archives, and content managers, facing increasing variety and complexity in the digital content that they manage [46]. Mellon-funded development continues through 2007.

Fedora is implemented as a set of web services that provide full programmatic management of digital objects as well and search and access to multiple representations of objects [35]. All Fedora APIs are described using the Web Service Description Language (WSDL). As such, Fedora is particularly well-suited to exist in a broader web service framework and act as the foundation layer for a variety of multi-tiered systems, service-oriented architectures, and end-user applications. This distinguishes Fedora from other complex object systems that are turn-key, vertical applications for storing and manipulating complex objects through a fixed user interface (e.g., DSpace [48][49], arXiv [2], ePrints [16], Greenstone [22])

By providing both a model for digital objects and repository services to manage them, Fedora is also distinguished from work focused on defining and promoting standard XML formats for representing and transmitting complex objects (e.g., METS [31], MPEG-21 DIDL [25], IEEE LOM [23]). However, Fedora currently supports ingest/export of digital objects encoded using METS and also the Fedora XML wrapper format (FOXML). The support of MPEG-21 DIDL and possible other formats is currently under realization.

Actually, prior to version 2.0 of Fedora, all Fedora-related functionality was built into the core Fedora repository service. As of version 2.0, the Fedora Ser-vice Framework was defined to move the Fedora architecture in a direction where new services can easily be developed and plugged into the Framework. This is very consistent with the Reference Architecture that we have presented in which formerly tightly integrated systems are broken apart into atomic, modular ser-vices that can be flexibly aggregated into different multi-service compositions. At the time of writing, Fedora is migrated to the new service framework approach. Version 2.1 of Fedora includes a new OAI Provider and a new Search service as part of the Fedora open-source distribution. These functions were previously built into the core repository.

4.2.1 Fedora Information Space

The Fedora object model supports the expression of many kinds of complex objects, including documents, images, electronic books, multi-media learning objects, datasets, computer programs, and other compound information entities. Fedora supports aggregation of any combination of media types into complex objects, and allows the association of services with objects that produce dynamic or computed content. The Fedora model also allows the assertion of relationships among objects so that a set of related Fedora objects can represent the items in a managed collection, the components of a structural object like the chapters of a book, or a set of resources that share common characteristics (defined by semantic relationships).

The object models are templates for units of content, called data objects, which can include digital resources, metadata about the resources, and linkages to software tools and services that have been configured to deliver the content in desired ways. These software connections are provided as methods encoded into two kinds of interrelated behavior objects as described below. A Fedora repository provides access to the data objects by leveraging tools and services that are described by the behavior objects. The behavior objects store metadata that describes the operations of the tool/service and the runtime bindings for running the operations. The WSDL is used to describe the tool/service bindings.

The digital resources and the metadata are datastreams in an object model. The content of a datastream is identified using a URL. When an object is ingested into a Fedora repository, a URL for a managed datastream is used by the repository system to retrieve the content and store it in the file space under its control; the datastream in the object is updated to be this internal address. When an object contains a datastream defined as external, the URL is stored in the datastream and used by the repository to access the data whenever necessary. An in-line metadata datastream is a bytestream that is namespaced XML encoded data stored in the XML instantiation of the object directly, rather than as remote or managed content. From the user's point of view, the linkages to software tools and services (via disseminators) are seen as behaviors upon the units of content. These behaviors can be exploited to deliver varieties of prepared content directly to a web browser. Fedora makes it possible to describe abstract sets of behaviors that constrain a corresponding set of specific processes or mechanisms delivering the behavior described for a given unit of content. One abstract set of behaviors, a behavior definition (bdef) object, can be used to constrain many mechanisms, or behavior mechanism (bmech) objects, ensuring a standardization of behaviors for different units of content that are equivalent in type, but differing in format. A bdef object formally defines the terms of a behavior contract that must be upheld by any bmech object to be paired with it. In turn, the bmech object contains a data contract, the terms of which any data object model subscribing to it must meet. Bdef objects and bmech objects are analogous to interfaces and implementations in object-oriented programming.

A data object model subscribes to a set of behaviors by linking to a bdef object and pairing it with a link to an appropriate bmech object. This pair of links defines a disseminator; an object model can contain any number of disseminators. In practical terms, this means a specific data object conforming to the model can have sets of behaviors for a variety of purposes, or sets of behaviors equivalent in purpose but that prepare the object's content to be delivered to applications with different format requirements. In summary, a data object model specifies the number and types of datastreams as well as the set of disseminators every conforming data object will have.

4.2.2 User

Users interact with the content held in the repository by means of client applications (e.g., web browsers, batch programs, or server applications). These applications access the repository's data by means of the four APIs by which Fedora is exposed: management, access, search (which are exposed via HTTP or SOAP), and the OAI provider API (exposed via HTTP). As a consequence their management is outside of the scope of the system.

4.2.3 Functionality

The main functionality provided by the Fedora system are a direct consequence of its rich and flexible information object model. In particular, the system's ability to represent rich objects is enabled to

(i) manage multiple types of objects,
(ii) control versioning of objects by linking all those related to each single object providing a history of how it changes over time,
(iii) regulate access to them via IP based restrictions access.

Moreover, each Fedora repository implements the OAI-PMH protocol and maintains a Dublin Core record. This enables each Fedora object to act as OAI data provider. Other than the classic functionality a repository provides for accessing objects, Fedora provides search functionality. In particular, its capability to represent complex objects and their relationships in terms of RDF representations, it supports iTQL and RDQL queries which provide clients with a powerful and flexible discovery mechanism.

4.2.4 Fedora Architecture

Fedora digital objects are managed within the Fedora Service Framework, which consists of a set of loosely coupled services that interact and collaborate with each other. At the core of the framework is the Fedora repository service. Other services exist around the core to provide additional functionality that is not considered a fundamental function of a repository. Any number of services can be developed to collaborate with the core Fedora repository service. Examples of these services are the Fedora OAI provider and the Fedora Search service.

The framework approach anticipates that new services will be added over time. Outside of the boundaries of the Fedora framework are external services that can either call upon Fedora services, or that Fedora can leverage in someway. The distinction between services within the Fedora Service Framework, and those outside, is that those within the framework are in a trusted relationship with the Fedora repository service, and are designed to specifically interact with Fedora repositories. Services outside the framework are typically general-purpose services, or organization specific services that call upon Fedora as an underlying repository for digital content.

At the core of the Fedora Service Framework is the Fedora repository service, which exposes interfaces for managing and accessing digital objects in a repository. In Figure 7, the repository service is deconstructed so that its internal modules and public service interfaces are visible.

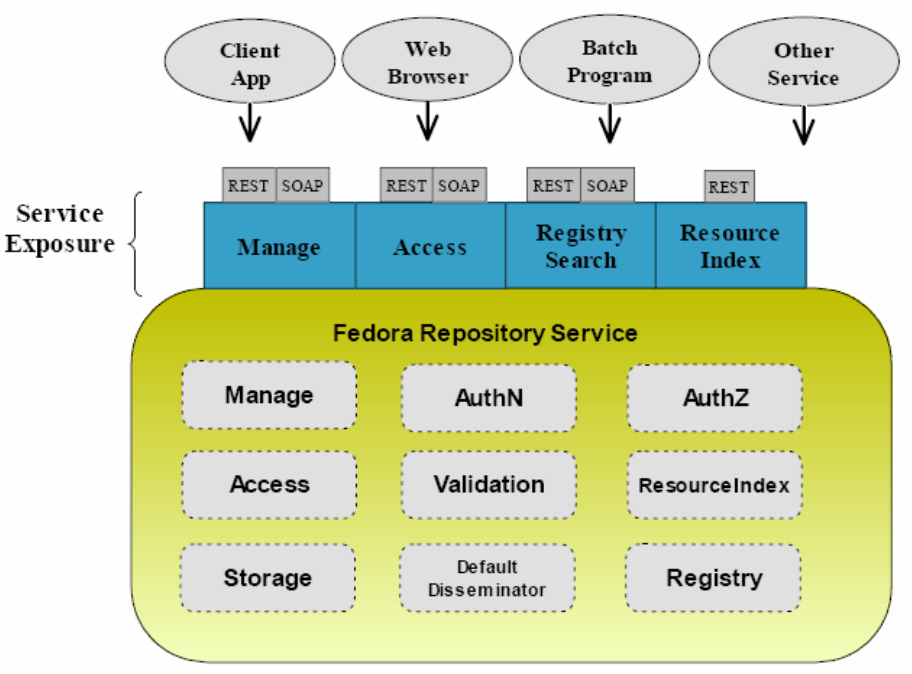

Figure 7. Fedora Architecture

At the top layer of the architecture there are alternative client scenarios for accessing the Fedora repository through its four web service interfaces. Each service interface is defined using the Web Service Description Language, with both SOAP and REST bindings. The internal implementation of the Fedora repository service consists of a set of internal java modules that can be configured, and optionally replaced with alternative implementations. The internal modules are not directly exposed to accessing clients; instead clients interact with the repository only through the defined web service interfaces.

The Management service interface (API-M) contains read/write operations necessary to manage a repository of digital objects. API-M operations exist for ingesting and exporting digital objects in an XML format, either Fedoras FOXML, or alternatively METS or MPEG21/DIDL. Also, objects can be created and modified using component-level operations that reflect the functional view of the Fedora object model described earlier.

The Access service interface (API-A) contains read-only operations for ac- cessing digital objects. The two main purpose of the Access interface is to (i) introspect on a digital object (i.e., to discover what datastreams and disseminator methods are available) and (ii) request disseminations on an object (i.e., access particular representations of the objects content).
The final two access points to the Fedora repository service are the Registry Search and Resource Index interfaces. These provide discovery capabilities to locate digital objects. The Registry Search interface exposes service operations to perform a simple search of the digital object registry based on object properties. The Resource Index interface is the service entry point to an RDF-based index of the entire repository. As such it contains all representations and relationships of objects, plus object properties and Dublin Core metadata elements.

4.3 aDORe

aDORe is a repository system designed and implemented at Los Alamos National Laboratory for enabling self-supporting access to digital information. It hosts the vast collection of digital scholarly assets that the LANL Research Library acquires or licenses (approximately 80,000,000 on June 2005) and makes them accessible through locally developed user services.

4.3.1 Information Space

In the aDORe architecture, digital objects are represented by means of the XML-based format of the MPEG-21 Digital Item Declaration Language (DIDL). A digital object can consist of multiple data-streams as Open Archival Information System Archival Information Packages (OAIS AIPs). According to OAI MHP, items in the distributed repositories may be organized into sets. A set have either flat or hierarchical structure; multiple and parallel set structures are also supported.

4.3.2 User

The aDORe repository system is accessible by downstream applications only.

4.3.3 Functionality

The aDORe digital object repository was designed and implemented for ingesting, storing, and accessing a vast collection of Digital Objects at the Research Library of the Los Alamos National Laboratory.

Digital objects to be stored in aDORe can, in principle, be obtained in a variety of ways including FTP, OAI-PMH resource harvesting, and Web crawling. The ingestion process represents each digital object according to MPEG-21 DIDL specification. Hereby, a DIDL XML document is created that functions as the OAIS AIP representing the digital object. When a new version of a previously ingested digital object needs to be ingested, a new DIDL document is created for it: the aDORe Identifier Locator component keeps track of all versions of a digital object.

The access function is only provided to downstream applications that access the repository through OAI-PMH protocol requests. Harvesters collect a DIDL document from the aDORe environment. As a result, identifiers contained in the harvested DIDL documents become available in applications such as search engines. The identified resource is then retrieved from the aDORe environment by its Identifier Locator component.

4.3.4 Architecture

The architecture has a natively component-based, distributed design: it operates on the basis of groups of autonomous components; interaction with those components is protocol-based.

The OAIS AIPs are stored in autonomous repositories and brought in through an appropriate ingestion process. A Repository Index keeps track of the creation and location of all the autonomous repositories, whereas an Identifier Locator registers in which autonomous repository a given Digital Object or OAIS AIP resides.

aDORe introduces an OAI-PMH Federator, which is used for requesting OAIS Dissemination Information Packages (OAIS DIPs). These dissemination packages can either refer to stored

OAIS AIPs, or to transform AIPs. The Federator acts as a front-end to the overall architecture and allows OAI-PMH harvesters to collect batches of OAIS DIPs from aDORe.

aDORe provides a second front-end, the OpenURL Resolver, which is used for requesting OAIS Result Sets. An OAIS Result Set is instantiated through the dissemination of a Digital Object or of its constituent data-streams.

Both front-ends make use of an MPEG-21 Digital Item Processing Engine to apply services to OAIS AIPs, Digital Objects, or constituent data-streams that were specified in the request.

4.4 Categorisation

The table below summarizes the aspects of the different repository systems that have been analyzed. Those aspects for which we have found no description in the literature have been left unspecified.

	Fedora	DSpace	aDORe
User			
User Identifier	Yes	Yes	No
User Profile	No	Yes	No
Role	No	Yes	No
Policy	No		No
Group	No	Communities	No
Information Space			
Information Object	FOXML (supports METS, too)	Relational Database	MPEG-21 DID
Information Object Identifier	FEDORA Persistent Identifier (PID)	CNR Handle System	Identifier Locator
Content			
o Metadata	Yes	Yes	Yes
o Text	Yes	Yes	Yes
o Image	Yes	Yes	Yes
o Audio	Yes	Yes	Yes
o Video	Yes	Yes	Yes
o Composite	Yes	Yes	Yes
Version	Versioning of individual datastreams (linear)	ABC Harmony Datamodel	Yes
Manifestation	datastream	bitstream	datastream
Annotation			
Metadata			MPEG21 DIDL
o Descriptive Metadata Format	DC (searchable), any other format non-searchable	Qualified DC	
o Structural Metadata Format	FOXML	Relational Database	
o Administrative Metadata Format	FOXML	Relational Database	
o Preservation Metadata Format	FOXML, Functional preservation is implemented by the user through the	Relational Database, Bundles and Bitstreams. Functional preservation through supported file formats, bitstream preservation otherwise	
Collection	RDF based Object-to-object relationships	Native support for Collections in the Relational schema	OAI-PMH sets
Functionality			
Access			OAI-PMH
o Search	Yes	Yes	No
• Full Text	No	No	No
• Metadata	Yes	Yes	No

• Image	No	No	No
• Audio	No	No	No
• Video	No	No	No
• Speech	No	No	No
• Single-Object, Single-Feature	No	No	No
• Multi-Object, Multi-Feature	No	No	No
• Compound Document Match	No	No	No
• Predicates	No	No	No
• Query Expansion	No	No	No
o Cross-language	UTF-8	UTF-8	No
o Relevance Feedback	No	No	No
o Browse	Very Basic	Yes, built-in support in the Web UI	No
o Visualize	Very Basic	Yes, built-in support in the Web UI	No
o Translate	No	No	No
Content Management			
o Submit	Administrator only, perhaps in FEDORA 2.1 updated Authorization model other roles may exist too	Yes	
o Update	Administrator only	Yes	
o Annotate			
o Review	Administrator only	Yes	
DL Management			
o Annotate			
o Update	Yes	Yes	
o Withdraw	Yes	Yes	
o Describe	Yes	Yes	
o Disseminate	Yes	Yes	
o Preserve	Yes	Yes	
o User Management			
• Registration	No	Yes	
• Role Management	No	Yes	
o Policy Management		Yes	
Personalize			
o Collection Management	No	Yes but basic	No
o Personalised access	No	Yes	No
o Notification	No	Yes	No
o Others		Online registration	No
Enabling			
o Authentication	Yes	Yes	
o Authorization	Yes (XACML-based Policy Enforcement)	Yes (Relational Database)	
o Encryption	No	No	
o Subscription	No	Yes	
o Notification	No	Yes	
o Process composition	No, only dissemination of behaviors on datastreams	Yes (simple 3-step workflow)	
Others	• OAI-PMH harvesting • Most functions are exposed as SOAP-based Web Services • Support for LDAP, IP-based authentication, HTPP basic authentication and SSL-based authentication	• OAI-PMH harvesting • Support for LDAP	• OAI-PMH Federator • Open URL Resolver
Quality of Service			
Security			

Economics			
Availability			
Reliability			
Performance			
Response time			
Security			
Authentication			
Integrity			
Data Protection			
Message Protection			
Robustness			
Capacity			
Load balancing			
Recoverability			
Messaging			
Consistency			
Scalability			
Architecture			
Characteristics	Loosely coupled services acting on top of the repository service.	Layered architecture of components interacting through API.	Component-based and standard-based (XML, MPEG21 DID and DII, OpenURL, OAI-PMH) architecture where interaction is protocol-based.

5 Systems for specific digital libraries

This section analyses the systems that support three among the most significant initiatives that aim at building large scale DLs by federating content published by different providers, DAREnet, TEL and NSDL. Note that most of the descriptions of these initiatives found in the literature focus more on the aspects of the digital library than on the system used to support it. In many cases we have thus inferred the characteristics of the systems from the description of the DL itself.

5.1 DAREnet

DARE [12][51] is a joint initiative by Dutch Universities, National Library of Nederland, Royal Nederland Academy of Arts and Sciences (KNAW), and Nederland Organization for Scientific Research (NWO). Its aim is to store the digital outputs of all Dutch research in a common network of *Institutional Repositories* in order to facilitate its accessibility.

The Project aims to engage all academic institution wishing to join the federation, as long as its repository adheres to the DARE specification and sharing philosophy. Currently, the federation encompasses the institutional repositories of Dutch universities.

In the beginning of the project not all universities involved had an operative institutional repository, and not all the extant institutional repositories were Open Archive compliant. For example, three of them were partners in the ARNO project (a SURF funded project) and had the system in use; others had a DSpace repository operative; others had to implement a repository from scratch or have their digital objects hosted by other partner repositories. All of them had to convert their internal metadata format to the DC qualified metadata.

DARE lays the foundation for a federation of institutional repositories—a network of interoperable digital libraries—by setting out guidelines for the cooperation and interoperability of otherwise independent institutional repositories. Interoperability, based on qualified DC metadata and OAI-PMH standards, enables the construction of services across the federation. The proof-of-concept of the project is *DAREnet*, a demonstration portal-service for searching and browsing all institutional repositories as a whole.

5.1.1 Information space

Metadata

DARE employs a simple DC as the mandatory metadata set, plus DARE-qualified DC elements (*dare-qdc*) as an optional metadata set. Such a set features elements describing the unique digital object identifier (e.g., DOI of IDF or CNRI handle system), the properties required to establish user rights in relation to objects, and the distinction between peer-reviewed and non-peer reviewed content. All federated institutional repositories partaking convert their metadata format into the DARE format and become OAI Providers.

Manifestations and document format

Digital objects should respect *SPARC content requirements* and be validated accordingly before being stored in an institutional repository; DARE acknowledges the importance of differentiating between published and grey-literature, as well as peer-reviewed and non-peer reviewed research. However, no restrictions on data formats are enforced by DARE; manifestations can be freely handled by the individual repositories and according to their

implementation functionality. Instead, in the style of harvesting, a link to the manifestation or to a jump-off page must be made available through the metadata.

5.1.2 Users

Individual repositories
DARE's guidelines state that the administration of the individual repository should include a process to apply for and supply user ID's and passwords. The specification anticipated metadata fields for user rights and access information, so as to allow the future construction of appropriate personalization services (none are built at the moment).

DAREnet
At the service level, users are not registered but can still freely query the repositories via the Web site DAREnet. Instead, DAREnet administrators have restricted access to the functionality of harvesting settings and DAREnet site pages configuration.

5.1.3 Functionality

Access
Since 2004, DAREnet service-portal demonstrates the benefits of the interoperability provided by the DARE content network. The portal allows users to run *Google* like queries on metadata or on full-text over data stored on a local repository, whose content is obtained by OAI harvesting the repositories. Full-text search can be performed on all type if sources, be it Web pages, PDF documents or database records. As a consequence, the repository's content is also accessible by search engines such as *Google*.

In some cases service providers may wish to retrieve the object and not just the metadata. DARE's specification claims that one solution *might* be the use of MPEG21-DIDL; so far, this solution has only been implemented at Los Alamos in the context of aDORe [6].

Data ingest
Management of digital objects is left to the individual institutional repository administrations.

Preservation
DARE's content network offers a common service for digital objects preservation. The service, *E-Depot*, is delivered by the Koninklijke Bibliotheek (The Royal Library) and guarantees long-term preservation of its content. Repositories can send digital objects to E-Depot, which stores them securely and free of charge. Stored objects retain their unique identifiers and can be retrieved by their original institutional repositories at any time.

DL management and expandability
Security
On of the main issue of DARE is that of retaining university copyright and right over their publications while allowing their free access and dissemination. However, no policy is imposed nor suggested on data protection issues, messaging and integrity. Each repository implements its own security policy and exports metadata and/or documents accordingly.

Management services
Specific services can be built over the data layer to access and publish the network content for different use. One of the main goals of DARE is to initiate and stimulate the supply of academic research material to the repositories. In order to do that, grants are being awarded

yearly to relevant projects and initiatives which offer services to academics or direct benefits to their work. Among these, some examples are:

- *DARLIN* (Dutch ARchive for Library and Information sciences). It gives subject specific access to Dutch publications on library and information science. Additionally, authors can self-archive their own publications and an Open Access multimedia journal for new publications in this field, DARLIN journal, will be available shortly.
- *NARCIS*. It offers central access to Dutch research information.
- *P-Web*: P-Web is a web-based tool for publishing conference proceedings on-line by using Institutional Repositories for the entry and storage of documents. The organization of a conference can add additional conference-specific information or use it as the website of the conference.
- *Theses Online*: is the start of a national database of graduate papers or extended theses taken from OAI-compatible institutional repositories, classified according to discipline, including a search engine.

Furthermore, DARE's content is made available to *OAIster* (University of Michigan), which enables easy searching in a collection of freely available, academically-oriented OAI-compliant digital repositories currently including more than six million documents.

5.1.4 Architecture

The architecture proposed by DARE is composed of a data layer and a service layer as depicted in Figure 8. The data layer consists of a federation of OAI data providers, exposing Institutional repositories metadata according to an agreed DARE DC metadata format and, on demand of an OAI harvester, feeding a centralized repository. In an OAI architectural style, this OAI repository becomes the mid-layer for the construction of services over the whole federation.

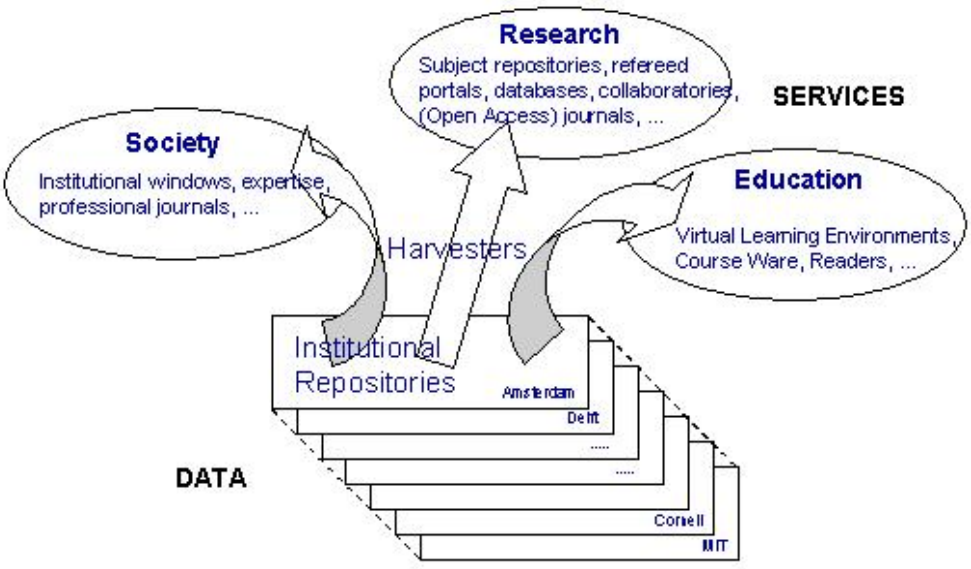

Figure 8. The DARE Architecture: Data and Services Model

DAREnet presentation layer, search, indexing, and harvesting are implemented by *i-Tor* technology [24] (funded by SURF project). i-Tor is an open-source tool (based on free components like Linux, Java, MySQL and Lucene) for the implementation of services,

including information portals (web access to heterogeneous information sources, e.g. database systems, document repositories, file systems, etc.) and repositories, intended as combination of an OAI Repository, to make local data (e.g. relational database) and files (e.g. academic publications) accessible as records of an Open Archive and an OAI Harvester.

i-Tor permits the restricting of access to local data and HTTP services by username/password permissions over an SSL connection. Exploiting such mechanism, DAREnet administrators can remotely manage harvesting settings and DAREnet site pages from a Web interface.

5.2 TEL—The European Library

The European Library (TEL) web service is a portal which offers access to the combined resources (books, magazines, journals.... - both digital and non-digital) of the forty-five national libraries of Europe [50]. It offers free searching and delivers digital objects - some free, some priced. The European Library service is aimed at informed citizens world-wide (both professional and non-professional) who want a powerful and simple way of finding library materials. Moreover, it is expected to attract researchers as there is a vast virtual collection of material from all disciplines. It offers anyone with an interest a simple route to access European cultural resources.

The European Library originates from the TEL project, which was finished successfully on January 31, 2004. The key aim of TEL project was to investigate the feasibility of establishing a new Pan-European service (named The European Library) which would ultimately give access to the combined resources of the national libraries of Europe. The project was partly funded by the European Commission and it ran for 3 years. During the project it became clear that Gabriel, the current website of the European national libraries, Gabriel (GAteway and BRIdge to Europe's National Libraries) would be integrated into the new European Library website. The Gabriel website was integrated into The European Library in the summer of 2005.

Two kinds of participants are involved in the in development of The European Library:
Full participants are the national libraries of Austria, Croatia, Denmark, Estonia, Finland, France, Germany, Italy, Latvia, Netherlands, Portugal, Serbia, Slovenia, Switzerland and the United Kingdom, along with ICCU (the national central cataloguing institute from Italy) and CENL (the Conference of European National Librarians). The collections of the Full Participants are the first to be included (searchable) on the European Library website.
Basic participants are the other 30 European national libraries. Their collections will be included at a later stage.

The day-to-day work (management, marketing, implementation, maintenance, design, editorial work, development, technical helpdesk etc.) is done by the European Library Office team, based at the Koninklijke Bibliotheek, the National Library of the Netherlands.[8]

5.2.1 Information Space

The European Library offers a default information space, consisting in a set of collections selected from The European Library's entire list of collections. This currently contains one collection from each of our partner libraries plus a virtual collection of digitized material from

[8] http://libraries.theeuropeanlibrary.org/contactus_en.html

several of our partner libraries. Users can explicitly select other collections from the national libraries of Austria, Croatia, Denmark, Estonia, Finland, France, Germany, Italy-Florence, Italy-Rome, Latvia, Netherlands, Portugal, Serbia, Slovenia, Switzerland, United Kingdom.

The collections in Online books, images, maps, music, etc. can also be searched separately.

Metadata
The TEL metadata group started with the objective of creating a TEL data model capable of assuring integrated access to inhomogeneous resources. The following decisions were taken: Metadata should be encoded in XML; the *Library Application Profile* (DC-Lib) should be adopted as starting point for TEL data model; when the Library Application Profile is not sufficient TEL will create its own *TEL Application Profile*. The TEL Application Profile should consist of terms from Dublin Core plus qualified terms from Dublin Core plus TEL terms plus a subset of Collection Description terms.

A metadata registry, i.e., an On-line XML file containing all the terms in the TEL profiles or under consideration should provide:

- Details of all the characteristics of the terms including: which Application Profile they are in; what their status is (accepted, proposed); translations of labels
- Using XSL style sheets to generate: a view of each Application Profile; structured information for data entry forms.[9]

Presently, the TEL Advanced Search function offers specifying queries on the following fields: Author, Title, Subject, Type, Format, Language, ISBN, ISSN.

5.2.2 User

According to its mission, TEL offers anyone with an interest a route to access European cultural resources. No user registration exists. Research functions are freely available, delivery may be priced.

5.2.3 Functionality

The main functions of TEL are searching for objects and/or for information on National Libraries and National Libraries' collections.

Search for information objects
The portal allows users to access i) a centralized collection, named "The European Library Harvest" whose content is everything harvested by The European Library from National Libraries and held in a central index. As officially stated, this duplicates collections held under the libraries themselves, however it has the advantage of creating a general search under one target for the National Library Collections of the partner libraries; ii) the single collections of each partner library.

The User Interface offers simple and advanced search capabilities on the TEL default list of collections or on one or more collections to be selected by the user from the TEL information space. Search results are not merged nor ordered, they are first presented in a brief format and, when selected, they are returned with the following metadata: Title, Author, Type,

[9] Woldering, B., 2004, Presentation on TEL, Online Information and Education Conference 2004, Bangkok.

Language, Publisher, Date, Rights, Identifier. Automatic research expansion can be activated on Title, Author and Publisher. Possible linking service services can be found (for example, (paper) document delivering). The digitized version of the document, when present, can be seen online. Collection descriptions can be discovered like any other object.

Searching for information on collections and/or Libraries
Users are assisted in selecting the collection they are interested in by the following functions: search collection descriptions, browsing all the collection, or browsing collections by subject.

Each participating National Library can be selected to get information on its History, Collections, Access & Opening hours, Online services.

5.2.4 Architecture

TEL architecture[10] [52] is composed by a PORTAL that, through the SRU protocol, provides distributed searching of national collections accessible through Z39.50 and a searching in a central index that comprises records from other collections harvested by the OAI-PMH protocol. The development process of such architecture is depicted in Figure 9.

The TEL Portal runs in a standard browser using JavaScript and XSLT. A set of service and collection descriptions encoded in XML are loaded into the browser either from a URL or from a local file. An XSLT style sheet presents the collection descriptions to the user and the user selects the services or collections required for the search.

[10] The architecture supporting the TEL digital library has been modified since the beginning of the TEL project. Here we described the architecture reported in [52].

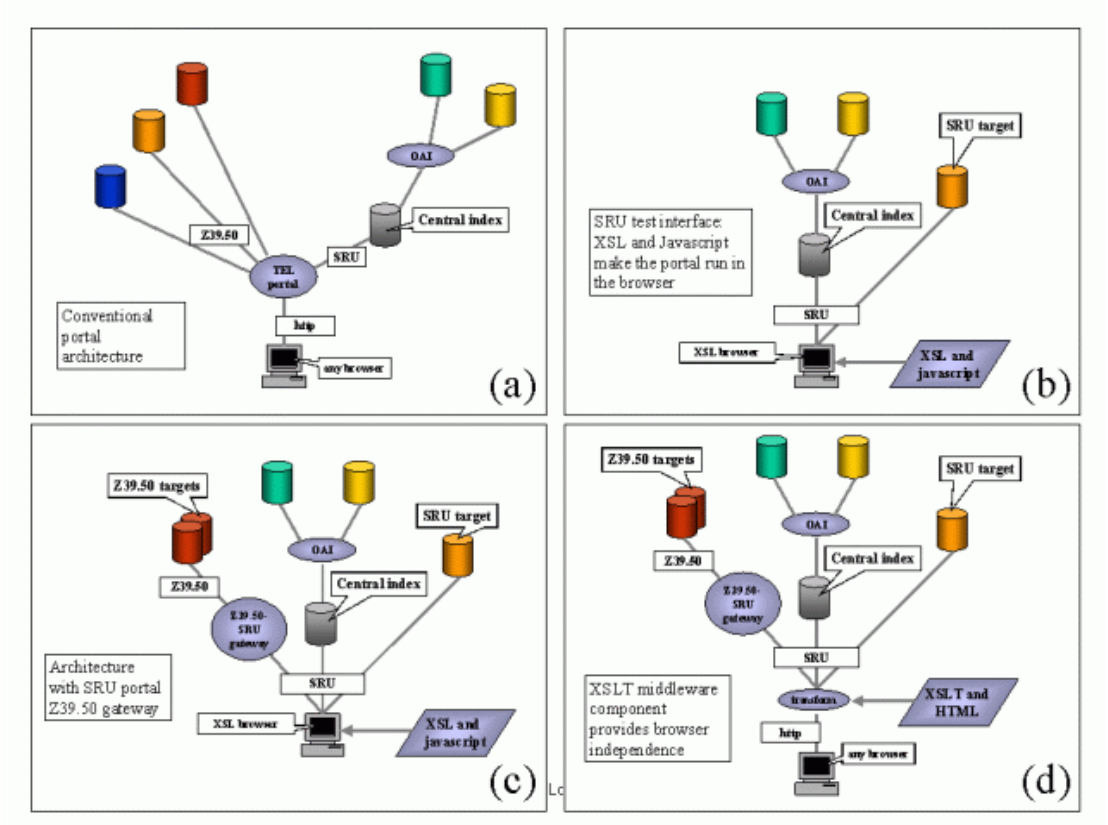

Figure 9. Development of the TEL portal architecture

The search query is defined using CQL (Common Query Language) [CQL]. CQL may vary from simply searching for one or more words to complex Boolean expressions using different index sets.

The Z39.50-SRU gateway allows the adoption of the SRU protocol by TEL partners whose library systems can only support Z39.50. To do this, the gateway has to perform a number of transformations. For example, it will perform the necessary transformation between the SRU search query language and the Z39.50 query accepted by the target. This includes character set processing and matching with appropriate indexes. Transformations also take place on output, (for example, conversion of the record format from MARC to the TEL metadata profile and/or a character set conversion from MARC8 to Unicode). The implementation of the gateway is straightforward and the source code and information about installation can be downloaded from the British Library.

Presently, however, the official TEL website reports that " During the project it became clear that Gabriel, the current website of the European national libraries, Gabriel (GAteway and BRIdge to Europe's National Libraries) would be integrated into the new European Library website. The Gabriel website was integrated into The European Library in the summer of 2005.

> "Besides practical information about the National Libraries (access and opening hours, functions, history etc.) Gabriel provides information about for instance the collections, the online public access catalogues (OPACs) and specialised webservices of each library. Furthermore Gabriel offers a central search engine

with which the websites of all National Libraries can be searched simultaneously."[11]

The objectives of the Gabriel service are to:

- to provide information on the World Wide Web about National Libraries in a uniform way in several languages;
- to provide convenient online links to sources of information about their services and collections;
- to give access to all their online services where appropriate;
- to be a bulletin board with news items about the National Libraries
- to give access to all the WWW servers of the National Libraries through a single search service;
- to build collaborative links between European National Libraries in the networking field;

The first version of Gabriel was launched in 1995[12]. After six years the original Gabriel version became outdated. In 2001-2002 Gabriel has undergone a radical refurbishment both in appearance and functionality. A major improvement is the implementation of a central database holding all vital information about all National Libraries. Each library can view - and if necessary - edit its own data at any time, thus making the information on Gabriel always up-to-date.

5.3 NSDL

The National Science Digital Library [33] (NSDL) was created by the National Science Foundation to provide organized access to high quality resources and tools that support innovations in teaching and learning at all levels of science, technology, engineering, and mathematics education. Because of the heterogeneous community of participants and technologies, the library is being developed with two key notions: a *spectrum of interoperability* and *one library, many portals*.

As reported in its 2005 Annual Report, the NSDL Program has received $100,694,398 in support of 193 awards in 33 states to digital library projects since 2000.

5.3.1 Information Space

The NSDL information space consists of 610 thematic collections[13]. These collections are organised in six topics, i.e. Education, Health, Mathematics, Science, Social studies, and Technology. In the NSDL project much effort is spent in creating such collections.
The information objects managed by the NSDL include images, video, audio, animations, software, datasets, besides text documents such as journal articles and lesson plans.

5.3.2 Users

For access the NSDL portal does not require either a user name or password and as a result no personalisation or user management is provided at the portal level. The management of users,

[11] http://www.kb.nl/hrd/netwerk/gab-en.html
[12] http://www.kb.nl/gabriel/
[13] As on March 2006.

as well as the enforcement of policies, is up to the content providers (see Access Management Service).

5.3.3 Functionality

The main functionality provided by the NSDL portal is dedicated to the access of the information objects. In particular a simple keyword based search facility is provided enabling the users to express their information needs in a Google-like stile. In addition, selectors on the resource format (e.g. text, image, audio) and grade level (e.g. graduate, college, high school) are supported. In addition to the search functionality, three type of browse functionality are provided, i.e. browse by topic, browse by collection name, browse by subject via an interactive visual view.

Basic mechanisms for *Recommend resources for inclusion in NSDL* and *Contribute learning materials created by the users* are provided.
Focused views of the NSDL resources are provided via the NSDL projects that are in charge of offering audience-specific views of selected NSDL resources (these specialized views are also known as portals). Pathway audiences may be grouped by grade level, discipline, resource or data type, or some other designation.

5.3.4 Architecture

The NSDL Architecture, as depicted in Figure 10, consists of several components that interact via web service interfaces and protocols.

- NSDL Data Repository (NDR) – Represents the core of the whole NSDL, it is the central repository over which the various portals are built. Initially, the NSDL Repository stored only metadata ingested from participating projects, as well as metadata gathered from open-access Web resources. During the NSDL activity it became apparent that the repository needs to represent a more resource-centric view, including the need to support: (i) content, such as annotations, reviews, or information on structuring a set of resources in a lesson plan; (ii) explicit relationships among resources in the repository; and (iii) information about who or what organization provided a particular piece of information about a resource. A repository with such characteristics has been created using a Fedora digital object repository. The NDR represents resources as digital objects, and associates with them multiple metadata records from different sources. It represents the organizations and individuals that provide metadata or select resources, and relates them to the appropriate metadata and resources. Since Fedora is fundamentally a content repository, it can also represent content such as annotations or reviews. Finally, Fedora provides an RDF-based flexible relationship structure that supports arbitrary relationships among resources, for example relating all those that match a particular educational standard, or structuring the resources that are assembled into a lesson plan.

- Metadata Harvesting – Represents the component supporting the ingestion of metadata both via the OAI-PMH and via the NDR API.

- Search Service – Provides fundamental capabilities for locating resources and collections within the library. Search services allow any item represented in the repository to be found, but in reality, the metadata provided by collections vary dramatically in formats, quality, and comprehensiveness. To address this challenge,

the search service combines indexing of metadata with indexing of full text content acquired by using network protocols where the content is linked via the identifier in the metadata record and freely available. The underlying technology uses Jakarta Lucene search engine.

- Access Management Service – While many items in the library will be freely available and anonymous user access is permitted, access to some materials is restricted. The NSDL core access management system relies on the Shibboleth protocol [44] to distribute identity verification (authentication) and cohort membership (authorization) to the administrators of distinct communities of users. In other words, the user's "home" institution performs user identity and capability management. Federated communities performing user identity and capability management can easily tie-in to this system using standard protocols (e.g. Kerberos and LDAP).

- Main User Interface – NSDL provides access to collections and services via portals. The main portal is represented by nsdl.org built by using PHP, MySQL, and the Internet Scout Portal Toolkit[14].

[14] http://scout.wisc.edu/Projects/SPT/

Figure 10. NSDL Architecture

- Archive Service – Provides persistent archive services to retrieve materials represented in the NDR from public sites (with a crawl depth of 10 levels) and archive both metadata and content for future retrieval. Web materials that are deleted or "lost" will be recoverable through archive services, and users will be provided options in NSDL search results to retrieve prior versions of resources.

- iVia – Provides focused crawling, an automated mechanism for including resources in the NSDL collection. In the near-term, this will allow including the resources in human expert-selected web sites, essentially "seeding" the crawler with the home pages of those web sites. iVia work also includes developing and deployment mechanisms for automatically generating basic Dublin Core records from textual resources. The iVia tool also provides automatic assignment of Library of Congress Classification to NSDL resources [36].

- User Help – Provides text based user help to the users of the main portal. In adjunction a virtual reference desk is provided via the AskNSDL functionality. This system aids

users in gaining assistance from one another, e.g., from more experienced colleagues who have expertise in using a specific collection or can answer discipline-specific pedagogy questions for example. This service is e-mail based and questions are distributed to a large group of experts.

5.4 Categorisation

The table below summarizes the characteristics of the systems supporting the three DLs analyzed in the previous sections.

	DARENet	TEL	NSDL
User			
User Identifier	No	No	No
User Profile	No	No	No
Role	No	No	No
Policy	No	No	No
Group	No	No	No
Information Space			
Information Object	OAI providers have no restrictions on the types, content, or format of the digital objects they export		
Information Object Identifier	Yes	Yes	Yes
Content			
o Metadata	Yes	Yes	Yes
o Text	Yes	Yes	Yes
o Image	Yes	Yes	Yes
o Audio	Yes	Yes	Yes
o Video	Yes	Yes	Yes
o Composite	Yes	Yes	Yes
Version	No	Yes	No
Manifestation	The OAI Harvester locally stores textual objects (when possible) to enable full text search, otherwise an HTTP link to the objects in the original repository is available	Yes	
Annotation	No	No	No
Metadata	Yes		
o Descriptive Metadata Format	Yes	Yes	Yes
o Structural Metadata Format	No	No	No
o Administrative Metadata Format	No	No	Allowed.
o Preservation Metadata Format	No	No	Allowed.
Collection	No	Yes	Yes
Functionality			
Access	Searching the DL can be done through a web portal (DAREnet) in the style of search engines. By default, key-words search is performed into all metadata fields. However, the query is implicitly extended to full-text search for all textual objects locally available. Users may choose to query only one of the contributing repositories and to sort the result		

	either by relevance or date of publication.		
o Search	Yes	Yes	Yes
• Full Text	Yes	No	No
• Metadata	Yes	Yes	Yes
• Image	No	No	No
• Audio	No	No	No
• Video	No	No	No
• Speech	No	No	No
• Single-Object, Single-Feature	Yes	No	Yes
• Multi-Object, Multi-Feature	No	No	No
• Compound Document Match	No	No	No
• Predicates	No	No	No
• Query Expansion	No	No	No
o Cross-language	No	Yes	No
o Relevance Feedback	Digital objects in the results of a query are presented by title, authors, and an abstract, if it exists. Result refinement cannot be applied. However, by clicking on an author name it is possible to perform a query returning all related digital objects.	No	Yes
o Browse	No	Yes	Yes
o Visualize	See Relevance Feedback.	Yes	Yes
o Translate	No	No	No
Content Management	An OAI Harvester withdraws metadata from the repositories and, when possible, the original documents, so as to enable full-text search.		
o Submit		Yes	Yes[15]
o Update		Yes	No
o Annotate		No	No
o Review		No	No
DL Management	DL management is left to the local policies of individual repositories. Instead preservation services are provided to all repositories by the National Library E-Depot, a persistent repository. Individual repository can, at their wish, send copies of their digital objects to the E-Depot for long-term storage.		These functionality are provided in the back end of the portal.
o Annotate		No	No
o Update		Yes	No
o Withdraw		Yes	No
o Describe		Yes	No
o Disseminate		No	No
o Preserve	Yes	No	No
o User Management	According to DARE		

[15] It is possible to *Recommend resources for inclusion in NSDL* and *Contribute learning materials created by the users.*

	federation guidelines, participating repositories should guarantee local user access rights management. Digital object controlled access will allow for the construction of services with personalization features. Currently, no DARE service functionality takes users issues into account.		
• Registration	No	No	No
• Role Management	No	No	No
o Policy Management		Yes	No
Personalize	No[16]		No[17]
o Collection Management		No	
o Personalised access		No	
o Notification		No	
o Others			
Enabling	In order to participate to the federation, repositories must provide an OAI data provider service, disseminating metadata according to a qualified DC format.		The enabling framework is represented by the OAI-PMH and NRD API with respect to the metadata harvesting and by the Shibboleth protocol for authentication and authorization issues.
o Authentication	No	No	
o Authorization	No	No	
o Encryption	No	No	No
o Subscription	Static subscription of an OAI data provider to the centralized OAI harvester	No	No
o Notification	No	No	No
o Process composition	No	No	No
Others			
Quality of Service			
Security	No	No	
Economics	No	Yes	
Availability	Yes[18]	Yes	
Reliability		Yes	
Performance			
Response time	Yes[19]	No	
Security	No		
Authentication	No	No	
Integrity	No	No	
Data Protection	No	No	
Message Protection	No	No	
Robustness		No	
Capacity		No	
Load balancing	No	No	
Recoverability		No	

[16] At the moment, no service provides personalization tools.

[17] The portal can provide focused views of the NSDL content via the Pathway project, users are not enabled to customise the information space.

[18] Search functionality is always available from the DARElnet Web site.

[19] Search engine compatible response time.

Messaging		No	
Consistency		No	
Scalability	No[20]	No	
Architecture			
Characteristics.	OAI Harvesting of independent repositories, federated by a common qualified DC metadata set.	Portal and distributed protocol based.	The architecture follows both the service oriented approach and the open archive approach where there the collections can be considered content providers while the NSDL portal is the service provider.

[20] Due to the static centralized approach, scalability depends on the storage and processing resources manually installed at the harvester site.

6 Digital Library Systems

Digital Library Systems are software systems providing digital library functionality on a set of information objects. They are different from repository systems because their goal is to provide a broader range of functionality than is provided by repository systems. They are of a general purpose nature, have been implemented to fulfill the requirements of particular types of DL building user communities. In this section we examine OpenDLib, OSIRIS/ISIS, and Daffodil, three digital library systems developed by DELOS partner institutions and discuss their capabilities and facilities for building digital libraries.

6.1 OpenDLib

OpenDLib [34] is a software toolkit developed at ISTI-CNR that can be used to easily create a digital library, according to the requirements of a given user community. This can be done by first instantiating the software appropriately and then either loading or harvesting the content to be managed. The toolkit consists of a federation of services that implement the digital library functionality making few assumptions on the nature of the information objects to be stored and disseminated. Using the toolkit it is possible to handle a wide variety of information object types with different formats, media and structures. In particular, the toolkit can manage new types of information objects that have no physical counterpart, such as composite information objects consisting of slides, video and audio recordings of lectures, seminars or courses. OpenDLib can also maintain multiple editions, versions, and manifestations of the same information object, each described by one or more metadata records in different formats. The information objects can then be organized in a set of virtual collections, each characterized by its own access policies. Authorized people can dynamically define new collections by specifying appropriate definition, can share private content with other selected users, and can access the digital library management functionality. The basic release of OpenDLib provides services to support the submission, description, indexing, search, browsing, retrieval, access, preservation and visualization of information objects.

From a deployment point of view, the entire set of services can be managed by a single or by a multitude of organizations that collaborate on the maintenance of the shared digital library, each according to their own computational and human resources. Moreover, the toolkit has been designed to easily support the plug-in of other services, when requested to meet particular and unpredictable needs.

6.1.1 User

OpenDLib maintains information about the users, groups, and communities. In particular, it regulates the access to the digital library via the user name and password mechanism and stores the user credentials using cryptographic techniques allowing user requests to be authenticated and authorized. Moreover, user and group profiles to be managed can be customized with respect to the information to be maintained.
With respect to policies and role, OpenDLib does not provide explicit support for roles and provide mechanisms for associating policies with users and groups directly. At present, the system provides a standard configuration through which it is possible to express and manage policies on: (*i*) groups, collections, information objects, and services resources; (*ii*) *create*, *edit*, *delete*, *access*, and *manage* actions; (*iii*) user and group actors. For example, this model permits to establish that a specific user that becomes its administrator manages the Collection Service. Then this administrator could: (*i*) grant a set of users permissions to create only private collections; (*ii*) grant another set of users rights to create public collections; (*iii*) grant

users of both sets permissions to edit and delete their own collections; and, finally, (*iv*) decide that the public collections are discoverable and accessible by either registered or unregistered users.

6.1.2 Information Space

OpenDLib supports an information space as composed by information objects compliant with a proprietary document model (DoMDL [9]) organised in collections. This flexible document model allows the system to deal with structured, multi-editions and multimedia objects that can be disseminated in multiple manifestation formats. According to this model, depicted in Figure 11, OpenDLib objects are modelled in terms of four entities, i.e. Document, Edition, View, and Manifestation. The *Document* entity represents the abstract object as distinct intellectual creation, capturing the more general aspects of it in very abstract terms. The *Edition* entity represents a specific expression of the distinct intellectual creation, thus being able to model an instance of the document along the time dimension. The *View* entity models the different ways in which a digital object can be organised, viewed and disseminated. Finally, the *Manifestation* entity corresponds to the physical format through which a document is disseminated. The model does not constrain the media types that can be stored as manifestations. View entities are further classified into metadata and content entities. The former is a view representing a document edition through its metadata, the latter represents the actual data that the object maintains. This is further divided in Body and Reference entities. The *Body* is a view of the object content either as a whole entity or as an aggregation of other views. Body views may be specialised by other views representing specialised perceptions of the same content. *Reference* views instead do not have explicit manifestations but represents links to views with already existing manifestations.

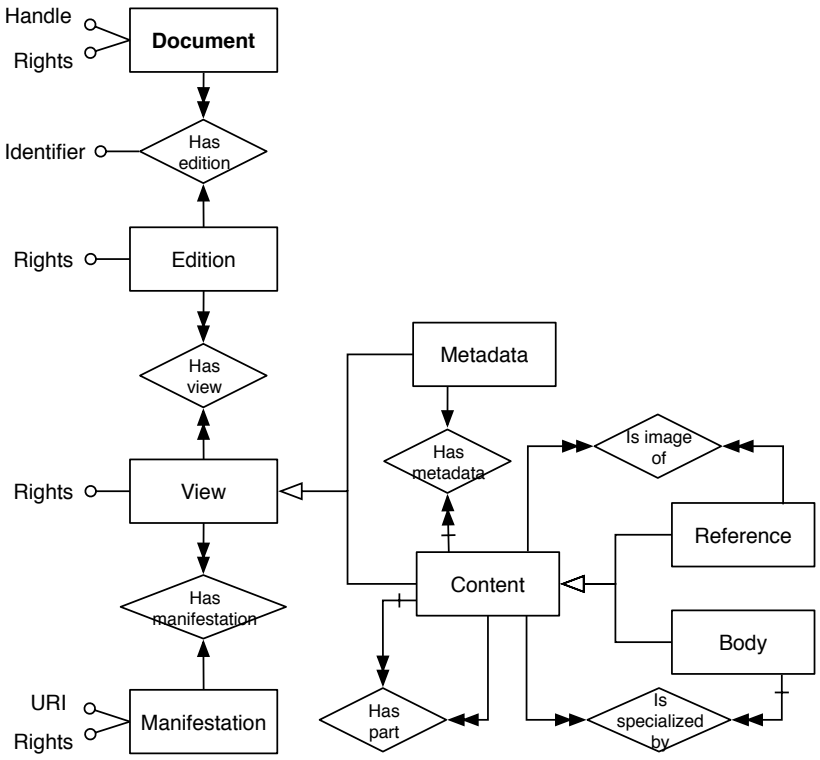

Figure 11. DoMDL - The Entity-Relationship model

The presence of the metadata view allows having information objects with multiple metadata associated with various parts of the objects. Moreover, the model does not constrain nor impose any metadata format, thus making OpenDLib capable to deal with any metadata the digital library community is interested in.
With respect to annotations, OpenDLib does not provide explicit functionality enabling users to annotate information objects even if its document model is powerful and capable enough to represent and store them.

6.1.3 Functionality

In OpenDLib all the aspects related to the presentation of the system to its users are provided via a *web-based user interface* that provides access to the system functionality.
With respect to the access class of functionality, OpenDLib supports two types of queries: *standard keyword based* and *advanced search* which accepts field based conditions that can be combined in complex logical expression. It is worth noting that the system is configurable with respect to the metadata fields to be indexed and thus on the fields that can be used to express user information needs. Moreover, the system supports a *relevance feedback search* which enables the users to mark as relevant some of the item constituting the result set of a search and to resubmit the query by expressing interest in document similar to those marked. The *browse* functionality allows the user to explore the digital library information space exhaustively. It is worth noting that the system is customizable with respect to the fields that can be browsed. Moreover, both the search and the browse functionality are implemented by relying on the collections mechanism and thus enabling users to have access to focused views of the whole information space. The *visualization* of the information objects is provided through two *visualization* paradigms that have been appropriately designed to take into account the complexity of the objects that are composed by multiple parts and whose structure can be defined according to the application framework. These paradigms are: *tab based*, which shows the parts of an information object in multiple pages of a single window by creating tabs; and a *window base*, which shows each part of an object as a page of a new window.
The OpenDLib *submission* functionality class comprises *submission*, *update*, and *review* functions on information objects compliant with DoMDL. It is worth noting that submitted objects reside in a private incoming area until an authorized user decides whether to publish them as digital library objects.

The *DL management* functionality class provides functionality allowing *publishing*, *updating*, and *withdrawing* digital library information objects. In particular, the system provides a management environment reporting the status of new submissions, of new editions of already available objects, of objects corrected, and of objects withdrawn and it provides facilities for dealing with this information.

With respect to the *personalization*, OpenDLib provides facilities for creating personal collections and for defining the user information space, i.e. for identifying the collections of the whole digital library information space a user is interested in. The access functionality, by default, acts on the personal user space.

6.1.4 Quality of Service

OpenDLib is a system that once configured can provide different levels of quality of service. In particular, being a federation of services where the services can be replicated and distributed, the system can be tuned to fulfil the specific requirement of the communities that

create their own DL. Clearly, this consideration applies to those aspects of the quality of service whose level can be improved via replication and distribution, i.e. availability, reliability, performance, load balancing, and scalability.

6.1.5 Architecture

The OpenDLib architecture depicted in Figure 12 consists of an open and networked federation of cooperating services. This architecture has been explicitly designed to support plug-and-play expansions. The OpenDLib federation is composed by the following services:

- *Manager:* maintains and continually updates a picture of the status of the DL service federation and disseminates it on request to all the other services;
- *Registry:* maintains information about the users and group communities;
- *Repository:* stores and disseminates documents that conform to the DoMDL document model;
- *Collection Service:* mediates between the virtual dynamic organization of the content space, built according to the requirements of the DL community of users, and the concrete organization into basic collections of documents hold by publishing institutions;
- *OAI Harvester:* gathers the content published by OAI-PMH compliant archives;
- *Library Management:* supports the submission, withdrawal, and replacement of documents through a complete review workflow;
- *OAI Publisher:* provides the content of the OpenDLib DL through the OAI-PMH protocol;
- *User Interface:* mediates between human actions and all the OpenDLib services. As result of their search or browse operations, users obtain a set of results pages with the list of information objects that satisfy their requests. The User Interface provides multiple and customisable ways to visualise these objects;
- *Query Mediator:* dispatches queries to index service instances, according to availability and replica priorities;
- *Browse:* supports the construction of indexes to browse the entire library content. The Browse function is parametric with respect to the metadata formats, to the set of fields to be browsed, and to the set of formats for result sets;
- *Index:* accepts queries and returns information objects matching those queries. The Index is parametric with respect to the metadata formats, to the set of indexed fields, to the set of result sets formats and the language of the terms. It offers different search options: free text or advanced (with fields selected from a variety of configurable metadata formats); single or cross-language; with or without relevance feedback.

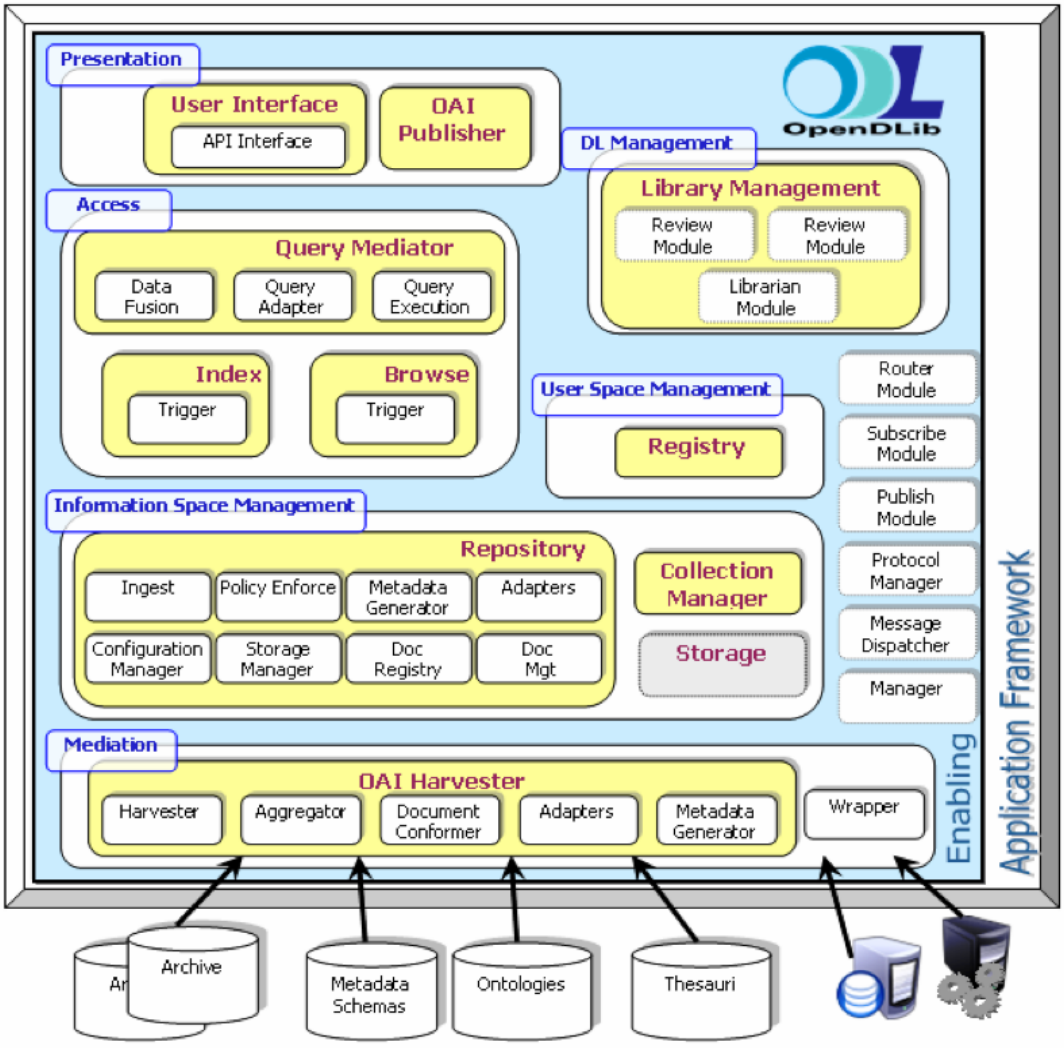

Figure 12. OpenDLib Architecture

A HTTP-based communication protocol, named OpenDLib Protocol (OLP), has been designed in order to regulate the communication among the services. This protocol imposes a set of rules governing how information is exchanged in the system. These rules must be satisfied both by the consumer and by the producer of the information.

6.2 OSIRIS/ISIS

OSIRIS (Open Service Infrastructure for Reliable and Integrated process Support) [40][41] is a platform that allows combining different distributed services into processes. The OSIRIS platform itself does not provide any application functionality but, by combining specialized application services, supports the definition and reliable execution of dedicated processes (this is also known as “programming-in-the-large”). ISIS stands for Interactice SImilarity Search and is an application for information retrieval in multimedia collections built at ETH Zürich [32]. It supports content-based retrieval of images, audio and video content, and the combination of any of these media types with sophisticated text retrieval [45].

6.2.1 Information Space

ISIS is efficiently searching and maintaining a collection of more than 600.000 images used within the ETHWorld project, the virtual campus of ETH Zürich. The images have been extracted from websites of the university and all its institutes. OSIRIS/ISIS operates on the following collections:

- ETHWorld: 625'000 images extracted from ETH websites plus corresponding textual information,
- ISIS: 53'837 images plus corresponding textual information,
- ISIS Video: 1'200 video sequences from five movies plus gathered textual meta information (cast, taglines, subtitles, keywords...),
- ISIS Audio: 1'185 MP3 music files plus gathered textual meta information (artist, title, album, lyrics…),
- ISIS Med: 50'143 medical images plus textual annotations.

6.2.2 Functionality

A sample query process (including user feedback) consists of the steps *Query Reformulation* (based on relevance feedback the user has issued), *Query Execution* (index access), and *Result Filtering* (which may again take user feedback into account). In Figure 13, this process is shown in the design view of the O'GRAPE tool.

Any content-based retrieval system is commonly exposed to heavy load under two distinct circumstances:

- Content-based queries, e.g., for similar images, are based on comparison of features of the object like colour histograms or texture. Because these features form high-dimensional retrieval spaces, determining the similarity for ranking the results is computationally expensive. One approach to reduce the system load would use efficient data structures as indexes, e.g., as described in [55]. Another approach would replicate data on several nodes to serve more requests in parallel, employ load-balancing, or try to handle parts of the request on several nodes [7]. ISIS follows both approaches.

While replication helps to cope with query load, it increases complexity of modifying a collection by inserting, deleting, or updating objects since the updates of all indexes have to be coordinated to ensure consistency. In ISIS, this is done by appropriate system processes, i.e., processes that have been designed by system administrator and which run automatically to guarantee consistency over several replicas of the index. The extraction of features itself can be a time-consuming task, therefore monitoring constantly changing collections and providing access can be challenging as well. If the insertion of multimedia objects can be divided in several sub-tasks and those can be executed on different nodes while using an infrastructure ensuring correctness of the distributed execution, this can improve the performance significantly [56]. In case of a web document, the object will not only contain an image, but also some text surrounding this image on the page. Later on, this text is used to determine textual descriptions related to the image. Independent of the image context, the feature extraction service uses raw pixel information of the image. Finally, the store features service hands all derived object information over to a metadata service, which makes it available for indexing and search in a suitable way.

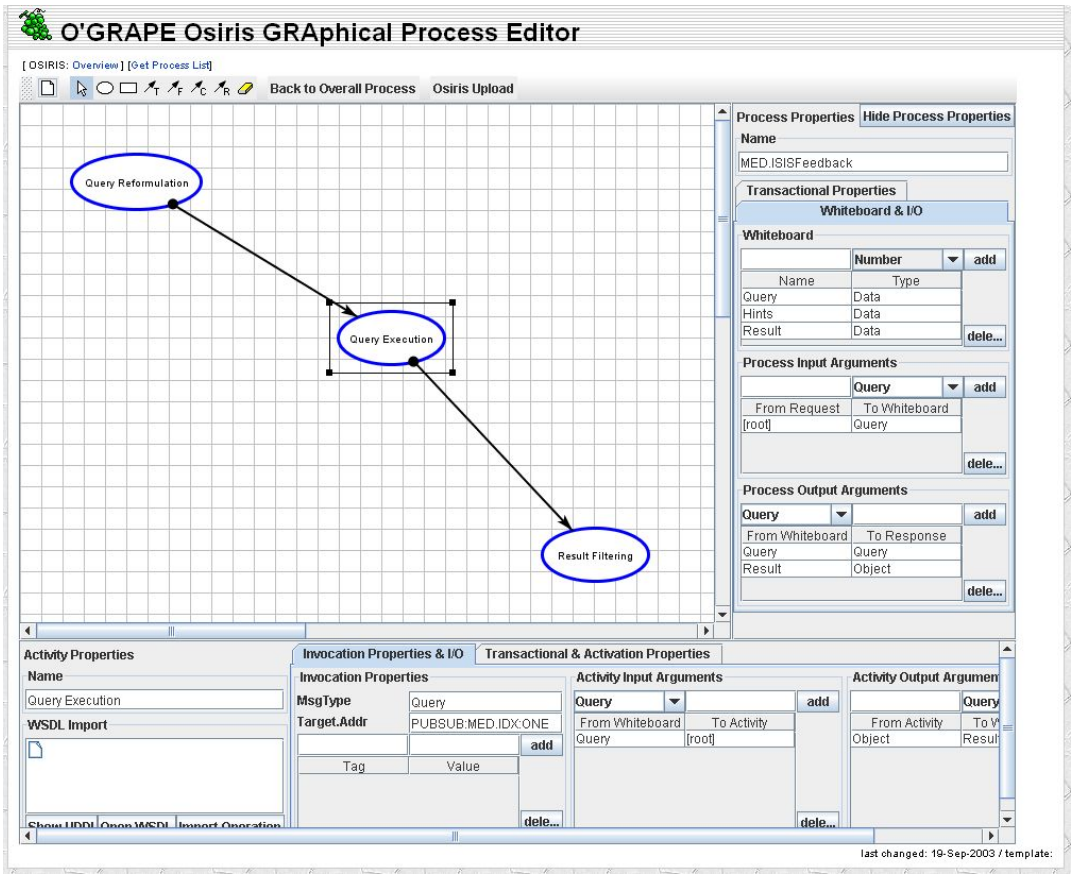

Figure 13. Design View of an ISIS Search Process (Encompassing Relevance Feedback) in O'GRAPE

Worth mentioning in this context, ISIS is efficiently searching and maintaining a collection of more than 600.000 images used within the ETHWorld project, the virtual campus of ETH Zürich. The images have been extracted from websites of the university and all its institutes.
The complete ISIS Digital Library application has been implemented completely as application services based on OSIRIS. A selection of ISIS application services comprises: Collection Management (Meta Database, Storage, Web Crawler), Search Interface and Query Processing (Session Management, Relevance Feedback, Indexing), Feature Extraction (Meta Database, Feature Extractor, Face Detector, Audio Feature Extractor, Hypertext Feature Extractor, Term Frequency Extractor).

6.2.3 Quality of Service

One of the main considerations in designing ISIS was to ensure high scalability and flexibility. Therefore, instead of implementing one monolithic application, ISIS consists of a set of specialized application services for similarity search which are combined by the OSIRIS middleware. The ISIS services can be easily distributed among several nodes in a network [54].

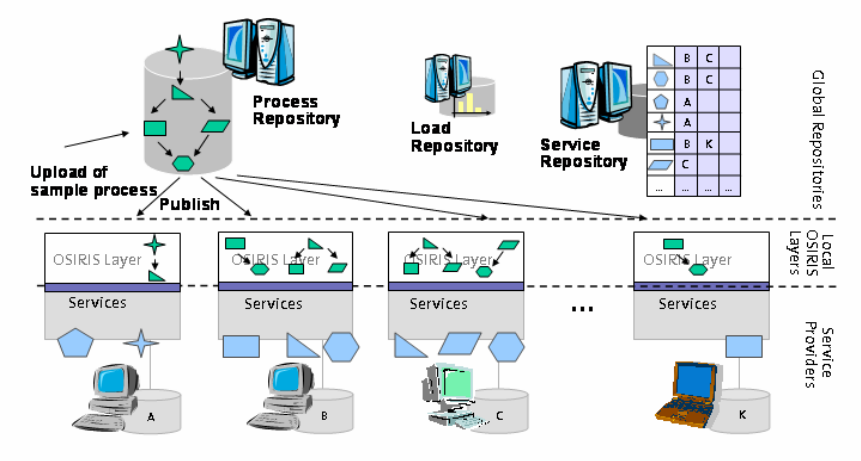

Figure 14. OSIRIS Architecture

Queries in ISIS are therefore implemented as processes. It is important to note that the process specification just contains the details of all application services it encompasses (WSDL description) and the orders within the process. The actual service providers where the service is invoked are determined at run-time. Therefore, information on the location of these providers is not part of the process description. Hence, each step of the process can be executed by any node providing the required service. After issuing the query a first time, a user can refine and re-issue her query.

6.2.4 Architecture

When different specialized digital library application services are made available to the OSIRIS platform, users can define and run powerful digital library processes by making use of these services. OSIRIS processes themselves are wrapped by a service interface. Therefore, a process can be invoked just like any other service (and used in other processes as well).
Following the model of transactional processes [42], processes in OSIRIS contain two orders on their constituent services: a (partial) *precedence order* specifies regular execution while the *precedence order* is defined for failure handling purposes (alternative executions). Data flow between services of a process can be defined independently of control flow. Activities in a process are invocations of application services. Ideally, the transactional behaviour of each application service is known. This transactional behaviour includes information on compensation (how can the effects of a service execution be semantically undone; this is needed for compensation purposes in case a failure in a process execution exists) and on whether a failed service can be re-invoked (retriability).

In addition to transactional guarantees and reliability, OSIRIS focuses on scalability of process execution. The decentralized peer-to-peer approach for process execution in OSIRIS, which is realized by sophisticated replication mechanisms for control flow dependencies, avoids any single point of failure during process execution and provides a high degree of scalability (see Figure 15). Peer-to-Peer process execution also incorporates sophisticated load balancing in order to distribute process load among available, suitable peers.

Finally, OSIRIS is equipped with the O'GRAPE (OSIRIS GRAphical Process Editor) [53] user interface for process definition. It allows for easy creation of process descriptions without programming skills. In addition, O'GRAPE supports the integration of existing application services by leveraging existing Web service standards like SOAP and WSDL.

ISIS stands for Interactice SImilarity Search and is an application for information retrieval in multimedia collections built at ETH Zürich [32]. It supports content-based retrieval of images, audio and video content, and the combination of any of these media types with sophisticated text retrieval [45].

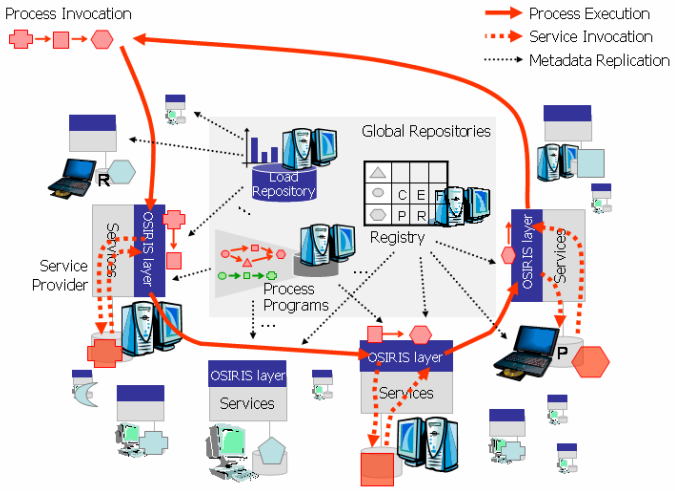

Figure 15. Overview of Peer-to-Peer Process Execution in OSIRIS

6.3 Daffodil

Daffodil [11] is a virtual DL targeted at strategic support of users during the information search process (see [43], [27], [20]). For searching, exploring, and managing DL objects, Daffodil provides information seeking patterns that can be customized by the user for searching over a federation of heterogeneous digital libraries. Searching with Daffodil makes a broad range of information sources easily accessible and enables quick access to a rich information space.

6.3.1 Information Space

Most information objects in Daffodil are descriptive metadata, which are described using the BibTeX scheme, formatted in XML. Objects within Daffodil can be queries, metadata, fulltext, authors, terms, journals and conferences with their aggregations or URLs.
The metadata is extracted from the connected digital library with a wrapper toolkit. The toolkit can extract and transform data in any format. The currently connected digital libraries are from the computer science area, such as ACM, CiteSeer, Achilles, DBLP, SpringerLink, Scirus, HCIBib, LeaBib, CompuScience. The information object identification within Daffodil is the title of the information object, but can be extended to include more attributes.

For including a new source into the Daffodil system, there are two possibilities:

- If the source data is available, then it is converted into XML and indexed with Lucene, so that it can be accessed via the standard Lucene wrapper of Daffodil
- Alternatively, a wrapper for the corresponding digital library system has to be implemented (which can be performed quickly using Daffodil's own wrapper toolkit), which transforms queries and result lists.

6.3.2 Users

The current Daffodil system aims at support information searching by students and researchers in all domains. Everybody who installs the system gets full access to the complete

functionality, using the system via the visitor account. In order to get personalized access (e.g. a personal library or collaboration support), users must register.

6.3.3 Functionality

DAFFODIL combines browsing and searching strategies in a natural way. The system supports collaborative search and provides the user with awareness, i.e., it provides information of new or changed objects related to previous searches. Users are free in choosing a search strategy, but the system assists them by providing easy access to well-known search tactics or stratagems, and in helping them to combine these for creating a comprehensive search plan. Based on empirical observations of the information seeking behaviour of experienced library users, Bates [3][4] identified a number of successful tactics for the information search. In [5] tactics referring to monitoring, file structure, search formulation, term selection, and ideas are described.

The graphical client, depicted in Figure 16, combines a set of high-level search activities as integrated tools according to the WOB model for user interface design. The current Daffodil prototype for the domain of computer science provides a variety of tools and functions, which are described in the following.

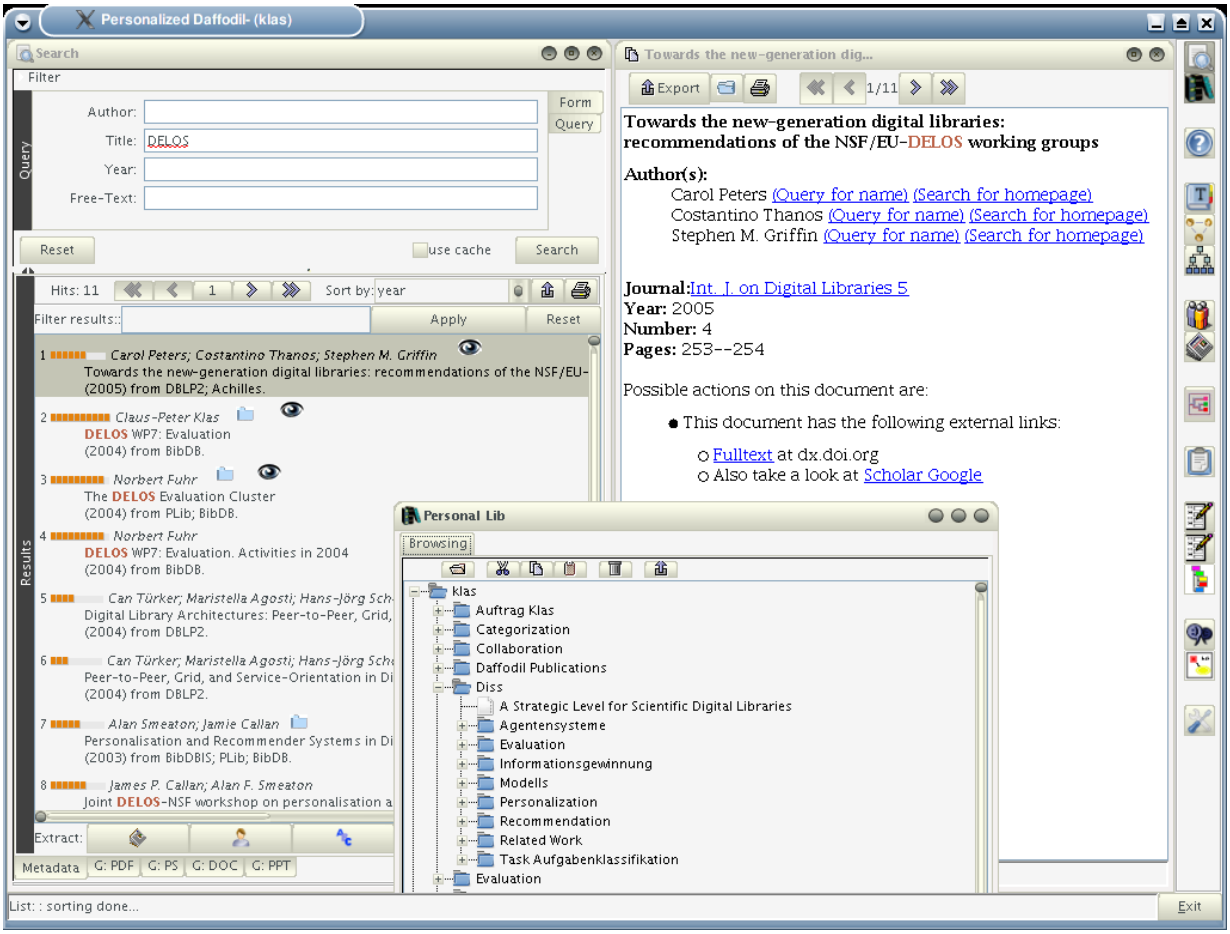

Figure 16. Daffodil desktop

Search Tool

The search tool is the usual starting point for searching within DAFFODIL. It provides a form-based interface for formulating queries to the federated digital libraries in a uniform way, and allows specification of the search domain by selecting some or all of the available libraries.

Wrappers map the uniform queries onto the query languages of the information providers. Results are merged and presented to the user (along with a paraphrase of the submitted query) in a homogeneous way for viewing and navigation. Unintrusive icons are used to mark documents that have previously been seen, stored or that have been interacted with in other ways (as seen in Figure 16). On the list of results, feature extraction can be used to get commonly occurring terms, authors, journal or conference titles.

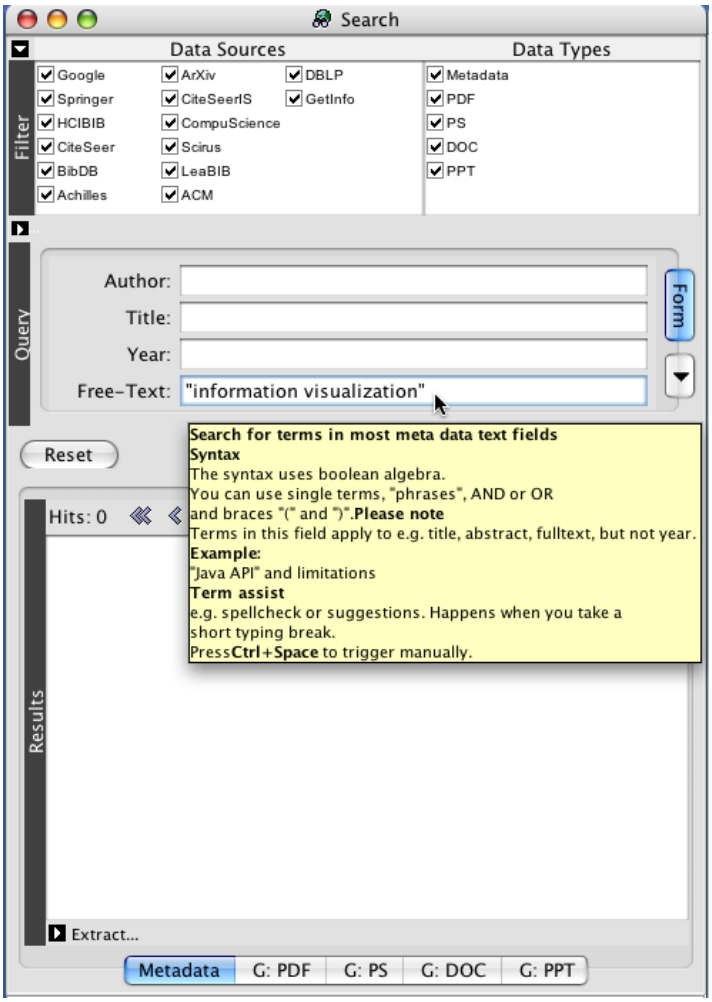

Figure 17. Search tool

Proactive query formulation

The search tool supports query formulation with pro-active functionality, like checking for spelling errors and overconstrained queries, e.g. a query like „year=2001 and year=2002“.

The related terms service suggests additional/alternative query terms to the user. These term hints are indicated by blue curly lines, and mouse clicking opens a popup list at the term, as depicted in Figure 18.

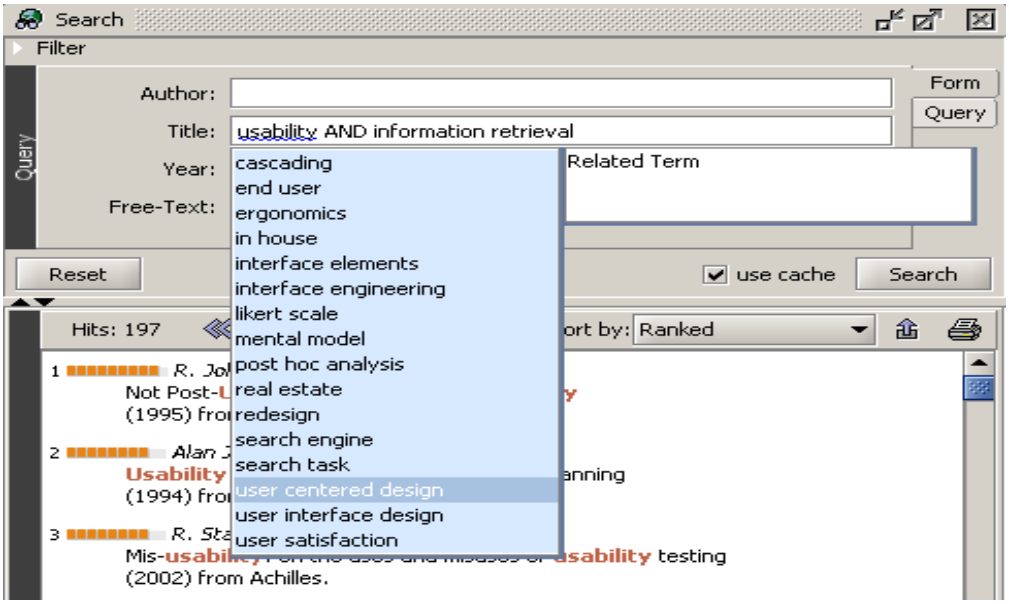

Figure 18. Related terms

Personal library

Queries, results, fulltexts, terms, authors, conferences or journals can be stored in the personal library, where they are saved for accessing them beyond the context of the current search. With personal folders users can structure their results, and can build a personal archive of interesting documents and other objects. For using search results outside of the DAFFODIL system, export functions are available. Group folders with the possibility for annotations on objects, including previous annotations, provide support for collaborative information access. Based on the personal folders of a user, the system provides awareness and recommendations:

- Awareness comes in two forms. In group folders the system highlights new objects or annotations added by other users. In addition, a search profile service can be enabled for queries as well as for authors, journals or conferences, in order to inform the user when new publications for these searches are available.
- Recommendation compares the personal library folders of pairs of users; if they are sufficient similar, items filed by only one of the two users are suggested to the other.

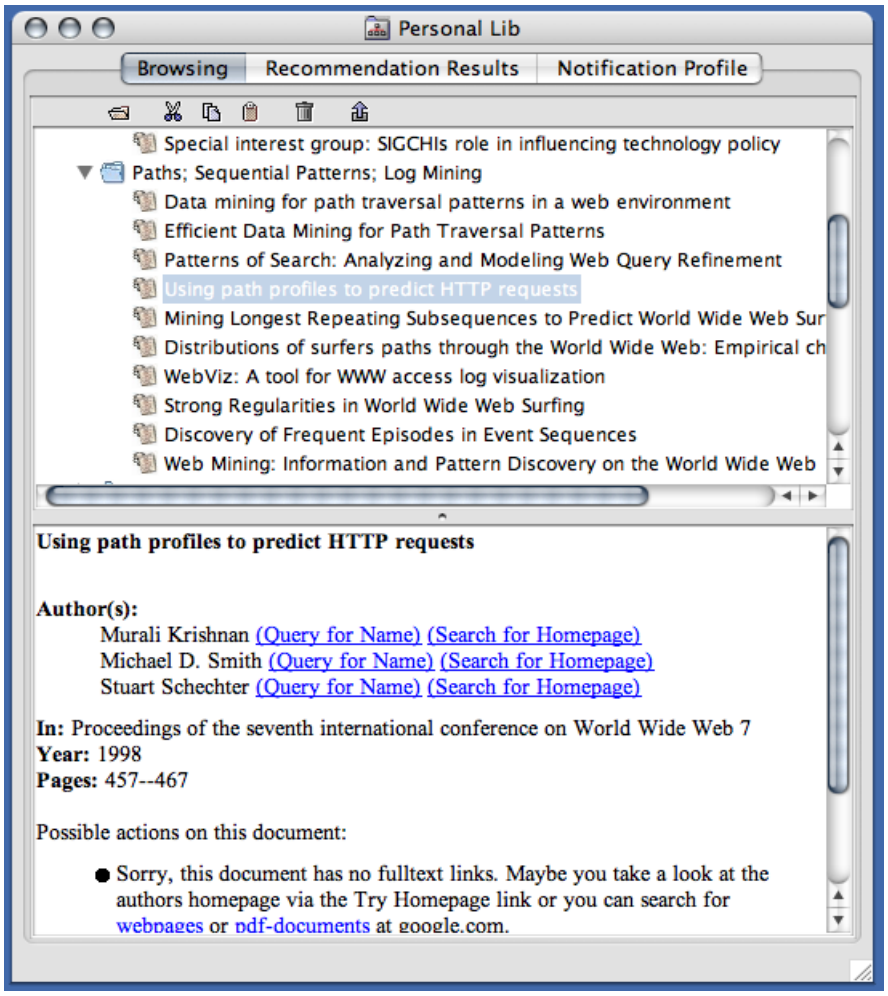

Figure 19. Personal library

Journals and Conferences

For browsing and searching in journal volumes and conference proceedings, the journal and conference tools (Figure 21) are available. Users can search for titles of scientific journals or conferences to browse within the results—often with direct access to metadata on articles or even links to fulltexts.

These browsers can be used as a starting point or an intermediary step in a larger search plan. Explicit links in the detail views of search results point to the journal or conference proceeding where a document was published. Activating them will open the respective tool at the corresponding journal or proceedings.

Author Network

Another stratagem directly supported by DAFFODIL is the author search. Starting from an author whose relevance to the search interest is known, it is possible to search for further publications by this author, or to exploit the co-author relationships of the author for deriving a collaboration network [20].

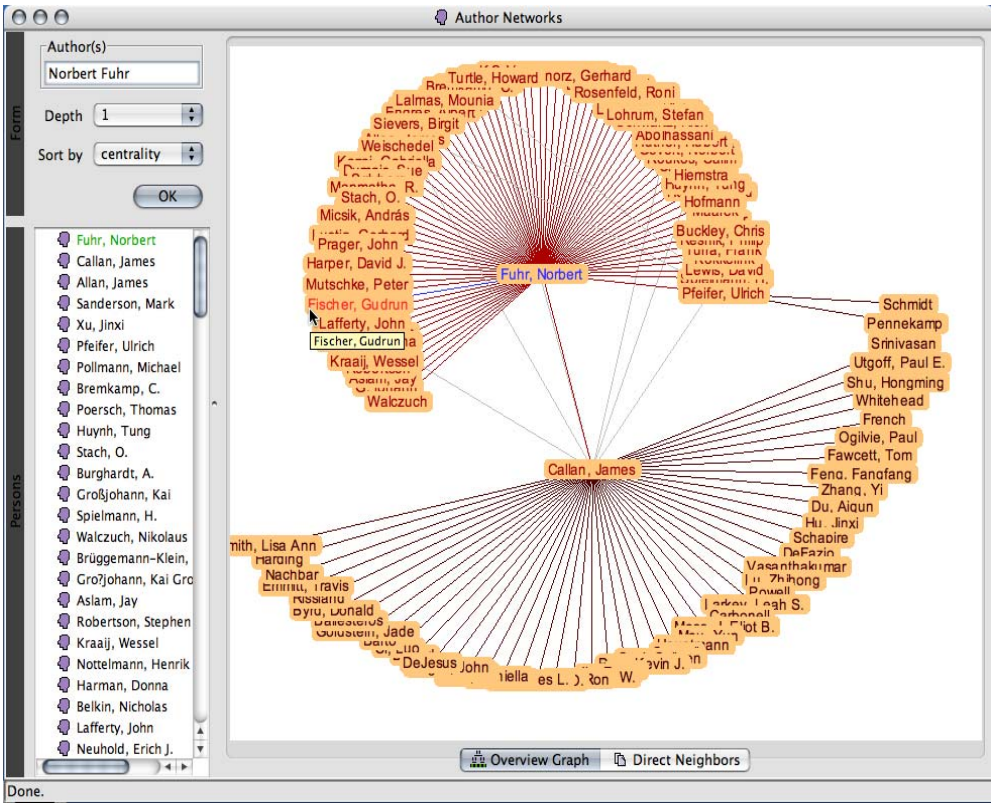

Figure 20. Coauthor graph

In such a network which can also be visualised in a relationship graph (Figure 20) it is easy to identify central authors and to find authors who often publish together. Since the authors are ranked by centrality, the user may also find that other authors than the original one are more central to the search topic, and can use these for further searches.

References and Citations

With the reference tool (Figure 21) it is possible to find referencing or referenced documents for an existing document. The tool can be activated intuitively by dragging and dropping a document from other tools. The results can be reused to find more references, used within another tool for further searches, or they can be stored in the personal library.

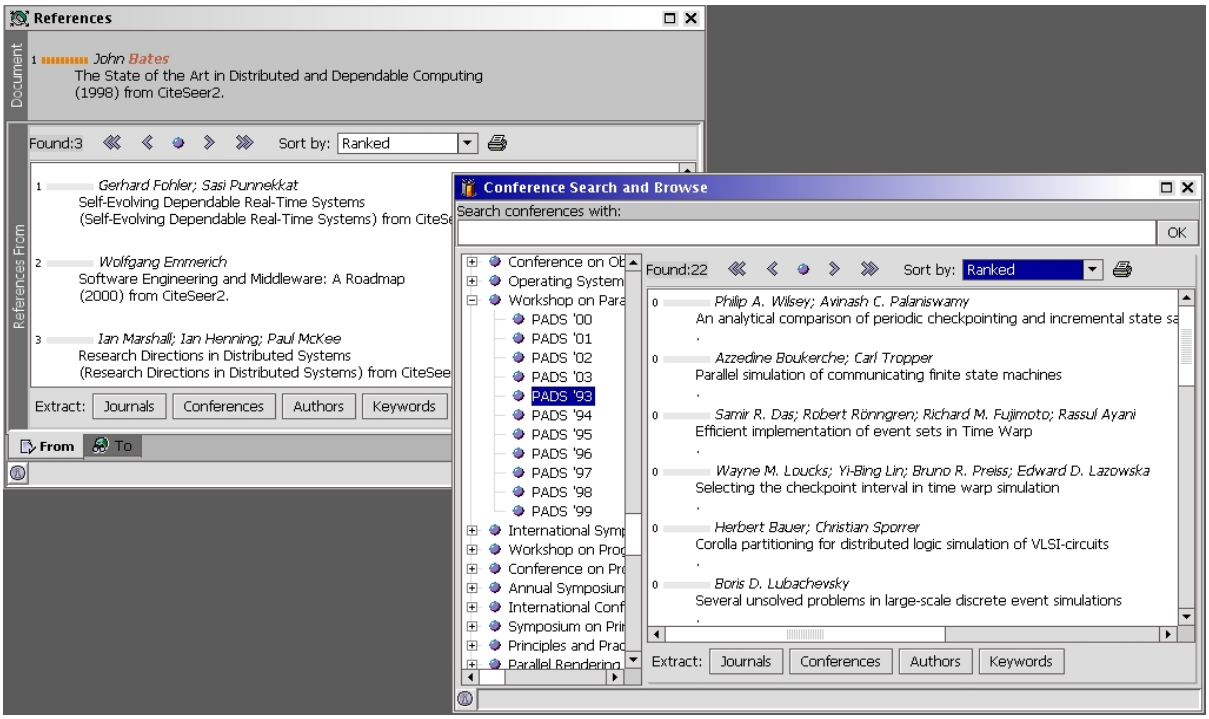

Figure 21. References (left) and Conferences (right)

Classification Browser

A *classification browser* that provides hierarchical, topic-driven access to the information space and enables browsing of classification schemes such as the ACM Computing Classification System.

Thesauri

The *thesaurus tool* can be used to get more general or more specific terms (hypernyms or hyponyms), or semantic definitions for a search term. Subject specific and web-based thesauri are used for finding related terms. The resulting terms can then easily be used in other tools for further queries.

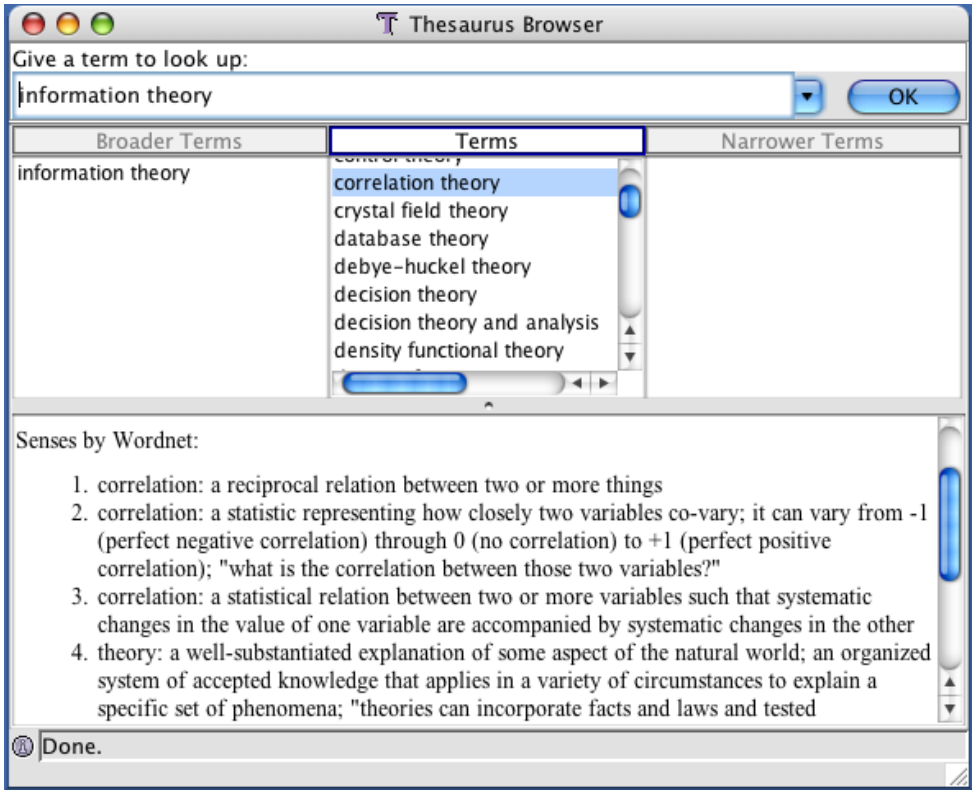

Figure 22. Thesaurus incl. WordNet

6.3.4 Architecture

The Daffodil architecture is mainly divided into two parts, the graphical user interface and the agent-based services.

The front-end client (Figure 23, red frame) is a Java application, which is deployed to the user via Java Webstart technology. The graphical user interface is the binding component between the Daffodil services and the user (see Figure 16). The main goals are easy user access, open communication and interfaces, modular modelling, adaptability, pro-activity and enabling higher-level search functions for the users. The underlying interface design follows the so-called WOB model [26], which is based on a tool metaphor. This model attempts to solve the inherent contradictions in the interface design process -- like that between flexible dialog control and conversational prompting -- using a set of co-ordinated ergonomic techniques. The general software ergonomic principles of the WOB model are:

- *Strict Object Orientation and Interpretability of Tools*: Strongly related functionality of the system is encapsulated in tools that are displayed as icons (not as menus). The tools open views, which are 'normal' dialog windows. Due to well-defined *dialog guidelines*, the chain of views a user is working on can be interpreted as a set of forms to be filled. In contrast, experienced users will prefer the tool view, which enables them to perform tasks more quickly; however, this view is cognitively more complex, and it is not required for interpretation. The user can manipulate objects on the surface in a direct manipulative manner. It is essential that consistency is guaranteed for the direction of the manipulation. Thus, the model requires object-on-object interaction style with a clear direction and semantics. The generally recommended interaction style is as follows: To apply a function on an item, the latter has to be dragged to a tool.

- *Dynamic Adaptivity*: The interface adapts its layout and content always to the actual state and context. This is mostly used for a reduction of complexity in non-trivial domains, like browsing simultaneously in several relevant hierarchies at once. For example, the user may set the relevant context by choosing a classification entry; when activating the journal catalogue as the next step, the journals are filtered according to the valid classification context, to reduce complexity.
- *Context Sensitive Permeability*: When known information is reusable in other contexts, it will automatically be reused.
- *Dialog Guidelines*: The views of the tools are functionally connected e.g. by means of action buttons, hypertext links or rules which are triggered by plan recognition. A tool can also open its view proactively if the user needs its function in a given situation.
- *Intelligent Components*: Tools and controls in the interface have access to context and state, in order to decide, if their function is valuable for the user. If applicable, they shall interact pro-actively with the user or the shared environment (the desktop), respectively.

Two principles of the model are information system-specific:

- *Status Display with Edit Mode*: The system shall always display a paraphrase of the current state for the user. It can be shown as a natural or formal language string or even by using some visual formalism (like a table). The most obvious use case is query formulation. With a form-based interface some aspects (e.g. boolean operators) are always hidden. Thus, Daffodil also displays the paraphrase (e.g. the formal query) in order to prevent the user from forgetting parts of his/her query (re-)formulation. It enables easy access to all aspects of the systems state, e.g. for iterative query formulation. Novice users can learn details from the paraphrase they would otherwise have to guess. They also can see if the system interprets their input in the way they expect it to.
- *Iterative Retrieval and Query Transformation*: Initial query formulations tend to be inadequate for the user's intentions, due to uncertainty or unconscious goals in the search process. Therefore applications shall simplify iterative query formulation for the user. This can be achieved e.g. by summarizing the query when displaying results. Furthermore, methods for automatic transformation have to be provided, in order to address the 'zero result' problem and to allow for the handling of semantic or syntactical heterogeneity of underlying data sources.

In accordance with the *Dialog~Guidelines* principle, a particular feature of Daffodil's interface is *Multi-Level-Hypertext* [18] interaction that allows for switching the level of information, e.g. from a document to the journal or to the authors institution or homepage. External links are provided for giving strategic support when Daffodil's services supply no results. In these cases, queries for external search engines like Google[21] or HPSearch (home page search service)[22] are generated dynamically and executed from within Daffodil. This results in an external browser being invoked, where interaction may continue.

The backend of the Daffodil architecture (Figure 23, blue frame) is based on an CORBA agent architecture. Each front-end tool (see section Functionality) is represented by one or more agent-based services that provide the actual functionality. Currently, more than 30 services and 15 wrapper agents are used by the Daffodil system to provide the services. For efficiency reasons, these services communicate via CORBA, but they can also be accessed via

[21] http://www.google.com

[22] http://hpsearch.uni-trier.de

SOAP. For communication between two agents or an agent and the user interface, XML messages are used. The agent framework itself is modelled very simplistically (for high performance) and provides parallel threads for each user. There are two different kind of agents, namely Internal Agents and External Agents. For communication, the first kind uses CORBA and the latter HTTP.

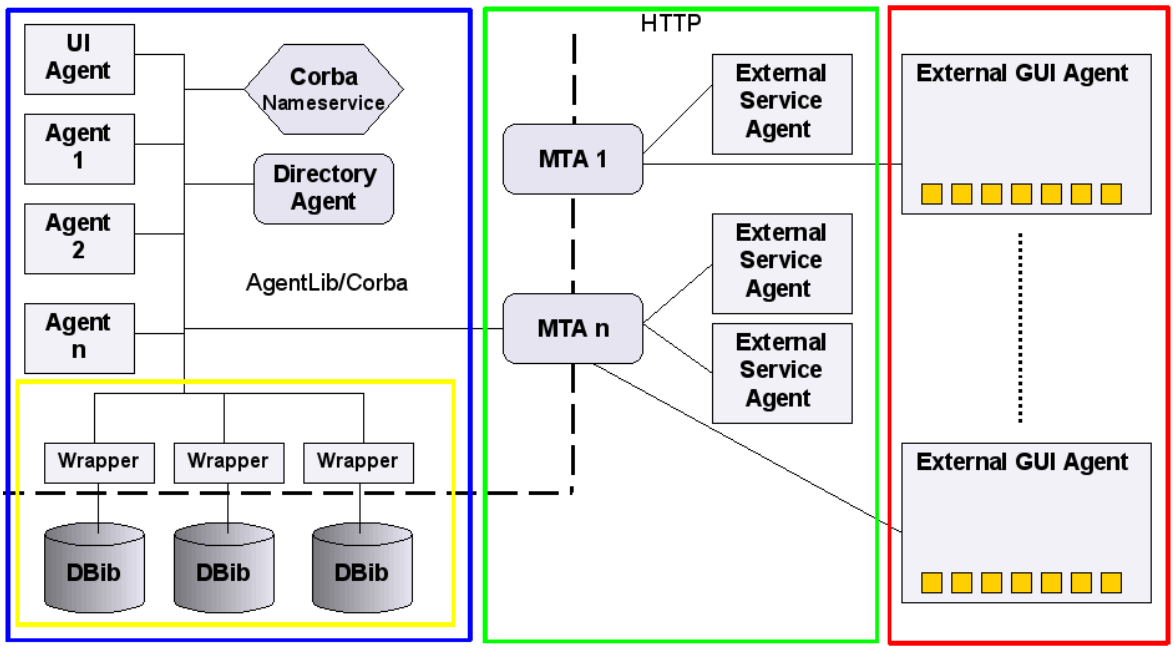

Figure 23. Daffodil architecture

DL management and expandability
Currently, management of digital libraries management is done manually. A new source can be easily added to the Daffodil system by providing a new wrapper for the specific source. A new functionality can be added by implementing a new backend service while reusing an existing or implementing a new graphical tool on the desktop. Due to the fact that the Daffodil framework is service-oriented and agent-based, it is very flexible and extensible.

6.4 Categorisation

The table below summarizes the features of the DL systems described above.

	OpenDLib	OSIRIS/ISIS	DAFFODIL
User			
User Identifier	Yes	Partial (login Username / Password)	Yes (login Username/Password)
User Profile	Customisable.		Customisable.
Role	No[23]	User / Admin	User
Policy	Yes[24]		
Group	Yes	Yes	Yes
Information Space			
Information Object	Compliant with DoMDL.		XML BibTex.
Information Object	Yes	Yes	Yes

[23] Not explicitly supported. Policies assigned per User/Group.

[24] Resources subject to policies are groups, collections, information objects, and services. Default actions are *create*, *edit*, *delete*, *access*, and *manage*.

Identifier			
Content			
o Metadata	Yes	Yes	Yes
o Text	Yes	Yes	Yes
o Image	Yes	Yes	No
o Audio	Yes	Yes	No
o Video	Yes	Yes	No
o Composite	Yes	Yes	No
Version	Internal.	Partial (No full version control, but date of last update)	
Manifestation	Multiple manifestations per object, multiple media formats.		
Annotation	No	No	Yes
Metadata			
o Descriptive Metadata Format	Yes[25]		
o Structural Metadata Format	Yes[26]	Yes	Yes, XML
o Administrative Metadata Format	Yes[27]		
o Preservation Metadata Format	No		
Collection	Yes	ETHWorld: 625,000 images extracted from ETH websites plus corresponding textual information, ISIS: 53,837 images plus corresponding textual information, ISIS Video: 1,200 video sequences from five movies plus gathered textual meta information (cast, taglines, subtitles, keywords...), ISIS Audio: 1,185 MP3 music files plus gathered textual meta information (artist, title, album, lyrics…), ISIS Med: 50,143 medical images plus textual annotations	computer science sources: Achilles, DBLP, LeaBib, Citeseer, ACM, HCIBib, SpringerLink, CompuScience, Scirus
Functionality			
Access			
o Search	Simple and advanced. Fields set customisable.	Yes (Text + Multimedia Part)	Yes (Structured text based)
• Full Text	Yes	Yes	
• Metadata	Yes	Yes	Yes
• Image	No	Yes	No
• Audio	No	Yes	No
• Video	No	Yes	No
• Speech	No		No
• Single-Object, Single-Feature	Yes	Yes	
• Multi-Object, Multi-Feature	No	Yes	
• Compound	Yes	Yes	

[25] Any descriptive metadata format can be managed.

[26] Used to represent DoMDL documents and thus compliant with a proprietary XML schema.

[27] Proprietary format.

Document Match			
• Predicates		Yes	
• Query Expansion	No		
o Cross-language	Yes	No	UTF-8
o Relevance Feedback	Yes	Yes	Implicit
o Browse	Yes. Fields set customisable.	Yes	Yes
o Visualize	Window based and tab based.	Yes (Ranked List, Fastmap)	Yes
o Translate	No	No	No
Content Management			Done by DL sources
o Submit	Yes	Yes	
o Update	Yes	Yes (automatic crawling)	
o Annotate	No		Yes
o Review	Yes		
DL Management			Done By adding a new wrapper.
o Annotate	No	No (but process-supported information enrichment)	
o Update	Yes	Yes	
o Withdraw	Yes	Yes	
o Describe	Yes	Yes	
o Disseminate	No		
o Preserve	No		
o User Management			
• Registration	Yes	Yes	
• Role Management	No	Yes	
o Policy Management		Yes	
Personalize			
o Collection Management		Yes	Yes
o Personalised access		Partially (supports different namespaces; for each namespace, a new configuration can be defined and the templates for displaying the results can be exchanged)	Yes
o Notification	No	Yes	Yes
o Others			Annotations
Enabling			
o Authentication	Login via user name and password	Yes (by username / password)	
o Authorization	Yes	Partially (by username / password; yet it is not possible to block one single service for one user or group)	
o Encryption	No		
o Subscription	No	Yes	
o Notification	No	Yes	
o Process composition	No	Yes	
Others			
Quality of Service			
Security			
Economics			
Availability	Yes	Yes (replication of services)	
Reliability	Yes	Yes (sophisticated failure handling)	
Performance	Yes	Yes	
Response time		Yes (good response time behaviour, even for highly complex similarity queries with several reference objects in	

		collections with > 600,000 documents, response time is acceptable)	
Security			
Authentication	Yes		
Integrity		Yes	
Data Protection			
Message Protection			
Robustness	Yes	Yes	
Capacity	Yes	Yes	
Load balancing	Yes	Yes	Yes
Recoverability		Yes	
Messaging			
Consistency		Yes	
Scalability	Yes	Yes	
Architecture			
Which kind of architecture? Which are the main components and their functionality? See the reference architecture for the description and the terminology	OpenDLib is a federation of services. These services cooperate (*i*) through the OLP protocol that regulates the exchange of information about services status and (*ii*) with the support of the Manager service that gathers these information and provides a picture of the whole federation, checks their consistency, and controls the flow of communication.	P2P Workflow Execution (OSIRIS)	Agent-based service oriented architecture, with backend- services and full application, java based graphical frontend.

7 Emerging Models

The changes in the technology landscape, the new opportunities in creating interoperability between heterogeneous types of data, the developments in processing and information representation mechanisms, and the expectation that service architectures will be central to the design of DLs means that new kinds of digital library models, such as those which are grid enabled, open new opportunities.

7.1 DILIGENT

The DIgital Library Infrastructure on Grid ENabled Technology [14](DILIGENT) an on-going EU funded project aims to deliver a test-bed DL infrastructure.[28] This infrastructure will support the on-demand creation and management of multiple transient DLs activated on the same set of shared resources. Resources covered here include information sources (i.e. repositories of accessible information), services (i.e. software tools which implement a specific functionality and whose descriptions, interfaces and bindings are defined and publicly available), and hosting nodes (i.e. networked entities that offer computing and storage capabilities and supply an environment for hosting information sources and services).

The inclusion of this infrastructure is motivated by two main observations:

(i) the role of digital library is evolving far beyond the connotation of the term "library" in favor of environments where groups of individuals, collaborating towards a common goal, can be authorized to access, discuss and enhance on-line shared information, and

(ii) there is growing demand for digital libraries able to support, and possibly improve, the way in which scientific research is conducted.

One of the main purposes of the DILIGENT infrastructure is to respond to the communication and collaboration needs of these virtual groups of individuals or, more generally, to the need of virtual research organizations, such as organizations composed of dynamic groups of individuals, institutions and resources distributed worldwide. In order to satisfy the requirements of these virtual organisations, all the services designed to implement the infrastructure must support generic, multi-type and multimedia information objects. From the technical point of view these services exploit a number of Grid technologies (EGEE [15] and gLite [21]) which provide an appropriate framework for transparently accessing and processing the type of content embedded in these new information objects.

The requirements for refining the DILIGENT infrastructure functionality have been expressed by two user communities that actively participate in all the phases of the project:

- ImpECt, an environmental e-Science domain community, and
- ARTE, a cultural heritage domain.

These communities have selected a number of operational scenarios that will be implemented as part of the project. In particular, the ImpECt community, composed of specific users (e.g. WWF, REMPEC, IOC/UNESCO, ESA), has identified a scenario that deals with the implementation of the Barcelona Convention whose main objectives relate to protecting the marine environment of the Mediterranean Sea against pollution. In order to support this operational scenario the ImpEC community requires DLs able to improve accessibility, interoperability and usability of environmental data, models, tools, algorithms and instruments

[28] DILIGENT is an acronyms that stands for "A DIgital Library Infrastructure on Grid Enabled Technology", http://www.diligentproject.org.

by integrating the distributed information sources with specialized information handling services.

The scenario identified by the ARTE community mainly refers to the activities conducted within the ARTE project, which is one of the projects managed by CTL-Center for the Data Processing of Texts and Images in the Literary Tradition, at Scuola Normale Superiore in Pisa. The ARTE community is engaged in the exploration of the broad zone of interaction between words and images that across different periods and genres has characterised the literary tradition. The ARTE scenario points out requirements for powerful collaboration tools and for demanding search facilities capable of identifying similar objects across different media.

Below we briefly summarize the requirements collected from the users of these two scenarios. The complete list of requirements is publicly available at that http://diligentproject.org/content/view/98/114/.

7.1.1 Information space

Both communities are interested in complex and multimedia information objects. In particular, it is mandatory the capability to represent and manage objects: (i) composed by multiple parts in different media, (ii) having multiple manifestations, (iii) containing parts of other objects, thus allowing the reuse of information.
The information space should allow: i) information objects be organized into collections, a collection being seen as an instrument to aggregate logically related objects starting from user specifications; ii) personal workspaces be available, so that users can keep their own information objects private (e.g., collections, computed results, contents of interest, etc.).
In particular:

- ImpECt is interested in representing and managing objects dynamically generated. This point is particularly innovative and it is related to the capability to associate a behavior to the objects. Thanks to this behavior objects may personalize their presentation at access time, e.g. dynamically generate a manifestation or dynamically update their content. Behavior allows realizing living documents, i.e. documents capable to be continuously updated and evolve accordingly to the status of their constituent objects.
- ARTE is interested in handling course material which comprises composite objects made by multiple media. A particular attention is dedicated to the handling of audio-video material. This material is already distributed by one of the ARTE content member organization (RAI-Radio Televisione Italiana) to schools as teaching material.

7.1.2 Users

Both communities require controlled access and use of data, information objects (or parts of them), tools and services.

7.1.3 Functionality

DL Management
Both ImpECt and ARTE scenarios envisage the building of a digital library as a simple aggregation process by which a user community could request a new DL through specifying a number of characterizing criteria on: the information space (e.g. publishing institutions,

subject of the content, types of information objects); the operations that manipulate the information space (e.g. type of search, tool for data analysis); the services for supporting the work of the users (e.g. type of personalized dissemination, type of collaboration); the quality of service (e.g. availability, response time), and on many other different aspects, like the maximum cost, lifetime, etc. In particular:

- *ImpECt* is interested in building digital libraries for supporting the production of environmental reports. This scenario is characterised by the following main characteristics: (ii) the participating users are spread worldwide; (iii) the reports, once defined, needs to be generated periodically on different pool of data; (iv) an important type of information object to be made accessible via the DL is represented by raw data; (v) the report contains advanced products that can assume various forms, e.g. raw data, maps, graphs, and can be reused for the production of other products.
- *ARTE*: ARTE members deem it impractical that people in the Humanities be able to create a DL, therefore they have expressed requirements concerning how they can define a DL and then request a technician (the ARTE DL Administrator) to create it. In particular, requirements are about i) how ARTE members can select known archives and services or discover resources in the DILIGENT infrastructure; ii) how they can suggest the inclusion of these resources in the infrastructure in the case that they are not already registered in DILIGENT. In this context, much attention is addressed to how virtual collections can be managed, as such collections become a key means to virtualize and personalize the DL environment that needs to be dynamic and changeable on demand to satisfy specific constraints.

Access

Advanced access functionality is of fundamental importance in order to support users in dealing with the new types of information objects envisaged. This functionality must offer the possibility to discover both whole complex objects and single parts of them, thus enabling the users to fruitfully identify the information they are interested in. Effort must be spent in identifying tools for easily specifying complex and precise queries on objects very different from the traditional textual documents or metadata records.

In particular:

- *ImpECt:* Advanced and computational intensive searches like those based on spatial and temporal query criteria constitute ImpECt's high desirable requirements, as they need searching for information objects related to a given region/period of time/topics, etc.
- *ARTE:* Searching by images is stressed as a basic requirement but many other advanced and computational intensive search functionalities constitute high desiderated requirements, e.g. "Search by Video scenes" and "Search part-of objects by Image". When documents contain images, these types of search can give a solution to the problem of searching language-specific files as texts and audio files are.

Content Management

Advanced cooperative functionality is needed to support the joint work of the digital library users. Annotations represent one of the mechanisms enabling users to cooperate in different contexts, e.g. during the creation of information objects, to express judgments on objects about their quality, pertinence, their relationship with other objects, etc., improving thus the access to other users. Moreover, annotations are to be expressed in multimedia formats so that they become complex objects and effective instruments for enriching objects.

In particular:

ImpECt special requirements regard the capability to process, merge and elaborate the digital library's contents into elaborated information products like reports, consolidated multi-temporal images analysis, documents, animations and simulations; part of these activities includes the definition of ad-hoc compound services.

The ARTE the community asks for the capability to create complex objects representing courses, exhibition catalogues, and workshops cooperatively, and to manipulate audio/video files for making their content searchable.

Personalization-Collection management
Users ask for having personalized views of single objects and personalized views of digital library functionality. Personalized views of the information space are also requested in order to have focused views of the huge information space that potentially becomes available by reusing pre-existing information. The mechanism of collections represents one of the modalities for reducing and customizing this space.
In particular:
The ARTE community sees collection management/personalization as a mechanism to support the organization of courses. The related requirements describe how specific collections that address the knowledge needs of the students can be created by reusing content and services maintained in an ARTE DL or available in DILIGENT. These collections are considered able to automatically update their content following the changes in the original archives. As a result, the students of each course have access to the most updated material on the topic of each course.

7.2 BRICKS

BRICKS [8] is an on-going EU funded Integrated Project (IP) that aims to establish the organisational and technological foundations of a Digital Library at the level of a European Digital Memory (EDM). In this context, a "digital library" refers to a networked system of services over globally available collections of multimedia digital documents, providing a variety of knowledge layers for a variety of users and access modalities. The BRICKS vision is an integrated system that offers functionality for new generation of Digital Libraries, a comprehensive term covering "Digital Museums", "Digital Archives" and other kinds of digital memory systems. The results of the Project will constitute the main assets of a Factory, which has been subsidised by the Consortium partners and the EU for the duration of the Project, but will sustain itself in the future. The mission of the BRICKS Factory is the definition, development, and maintenance of a user- and service-oriented space to share knowledge and resources in the Cultural Heritage domain.

A key motivation behind BRICKS is the drastic reduction of the costs of developing and deploying DL services over the entire DL lifetime. The BRICKS infrastructure follows a component-based software architecture that allows interoperability of heterogeneous content and services, which are, thus, reusable in defining new services. In particular, a Brick is a software component (possibly encapsulating content), whose functionality is made available through a formally defined interface; hence, it can be integrated with other Bricks to create functionally-richer Bricks. Furthermore, BRICKS is a running platform on which services obeying the rules of the architectural model can be deployed and, thus, made available to the entire DL, with minimal effort. The ambition is for this infrastructure to include a set of

services that can be the basis of future advancement of the BRICKS factory and an attractive lighthouse for the creation of a future cultural heritage community.

In order to be adequate to its role, the architecture attempts to fulfil the following requirements:

- *expandability* (ability to acquire new services, new content, or new users, without any interruption of service);
- *scalability* (ability to maintain excellence in service quality, as the volumes of requests, of content and of users increase);
- *availability* (ability to operate in a reliable way over the longest possible time interval; *incrementality of engagement*, (ability to offer a wide spectrum of solutions to the content and service providers that want to become members); and,
- *interoperability* (ability to make available services to and exploit services from other DLs).

In order to manage a complex and ambitious set of services that exhibit these five qualities, BRICKS has defined a suitable set of user and pilot scenarios. These serve as concrete targets for the entire development of the infrastructure. These scenarios form a relevant and well-balanced distribution between different user typologies and methods of working in a digital content context. These are grouped according to their primary motivation (BRICKS community, BRICKS content, etc.) and are briefly summarized below:

.

- *Archaeological Sites*: The objective of this group of scenarios is to enable intelligent access to shared knowledge and information about the European cultural heritage for education and other uses. These scenarios will explore the potential of the BRICKS platform for cross-language information retrieval, geographic information retrieval and presentation, visualisation, and eLearning. The scenarios included are the following: Cultural Landscape Discoverer (focusing on sharing knowledge and information about cultural landscapes); Finds Identifier (focusing on improving the identification of archaeological finds and the creation of archaeological reference collections while supporting education); Landscapes Reconstructed (focusing on reconstruction of cultural landscapes via innovative and intelligence access); and Pompeii and Roma (two scenarios focusing on making high-resolution images of the two sites accessible via BRICKS).

- *Small and Medium Museums*: The objective of this individual scenario is to introduce a digital application process for the "European Museum of the Year" award. Each year, the judging committee for the award is looking for enterprise and innovation likely to have a significant influence in the national and international museum field. Special attention is paid to imaginative interpretation and presentation, amenities, financial organization, social responsibility, educational work, marketing and management. The main purpose of the scenario is to create a database with a collective memory of museum innovation in the Council of Europe area. This will help potential candidates and judging committee members to identify innovative practices in museums to benchmark their own practices, make use of experiences elsewhere, share knowledge on innovation, access suppliers of innovative projects, etc., things impossible with the current manual processes. Using BRICKS is an opportunity for content providers to become a European hub of museum innovation. It can raise their profile and therefore offer opportunities for income generation and new partnerships (e.g., in tourism).

- *Living Memory*: The objective of this group of scenarios is support the general public, students, researchers, and anyone else interested in collaborative environments in creating their own cultural contents and making them available to the BRICKS community. The scenarios included are the following: Online Exhibition (focusing on cultural institutions preparing online exhibitions of their contents by using the BRICKS infrastructure and BRICKS tools developed for this purpose); Expert Forum (focusing on facilitating cultural institutions in collecting contributions of people who have specialized knowledge in a certain field); E-Learning Forum (focusing on transforming cultural content into learning content for academic learning environments as well as on introducing students to real-life scenarios in cultural heritage management and related fields); Projects, which is a generalization of Online Exhibition (focusing on creating "niches" where selected users can access and work on selected resources); and The Story Album (focusing on museum visitors being transformed into memory creators themselves, leaving to the museum their own oral contributions, photos, and written annotations).
- *Scriptorium*: The objective of this group of scenarios is to facilitate the works and scientific activities of target users by the definition of a new way of fruition and management of distributed digital texts and historical documents. The target users are historians and archive professionals, universities, cultural centers, libraries, history professor and teachers, and other history-inclined individuals. The main scenario here is the Critical Editions scenario, whereas others will be specified in time. The Critical Editions scenario will describe the creation of a critical edition of an ancient work, in particular, the opera by the Italian scientist Francesco Maurolico about conics.

Below we briefly summarize the requirements collected from the main actors of these four groups of scenarios.

7.2.1 Information space

Across all scenarios, the requirement for dealing with complex and flexible information is evident. Each participating institution should retain ownership and maintain its own data collections, which are offered for shared use through the BRICKS infrastructure. The latter should be responsible for the management of each institution's content and metadata visible in BRICKS: (i) it should offer flexible mechanisms for importing older data from legacy systems; (ii) it should plug the legacy systems into the overall architecture so that there is transparent access to the old data; and (iii) it should manage all new (i.e., data born BRICKS-aware) content and metadata.

Each document in the BRICKS world can be a complex object, capturing whole-part relationships, and may be (or include) including multimedia. The system should support different editions of a document as well as unique identifiers. In addition, it should offer the ability to generate external object representations (e.g., different visualizations, different resolutions, generated maps, thumbnails) on the fly.

The key requirement regarding the information space of BRICKS users is that, based on their privileges, they should be able to create, organize, and manage sets of document at various levels. The main concepts in this direction are the following:

- Physical and logical collections: are actual collections that are either physically stored or virtually defined for the purposes of being used by BRICKS users. They can be

nested to form a hierarchy of collections, which is ideal to structure content; they can be browsed like operating-system directories in a windows-based interface; they can be created from query results or "manually"; they can have metadata that can be edited and used for search; or they can be associated with visibility rights at different levels.

- Folders: are similar to collections but can be created by every registered user for their personal use. In addition to BRICKS content, they can hold external content as well, i.e., content that is not checked into the system. Folders can be made accessible to non-registered users, but can have no metadata and, hence, cannot be searched.
- Projects: These are roughly transient collections. A project consists of a "corpus" (i.e., a collection, a set of annotations, and additional resources, e.g. Thesauri) and a user group (consisting of the members of the project). Examples of projects may be online exhibitions (with extended rights for registered users) or student-teacher projects.

7.2.2 Users

All scenarios, from all user communities involved, require that the BRICKS infrastructure takes responsibility for a rather sophisticated approach to several security-level issues. Besides standard user management, user rights should be managed at a fine level of granularity (e.g., enabling individualizing rights depending on the different groups to which a user may belong, the different roles a user may be play at a given time or within a given context, or combinations of these). Each local institution should be able to maintain its local user management practices. At the same time it should be able to easily adopt any available BRICKS-specific user management mechanisms, if it were so decided. Important specializations in groups and roles include cultural end user (interested just in content search & browse), user manager, query & retrieval manager, annotation manager, media manager, and of course, content manager.

In addition, a comprehensive approach has been taken in respect of Digital Rights Management (DRM) on the part of the organizations that own the content that becomes available through BRICKS.

Finally, the overall security model should be open, so that extensions in the overall structure may be conveniently implemented.

7.2.3 Functionality

DL Management

Given that one of the objectives of BRICKS is to make it possible for small and medium-size cultural institutions to organize and offer collections to interested users, a critical issue for all potential members is maintaining a balance between the flexibility in the management of those collections and protecting the rights of their owners. Collections should be created by assigning a schema to the resources of interest (with respect to their content and application services, and according to their processing power and storage).

Focusing on the sustainability model for both the BRICKS infrastructure as well as the participating memory institutions, the system offers functionality for tracking resource use where they are not freely accessible. This functionality supports management of appropriate pricing schemes and generating the relevant bills each time.

Access
Based on users' privileges, they should be able to use content access services offered by the BRICKS infrastructure. Such access can have the form of a targeted search or a serendipitous browse of the content available by the BRICKS community. Each access form, especially search, may be further subdivided into several diverse types:

- Keyword search: a keyword-based search over the metadata available or the free text appearing in the collections' documents (in the Information-Retrieval philosophy)
- Structured query search: a boolean search using query expressions over the metadata available. For this a high-level structured query language should be offered by the system.
- Ontology-based search: a keyword-based or query-based search over an ontology (or collection of ontologies) in the cases where content has indeed been classified ontologically
- Combinations of above: search that combines some of the above characteristics, combining the inquiring features that are most appropriate for each content type.

Content Management
Supporting the information space as described above is the main requirement with respect to content management. Composing complex content (including multimedia) from simpler objects/documents is at the heart of this. Constituent objects should be sharable across all complex collections without, of course, necessarily having separate physical copies of the shared content (replication is an orthogonal issue) with its use.

A major enhancer of the experience a user has when examining the contents of documents is the ability to interact with the contents themselves. This holds whether the user is a scholar or a curious 10-year old or anything in between. Creating and retrieving users' annotations on specific documents is one of the most critical forms of such interaction; in fact, it is a very useful form of collaboration as well. BRICKS should offer advanced annotation management services. These include the ability to annotate not just whole but also parts of the content item (pictures, text excerpts, audio clips, etc.) to be annotated, which can be selected using graphical tools; with respect to the latter, users should be able to install software on their environment to take advantage of such functionality. Annotations should be offered in different types: free text annotations, structured annotations (using controlled vocabularies, thesauri, ontologies, etc.), associations and links, and other forms. Naturally, access to annotations should follow all the rules applied on the content itself. (e.g., annotations should be "private", "shared", "public").

Personalization-Collection management
Complementary to annotation, is personalization: whereas the former is more directed towards servicing interactions among multiple users (for the most part), the latter is servicing interactions between individual users and the system. Personalization services in BRICKS should customize the system behavior depending on the user concerned at any one point. Personalization should be supported at various levels: based on different characteristics of the user (e.g., user preferences, demographic data), whether as an individual or as a member of different groups. Such characteristics should be stored in a user's profile (for every user or group of users), which should be managed by the BRICKS system. Furthermore, there are several aspects of system behavior that should be affected by personalization. The most prominent ones are regarding the results of searches, whether with respect to the order in

which documents appear (based on personal relevancy) or with respect to the actual set of documents returned. Such personalization will both be reducing the execution time of user queries and will be increasing the recall and precision of queries results (according to user preferences).

8 Conclusions and Lesson Learned

Digital Libraries are the core of the information society of the 21st century. In a vast variety of applications and application domains, DL functionality plays an important role.

DL applications usually have particular requirements, especially with regard to the DL functionality needed. Therefore, in most cases custom DLs are built for individual applications. While they are doing very well in providing functionality they have been developed for, most of them lack support for other features. In addition to limited functionality, from the tables given in Section 4-7 which summarize the comparison of existing systems, it becomes obvious that systems also consider only a limited number of information object types. While there is already a significant gap in most systems between provided functionality and user requirements, it is already known that future requirements for DLs will address an even richer set of functionality, new object types, etc. which makes this gap even larger.

Currently, there is no universal DL that supports all requirements and expectations coming from users and user groups. Rather, a large number of systems in the DL area currently exist, ranging from repository systems to sophisticated systems being based on novel architectures and technologies such as Grid or peer-to-peer infrastructures. In this document, we have provided a classification of these systems as a first attempt to systematically order the plethora of systems from the DL field. A thorough analysis of selected systems which is provided in this document reveals heterogeneity at several levels:

- at the level of users (and user groups) supported
- at the level of the information model they consider
- at the level of the architecture these systems follow
- at the level of the DL functionality provided, and finally
- at the level of the quality, this functionality is provided to the user (quality of service)

Due to this heterogeneity, it turns out to be a hard and challenging task to compare DLs in terms of their support for particular DL requirements, not to speak of evaluating or benchmarking these systems. In addition, when taking a closer look at the individual levels, significant differences among systems even in the same class can be found. Support for users and user groups, for instance, is mostly not considered at all or if so, only available in a rather limited way. In terms of the information space, systems usually focus on special object types and/or collections. The same is true for the DL functionality provided. Since systems are mostly custom-made for supporting their particular object types and functionality, extending them to new object types and/or more sophisticated functionality would be a very complex task. Major differences between the systems we have compared exist also at the architecture level and especially at the level of quality of service. Most systems have put the focus on the DL functionality they provide but do neglect the quality in which the functionality is offered. This is especially true for non-functional requirements like availability, failure resilience, reliability, etc. However, in future DL with an increasing number of users, a richer set of DL functionality that needs to be provided, a larger set of heterogeneous objects of different types that need to be managed, these quality characteristics will increasingly become important.

The analysis of DLs presented in this report confirms a sense of the urgent need for having a reference model of Digital Libraries as it is currently being developed in DELOS WP 1. This

reference model needs to be supported by the capabilities and features of the systems in the current DL landscape. Starting with a careful analysis of user requirements and expectations, a reference model will clearly identify the fundamental concepts, core functionality, building blocks, and processes underlying digital libraries. The reference model will be a major step in supporting the analysis and comparison of systems, but also in increasing the consistency across digital libraries and in measuring their technical qualities.

Currently, the implementation and deployment of a new DL is a complex and time-consuming task. With the help of the reference model, the core building blocks that make up a DL and their interrelations are identified. In addition, the DL reference model also addresses a reference architecture of a modular service-oriented DL that provides the technological basis for putting together different DL services. Once the reference model is in place, it is expected that basic DL functionality will be available by means of well tested, highly sophisticated, specialized, and even certified services from different providers. In addition to the economic benefits for the DL area, also the complexity and effort to set up a new DL will be significantly decreased. In contrast to building a new DL mainly from scratch, the vision is to have a market for core DL services that can be easily combined in a customized system for a particular user group or application.

References

[1] S. Anderson and R. Heery. Digital Repositories Review. JISC Digital Repositories Programme, 2005. http://www.jisc.ac.uk/uploaded_documents/digital-repositories-review-2005.pdf

[2] arXiv e-print service. http://arxiv.org/

[3] M. J. Bates. Idea tactics. *Journal of the American Society for Information Science*, 30(5):280-289, 1979.

[4] M. J. Bates. Information search tactics. *Journal of the American Society for Information Science*, 30(4):205-214, 1979.

[5] M. J. Bates. Where should the person stop and the information search interface start? *Information Processing and Management*, 26(5):575-591, 1990.

[6] J. Bekaert, P. Hochstenbach, and H. Van de Sompel. *Using MPEG-21 DIDL to Represent Complex Digital Objects in the Los Alamos National Laboratory Digital Library*. D-Lib Magazine, 9(11), November 2003.

[7] K. Böhm, M. Mlivoncic, and R. Weber. Quality-Aware and Load-Sensitive Planning of Image Similarity Queries. In Proceedings of the 17th Intl. Conf. on Data Engineering, pages 401–410, Washington, DC, USA, 2001.

[8] BRICKS. Building resources for Integrated Cultural Knowledge Services. http://www.brickscommunity.org/ IST-2003-507457.

[9] Candela, L., Castelli, D., Pagano, P., Simi, M. *From Heterogeneous Information Spaces to Virtual Documents*. In Proceedings of the 8th International Conference on Asian Digital Libraries, ICADL 2005, pages 11-22, Bangkok, Thailand, December 2005.

[10] COLLATE. Collaboratory for Annotation, Indexing and Retrieval of Digitized Historical Archive Material. http://www.collate.de/. IST-1999-20882.

[11] Daffodil. Distributed Agents for User-Friendly Access of Digital Libraries. http://www.daffodil.de

[12] DAREnet Digital Academic Repositories http://www.darenet.nl

[13] DELOS Workpackage 4 User Interfaces and Visualization. D4.1.1: Report on functional and non-functional digital library requirements. September 2004. http://delos.dis.uniroma1.it/docs/Delos_D4.1.1_v1.7.pdf

[14] DILIGENT A DIgital Library Infrastructure on Grid ENabled Technology. (IST-2003-004260) http://www.diligentproject.org

[15] EGEE: Enabling Grids for E-science in Europe. http://public.eu-egee.org

[16] ePrints Supporting Open Access. http://www.eprints.org

[17] Fedora Flexible Extensible Digital Object and Repository Architecture. http://www.fedora.info.

[18] N. Fuhr. Information Retrieval in Digitalen Bibliotheken. In: 21. DGI-Online-Tagung -- Aufbruch ins Wissensmanagement. 1999.

[19] N. Fuhr, N. Gövert, and C.-P. Klas. An agent-based architecture for supporting high-level search activities in federated digital libraries. In *Proceedings 3rd International Conference of Asian Digital Library*, pages 247-254, Taejon, Korea, 2000. KAIST.

[20] N. Fuhr, C.-P. Klas, A. Schaefer, and P. Mutschke. Daffodil: An integrated desktop for supporting high-level search activities in federated digital libraries. In *Research and Advanced Technology for Digital Libraries. 6th European Conference, ECDL 2002*, pages 597-612, Heidelberg et al., 2002. Springer.

[21] gLite A Lightweight Middleware for Grid Computing. http://glite.web.cern.ch/glite/

[22] Greenstone Digital Library Software. http://www.greenstone.org

[23] IEEE P1484.12 Learning Object Metadata Working Group. http://ltsc.ieee.org/wg12/

[24] i-Tor Tools and Technology for Open Repositories. http://www.i-tor.org
[25] V. Iverson, Y.-W. Song, R. Van de Walle, M. Rowe, C. Doim, E. Santos, and T. Schwartz. MPEG-21 Digital Item Declaration. ISO/IEC JTC 1/SC 29/WG 11 N397, http://xml.coverpages.org/MPEG21-WG-11-N3971-200103.pdf
[26] J. Krause. Das WOB-Modell. In: Vages Information Retrieval und graphische Benutzeroberflächen: Beispiel Werkstoffinformation. Konstanz: Universitätsverlag, Konstanz (1997) 59—88
[27] S. Kriewel, C.-P. Klas, A. Schaefer, N. Fuhr. Daffodil - Strategic Support for User-Oriented Access to Heterogeneous Digital Libraries. D-Lib Magazine, 10(6), June 2004.
[28] C. Lagoze, S. Payette, E. Shin, and C. Wilper. *Fedora: An Architecture for Complex Objects and their Relationships*. Journal on Digital Libraries, Special Issue on Complex Objects, 2005.
[29] C. Lagoze and H. van de Sompel. The open archive initiative: building a low-barrier interoperability framework. In Proceedings of the first ACM/IEEE-CS Joint Conference on Digital Libraries, pages 54-62. ACM Press, 2001.
[30] G. Marchionini and G. Geisler. The Open Video Digital Library. D-Lib Magazine, 8 (12), December 2002.
[31] Metadata Encoding and Transmission Standard (METS). http://www.loc.gov/standards/mets/
[32] M. Mlivoncic, C. Schuler, and C. Türker. Hyperdatabase Infrastructure for Management and Search of Multimedia Collections. In DELOS Workshop: Digital Library Architectures, pages 25–36, S. Margherita di Pula, Cagliari, Italy, 2004.
[33] NSDL The National Science Digital Library. http://nsdl.org/
[34] OpenDLib A Digital Library Service System. http://www.opendlib.com
[35] S. Payette and S. Thornton. *The Mellon Fedora Project: Digital Library Architecture Meets XML and Web Services*. In Proceedings of 6th European Conference on Research and Advanced Technologiey for Digital Libraries, ECDL 2002, pages 406-421, Rome, Italy, September 2002.
[36] G. W. Paynter. Developing Practical Automatic Metadata Assignment and Evaluation Tools for Internet Resources. In Proceedings of the Fifth ACM/IEEE Joint Conference on Digital Libraries JCDL'05, pages 291-300, Denver, Colorado, ACM Press, 2005.
[37] RENARDUS Academic Subject Gateway Service Europe. http://www.renardus.org/ . IST-1999-10562
[38] S. Ross. Digital Library Development Review. National Library of New Zealand, Wellington, 2003, ISBN 0-477-02797-0, http://www.natlib.govt.nz/files/ross_report.pdf
[39] SCHOLNET. Developing a Digital Library Testbed to Support Networked Scholarly Communities. http://www.ercim.org/scholnet. IST-1999-20664.
[40] C. Schuler, H. Schuldt, C. Türker, R. Weber, and H.-J. Schek. Peer-To-Peer Execution of (Transactional) Processes. In: International Journal on Cooperative Information Systems (IJCIS). Vol. 14, No. 4 (2005) 377-405.
[41] C. Schuler, R. Weber, H. Schuldt, and H.-J. Schek. Scalable Peer–to–Peer Process Management – The OSIRIS Approach. In Proc. of ICWS Conf., pages 26–34, San Diego, CA, USA, 2004.
[42] H. Schuldt, G. Alonso, C. Beeri, and H.-J. Schek. Atomicity and Isolation for Transactional Processes. ACM Transactions on Database Systems, 27(1):63–116, 2002.
[43] A. Schaefer, M. Jordan, C.-P. Klas, and N. Fuhr. Active Support For Query Formulation in Virtual Digital Libraries: A case study with DAFFODIL. In: *Research and Advanced Technology for Digital Libraries. 9 th European Conference, ECDL 2005*, Springer.
[44] Shibboleth Project Website http://middleware.internet2.edu/shibboleth/

[45] M. Springmann. A Novel Approach for Compound Document Matching. Bulletin of the IEEE Technical Committee on Digital Libraries (TCDL), 3, 2006. To appear.
[46] T. Staples, R. Wayland, and S. Payette. The Fedora Project: An Open-source Digital Object Repository Management System. DLib Magazine, 9 (4), April 2003.
[47] A. Stein, et. al., COLLATE - Collaboratory for Annotation, Indexing and Retrieval of Digitized Historical Archive Material - Final Project Report Del. No D11.1, 2004, http://www.collate.de/D11.1_Final-Report_COLLATE-040203.pdf
[48] R. Tansley, M. Bass, and MacKenzie Smith. *DSpace as an Open Archival Information System: Current Status and Future Directions*. In Proceedings of the 7th European Conference on Research and Advanced Technologiey for Digital Libraries, ECDL 2003, pages 17-22, Trondheim, Norway, August 2003.
[49] R. Tansley, M. Bass, D. Stuve, M. Branschofsky, D. Chudnov, G. McClellan, and MacKenzie Smith. *The DSpace Institutional Digital Repository System: Current Functionality*. In Proceedings of the 3rd Joint Conference on Digital Libraries, JCDL 2003, pages 87-97, IEEE Computer Society, 2003.
[50] TEL The European Library. http://www.theeuropeanlibrary.org/
[51] A. van der Kuil and M Feijen. The Dawning of the Dutch Network of Digital Academic REpositories (DARE): A Shared Experience. Ariadne, 41, October 2004.
[52] T. van Veen and B. Oldroyd. *Search and Retrieval in The European Library*. D-Lib Magazine, 10(2), February 2004
[53] R. Weber, C. Schuler, P. Neukomm, H. Schuldt, and H.-J. Schek. Web Service Composition with OGrape and OSIRIS. In Proc. of VLDB Conf., Berlin, Germany, 2003.
[54] R. Weber, J. Bolliger, T.R. Gross, and H.-J. Schek. Architecture of a Networked Image Search and Retrieval System. In Proc. of CIKM Conf., pages 430–441, Kansas City, Missouri, USA, 1999.
[55] R. Weber, H-J. Schek, and S. Blott. A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Spaces. In VLDB '98: Proc. VLDB'98, pages 194–205, San Francisco, CA, USA, 1998.
[56] R. Weber and H-J. Schek. A Distributed Image-Database Architecture for Efficient Insertion and Retrieval Tasks. In 5th Intl. Workshop on Multimedia Information Systems (MIS'99), pages 48–55, 1999.