Chapter: 7
Unified Intranet Models for sharing Electronic Information resources: a review

7.1 Introduction

7.2 Intranet portal as a unified interface for organising and sharing electronic information resources

7.3 Unified Intranet Portal Solutions (Models)

7.4 Discussions

7.5 Conclusion

7.6 References
7.1 INTRODUCTION

In the 1990s libraries changed significantly, developing new services and processes primarily to deal with the expansion of electronic resources in their collections. These resources have resulted in changes in the management of acquisition, access, bibliographic records and resources sharing between libraries. Previous changes in libraries have tended to be evolutionary, with the movement from hand duplicated to printed books the most significant. This development resulted in increased collections and new libraries. The Internet has brought a far more radical change, with the number of resources now available wildly surpassing the changes in traditional printing, and requiring rapid change in all library services and systems (Missingham, 2001).

Today's 'hybrid' libraries provide substantial resource functions, providing pathways for users through the maze of knowledge sources and networked information systems that are now available in the modern academic library. Libraries are becoming combined information technology and information access points with services no longer restricted to time and place. Information delivery to the desktop is the expectation rather than the hope. The knowledge mediation function of the library comprises three basic functions (Leigh, 2000):

- Providing a window on available knowledge through a wide variety of sources within the library holdings;
- Actually providing materials to the user once they have decided exactly what they need; and
- Offering users various kinds of support related to the complex processes of knowledge mediation and acquisition.

Information is as much about access as about quality. Information discovery, information organisational and information sharing is crucial for the sustainability of information services within libraries.
Access to distributed information is hampered by the multiplicity of material available on-line from a network of public, private and commercial organizations, libraries, publishers, vendors and individuals. There is a great need for the development of a system infrastructure that facilitates navigation and retrieval, and that provides mediating support for the maze and variety of information available on-line. This system infrastructure should be capable of identifying, accessing, and retrieving the digital resources available. Furthermore, it needs to provide a coherent and consistent view of as many of the information repositories as possible. The overall goal of system architecture is to ensure acceptable behavior in the face of limited and varied computation resources and connectivity and decentralized control. The goal of system architecture is also to attempt future proofing where possible and permit scalability as the system grows. Several properties of distributed, federated information systems complicate service guarantees (Fuhr, 1999):

- The variety of hardware (processors, network, protocol, display devices) complicates system optimization.
- Collection size and diversity, and the number of collections affect system behavior.
- Diverse requirements for privacy and security affect performance.
- Different models of system cooperation imply a spectrum of solutions for any given problem.

In otherwords, most libraries currently provide access to a huge range of unrelated and seemingly unconnected electronic resources that have no common interface. To find that vital piece of information, you have to search each resource individually.

7.1.1 NEW COMPLEX INFORMATION ENVIRONMENT

The number of information resources in the scholarly environment is growing, and with it the need for integrated and easy access. Users expect to enjoy a single point of access to all the information resources that constitute their search environment. Furthermore, the typical library patron is not aware, and does not want to be aware, of differences between
resources. Institutions strive to provide both novice and experienced users with an interface that will enable them to obtain high quality materials from appropriate resources with minimal effort. The challenge for all the stakeholders in the industry – libraries, software vendors, and information providers – is to provide a friendly, integrated environment in a heterogeneous world, an environment that is as easy to use as a Web search engine and also returns authoritative results from library-defined collections (Sadeh, 2004).

In addition to the rapid transformation in computer, information, and telecommunication technologies, libraries everywhere also face with following challenges (Lee, 1998):

- exploding information resources,
- skyrocketing costs of library materials,
- growing diversity in information formats,
- shrinking library funding,
- high costs of library automation,
- increasing demand for library staff skilled in information technologies,
- changing nature of library collections, and
- expanding user demands and expectations.

A dramatic development of telecommunication and information technologies, especially the great popularity of World Wide Web, significantly contributed to the dynamic implementation of the latest technological solutions in library information processes, and brought about large, almost revolutionary changes at libraries. The emergence of modern information technologies forced libraries to look for new methods of servicing their clientele; especially, to grant the new users access not only to own resources and services, but also to outside (virtual) sources of information. A new process has begun the process of library transformation that makes libraries more access-oriented, as opposed to content-oriented. In other words, while there is information proliferation at exponential rate, there is also growth in technologies to support acquire, store, organize, distribute, access and use these resources. All these make us to believe that we are in a new complex information environment.
We live in a world of multiple, heterogeneous information repositories, resources, portals, and IR systems (Mischo, 2002):

- OPACs – local, regional, national shared bibliographic databases.
- Local and remote A & I Services.
- Discrete publisher and vendor repositories (full-text).
- Web search engines, vertical portals, custom portals Local metadata, digital objects, GIS, finding aids.
- Preprint servers and institutional repositories (D-Space, Eprints).
- Harvestable (OAI) sites and services.

The figure (Fig.7.1) below outlines the framework that supports today library’s needs for unified access (DSTC).

![Figure 7.1: Complex Information Environment (Adapted from DSTC)](image)

**7.1.2 NEW ENABLED INFORMATION ENVIRONMENT**

Two trends in information technology are rapidly transforming the way many organisations process data. On the one hand, the expansive networking of computer
systems has brought together workstations and mainframes to form vast collections of
distributed, online information services. The prime example and manifestation of this
trend is the World Wide Web (WWW). On the other hand, information units published in
these networks are no longer homogeneous, but may be compositions of heterogeneous
data types, e.g. text documents, images, and database records (Barja, et al. 1998)

Today, we are more fortunate, although the array of technological possibilities can
sometimes be daunting. The Internet is at our constant disposal for research, and
intranets, extranets and portals provide the means to collaborate with colleagues,
exchange documents, and manage business processes, market services and much more.
There are systems that have (ILTA, 2005):

- Broad accessibility, highly controllable at role and group levels;
- Strong matter-centric relational structure for navigation, context and relevance;
- Integration capabilities to consolidate information from source applications;
- Adaptable information structure for managing and viewing information; and
- Functional automation for communication and information flow.

We now have at our disposal a set of standards and best practices that allow us to create
integrated digital libraries and address some of these classic problems of information
retrieval. We have a standard retrieval environment (Web) and interface/client (Web
Browser), standard transport mechanisms to connect heterogeneous content (HTTP, OAI,
SOAP, WebHTTP), standard metalanguages and tools for describing and transforming
content and metadata (XML, DTDs & Schemas, XSLT, DC/DCQ, RDF, METS),
standardized search/retrieval mechanisms (HTTP Post/Get, SQL, Z39.50), and standard
linking tools and infrastructure (DOI, OpenURL, CrossRef)(Mischo, 2002).

7.1.3 TOWARDS INTEGRATED SOLUTIONS

The term integration may have part of its origin in the library world as an application and
concept called integrated library systems. As a further use of the word integrated, Carol
Tenopir (1995) has written that in an integrated reference environment, a common
interface leads users seamlessly to the best resource for their needs. It may not be obvious to the user whether the database is on his home institution's library computer, an another institution, at an online vendor's office several thousand miles away, or on the other side of the world at an Internet site.

In an increasingly complex and global information environment, an integrated library is of vital importance in enabling end users to search through large quantities of information. An integrated library means creating intelligent search, retrieval and presentation tools and interfaces for users. It means incorporating new information types, metadata and document encoding schemes. It also means new hardware and software systems which are capable of interpreting users' requests, including selecting from multiple databases. In the integrated environment a user interface should be a simultaneous gateway to the electronic and traditional collection of the library and to all central resources. An end user will not have to install different software clients and he will not be assigned different usernames and passwords (Swets, 1998).

In many digital library projects the access to heterogeneous information resources is a major issue, as users prefer a unified interface to the available information and staff will be relieved of too many routine instructions to individual databases. The so-called unified access means access to heterogeneous types of resources via HTTP and Z39.50 protocols. The users will have authorized access via a unified, integrated and comfortable interface. Most access systems consist of some type of metadata database access to the (linked) heterogeneous information resources (Halm, 1998).

In an attempt to harness the power of these distributed, heterogeneous systems into the hands of an end-user, systems have been built which, at least technically, allow the user to tap into any one of an array of information sources. These solutions do not really make information seeking much easier or simpler, however, because they require the user to learn the intricacies of a large number of interfaces, system features, and data formats. Thus, users are still expected to know how to receive, manipulate, and combine ever more complex types of data from ever more sources.
Both the early computer science experiments in digital libraries and the earlier initial efforts to build online public access catalogs (OPACs) followed a model similar to that shown in the figure 7.2. Under this model, a user needed to interact with each digital repository independently, to learn the syntax supported by each digital repository, and to have installed on their own computer the applications software needed to view the types of digital objects supported by each digital repository (Besser, 2002).

![Figure 7.2: Traditional Digital Collection Model (Adapted from Besser, H, 2002)](image)

Obviously the model in Fig.7.2 was not very user-friendly. Users don't want to learn several search syntaxes, they don't want to install a variety of viewing applications on their desk, and they want to make a single query that accesses a variety of different repositories. Users want to access an interoperable information world, where a set of separate repositories looks to them like a single information portal. A user-friendly model is outlined in Fig.7.3. Under this model a user makes a single query that propagates across multiple repositories. The user must only learn single search syntax. The user doesn't need to have a large number of software applications installed for
viewing. And retrieved sets of digital objects may be looked at together on the user's workstation. The model in Fig. 7.3 envisions a world of interoperable digital repositories, and is a model we need to strive for (Besser, 2002).

Figure 7.3: Ideal Digital Collection Model (Adapted from Besser, H, 2002)

Over the years we have made some significant progress towards the Fig. 7.3 model, particularly in the area of OPACs. Web browsers have given us a common "look-and-feel" between different repository user interfaces. The Z39.50 protocols have allowed users to employ single familiar search syntax, even when the repository's native search syntax appears foreign. Z39.50 has also promised to let user queries propagate to different repositories. But when one leaves the world of OPACs and enters the world of digital repositories, much work still needs to be done to achieve real interoperability. Most of this work involves creation and adoption of a wide variety of standards: from standards for the various types of metadata (administrative, structural, identification, longevity), to ways of making that metadata visible to external systems (harvesting), to common architectures that will support interoperability (open archives) (Besser, 2002).
7.2 INTRANET PORTAL AS A UNIFIED INTERFACE FOR ORGANISING AND SHARING ELECTRONIC INFORMATION RESOURCES

7.2.1 INTRODUCTION

Intranet Portal is an intelligent intranet. In addition to providing a unified interface, intranet portal can give a single login challenge to a user and then propagate the user's credentials through multiple applications. Successful intranet sites assemble useful information, organize it into logical systems, and deliver the information efficiently. What the user perceives as intranet portal is a library's intranet and/or extranet accessed through the web browser.

Intranet portals include direct access to applications, as well as a rights-and-permissions structure that allows for personalization of portal views to different audiences. A key benefit of intranet portals is the ability either to browse the contents of the portal using a site-specific (and therefore context-specific) taxonomy, or the ability to search the content. It is this search capability that in effect becomes a universal search (Cox and Yeates, 2002).

The core feature of any portal will be integrated, cross-database searching of a local catalog, other library catalogs, selected web sites, locally licensed full-text and abstracting/indexing databases, and public domain or publicly accessible abstracting and indexing services. Cross-database searching or integrated searching is a key feature of a portal. It distinguishes it from a web site. Many library web sites provide access to the online catalog, licensed resources, vetted web sites, and links to one or more commercial search engines. However, access to these disparate resources is most frequently accomplished by searching one source at a time. Library web sites usually do not permit users to conduct a single search of multiple resources, nor do web sites deliver integrated results. Users usually have to integrate the results from their separate searches as another step. Both the multiplicity of standards and the lack of standards are challenges in developing integrated, cross-database searching (Jackson, 2002).
True search interoperability will require the standardization of database content and the development of common search gateways among vendor online systems. It leads easily to the notion of unified and standardized formats for all electronic journal materials and other types of e-content as well. Accomplishing these changes will be no small task (Online, 2004).

7.2.2 UNIFIED ACCESS TO HETEROGENEOUS RESOURCES

Unified access refers to a process in which a user submits a query to numerous information resources. The resources can be heterogeneous in many aspects: they can reside in various places, offer information in various formats, draw on various technologies, hold various types of materials, and more. The user's query is broadcast to each resource, and results are returned to the user. The development of software products that offer such simultaneous searching relies on the fact that each information resource has its own search engine. The simultaneous searching product transmits the user's query to that search engine and directs it to perform the actual search. When the simultaneous searching software receives the results of the search, it displays them to the user. Simultaneous searching is also known as integrated searching, metasearching, cross-database searching, parallel searching, broadcast searching, and federated searching. Following diagram explains the environment better in the following figure (Fig.7.4):
User Level:

Consider a user of a participating library who wishes to search and find out all relevant information from any available sources that might satisfy his or her particular information need in a distributed library consortia environment. The user is most often concerned about the relevance and timeliness of information, not about, which source the information comes from, which data model it adheres to, or which query interface was used to retrieve the information. The user should not need to be aware of the technical underpinnings of the system, nor be limited as to what type of information he or she can access or from what type of sources. Ideally, there should be a uniform interface for expressing common queries for multiple information types and a single, consolidated set of results consistently ordered regardless of the particular scoring mechanism used by each source. The user should also be allowed to select, compare, cluster, and otherwise analyse information sources at a meta-level.
System Level:

Unified access at system level will provide:

- A single point of access
- Unified login (including one user ID)
- One common user interface, i.e. one presentation structure
- One uniform user-friendly retrieval system
- Direct access to electronic media and unified request service
- Patron-initiated online requests of resources and interlibrary loan facilities

7.2.3 UNIFIED INTERFACE: CHALLENGES

Many types of heterogeneity are due to technological differences; for example, differences in hardware and operating system platforms. Researchers and developers have been working on resolving such heterogeneities for years. These are referred to as system heterogeneities. Differences in the machine-readable aspects of data representation, formats, and storage for digital media may be referred to as syntactic heterogeneity. We consider representational heterogeneity involving schematic mismatches, and differences in data modeling constructs as structural heterogeneity. Differences in meaning are dependent on the vocabulary and terminology used to express information, and the contexts in which it is interpreted. They are referred to as semantic heterogeneity. (Ouksel and Sheth, 1999).

Barriers to access to heterogeneous resources in a consortia environment include multiple and confusing authentication and registration procedures. Users will not return to services that are time-consuming and difficult to use. In the print environment issues of ownership and access are very clear to our users, this is not true in the electronic environment where access may not necessarily include the ability to browse, download and print. Our users may also not have the necessary knowledge and skills to utilise the diversity of resources, and they may simply not be aware of the range of resources available.
The success of the unified access to heterogeneous resources will be heavily dependent on the use of agreed standards implemented in agreed ways. Standards will:

- Provide interoperability and communication across resources and services
- Provide for the consistent description of electronic data
- Provide consistent, unambiguous interpretation of queries and results
- Provide consistent return and display of results and data

However, at the moment there is an absence of standards in key areas. One of the main influences here is the fact that commercial interests are controlling the developing technologies such as the web, distributed processing, broadband networks etc. It is easy to make the assumption that the primary challenges are technological but this is to ignore the impact of a technologically dynamic and diverse environment on users.

7.2.4 UNIFIED ACCESS – OPPORTUNITIES

At the lowest level of integration, the World Wide Web now enables heterogeneous information services to be presented to a user through a simple menu-driven interface. In this model, the developer's role is to select appropriate services to list and to ensure the links to online services remain current. This is an appropriate way of building an information map to existing services where standards or interfaces are not yet in place to provide a higher level of integration; or where there is no business need for a higher level of integration. The services may be so heterogeneous that it is always appropriate to search them through a separate interface. One way of providing unified access is to use broker architectures to integrate access to the library catalogue and the library's digital collections through standard protocols. Another is to build a central set of indexes for resource discovery purposes. Such architectural model may have the following major components, which facilitates building robust, scalable and interoperable heterogeneous distributed library systems (CDL, 2002):

- Client Desktop
Chapter 7: Unified Intranet Models ...

- World Wide Web browser
- Telnet client
- Application-specific desktop client

- Service-specific functionality
  - User profiles
  - Application services
  - Searching
  - Analysis
  - Result set management

- Integrating components
  - Citation linking
  - Search management
  - Format & display normalizing

- Server tools
  - Middleware (web/database integration)
  - Database
  - Custom tools
  - Mediators and converters

- Protocols and infrastructure
  - Transport
    - HTTP
    - TCP/IP
    - Telnet
    - Storage Request Broker
  - Information management
    - Z39.50
    - ODBC, Corba
    - SDLIP
  - Directory services
  - Security
    - X.509 authentication, LDAP and Shibboleth
7.2.5 UNIFIED ACCESS – CHOICES

Libraries now face a dilemma. On one hand, libraries subscribe to many types of database retrieval systems that are produced by various providers. The libraries build their data and information systems independently. This results in highly heterogeneous and distributed systems at the technical level (e.g., different operating systems and user interfaces) and at the conceptual level (e.g., the same objects are named using different terms). On the other hand, end users want to access all these heterogeneous data via a union interface, without having to know the structure of each information system or the different retrieval methods used by the systems. Libraries must achieve a harmony between information providers and users (Fang, 2004).

In order to bridge the gap between the service providers and the users, it would seem that all source databases would need to be rebuilt according to a uniform data structure and
query language, but this seems impossible. Fortunately, however, libraries and information and technology providers are now making an effort to find a middle course that meets the requirements of both data providers and users (Fang, 2004).

The process of finding relevant materials falls into two stages. First is the resource discovery phase, when the user locates the resources most relevant to the specific search. Next comes the information discovery phase, when the search is executed in the various information resources and the results are retrieved.

Once the user sets the scope of the search and submits a query, the information discovery phase begins. The metasearch system delivers the query to the selected information resources and returns the results to the user. The process requires that the system 'understands' the expectations of the resources regarding the form of the query, on the one hand, and the nature of the results, on the other. It is up to the system to convert the unified query and adapt it to the requirements of each searched resource, deliver the query in the form appropriate to each resource, receive the results, and manipulate them so that they comply with the system's unified form (Sadeh, 2004).

Unified access to heterogeneous digital libraries (DLs) in a distributed environment can be achieved in three ways: modifying the existing DLs to interoperate, extracting metadata from each of the DLs and indexing it as a separate DL; or treating each DL as a separate entity and performing distributed searches. The first approach to interoperability requires digital libraries to use the same DL protocol or software suite. However, there are enough significant DL systems in use to assume that the DL community will continue to support a collection of heterogeneous systems and protocols. The second method of metadata extraction has advantages, but it assumes that metadata can be extracted and reindexed with no technical or legal barriers, neither of which assumptions are often true in our experiences. The third method creates more work for the provider of the federated digital libraries (FDLs), but allows for the inclusion of a greater number of DLs (Zubair, et al).
Underlying technologies for distributed search across heterogeneous, networked digital libraries are well researched. Client/server-based protocols for searching and retrieving information from remote databases have been widely used since the advent of Z39.50 to contemporary efforts such as SDLIP.

In case of Federation, a search request is distributed to other DLs that in turn distribute the request to others in a recursive fashion. The results are then accumulated together and returned to the originator of the request. Alternatively, in federated search, a client or DL front-end may broadcast queries and collect results directly. Federation has the advantage of distributing the load of searching across various DLs, but suffers from the fact that it depends on the high availability of the content providers. In addition, there may be significant latency in generating the search results.

In contrast, Harvesting is the process of pre-fetching the metadata from other DLs and then providing services on the local copy of harvested metadata. Such an approach is used in major commercial search engines like Google. Since the search is performed on a local copy of the metadata, results can be returned with very low latency. Moreover, since harvesting is carried out on a periodic basis, even if any of the content-provider DLs is unavailable, we can still search its metadata.

Figure 7.5 illustrates the concepts of Harvesting and Federation (Suleman, 2002):
7.2.5.1 UNIFIED ACCESS THROUGH FEDERATED SEARCHING

Federated searching—also known as parallel search, metasearch, distributed search or broadcast search—can be seen as an extension of the common user-interface research done decades ago. Federated searching is a technology that allows users to search many networked information resources from one interface. Federated searching aggregates multiple channels of information into a single searchable point (Tennant, 2003).

The objectives of federation services are to provide a uniform interface hiding the special features and restrictions of the individual sources, and to supply an integrated view on the data. Queries on the global views have to be decomposed, rewritten, and possibly translated in order to be processible by the sources. Finally, the results of the sub-queries have to be combined which means resolving conflicts caused by different representations of data (Endig, et al, 2000).

Given the multidisciplinary nature of research, users looking for "the" answer will find that federated search facilitates discovery of databases that they otherwise may not have consulted. The novice user looking for "an" answer can find results in multiple sources with a single search (Luther, 2003).

In the library space, federated search naturally evolved from broadcast searching, which simultaneously searches OPAC targets via Z39.50 protocol. Libraries moving beyond virtual online catalogs find that federated search engines give them the ability to include subscription databases, the Internet, or virtually anything in the electronic arena in which the access point can be authenticated. The types of resources that can be searched include local and remote library catalogs, abstracting and indexing databases, full-text aggregator databases, and digital repositories. From a technical standpoint, the software uses a distributed search method across heterogeneous databases using multiple search protocols.
When a federated search engine is implemented at a particular library, it then becomes a unique service (Tennant, R. 2001). Federated-searching software allows customization, so no two implementations are exactly the same (Gerrity, et al., 2002).

Hane notes the limitations of the current generation of federated search engines. These include (Hane, 2003):

- The lack of a uniform authentication standard means that some databases are inaccessible to federated search engines.
- True, full, deduplication is impossible because databases download results in small sets and metadata standards vary by resource.
- Relevancy ranking is limited by the quality of the metadata, which usually does not include abstracts or full-text information.
- Although federated search systems are fundamentally software, they must be implemented and managed as a service, which takes a great deal of resources.
- Federated search engines cannot improve on the native interface in terms of search accuracy and precision.

As both a technique and technology, federated search enables search and content integration across internal and external subscription content. A sophisticated federated search system can be tuned to meet the specific needs of an organization. Federated has the untapped potential to provide good Return On Investment (ROI) through improved efficiency in searching and improvement in the quality of the information it delivers. Perhaps the most important perceived shortcoming of federated search is its lack of relevancy ranking. When content is retrieved via searching multiple content sources in real time, one of the fundamental questions is how the system organizes the consolidated results by relevance. Due to a scarcity of rich metadata, the filtering of heterogeneous data remains a major challenge throughout the federated republic. Building federated search capabilities poses infrastructure challenges and organizational conflicts. The road to federated search begins with more than wishes for power and simplicity. It begins with acceptance of the need for standards to evolve (Solomon, 2004).
7.2.5.2 UNIFIED ACCESS THROUGH METADATA HARVESTING

Harvesting is a technique for extracting metadata from individual repositories and collecting it in a central catalog. The "harvesting process" relies on the metadata produced by humans or by full or semi-automatic processes supported by software.

The OAI-PMH uses harvesting as its approach. The assumption is that though there is an increased burden on the service provider, it can be overcome by increasing the resources of the service provider, thereby leveraging the other advantages of the harvesting approach, such as increased availability, low latency, etc.

A typical harvesting data using OAI-PMH and loading it into the database is given in Figure 7.6 (Arms, et al., 2002):

![Diagram illustrating the harvesting process](image)

**Figure 7.6: Importing metadata into the repository using OAI-PMH (Adapted from Arms, W.Y et al., 2002)**

The native metadata records that are harvested from collections are encoded in XML. They are stored in a temporary staging area. In the staging area, the records pass through three stages. First they go through a cleanup step, locally known as "caressing". Tasks at this stage include combining ListRecords responses and possibly stripping off some of the OAI-PMH wrapping. Then a crosswalk is used to generate a metadata record in the normalized format.

Figure 7.7 (Arms, et al., 2002) shows the steps in taking metadata from the database and serving it for harvesting. Within the repository there are relational tables configured for
the OAI server. When harvesting requests are received from service providers, SQL queries are sent to the repository and the appropriate records sent to the OAI server.

Ya-ning Chen, et al (Chen, etal, 2003) have provided the following requirements for a metadata system

1. Input and maintenance
   - User can create, delete, modify, and cut and paste metadata records through a web-based interface.
   - Multi-value attributes.
   - Select-list menu: including single column menu, hierarchy menu, and one-to-one menu.
   - Maintenance of values and codes for select-list menu content.
   - Preview of images.

2. Retrieval
   - Including the simple and advanced search options.
   - Keyword, Boolean operator, and limitation.

3. Display
章 7: 统一内联网模型

- 包括简单和高级显示选项。
- 内容专家可以选择显示格式（包括标签和信息）。

4. 兼容性

- 为提供基于 XML 的元数据导入和导出机制。
- 除了基于 XML 的方法外，还应导入和导出其他文本文件。
- 元数据系统可以链接到外部系统，如人员权威文件系统、地名词典和管理系统。

5. 管理

- 跟踪记录活动的事务，如创建者、修改者、验证者和相关日期和时间。
- 验证元数据记录的准确性基于标准要求，并提供相关消息用于记录修改。
- 提供基于指定标准的去重功能，用于记录创建和修改过程。
- 在私有和公共组中设置属性，以保持仅限内部使用的指定信息。
- 提供不同级别的授权，如基于任务或工作职责的标准。

6. 自动化运营活动

- 提供库存功能，用于执行与装饰、修理、检查等相关的常规任务。
- 提供流功能，用于执行与借出/归还、索赔等相关的任务。
- 提供库存功能，以便执行与装饰、维修、检查等相关的常规任务。
- 提供流功能，用于执行与借出/还、索赔等相关的任务。
To provide the function of a report generator in order to generate related reports or statistic data based on specified criteria from metadata elements.

Following are the some of the tools used for metadata harvesting (OAI):

<table>
<thead>
<tr>
<th>Tool</th>
<th>Implementer</th>
<th>Description</th>
<th>Version(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arc source</td>
<td>Old Dominion University</td>
<td>Arc is released under the NCSA Open Source License. Arc is a federated search service based on OAI-PMH. It includes a harvester, a search engine together with a simple search interface, and an OAI-PMH layer over harvested metadata. Arc can be configured for a specific community, and enhancements and customizations by the community are encouraged. Arc is based on Java Servlet technology and requires JDK 1.4, Tomcat 4.0.x, and a RDBMS server (tested with Oracle and MySQL).</td>
<td>2.0</td>
</tr>
<tr>
<td>CDSware</td>
<td>CERN</td>
<td>The CERN Document Server Software (CDSware) allows you to run your own electronic preprint server. It complies with OAI-PMH and uses MARC 21 as its underlying bibliographic standard.</td>
<td>2.0</td>
</tr>
<tr>
<td>Celestial</td>
<td>University of Southampton</td>
<td>Celestial is an OAI 2.0-compliant OAI aggregation/caching/mirroring tool. Using MySQL and Perl, Celestial allows the administrator to mirror 1.0, 1.1, and 2.0 repositories in their own 2.0-compliant repository.</td>
<td>2.0</td>
</tr>
<tr>
<td>DP9</td>
<td>Old Dominion University</td>
<td>An OAI Gateway Service for Web Crawlers</td>
<td>2.0</td>
</tr>
<tr>
<td>DSpace</td>
<td>HP Labs and MIT Libraries</td>
<td>DSpace is an open source digital asset management software platform that enables institutions to capture and describe digital content. It runs on a variety of hardware platforms and supports OAI-PMH version 2.0.</td>
<td>2.0</td>
</tr>
<tr>
<td>sprints.org</td>
<td>University of Southampton</td>
<td>Software to run centralised, discipline-based as well as distributed, institution-based archives of scholarly publications. The software is OAI compliant, i.e. metadata can be harvested from repositories running the software using the OAI metadata harvesting protocol.</td>
<td>2.0</td>
</tr>
<tr>
<td>Fedora</td>
<td>Cornell University</td>
<td>An open source digital repository architecture that allows packaging of content and distributed services associated with that content. Fedora supports OAI-PMH requests on content in the repository.</td>
<td>2.0</td>
</tr>
<tr>
<td>MARCXML framework</td>
<td>Library of Congress</td>
<td>A suite of tools, stylesheets, guidelines and XML documents to support MARC21 records in the XML environment. Includes University tools to support transformation/migration from oai marc to MARCXML, including an XML schema for MARC21 records.</td>
<td>2.0</td>
</tr>
<tr>
<td>my.OAI</td>
<td>my.OAI</td>
<td>Very simple Perl based OAI Harvester</td>
<td>2.0</td>
</tr>
<tr>
<td>Net::OAI::Harvester</td>
<td>Ed Summers</td>
<td>Net::OAI::Harvester provides an object-oriented client interface to the data found in OAI-PMH repositories</td>
<td>2.0</td>
</tr>
<tr>
<td><strong>OAIA</strong></td>
<td>University of Southampton</td>
<td>Based on PERL and MySQL, OAIA is a simple mechanism for providing caching and aggregating of OAI repositories.</td>
<td>2.0</td>
</tr>
<tr>
<td><strong>OAI Java Implementation for Linux</strong></td>
<td>University of Illinois, Urbana-Champaign</td>
<td>This is a simple, illustrative implementation of the OAI metadata protocol, using Java. The code is available on Source Forge (<a href="http://sourceforge.net/project/showfiles.php?group_id=47963">http://sourceforge.net/project/showfiles.php?group_id=47963</a>).</td>
<td>2.0</td>
</tr>
<tr>
<td><strong>OAI Implementation for Windows NT/Windows 2000</strong></td>
<td>University of Illinois, Urbana-Champaign</td>
<td>This is a simple, illustrative implementation of the OAI metadata protocol, using Microsoft Windows NT server technologies. The code is available on Source Forge (<a href="http://sourceforge.net/project/showfiles.php?group_id=47963">http://sourceforge.net/project/showfiles.php?group_id=47963</a>).</td>
<td>2.0</td>
</tr>
<tr>
<td><strong>OAIbiblio PHP Data Provider</strong></td>
<td>ibiblio</td>
<td>OAIIbiblio is a data provider implementation of the OAI-PMH, version 2.0. Written in a PHP object-oriented manner, this toolkit can be easily customized to communicate with an already existing, multi-table database. Some of the features include: rendered metadata mappings/transformations using separate XSL files, retention of highly expressive storage-level metadata with WDDX, storage-level separate from application &amp; DBMS independence.</td>
<td>2.0</td>
</tr>
<tr>
<td><strong>OAI Cat</strong></td>
<td>OCLC</td>
<td>OAI Cat is a Java Servlet web application providing an OAI-PMH v2.0 repository framework. The framework can be customized to work with arbitrary data repositories by implementing some Java interfaces. A demonstration implementation is available for download on the OAICat home page.</td>
<td>2.0</td>
</tr>
<tr>
<td><strong>OAI Harvester 2</strong></td>
<td>OCLC</td>
<td>The OAI Harvester 2 Open Source project is a Java application providing a combined OAI-PMH v1.1 and v2.0 harvester framework. A sample harvester application is included, or the included classes can be used within custom Java applications.</td>
<td>2.0, 1.1</td>
</tr>
<tr>
<td><strong>oai-perl library</strong></td>
<td>University of Southampton</td>
<td>A library of PERL language classes that allow the rapid deployment of an OAI compatible interface to an existing web server/database</td>
<td>2.0</td>
</tr>
<tr>
<td><strong>PEAR:OAI</strong></td>
<td>ZZ/OSS Information Networking</td>
<td>A Perl implementation of the OAI-PMH Data Provider which is a PHP class library based on the PEAR classes.</td>
<td>2.0</td>
</tr>
<tr>
<td><strong>Perl Harvester</strong></td>
<td>Virginia Tech.</td>
<td>Object-oriented harvester class with support for OAI-PMH v1.0, v1.1, and v2.0. Includes sample code to illustrate usage.</td>
<td>1.0, 1.1, 2.0</td>
</tr>
<tr>
<td><strong>PHP OAI Data Provider</strong></td>
<td>University of Oldenburg</td>
<td>This implementation completely complies to OAI-PMH v2.0, including the support of on-the-fly output compression which may significantly reduce the amount of data being transferred.</td>
<td>2.0</td>
</tr>
<tr>
<td><strong>Rapid Visual OAI Tool</strong></td>
<td>Old Dominion University</td>
<td>Rapid Visual OAI Tool (RVOT) can be used to graphically construct a OAI-PMH repository from a collection of files. The records in the original collection can be in any one of the acceptable formats. The formats currently supported</td>
<td>2.0</td>
</tr>
</tbody>
</table>
are RFC1807, Marc subset & COSATI formats. RVOT helps to define the mapping visually from a native format to oai_dc format, and once this is done the tool can respond to OAI-PMH requests. The tool is self-contained; it comes with a lightweight http server and OAI-PMH request handler and is written in Java. The design of RVOT is such that it can be easily extended to support other metadata formats.

<table>
<thead>
<tr>
<th>Static Repository Gateway</th>
<th>LANL</th>
<th>An implementation of a static repository gateway that complies with the specification at <a href="http://www.openarchives.org/OAI/2.0/guidelines-static-repository.htm">http://www.openarchives.org/OAI/2.0/guidelines-static-repository.htm</a></th>
<th>2.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>utf8conditioner</td>
<td>Cornell Univ</td>
<td>This is a small C program that will either check or 'fix' a UTF-8 byte stream. It was designed to be used within an OAI harvester to attempt to remove bad codes from supposedly UTF-8 byte streams so that they can then be parsed using a standard XML parser which would otherwise fail.</td>
<td>2.0</td>
</tr>
<tr>
<td>VTOAI OAI-PMH Perl</td>
<td>Virginia Tech</td>
<td>This toolkit implements the skeleton of the OAI-PMH v2.0 in an object-oriented fashion, thus hiding the details of the protocol from code that is derived from the predefined class.</td>
<td>2.0</td>
</tr>
<tr>
<td>XMLFile</td>
<td>Virginia Tech</td>
<td>Exposes a set of XML files in a directory as an OAI data provider. Supports: hierarchical sets mapped to directory structure, dates based on file modification, multiple metadata formats and on-the-fly conversions, symlinking corresponding to multiple set membership, arbitrary data locations, and no installation or external tools needed!</td>
<td>2.0</td>
</tr>
<tr>
<td>ZMARCO</td>
<td>University of Illinois, Urbana-Champaign</td>
<td>ZMARCO is an Open Archive Initiative Protocol for Metadata Harvesting (OAI-PMH) 2.0 compliant data provider. The 'Z' in ZMARCO stands for Z39.50; 'MARC' stands for Machine-Readable Cataloging; and the 'O' stands for OAI, as in the Open Archives Initiative. ZMARCO allows MARC records which are already available through a Z39.50 server to relatively easily be made available via the OAI-PMH.</td>
<td>2.0</td>
</tr>
</tbody>
</table>

Table 7.1: Metadata harvesting tools (Adapted from OAI)

### 7.2.5.3 Unified Access Using Broker or Mediated Architecture

Digital Libraries require the integration of highly heterogeneous data residing in external sources. To this end, they usually provide an architecture consisting of (i) one wrapper per integrated source in charge of transforming the data and the operations supported by the source into some commonly agreed formalism and (ii) one mediator component that communicates with the various wrappers and provides an integrated view of the system. Existing systems/prototypes mainly focus either on the flexibility to integrate data under
Chapter 7: Unified Intranet Models ...

various formats/forms or on the efficient query processing against heterogeneous and distributed sources.

Mediation is a technique used for integration of heterogeneous components. Lately, mediation have often been referred to as a layer residing between the end-user (client) and information sources; A mediator enables the user to interact with a source whose interface he does not know and helps the source understand queries expressed in a language not directly supported by the source. Thus, in general, mediation can be seen as information brokering over a broad variety of distributed, heterogeneous and autonomous components (Melnik, et al.).

To make the mediation task manageable, so-called wrappers are placed between mediators and information (re)sources. Wrappers shield mediators from some aspects of heterogeneity inherent in information sources. Task distribution between wrappers and mediators can be organized in different ways. The decision to be made is which part of heterogeneity has to be hidden from mediators. Wrapper design requires profound knowledge of the native interface of a component. Hence, wrappers often have to be written by the component's vendors. Therefore, it is crucial to simplify wrapper design by putting only minimal requirements on their interfaces (Melnik, et al.).

Broker architectures serve four main objectives (Pearce, et al., 2000):

- They provide a means of ensuring some level of semantic interoperability between existing or legacy provider systems pending the development of replacement systems that better meet local, regional or national information needs.
- They operate as an enabling technology to link essential building blocks that are best managed as distributed resources.
- They provide a single consistent user interface even when searching disparate resources.
- They enable service owners to authenticate users, track usage and deliver customised services through user profiles.
Using broker architectures to unite legacy systems under a single user interface is a valid way of solving immediate problems. However, while it is technically feasible to reduce the range of system interfaces to a small number, there are limitations to the extent of semantic integration that can be achieved through broker services. This will depend on the extent to which the provider systems (Pearce, et al., 2000):

- Support a requested function; for example proximity searching.
- Have a common understanding of the outputs required of the function; for example, the return of records containing titles beginning with a specified set of words, in alphabetic order.
- Use the same data standards; for example, MARC for resource description.
- Share business rules for the application of these standards; for example, by mapping the same data elements to a given access point, or using the same classification scheme.
- Share policies and guidelines; for example, local cataloguing rules and interpretations.

### 7.2.5.4 UNIFIED ACCESS THROUGH BOTH FEDERATION AND HARVESTING

In spite of its popularity with library users, there is some disillusionment with federated searching as implemented in the Library: not all databases can be available for crosssearching; searches are comparatively slow; searching is less precise and results not comprehensive; it is difficult to sort search results meaningfully; output options are poor. However, the OAI model for federated access, is perceived as less problematic, scalable, faster, and more standards-based (Tennant, 2004). Yet as the OAI protocol has become more widely adopted, service providers have discovered some major harvesting issues. Implementers of the OAI protocol for metadata harvesting (OAI-PMH) are experiencing problems not dissimilar to the ones experienced by federated search implementation teams. The central problem for both kinds of implementers is the diversity of metadata that the integrated framework must aggregate and make sense of for indexing, search,
sorting, and display (Tennant, 2004). Hence, any solutions that attempt to provide unified access may have to adopt both federation and harvesting approaches in order to achieve the desired level of unified access.

7.3 UNIFIED INTRANET PORTAL SOLUTIONS (MODELS)

Libraries and information providers are engaged in the design and development of intranet portal and gateway software to provide improved access to distributed information resources. These portals are attempting to provide seamless access services within the hybrid distributed information environment in which we work. These distributed resources environment includes: discrete publisher and vendor full-text repositories; locally mounted and remote Abstracting and Indexing (A & I) services; Web search engines and vertical portals; local collections of digital metadata, digital objects, and finding aids; preprint and other hidden Web sites and services; and local, regional, and national online catalogs and shared resource bibliographic databases (Mischo, 2002).

In the following pages, some of the unified intranet portal solutions from commercial, academic and research environment were reviewed.

7.3.1 COMMERCIAL MODELS

Large number of vendors are working on the problem of providing a unified interface and that there are at least partial solutions now being offered. Basically, these are library oriented portal solutions (enhancements or add-ons to integrated library management systems) that seek to integrate the holdings of libraries, from searching to access – regardless of format, the metadata standard in use, or the interface.

Features include:

- an integrated interface to multiple resources
- unified searching across different metadata schemes, including MARC records
unified or selective searching of different formats
unified or selective searching across multiple resources and systems
simplified access and/or authentication procedures
the ability to provide dynamic reference and citation linking
the ability to provide some sort of personalisation
the ability to provide management information and usage statistics.

In terms of the technology underpinning these developments, most are being developed using XML or like technologies, providing for searching across HTTP, Z39.50 and OAI (Open Archives Initiative) enabled resources, and most make use of OpenURL – a standard for providing context-sensitive links by specifying the syntax for sending metadata to services with Link Resolvers (servers specifically set up for the purpose) so that relevant article level items can be identified. Some of the leading commercial models with their descriptions are given in the following pages.

7.3.1.1 CHAMELEON IPORTAL (VTLS)

The Chameleon iPortal is an enhancement to the Virtua LMS, offering a Z39.50 based OPAC search engine; calendar of events; management/display tool; patron self-registration and updating tool; virtual reference and chat service; thesaurus browse interface; virtual news service integration with moreover.com; ability to maintain several "skins" (A skin is a term used by VTLS to indicate multiple user interfaces that cater to specific user classes); SDI Service; book locator service; ISO ILL Service

A range of distributed search protocols are supported, including various LDAP, SQL, Z39.50 and ISO ILL (via Virtua). Completely integrates with Virtua and supports SDI capability; thesaurus capability; book locator service and extended MARC service. VTLS also provide scanning services, digital library services, and graphic design and custom development services. Supported Platforms: Unix platforms: Intel Pentium (Red Hat Linux), Sun UltraSparc (Solaris 8), PowerPC 604 (AIX 4.3.3), DEC Alpha (Tru64 UNIX 5.1), PA RISC 1.1 (HP-UX 11.00). All platforms support the Oracle 8 RDBMS (www.vtls.com/Products/gateway).
73.1.2 ENCOMPASS (V2.0) AND LINKFINDERPLUS (Endeavor).

ENCompass is a complete XML-based solution for integrated end user searching across multiple data types and databases. Seeks to solve the problem of one search across multiplying silos of information. Support standards such as DC, EAD, TEI Lite, http, and ATHENS. ENCompass provides a framework that is designed to incorporate a wide range of searching and metadata types. It is claimed that this is the only product to provide access via Z39.50, HTTP, and XML gateways. This flexibility extends to the user interface by using XML/XSL style sheets to generate the public interface. ENCompass also provides flexibility for creating local collections by allowing the library to define the metadata supported for each local repository. LinkFinderPlus was designed for quick implementation of an OpenURL resolver, with a large knowledge base (11,000 resources). Supported Platforms: Unix on Solaris, with Oracle 9i or Windows 2000 and AIX (encompass.endinfosys.com/pdf/Encompass.pdf).

73.1.3 ILINK (SIRSI)

Sirsi’s Ilink is a gateway to a whole world of expertly organized, library-caliber information and services that can be delivered directly to users. Standards it supports are Bath profile, ISO ILL and SQL and ATHENS. SIRSI are particularly strong in ‘content enrichment’, having deals with a large number of book reviews, TOCs, cover images etc. They have also done a lot of work to make the initial interface and result screens more exciting and informative with favourite titles, or classifying returns by subject terms associated with hits. Cross searching functionality is confined however to Z39.50 searches. Supported platforms are all versions of UNIX and Windows NT/2000 (www.sirsi.com/Solutions/Prodserv/Products/ilink.html).

73.1.4 iPORT (OCLC)

iPort is a highly configurable, translatable, standalone portal product developed in Europe by and for academic libraries, based on current standards such as XML and HTTP, and
supporting ATHENS authentication. Based on Decomate II software, developed by Tilburg, Universitas Autonoma de Barcelona and London School of Economics. Support for other standards such as OAIPMH, LDAP, Bath Profile, SQL, and ISO ILL is under development. Supported platforms are Linux, Solaris or Digital Unix (http://www.ifnet.it/oclc/iportal/).

7.3.1.5 METALIB WITH SFX (EX LIBRIS)

Ex Libris's, MetaLib (including the SFX linking framework) is a library portal offering a gateway to the institution's resources, parallel, unified search in heterogeneous resources and a personalised scholarly environment and services. Links can be made with any resource catalogued in MetaLib's KnowledgeBase. MetaLib originated from a framework described by the members of the Cooperative Library Network for Berlin and Brandenburg (KOBV). The deduplication algorithm developed by ZIB, the mathematical institution responsible for the KOBV project, is the basis of the deduplication algorithm used in MetaLib. Both MetaLib and SFX were beta tested at several customer sites, including the Los Alamos Laboratories, the California Institute of Technology (Caltech), and KOBV. Supports Standards such as LDAP, Bath Profile, XQuery, ISO ILL. A range of distributed search protocols is supported, including various http (such as Entrez), Z39.50 and target proprietary APIs. SFX currently supports Athens and Kerberos. Supported platforms are Sun Solaris and Linux Red Hat (www.exlibris.co.il/).

7.3.1.6 MILLENNIUM ACCESS PLUS –MAP (INNOVATIVE)

Millennium Access Plus is made up of three independent components:

- WebBridge
- MetaFind
- Web Access Management

Together, these provide libraries with tools to manage and control access to all of their information resources by providing contextual linking, multi-protocol meta-searching, and authentication. Supported standards are Dublin Core, MARC, TEI, XML and.
ATHENS. Supported platforms are any versions of Unix (www.iii.com/pdf/paper_map.pdf).

7.3.1.7 ZPORTAL (FRETWELL-DOWNING)

ZPORTAL allows users to identify resources from libraries, museums, archives, and the Web, irrespective of how and where it's stored."(with plug-ins such as Zmbol to integrate non Z databases and Z2web to integrate search engines).Zportal concept evolved out of eLib and EC projects especially Agora. Supported Standards are LDAP, Bath Profile, SQL, ISO ILL and Athens. Z in the name implies the centrality of Z39.50 in their thinking. Supported Platforms are Unix: Solaris 7+, HP-UX, AIX and Windows (www.fdusa.com/products/zportal.html)

7.3.1.8 HORIZON INFORMATION PORTAL (DYNIX)

Access to online databases is available through Horizon Consolidated Searching where the library can define what databases are available to users with proper authentication. Users can search the local catalog, Z39.50 libraries, and online database with a single search entry. Other libraries not using Horizon Information Portal can access Horizon Information Portal catalogs via the Horizon Information Portal ZServer. Through a partnership with WebFeat, Horizon Information Portal allows to make non-licensed or licensed content available to users, and identify libraries through a variety of authentication schemes, including username/password, IP, and referrer URL. Horizon Information Portal is also built upon on XML and Java technologies. Horizon Information Portal includes a Z39.50 server and client. Supported platforms are Red Hat Enterprise Linux, Microsoft Windows Server and Sun Solaris (www.dynix.com/products/hip/).

7.3.2 ACADEMIC AND RESEARCH SOLUTIONS (MODELS)

7.3.2.1 DAFFODIL
Daffodil is a system for integrated search within the heterogeneous digital libraries of a scientific community, with merging of results. 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 (Fuhr, et al. 2002).

Daffodil is a federated DL system that offers a rich set of functions across a heterogeneous set of DLs. The current prototype integrates over 10 DLs (e.g. Achilles, Citeseer, DBLP, etc.) in the area of computer science, together with other information sources (e.g. Google, Scirus, Ispell, etc.). Since different DLs may contain various pieces of information about the same publications, the federation yields important synergies by combining this information in high quality detailed data for the user. For structuring the functionality, the concept of high-level search activities for strategic support is deployed. The higher levels base on the lower levels, so tactics usually relay on one or more moves, and so on. Based on these levels, strategic support during the information search process is the fundamental concept implemented within Daffodil. High-level search functions, based on the stratagem level, implement this strategic support for the user and provide functionality beyond today’s digital libraries. It is based on five phases, from discovering information resources, retrieving the information, collating the found information in a structured personal library, interpreting the found information through cognitive and collaborative processes, to re-presenting the new conceived information into new information. The workflow usually starts from the discovering through the re-presenting phase, but, of course, switching between the phases will occur often. Every phase consists of different stratagems, which can be combined to reach the information goal.

Daffodil’s high-level search activities, as outlined above, have been designed in close accordance with the WOB model as a range of tools that are integrated into a common workspace environment. The goal of Daffodil’s graphical user interface is to provide an environment for retrieval, searching, and browsing tasks, as well as collation,
organization and re-usage of the retrieved information in a user-friendly way. On the desktop, the set of available stratagems is represented as a set of tools. This design offers a wide range of synergies, starting from the information sources up to the visualization, whereby an optimal, strategy-supported information search process is presented to the user (Fuhr, et al, 2000).

![Figure 7.8 Daffodil Architecture Overview (Adapted from Fuhr, et al, 2000)](image)

### 7.3.2.2 Decomate II

Decomate II is partially funded by the European Commission DG XIII Telematics for Libraries programme. The program partners form an international group, spread over various European countries.

The goal of Decomate-II is to develop an end-user service, which provides access to heterogeneous information resources distributed over different libraries in Europe using a uniform interface, leading to a working demonstrator of the European Digital Library for Economics.

Decomate II aims to make a number of different types of information resources available to its users. These include:

- library catalogues
- bibliographic reference databases
- full-text databases of research papers, theses, reports, etc.
- multimedia/CD-ROM publications
The system handles three distinct categories of these information resources:

- resources stored locally at the user's own library access point
- resources stored elsewhere within the distributed Decomate II environment
- other resources available through the global network infrastructure (e.g. the Internet), including other libraries and document servers, WWW and FTP-sites, and publisher's networked digital archives

The project adopts three-tiered client-server architecture (Fig.7.9). Various clients, brokers, servers and their interplay can be identified. These include bibliographic database servers, search and retrieval servers, document servers, current awareness servers, as well as a knowledge broker. At this level the communication between objects is regulated by protocols such as Z39.50, HTTP, GEDI, FTP, SQL and ODBC.

The Decomate II system will assist the user in specifying a query and in choosing appropriate resource types. For sophisticated users complete control of all Decomate II parameters is available, i.e. the option to choose a specific resource type, location, payment method, delivery method, etc. An Information Broker provides this assistance. It assists both the end users and the Decomate II system itself in providing information that is necessary for their working. The end user is assisted by the retrieval of, for instance, a list of available databases (local or remote) or synonyms to the supplied keyword from a thesaurus. It further assists the system (in casu the query server) by adding database access data and authorization data to the query that was retrieved from the metadatabase.

The metadatabase is an important component of the system. This database contains all information concerning local and remote reference databases, document storage locations, access methods, remote sites, etc. It is the central store for all knowledge concerning resources available through the Decomate II services, and allows the system to manage distributed services and heterogeneous information types.
The basic functionality of DECOMATE is provided through configurable software modules and their specific implementation in the participating libraries. The main component for the end-user will be a standard WWW-browser at the end-user's workstation (e.g. a personal computer), which handles all interaction with the Decomate II system. The workstation can be located within the library or at the user's workplace, linked to the library through the institution's network (Decomate II).

7.3.2.3 THE MARIAN DIGITAL LIBRARY SYSTEM

MARIAN is an indexing, search, and retrieval system optimized for digital libraries. It was developed at the Virginia Tech Computing Center for VT Information Systems, with development continuing at the DLRL. Originally designed for library catalogs, it has been used successfully for collections of varying sizes and structures, and has been enhanced to support digital library and semantic web applications. The MARIAN data model combines three powerful concepts. First, structure and relationships in MARIAN collections are captured in the form of an information network of explicit nodes and links.
Similar graph-based models have proven effective in representing semi-structured data and Web documents, and for translating among different DL systems. Second, MARIAN expands this model by insisting that the nodes and links of a collection graph be members of object-oriented classes. Classes are an organizing method similar to link labels in semi-structured graphs, but are strictly more powerful because they form a full lattice of subsets and can support inheritance. Furthermore, since nodes in the collection graph are instances of information object classes, they can support complex behaviors. In particular, they can support approximate matching of the sort pioneered in information retrieval (IR) systems. Third, nodes or links can be weighted to represent how well they suit some description or fulfill some role.

MARIAN is specialized for a universe where searching is distributed over a large graph of information objects. The output of a search operation is a weighted set of objects whose relationship to some external proposition is encoded in their (decreasing) weight within the set. Weights are used in IR, probabilistic reasoning systems, and fuzzy set theory. The model grounds them firmly in a framework of weighted set operations and extends them throughout the entire MARIAN system. The use of object-oriented data and process abstractions in MARIAN helps to achieve physical and logical independence - common and useful concepts in the database field often neglected in IR. Most current IR systems emphasize the physical level of term indexes and weight metrics, making it difficult to integrate systems at a conceptual level. The flexibility of MARIAN's data model allows it to be used for object-oriented or semistructured databases, knowledge representation, or IR. Its power comes from the smooth combination of a number of successful concepts from such fields and programming languages or artificial intelligence.

The architecture of Marian system is presented in Fig. 7.10.
Middleware provides the tools for structural and semantic interoperability. System and syntactic differences are addressed by wrapping sources with special software modules. The 5SL language for declarative specification of digital libraries is used to describe capabilities of remote collections and their internal document structures. This information feeds data structures inside the mediator and allows semi-automatic generation of wrappers for harvested sources. Extended value-added services like searching, browsing, recommendation, personalization, and visualization are built on the top of the middleware (Gonçalves, et al. 2001)

### 7.3.2.4 THE EUROPEAN LIBRARY (TEL)

The European Library Project (TEL) is partly funded by the European Commission as an accompanying measure under the cultural heritage applications area of Key Action 3 of the Information Society Technologies (IST) research programme. TEL is a collaboration of a number of European national libraries under the auspices of CENL (Conference of European National Libraries).

At the start of the TEL project, the following primary objectives were formulated:

- Provide distributed searching of national collections accessible through Z39.50.
Offer a central index searched using an http/XML protocol. This central index comprises records from other collections harvested.

Investigate the feasibility of a system for integrated access.

Begin with separate testbeds in order to test interoperability of Z39.50 distributed searching and searching of the central index.

Later, combine these separate testbeds into a single testbed: the TEL portal.

Support interoperability using a common metadata model that meets current requirements but is also capable of evolving to meet future requirements.

In the interest of standardisation, TEL adopted SRU as the http/XML search and retrieval protocol by implementing the first draft specifications of SRU. At the start of the TEL project, SOAP did not provide the same advantages as SRU. In addition, SRU has a low implementation barrier compared to SRW. A test interface for SRU access to the TEL central index was implemented using XSLT and JavaScript. The architecture of this concept is shown in figure (fig. 7.11).
The advantages include scalability, functionality, low barrier of entry into TEL, and increased control of functionality for users, data providers and service providers. Last but not least, with the TEL approach there is no longer a need for a central portal. With the combination of concepts like DCX and the SRU extension mechanism, the TEL approach can undergo further development and could become an example for similar projects (Veen and Oldoyd, 2004).

7.3.2.5 INFORMIA

The goals of Informia are threefold: to provide abstraction from information sources, to support information retrieval and integration tasks, and to be scalable, extensible, and customisable. The Informia's internal data model and query language, its common access interface (CAI), and wrappers provide abstraction from locations of information sources and from their different query languages, access interfaces, data models, and schemas.
This gives a uniform view of the entire information space, across different types of information sources. Thus, to the internals of Informia, all information sources appear to have the same data model and query language (i.e. Informia's). Consequently, Informia is openended with respect to the number and types of information sources it can access, despite differences in their data models and query languages.

The described abstraction mechanisms allow different types of information objects and information sources to be treated uniformly, which is essential for supporting higher-level retrieval and integration functions effectively. Informia supports a number of such functions, such as browsing of source meta-information in its meta-information repository (MIR), transformation of queries, and combination of retrieval results.

![Figure 7.12: Informia's three tier mediated architecture](image-url)
Informia's internal architecture, depicted in Figure 7.13, follows the classical mediator model shown in Figure 7.12. It can be roughly separated into three parts: interaction with information sources, interaction with clients and user interfaces (top), and Informia's internals and components (middle).

The user interface is HTML-based and runs in a Web browser. Queries can be formulated at different levels of parameterisation detail. An agent query mode, where the system performs a more comprehensive analysis of results in the background and sends the results back to the user via email, is also available. Client-server communication in Informia is handled via a Common Gateway Interface (CGI) over a TCP/IP connection. Informia can also be operated in command-line mode locally or through a plain TCP/IP connection. The server runs on Solaris and consists of roughly 70,000 lines of C++ code (Barja, et al. 1998).

7.3.2.6 NLM GATEWAY (NATIONAL LIBRARY OF MEDICINE)

The NLM Gateway offers single-point access to multiple NLM knowledge resources. It allows users an overview scan of several NLM systems and will serve as a platform for exploring new functionality in federated database searching. The NLM Gateway is an intelligent gateway system to help some of NLM's users by letting them initiate searches...
in multiple retrieval systems from one interface at one address. The target audience for the new system is the Internet user who comes to NLM not knowing exactly what is here or how best to search for it. The goals of the gateway system are to:

- Provide "first-stop shopping" for an increasing number of NLM information resources,
- Help lead users to information they might not have known was present,
- Offer citations, full text, video, audio, and images,
- Ultimately, offer user profiles to guide searches in clusters of databases.

The NLM Gateway accepts the user's query and translates it as appropriate for different retrieval systems. The Gateway is fully object oriented, written primarily in Java and using the CORBA (Common Object Request Broker Architecture) distributed object-computing infrastructure. At the heart of the Gateway is its ability to accept a user's search and translate it into the series of search statements needed by each of the retrieval systems the Gateway can access. The Data Source Broker accomplishes this process. The search translation algorithms are created by a trained medical librarian on the Gateway team in consultation with the NLM experts responsible for the retrieval systems (Kingsland HI, et al, 2004).

7.3.3 OPEN SOURCES

Open source allows libraries to collaborate on designing and building systems that meet their needs, share them freely, and adapt them for local needs. It provides an alternative to the often-limited "black box" software available commercially.

7.3.3.1 HARVEST

Harvest is a modular, distributed search system framework with working set components to make it a complete search system. The default setup is to be a web search engine, but it is also much more and provides following features (Sourceforge):
• Harvest is designed to work as distributed system. It can distribute the load among different machines. It is possible to use a number of machines to gather data. The fulltext indexer doesn't have to run on the same machine as broker or web server.

• Harvest is designed to be modular. Every single step during collecting data, and answering search requests are implemented as single programs. This makes it easy to modify or replace parts of Harvest to customize its behaviour.

• Harvest allows complete control over the content of data in the search database. It is possible to customize the summarizer to create desired summaries, which will be used for searching. The filtering mechanism of Harvest allows making modifications to the summary created by summarizers. Manually created summaries can be inserted to the search database.

• The Search interface is written in Perl to make customization easy, if desired.

Although the HARVEST architecture is distributed, brokers do not support distributed information retrieval: the indexing software and user interface must run on the same machine. There is, however, a special central broker, called the HARVEST server registry that contains information about each gatherer, broker, cache, and replicator on the Internet. This central index can help users find a suitable broker that is likely to contain the information they are seeking. A broker consists of five software modules: collector, registry, storage manager, index/search engine, and query manager. The collector is responsible for obtaining new information, the registry stores information about each object, the storage manager archives the object on disk, the index/search engine indexes and retrieves the objects, and the query manager provides a World Wide Web interface to the index/search engine (Sourceforge).

7.3.3.2 ARC

Arc (http://arc.cs.odu.edu) service is the first federated search service based on the OAIPMH. Arc was initially released as an experimental service to investigate issues in metadata harvesting in October 2000. It has since then been used in a number of production and research projects.
The software developed for Arc service (http://oaiarc.sourceforge.net/) has been released as an open source system under NCSA-style license in September 2002. It has been used in a number of production services including MetaArchive.org (Halbert, 2003), ncstrl.org, RDN, and snelonline. The Arc system represents a comprehensive solution for communities to harvest, index, and search, as well as for third-party service providers to harvest from Arc.

The Arc architecture is based on J2EE, moreover, the changes required to work with different databases are minimal.

The current implementation supports two relational databases, one in the commercial domain (Oracle), and the other in the public domain (MySQL). Figure 7.14 outlines the major components: Harvesting service; Parser/Indexer; End-user service; administration interface, and infrastructure. Similar to a web crawler, the Arc harvester traverses the data providers automatically and extracts metadata. The significant differences include: normalizing the metadata to allow for complete and accurate searches and exploiting the incremental, selective harvesting defined by the OAI protocol.

The Arc parser/indexer service turns harvested data into internal representation for other services. The Arc parser is designed to be flexible to plug-in any metadata format parser. The Indexer uses full text indexing procedure of the underlying database.
OAI-PMH uses unqualified Dublin Core (DC) (Weibel, Kunze, Lagoze & Wulf, 1998) as the default metadata set, and most Arc end-user services are implemented on the data provided in the DC metadata. The current supported end-user services include simple search, advanced search, interactive search, annotation service, and browse/navigation over search result. Arc has a web based administration interface, which allows users to configure various parameters for harvesting and check harvester logs to handle various error situations such as erroneous XML replies from data providers.

The Arc system is based on Java/Servlet/JDBC technology. All required software is available at open source. Typical prerequisites are Apache Tomcat/Java/MySQL, or if the user selects it Oracle. It is a pure Java-based system and has been tested in Window/Linux/Solaris platforms. The software is designed to run a comprehensive solution, including harvester, indexer, search engine, and data provider. Using only selected individual modules it is possible but will require in-depth knowledge of the system. The software currently in SourceForge is well tested and stable (Maly, etal).

7.3.3.3 KEYSTONE DLS

The Keystone Digital Library (TKL) Suite is a family of Open Source digital content management, portal management and information discovery software packaged together to provide libraries, digital library services.

TKL consists of many modules and pluggable handlers used to add specialized functionality. Among the available modules are OAI harvesters and web harvesters. The existing handlers are specialized to integrate Z39.50 queries, SOAP queries, SQL queries and much more into the TKL framework. Custom designed pluggable components can be freely added.

The Keystone Library is based on XML and XSLT technologies. It currently uses Sablotron as processing engine, and PHP as application glue language. Helper utilities are written in Perl and Tcl.
Here are some of the functions supported by Keystone:

- Portal Creation and Management
- Federated Search Services
- Harvesting Metadata from Remote Repositories
- Link Resolver Services


Keystone Organizer is both a content management service and a portal management service that stores all forms of digital content and facilitates metadata creation to describe that content. It also serves as the knowledge base for the federated search services, the harvesting services and the link resolver services that are elements of Keystone Retriever, as well as for its own portal management services. This means that library staffs have one type of interface and one look-and-feel for all administration services and metadata creation throughout Keystone DLS.

Keystone Retriever is a family of information discovery services that includes federated searching, harvesting services, and link resolution for linking from citations to full text. Index Data is also offering its Keystone Resolver either as a stand-alone product or as part of its family of Keystone Retriever services. At this time Keystone Retriever uses third party electronic serials management systems to provide Keystone Retriever with the data it needs to perform its linking (Indexdata).

**7.3.3.4 OpenSiteSearch**

OpenSiteSearch is a java-based Z39.50 portal system. Its primary use is for libraries to build search interfaces to one or more Z39.50 targets, such as library catalogue Z39.50 servers, vendor database/ejournal Z39.50 servers, or local digital collections.
OpenSiteSearch also includes a component for building databases of digital content that are then searchable by the OpenSiteSearch search interface.

The OpenSiteSearch suite of software provides a comprehensive solution for managing distributed library information resources in a World Wide Web environment. It offers tools that integrate electronic resources under one Web interface, control access to resources, and build text and image databases locally.

![OpenSiteSearch Architecture](image)

**Figure 7.15: OpenSiteSearch Architecture**

The OpenSiteSearch suite includes the following software packages (Fig. 7.15):

- OpenSiteSearchWebZ software
- OpenSiteSearchDatabase Builder software
- OpenSiteSearchRecord Builder application

**OpenSiteSearchWebZ software**

WebZ provides a web-accessible interface to your library's electronic resources, and integrates the access to local and remote information. WebZ also provides the primary infrastructure for the Database Builder and Record Builder applications.

**OpenSiteSearchDatabase Builder software**
The Database Builder software is a complete set of software for building and maintaining information resources locally. Database Builder provides the following features and advantages:

- **Database Indexing.** You control the indexes and access points that make sense to your users and your data.
- **Host Databases Locally.** Host all types of data locally, including full text, abstracts, indexes, and images.
- **Powerful Searching.** Provide a full range of searching capabilities, including keyword and phrase searching, Boolean operators, truncation, wildcards, etc.
- **Powerful Searching.** Provide a fully functional interface that includes search history, record bookmarking, export, and browse.

**OpenSiteSearchRecord Builder Application**

Record Builder is an end-user application built on WebZ technology that harnesses Database Builder's administrative utilities to allow the creation, modification, and maintenance of Newton searchable databases for local collections of such things as images, sound files, web sites, and data. Through its own pre-built web interface, Record Builder offers an alternative to batch updating by providing cataloguers a way to add, modify, and delete single records online using a variety of predefined templates and workforms (OpenSiteSearch).

**7.4 DISCUSSIONS**

The basic premise of unified searching is that there are a large number of resources, which are searchable over the internet using standard protocols and formats. However, all the distributed resources we are interested in may not be searchable in a standard way, nor can they all easily be made searchable. Therefore, in some environments, it makes more sense to harvest a subset of resources, to build a consolidated index, and then to search that consolidated index through the portal at the same time as the distributed resources. The portal, in fact, ends up relying on a mixed model of searching, with some
resources distributed, and some harvested and centralized into a single searchable index. Open source tools are available to provide harvesting and search programs; as a result, there may not be enough of a profit margin to make it worthwhile for commercial product vendors to offer a harvesting solution. There may, however, be a small but significant opportunity for portal vendors or consulting companies to help institutions set up centralized, harvested metadata indexes to complement their library portals (Davies, 2004).

It is easy to connect library search services with products built on common standards like z39.50. The problem with z39.50 is that it's costly to run and not interoperable (z39.50 searching only searches z39.50 databases).

There are pitfalls in deploying federated search. A federated search user interface can provide single-point access to content from multiple sources — very useful in quickly understanding relevant content coverage across sources. By its very nature, it cannot perform the deep, source-specific searches with source-specific syntaxes. Super searchers should see this and help educate user audiences on the right and wrong uses of federated search.

One of the most common work-arounds between the current limitations, both of federated technology as well as limited library budgets, involves categorizing databases by resource types. Bundling content according to subject domain serves the more exhaustive searcher who seeks not merely to satisfy a requirement, but to plum the depths of their discipline in order to put their own theoretical knowledge to the test.

The cost of maintaining a cross-collection search service is of concern, where the cost is comprised of the development and maintenance of client and server components. The process of searching over remote digital libraries under the distributed federated search strategy can be decomposed into the following phases.
Discovery: Discovery, by the client, of the protocols supported by the server, such as query format, search syntax, and request format.

Action: Submitting the request from the client to the server.

Response: Parsing the response from the server and displaying the results (if any) to the end-user.

The principle impediment to the adoption of distributed federated search has been the lack of general agreement and adoption of "standards" for request and response protocol and query language. Outside of the (digital) library community, few online services have incorporated Z39.50. What is needed are technologies that give service providers flexibility in creating a cross-collection search service that suits their specific technical peculiarities while simultaneously lowering the cost to create clients by others to access the service. In the case of federated search, capabilities can be much easier to implement at the client level than comparable harvesting based facilities if the semantics of the search interface and response can be exposed. The federated search client (i.e., the site providing the search facility to the end user) need not be concerned with continuously updating and storing a mirror of the metadata repository at the local site. Instead, one program or component acts as a middleman passing queries to the federated search provider and collecting results in real time. The robustness of the federated search provider and the "cost" associated with end-user time-outs or long delays in response is also to be considered, which can be handled though in the software development of the client (Dong, et al., 2004).

The different service providers have very different kinds of metadata due to the different nature of the materials they hold. Z39.50, although it supports the idea of connecting the same client to different servers does not address the problem of combining metadata from different sources. This problem is left for the client software that has the combining task to do to deal with. Furthermore, Z39.50, by not specifying a particular meta-data format, but acting solely as a transport layer for the differing metadata does not materially assist in the resolution of the problem. Although client and server can attempt to negotiate a common metadata format, in practice, servers often do not have a good selection of
alternative formats available, so that it is quite possible that a single common format will
not be available from servers providing metadata on different data types (Bradley, 1998).
So, there are a significant number of proposed, or model, crosswalks specifications online
(Day, 1996).

As Day says (1996), in the translation from one metadata scheme to another there
inevitably arises the problem of lost attributes. DC provides a base set of metadata,
suitable for resource location only, whereas the specialised databases often have much
more extensive metadata available that cannot be translated well into Dublin Core. The
development of crosswalks is a difficult task, requiring in-depth knowledge of at least the
two-metata standards. Obtaining the expertise to develop a crosswalk is particularly
problematic because the metadata standards themselves are often developed
independently, and specified differently using specialized terminology, methods and
processes. Furthermore, maintaining the crosswalk as the metadata standards change
becomes even more problematic due to the need to sustain a historical perspective and
ongoing expertise in the associated standards (Bradley, 1998).

7.5 CONCLUSION

The commercial portal solution models are based on a cross-database search capability,
but they may not be as turnkey as those offered by ILS vendors. These products may
also: (1) offer a limited range of pre-configured support for commercial information
resources; (2) require more custom work to be able to support the full range of licensed
information resources that an organization will require; or (3) not offer OpenURL
services which facilitate fast access to licensed electronic full text. However, depending
on an organization’s needs, particularly in a limited corporate environment, one of these
products might just fit the bill. They are also expensive in terms of the initial purchase
price, their ongoing maintenance charges and the staff resources required to make them
work. There are also some open source components for library portals available for those
who are ambitious or adventurous. None of these open source components will give you
everything in a library portal, but under some circumstances, they may provide a starting
point for custom development. An OCLC product now available as open source,
SiteSearch supports search of multiple databases, using Z39.50 technology and a powerful and customizable web interface (the WebZ gateway). IndexData, a Danish firm that has been heavily involved in the Z39.50 community, also provides source code for Z39.50-based distributed portal search in an offering called Keystone. Neither offering, however, includes the gateways or connectors to link to non-Z39.50 databases (though IndexData is able to supply additional components and customization for a fee). For organizations looking to build a primarily Z39.50-based portal, either product might provide a good starting point for development. However, one is left with the uneasy sense that, in choosing solutions based internally on Z39.50, one might well be choosing yesterday's technology to solve today's needs.

One of the most original non-commercial approaches to a library portal has been that of The European Library (TEL). TEL is a new service promoting cross-searching of the collections of European national libraries. Through a clever use of the SRU protocol (Search and Retrieval via URL, one of a new generation of Z39.50-related protocols), the TEL project has offloaded most of the processing of the portal onto the browser. Javascript allows the sending of multiple SRU-encoded search requests to different information resources, and XSLT processing provides the appropriate formatting of the interface and search results. All that is really required outside the browser is a gateway that can translate back and forth between SRU and Z39.50 and can reformat the records that come back from a Z39.50 service (such as MARC21) into XML. While this service won't provide cross-searching of many commercial resources, and it is only just moving from a testbed to an operational service, it offers an interesting and innovative open source solution that may meet the specific needs of some library consortia.

These products are not without their problems, and there remain significant issues requiring further development. For example, although a huge range of resources can be incorporated into the portals, full integration is best achieved with those resources already complying with the relevant protocols. Even then, differences in metadata standards and data structures impinge on their ability to present a unified set of results – features such as removing duplicates being particularly vulnerable. The interfaces are not always
intuitive, and while most provide the ability to customise these – as well as a range of personalisation features – their implementation adds to the workload. They are resource-hungry not only in terms of the staff time required to maintain them, but also in terms of generating vastly increased levels of traffic to external resources. These models are not perfect by any means, nor are they comprehensive in the range of resources they can access. They do, however, show that the problem of providing seamless access to the wide range of resources now held and accessed by libraries is finally being addressed
7.6 REFERENCES


Chapter 7: Unified Intranet Models...


