1

Large Scale Distributed Knowledge Infrastructures

Wojtek Sylwestrzak
Interdisciplinary Centre for Mathematical and Computational Modelling,
University of Warsaw,
Poland

1. Introduction

Many tend to believe that the current process of development of science is one of the best established and most traditional research procedures that has evolved over years to reach its current almost ultimate perfection. Well, think again. In recent years we are witnessing the beginning of a paradigm shift in the scientific research conduct process. The traditional one-man-show way of performing research, most common in the first half of the 20th century is slowly beginning to disappear in the more advanced disciplines. Instead, modern research is increasingly based on the concepts of massively distributed collaboration, resource sharing, open access to knowledge and achieves new qualities through, often interdisciplinary, compilation research and data reuse. The conversion is becoming possible due to the rapid Internet technologies development of the last decades, electronic communication proliferation but also due to the subtle and indirect influence of the open access culture, originating from the open source software development but popularised through the Internet, and the resulting transformation in the human perception of notions of progress and scholarly work. The European Commission recognizes the importance of open access in research and, in anticipation of the open access mandate for its future programmes, establishes an OpenAIRE open repositories structure to house its future funded research output. One of the key prerequisites to the eventual success of the transformation process is a broad access to knowledge and, consequently, the existence of adequate tools and technologies making this access possible and effective. The availability of such technologies, however, is still lagging behind.

Due to the diversity of information they comprise, digital libraries are often considered to have become one of the major web services (Liaw & Huang, 2003). They are also assumed to be among the most complex and advanced forms of information systems, and interoperability across digital libraries is recognised as a substantial research challenge (Gonçalves et al., 2004; Candela et al., 2007). Moreover, it is commonly expected that the today's library, archive and museum services will converge in the future digital content repositories (Marty, 2008). Most of the current research activities in this area relate to metadata object description, inter-object relations (semantic similarity, citation references, near-duplicates identification, classification), text and data mining and automated content processing, user personalisation and community services, and large-scale distributed architectures and infrastructure interoperability and performance. In this chapter, we will

analyse several examples of the current state-of-the-art in digital library and repository infrastructure technologies. Only the very recent years have seen the rapid increase of the pace of research and development of adequate technologies necessary to build large scale knowledge management and content provisioning infrastructures to support the individual advanced digital libraries and repositories and the associated automated content analysing systems. In order to be able to find a common ground for evaluation and comparison of different solutions, a universal formal description framework is required. While there is still no single formal digital library reference model in wide use, several approaches have been recently proposed, most notably Streams, Structures, Spaces, Scenarios, and Societies (5S), DELOS Digital Library Reference Model, and MPEG-21, although the latter, aimed at defining an open framework for multimedia applications, is not directly related to digital libraries. The Reference Model for an Open Archival Information System (OAIS) provides a framework to address digital preservation.

The 5S model, proposed in a PhD dissertation by Marcos André Gonçalves, introduces abstract concepts of streams, structures, spaces, scenarios, and societies providing means to define digital library objects, services and other entities. In the model, the streams are understood as simple sequences of arbitrary items used to represent serialized content, the structures are labelled directed graphs, organizing the streams, the spaces are seen as sets with associated operations on them, the scenarios are sequences of actions performed in order to accomplish functional requirements, and the societies are defined as sets of entities and activities, and the relationships among them. (Gonçalves et al., 2004)

Based on these abstract notions, the model proposes a formal ontology defining the fundamental concepts, relationships, and axiomatic rules governing the digital libraries. Contrary to other approaches, the 5S has the ambition to describe digital libraries in an axiomatic formal way. The model can be used equally as a base to build a digital library taxonomy, a quality model, or to perform a formal analysis of specific case studies. The basic concepts of the 5S models are summarised in Table 1.

Streams are sequences of elements of arbitrary types (basically bitstreams), representing serialised content objects (which of course may be text, being a stream of characters) or data transfers (like in streaming video). 5S differentiates between "static" streams, which simply correspond to stored data, and "dynamic" streams, which are data in transfer.

S5 defines s*tructures* as the means to organize and arrange components of an entity. The purpose of structuring a document is to orientate the reader in the information. A typical representation of a structure of a digital text object is its embedded markup (for example in an XML file). Similarly, relations or graphs structure raw data, or hyperlinks define a structure of a web site.

Scenarios are sequences of events denoting transitions between states of the system. They can be seen as ordinary use cases, describing desired external behaviour of the system from end users' perspective. They provide functional description of the system and therefore may be considered vital in the process of its design. Since the scenarios can be perceived as user-level service contracts, in many cases they may provide enough specification for system prototyping purposes. Each scenario describes a part of the system's functionality in terms of what happens to the streams, to the structures, and in the spaces through a sequence of events. Scenarios allow to quickly comprehend the complexity of a digital library and they are a common way of specifying system's functional requirements in its design phase. Moreover, the scenarios are one of the most intuitive ways of describing the system's behaviour.

Models	Primitives	Formalisms	Objectives
Stream Model	Text; video; audio; software program	Sequences; types	Describes properties of the DL content such as encoding and language for textual material or particular forms of multimedia data
Structural Model	Collection, catalogue; hypertext; document; metadata; organizational tools	Graphs; nodes; links; labels; hierarchies	Specifies organizational aspects of the DL content
Spatial Model	User interface; index; retrieval model	Sets; operations; vector space; measure space; probability space	Defines logical and presentational views of several DL components
Scenarios Model	Service; event; condition; action	Sequence diagrams; collaboration diagrams	Details the behaviour of DL services
Societies Model	Community; managers; actors; classes; relationships; attributes; operations	Object-oriented modelling constructs; design patterns	Defines managers; responsible for running DL services; actors, that use those services; and relationships among them

Table 1. The 5S Digital Library Model (source: Wikipedia)

A digital library's role is to serve the information needs (collecting, preserving, sharing, etc.) of its *societies*. Therefore, a society can be seen as the highest-level component of a digital library. In the 5S model, it is defined as a set of users, computers or software and the relationships among them and between them and their activities. Examples of specific human societies in digital libraries include learners, teachers, patrons, authors, publishers, editors, maintainers, developers, and the library staff. The traditional role of hardware and software members of digital library societies have been to support and manage services used by humans, but recently, they can increasingly be perceived as the users themselves (e.g. processing content served by another software). Also societal governance issues, including policies for information use, reuse, privacy, ownership, licenses, access management, and information integrity, are of fundamental concern in digital libraries.

A *space* is a set of objects complete with operations on them that obey specified constraints. It defines logical and presentation views of several components. The concept of spaces is particularly useful because of the generality of its definition. It can be used when a feature of a digital library cannot be represented by any of the other four basic concepts of the 5S model. A space, as defined in the 5S, corresponds to a mathematical notion of a space (including specific cases such as topological, metric, linear or vector space). A document space or a virtual collaboration space may serve as examples in the digital library domain.

The DELOS Digital Library Reference Model (Candela et al., 2007) defines a three-tier digital library domain view, differentiating between a digital library, being the final system actually perceived by the end-users as being the digital library, a digital library system,

being the deployed and running software system implementing the digital libraries, and a digital library management system, being the generic software system supporting the production and administration of digital library systems and the integration of additional software offering more refined, specialised or advanced facilities. It also defines a number of digital library domain entities and relations between them, such as content, users, functionalities (actions), polices, etc. The DELOS Reference Model seems to be primarily focused on describing an autonomous digital library and does not cover the interoperability in a distributed environment.

The MPEG-21 standard, from the Moving Picture Experts Group is ratified in the standards ISO/IEC 21000 - Multimedia framework (MPEG-21). Its primary purpose is to define an open technology framework needed to allow users to exchange, access, consume, trade or manipulate multimedia in an efficient and transparent way. The standard is based on a definition of a Digital Item (DI), as a fundamental unit of distribution and transaction. A Digital Item is defined as a structured digital object with a standard representation, identification and metadata. The Digital Item is the digital representation of an asset and an entity that is acted upon within the MPEG-21 framework. Parties that interact with the Digital Items in the MPEG-21 environment are categorized as Users acting in different roles. The aim of MPEG-21 is to provide a set of tools allowing Users to interact between themselves and the objects of that interaction are Digital Items. These User-to-User interactions may include providing, modifying, archiving, or consuming content etc. In order to allow various parts of the standard to be used autonomously, MPEG-21 is organized into a number of independent parts, including: Digital Item Declaration (DID), Digital Item Identification (DII), Intellectual Property Management and Protection (IPMP) for license enforcing, Rights Expression Language (REL), Rights Data Dictionary (RDD), Digital Item Adaptation (DIA), or Digital Item Processing (DIP). While originally promoted for the media industry, mostly for its strong support of Digital Rights Management (DRM), it has also found some ground in the digital library domain, namely the aDORe project developed by Los Alamos National Laboratory (Van de Sompel et al., 2005).

2. Historical examples

The concept of a distributed content provisioning infrastructure is nothing new or unique to digital libraries. The Internet has seen successful, truly distributed, not centrally managed, large scale content infrastructures in operation for years. Internet Usenet, developed from the general purpose UUCP architecture (Novitz & Lesk, 1978) in the early 1980s, and still in massive use, may be one example here. Usenet is distributed among a large, changing and evolving conglomeration of servers that are loosely connected in a variable, yet robust mesh. Its servers store and forward messages to each other, and also provide read-write access to the clients (Salz, 1992). The servers may act in various roles, such as feeders, stampers, transit or storage servers. At the same time the Usenet content is structured into a hierarchy of groups with delegated management authorities. Current Usenet contents typically include text messages, images, computer software, as well as other multimedia objects. Infrastructure-wise, Usenet is governed by a set of protocols for generating, storing and retrieving its contents and for exchanging it among its widely distributed readership (Horton & Adams, 1987). The backend Usenet infrastructure employs the peer-to-peer architecture, that was rediscovered under that name only years later.

Another example of a truly distributed, massive information management system may be Archie, considered to be the first Internet search engine (Sonnenreich, 1998). Created in 1990, before the World Wide Web, Archie was the Internet search engine for the FTP sites' contents. While not directly dealing with content delivery, Archie focused on harvesting and indexing the content and providing search service. Similarly to Usenet, the independently operated, heterogeneous Archie servers, together with the archives' mirrors system created a complex grid-like infrastructure, in which they exchanged data about the harvested FTP servers and provided services to the users, several years before the "grid" term was coined and grid computing reinvented.

A more sophisticated example may be the Harvest Information Discovery and Access System, developed in 1994 (Bowman et al. 1995) part of which evolved over time into Squid Proxy Cache (Wessels, 2004). The Harvest infrastructure consisted of several, possibly replicated, distributed components: gatherers, brokers, indexers, replicators and caches. The gatherers, in their basic role being simply web robots, shared also some characteristics with the modern OAI-PMH servers/harvesters. At the time of the Harvest's implementation, there was still no HTTP/1.1 and the If-Modified-Since GET was not widely implemented. The Harvest system proposed that HTTP server managers (the providers) run gatherers periodically against their public contents, and this way maintain incrementally updated content summaries, bypassing the need of futile queries downloading all the contents, regardless if they changed or not. The brokers would then retrieve the information from the gatherers (or other brokers) through SOIF protocol, and would invoke the indexers through a unified interface to index them. The interface would allow for different broker and indexer implementations to communicate. Harvest provided also a weakly consistent, replicated wide-area file system called mirror-d, on top of which brokers were replicated. Finally, Harvest included a hierarchical object cache, with each cache server communicating with its neigbours and parents with an ICP protocol (and later through cache-digests). The Harvest system provided a complete, scalable, distributed content delivery and replication infrastructure.

The early digital library systems were largely centralized monolithic databases or search engines with minimal scalability, focused on providing access to bibliographic information. Often originating from libraries' online public access catalogues (OPAC), they soon were reaching their expandability limits. The early digital libraries were usually closed systems, accessible only to human users, through either text terminals, Z39.50 protocol or through their specific web user interfaces. In their evolution they changed from single library systems to large federated digital libraries, but their basic constraints remained.

While the resulting single-purpose centralized institutional systems were adequate for their intended usage, they also proved useful for examining issues such as better understanding what functionality a digital library should possess, and determining which interfaces users find most appropriate. At that stage, performance and scalability were still of secondary concerns.

An alternative approach was assumed by the digital libraries originating from open access repository systems (institutional, thematic etc.). The repository systems focused on storing and making available for download full texts, usually of research papers, either published or pre-prints, most frequently represented in PDF. A typical early open repository did not offer, or offered only limited full text search capabilities but its metadata were made available for batch downloading through an OAI-PMH interface, thus enabling federated

searches to be performed by meta search engines *(Pieper & Wolf, 2007*), and providing a use case for future distributed infrastructures development.

3. Digital library infrastructures

As already stated in the introduction, it is believed that digital libraries are going to be among some of the most complex large-scale system infrastructures of the future. Modern digital library infrastructure systems feature service oriented multi-tier architectures with a loose coupling of modules. This component-based approach allows tailoring of individual deployments through the selection and replacement of required modules. Components are more natural units and easier to reuse than complete monolithic implementations. They also provide an alternative pathway to digital library federation and scalability, as distributed implementations are easier to implement with components running autonomously on different machines.

It is observed that, while the basic textual information search and retrieval techniques have been already mostly mastered, relatively well understood and implemented, many of the current challenges lie in generic data reuse and the associated methodologies. While it may seem simple at the first glance, the mere diversity of the possible data types and structures, not to mention the different access methods, make it an enormously complex problem. Also, knowledge representation, while in many aspects already addressed in theory (e.g. semantic network concepts or topic maps representation), is still in its infancy as far as the practical large scale usable implementations are concerned. Besides, knowledge discovery and extraction techniques, finding interrelations between heterogeneous objects of often different provenance, similarity analysis and compound objects handling are still not standardized. An additional challenge is posed, surprisingly, by the increase in the growth of the volume of scientific output. It is believed that in the not so far future, machines and automata will become the primary consumers of scholarly publications, as the quantity of produced information will sooner or later render humans incapable of effectively absorbing it without automated assistance. Therefore, already now it is anticipated that the knowledge, whether represented in the form of traditional publications, data, or more complex relations thereof, should be stored primarily in machine-friendly formats to best allow for its subsequent mass processing.

In general, the two primary challenges of all large distributed digital library infrastructures are the requirement to integrate the heterogeneous data and the system's true multidimensional scalability. Both are the necessary prerequisites allowing for subsequent efficient processing and analysis of the distributed content. Scalability is the base feature on which other desired qualities of a digital library system depend.

Scalability, as a general property of systems, is difficult to define (Hill, 1990). Traditionally, it is understood as the system's ability to be enlarged and to handle increased load in a graceful manner (Bondi, 2000). In the context of digital library systems, scalability in a multitude of dimensions is required, not only limited to the system's performance but also its extensibility and manageability. In order to fulfil the evolving requirements, and at the same time to remain competitive on a functional level, any large scale digital library system has to be based on a dynamic framework, undergoing constant development.

In a large scale digital library context, the system's extensibility is achieved primarily through the infrastructural approach. A distributed open infrastructure allows for a multidimensional scalability by a modular system's design, where different functionalities

can be realised through implementation of new or alternative modules. In a large scale distributed environment, the communication and overall management may become an issue. Distributed, component-based architectures are obviously more scalable than monolithic architectures. With a component-based approach, it is be possible to install a simple digital library system quickly and inexpensively on a commodity hardware. At the same time it is possible to deploy a complex system with custom functionality, high availability, and a replicated, distributed architecture within the same infrastructure. A digital library, requiring a specialized capability not supported by the system, needs to customize only the adequate components, and can reuse the bulk of the infrastructure without modifications. At the same time, a digital library without the need for a particular feature can omit components for that service in its deployment. A component corresponding to a particular performance challenge can be upgraded, replicated, or distributed, with minimal modification elsewhere in the system. A component-based approach also proves advantageous with heterogeneity issues, that can equally be present in content types but also in capabilities or search mechanisms.

While a Service Oriented Architecture (SOA) allows to build firm and extensible infrastructure systems, it imposes certain overhead both in the development cost and in the system's (communication) performance. This is alleviated by a more lightweight Resource Oriented Architecture (ROA) approach, which generally reduces the time to implement a system and also in many cases may result in a lower communication overhead.

Resource Oriented Architecture, however, does not offer true scalability, and may render large scale systems continuous development difficult to manage. To this end, the best solution may be a hybrid approach, offering well architected and tested stable SOA for the core services in the backend, and ROA for more rapid implementation of the front end web based services.

An early Open Digital Library framework has been proposed in 2001 (Suleman and Fox, 2001), in an attempt to define component interfaces for functions such as searching, browsing, combining metadata of different provenance, reformatting metadata, or providing a sample of recently added items. A prototype implementation has been prepared and successfully deployed.

aDORe is an infrastructure system developed at Los Alamos National Laboratory aimed at managing collections of objects stored in OAI-PMH enabled repositories, and making them available to external applications. The objects are represented in the system in the MPEG-21 Digital Item Declaration Language (DIDL) format. The Digital Objects in aDORe can consist of multiple datastreams as Open Archival Information System Archival Information Packages (OAIS AIPs), stored in a collection of repositories. The location of the repositories is kept in a Repository Index and the identifiers of each OAIS AIP, its represented object, and the relevant OAI-PMH repository where the object is stored, are contained in an Identifier Locator. The Identifier Locator is typically populated through OAI-PMH harvesting. An OpenURL Resolver provides OAIS Result Sets (presentable digital objects) to NISO OpenURL requests, and an OAI-PMH Federator exposes aDORe OAIS Dissemination Information Packages (OAIS DIPs) to OAI-PMH harvesters. Some concepts of the aDORe architecture may seem to resemble the Object Brokers of the Harvest system. Notably, aDORe makes an extensive use of MPEG-21 specification, which is rather unusual for a digital library system, as the standard seems to be mostly promoted by the media industry, interested in its DRM capabilities. The distributed storage in multiple OAI-PMH repositories

should make aDORe a relatively scalable system on the storage level. While the system is basically intended for local deployment, its modular architecture should also make it easier to be implemented in a distributed environment. While a centralised registry in the form of Identifier Locator may seem to create a bottleneck and a single point of failure, the system is capable of supporting tens of millions of documents. Nevertheless, generally, the component-based design of aDORe makes it possible to migrate between different implementations of the software modules without affecting the overall system's functionality. (Van de Sompel et al., 2005)

SeerSuite is a set of tools constituting a framework of an academic digital library built automatically by retrieving scientific contents found on the Web. SeerSuite tools are used by a couple of Internet services, most notably by CiteSeer^{X}, an index of publications in computer and information science and related areas such as mathematics or statistics, comprising over one million objects (Teregowda et al., 2010). The tools support full text indexing of the harvested contents and automatic citation extraction, indexing and linking. The basic components of the suite include a crawler, text and metadata ingestion and extraction tools, XML and fulltext repositories, object and citation databases, full text index, user interface, personalisation database and workflow supporting scripts. One of the design goals of SeerSuite, replacing a previous CiteSeer software, was a possibly high level of content processing automation. Once found by the crawler, a research paper, usually in PDF or PS format, is harvested and its text payload is extracted and analysed. At this stage, the text is being filtered to avoid indexing non-academic documents, and metadata, including citations are automatically recognised and extracted. The document is assigned a unique identifier, and duplicates are identified and handled. All the generated information is stored either in a database or in a form of XML in the repository. Also a copy of the original retrieved document is kept, and the citation database is updated accordingly. The files in SeerSuite are versioned and time-stamped and the full text index is incrementally updated taking this information in the account. This approach allows to avoid costly rebuild of the whole index each time a document is added or changed. Independently, MyCiteSeer portal keeps user profiles, portfolios and queries and supports building private collections, social bookmarking, user alerts and other similar personalised services. Individual services exchange data access object (DAO) information, or communicate through SOAP or REST interfaces.

A differentiating factors of SeerSuite include extensive metadata extraction tools and a strongly synchronised standalone citation graph service. While the system is very focused and remains centralised, a notable design effort has been taken to decompose it into a collection of autonomous tools (services) that can be potentially used on their own as building blocks of a future distributed digital library infrastructure.

A more universal infrastructure system, YADDA (Yet Another Distributed Digital Archive), designed along the lines of open knowledge environment paradigm, originated as a replacement software for Elsevier's ScienceServer platform. To this end, not only the extensibility but also high performance and high scalability were among its main design goals. For a number of years, ScienceServer was the primary (and the only) platform providing online access to journals to Elsevier's subscribers. In this time its one instance provided access to several million fulltext articles from Springer and Elsevier to all Polish academic and research institutions. Elsevier's announcement to terminate the development and support of the platform led to the necessity of looking for an alternative solution. It was

decided that a new, open system would be developed, not only meeting the functional and performance requirements of the high traffic journal provisioning platform but also capable of supporting in house developed bibliographic databases and repositories, open access content, books and other media, and integrating them in a single unified point of access for the end users. (Zamłyńska et al., 2008)

Contrary to many other digital library management systems, the YADDA suite models a much broader environment beyond the simple content items, and YADDA objects equally include object hierarchies, compound objects, actors, roles, licenses or institutions, and relations between them.

The basic YADDA environment employs the web services framework acting as a collection of APIs to services that can be accessed remotely. YADDA infrastructure consists of a set of core services including Object Storage Service, Metadata Storage Service, Structured Browse Service, Index Service, Workflow Manager Service, and AA Service and a number of extension services.

The Object Storage Service intended to store large volumes of mostly binary data supports full synchronization and versioning. In addition to that, it supports hierarchical data storage, in a manner similar to a traditional filesystem. Specific backends of the Object Storage Service allow to access objects using either YADDA-specific optimised interfaces, or well-known standard protocols like FTP, HTTP, or rsync.

The Structured Browse Service is an OLAP cube concept based module for managing relations between stored objects. The service allows to define relations and to query their data. It furthermore supports a number of specific non-standard field types such as enumeration string fields or bit sets with fast mask queries, which are particularly useful in the case of license credentials. The service allows for effective querying of aggregated data, or fetching the count of objects fulfilling specific search criteria. It supports lazy materialization of aggregated views, in which the results of predefined queries are materialized and the materialized tables are updated when the contents are accessed. The service also allows to define indexes on both relations and aggregated views.

The Index Service provides a flexible, fast and effective full-text search capability without restrictions on the type of the indexed documents. Depending on a particular setup, a number of Index Service instances can co-exist simultaneously, for scalability, load balancing, or reliability purposes. Index groups can be defined and searched in a single query. The service is transactioned, and its performance can be improved by splitting the index and/or storing it in memory. The service provides effective iteration through search results and filtering of frequent logical conditions in queries. Frequently executed, big boolean queries can be defined as filters which, when used, speed up searching up to 10 times. Currently, two different implementations of the YADDA Index Service API exist, with different functionalities, based on Lucene and SOLR, that can be used interchangeably.

The Workflow Manager Service is a subsystem responsible for scheduling and executing predefined tasks on the objects stored in the repositories. The tasks are organized in "processes", which define the sequences of the events. A process consists of nodes, each being a relatively simple operation awaiting an input and producing its output. During its execution, a node can access other YADDA services, and invoke associated actions. A simple example of a process node accessing a service may be a metadata reader, which takes an object's ID as an input, queries the Metadata Storage Service, and provides the object's content as the output. Sets of predefined nodes can be configured into chains and executed.

Processes can be run manually, can be scheduled or can be triggered by operations on other services, particularly by changes in the Metadata Storage Service.
The Metadata Storage Service (formerly the Catalogue service) is primarily responsible for storing rich metadata. This service provides synchronization, version control and search for metadata objects meeting specified criteria. A number of processes, defined in the Workflow Manager make use of the Metadata Storage Service data, including:

- A general indexing process, retrieving object hierarchy information (for example an article belonging to a volume of a journal published by a publisher) from the metadata structure elements and storing it in particular relations of the Browse Service (hierarchical relations and contributor-publication relation) and in the fulltext index.
- A metadata extraction framework, which runs as a multi-level process. First, a PDF file or an image is converted to a set of characters with assigned locations through optical character recognition. Next, the page layout is discovered and finally particular zones are tagged as title, author, abstract, keywords, references, etc.
- Citation parsing and matching by a rule-based citation parser. A network of citations is created by matching parsed citations with entries in the repository.

The Authentication and Authorization Service is designed as an open and distributed system, providing sophisticated security that allows to support a network of repositories and clients. It implements a complex yet transparent authentication and authorization layer based on XACML and SAML standards. One of the service's most significant features is the separation of authentication, authorization and policy enforcing functions. Thus it is possible to separate authority providers (users databases, client institutions etc.) and content providers (repositories which rely on the authentication data provided by authority providers, and which serve particular content). Furthermore, the service allows to propagate trust relationships in the network of repositories and clients (so-called "webs of trust"). Since the service uses XACML as a policy definition language, it is possible to define a variety of rule-based access policies in a flexible way. Each YADDA service supports Authentication and Authorisation Service based security layers, which allows to assign specific licenses to each object maintained by these services. Using XACML, it is possible to define flexible ways of limiting access to all objects according to their particular licenses.
Besides the core services, the YADDA environment contains a number of optional extension services, including a categorisation service, a similarity service, a citation extraction service, a reference service (citation graph and index) and a choice of interface services, including web GUIs. A standalone tool, DeskLight (being in fact a YADDA instance itself) allows for content publishing and online or offline collaborative content curation. All YADDA services and tools, particularly the YaddaWEB user interface and the DeskLight application are fully multilingual – with full Unicode and left-to-right and right-to-left writing support. The underlying data model allows to maintain multilingual information about any given element. For example a single publication can have its corresponding abstracts or keywords in a number of languages at the same time.
A number of tools have been developed for loading and bulk converting imported data from proprietary formats to the internal YADDA format or to export the data using standard formats and protocols (like OAI-PMH).
Formal service contract definitions allow user-specific security to be introduced to any service. Repository descriptors in the form of XML files provide descriptions of all services available in a given repository, allowing automated discovery and connection. Besides, the

service contract definitions allow to automate the service concertation process, service conformance testing and troubleshooting.
A proof of the YADDA environment flexibility and its down-scaling capabilities may be its embedded instance, DeskLight, which consists of custom lightweight implementations of the core APIs together with a couple of specialized editing tools and a GUI, all packaged as a java application intended for desktop use. DeskLight may be used as a local metadata editor, synchronizing the data with other DeskLight and YADDA server instances, thus allowing for efficient collaborative editing.
YADDA is a remote facade based service system, rendering it indifferent to the underlying inter-service communication protocols. The approach allows the services to be easily used in different deployment scenarios, ranging from tightly-coupled high performance scale-up installations to extensive, large open standards based distributed systems with service-level redundancy. The resulting flexibility of YADDA allows for its various components (services) to be easily included individually or in groups in other digital library infrastructures. The feasibility of this approach has been confirmed by diverse employment of various YADDA components in a number of different systems and environments.
Besides the original Elsevier and Springer journals application, individual YADDA services have been used in a number of different deployments, four of which are briefly presented below: DRIVER's Network Evolution Toolkit (used in several individual installations itself), OpenAIRE service, European Digital Mathematics Library, and BazTech database.
D-NET (DRIVER Network Evolution Toolkit) is a Service Oriented Architecture (SOA) based software suite created for the DRIVER digital library, aggregating the contents of the European research open repositories. The web services based suite allows to build a distributed infrastructure composed of a number of services, including an index, browse, store, OAI-PMH, collection, transformation, similarity, citation, text engine service and a number of D-NET specific orchestration services such as authorisation and authentication, information or manager service. Notably, version 2 of D-NET supports compound objects handling. Depending on a particular instantiation of the software suite, D-NET services can be combined into larger applications. The same services can be also shared among different environments. Individual services active in a D-NET instance register with its Information Service, allowing other services to discover them. The D-NET system's workflow is managed by a dedicated manager service responsible for executing other services in a desired sequence. (Manghi et al., 2010; D-NET: release of the DRIVER Software, http://www.driver-repository.eu/D-NET_release) D-NET successfully employs a number of YADDA infrastructure components, including its index, object store, authorisation and authentication, citation and referencing, and similarity services.
Another digital library system, where YADDA modules are being used is EuDML - the European Digital Mathematics Library (Sylwestrzak et al., 2010), currently in prototype. The EuDML system will consolidate the European information space in mathematics, harvesting national and local digital libraries and repositories and unifying and enhancing their metadata. The system, which will follow Service Oriented Architecture, will reuse existing technology but also develop new modules acting as services. The EuDML background services will include metadata harvester, registry and conversion manager, storage, search and browse, AA, and workflow manager. Besides the core, there will be a number of enhancement tools and services including citations manager, content annotation, author matching, data enrichment, personalisation and user interface with accessibility features. EuDML will use structured browse, index, storage, AA and citation services from the

YADDA environment. It will also use REPOX and MDR services developed for Europeana for metadata harvesting, mapping and managing. (Reis et al., 2009) The primary design goals of the EuDML platform are its extensibility, allowing easy addition of new services (and content), and its scalability in many dimensions, including the content's volume, content's structure, number of services, number of concurrent users, etc., without performance or reliability degradation. To this end, the system will be designed in a modular, distributed architecture, allowing to replace, upgrade or provide alternative services realizing the same or similar functions in the future versions.

OpenAIRE is a European initiative to provide an open-access publication repository infrastructure for scientists conducting research fully or partially funded by the European Commission. It is intended that, after leaving its pilot phase, OpenAIRE will provide an infrastructure to mandate open-access to all output of any research funded by the European Union, including textual publications but also data and multimodal results. Similarly to DRIVER, OpenAIRE uses selected YADDA services, including the Object Storage, Index and the Authentication and Authorisation Services. Users can upload their publications either to a central OpenAIRE repository (run by CERN), to the supported thematic repositories, or to their local open-access repositories and register the upload with the OpenAIRE system through a portal available at http://www.openaire.eu/.

A different application scenario for YADDA is BazTech - the citation database of Polish research journals in technology and related disciplines. While the BazTech database is centralized, its creation and updating process is highly distributed, and organized in a hierarchical manner. BazTech is maintained by a consortium of the libraries of Polish technical universities. In each library, its employees update the data in a local copy of the repository. The metadata are edited and the fulltexts uploaded using the DeskLight version of YADDA. The new contents are supervised, and when approved, the local repositories are merged together to form the eventual central BazTech database, running on another YADDA instance. Similar YADDA setups are used by a number of other project with similar usage characteristics.

The diversity and the multitude of different YADDA services deployment scenarios may serve as a proof, confirming that an open digital library service infrastructure concept is feasible not only as a prototype but also it excels in real life heavily used production systems.

4. Conclusion

Digital libraries related technology has undergone significant changes in the recent years. While the evolution path from the simple, autonomous, single-purpose monolithic systems towards multi-tier open infrastructural solutions may seem obvious, a lot remains still open for future research and subsequent development. There is yet no single widely adopted and mature enough production quality solution that would fully warrant adequate development potential beyond the immediate needs. In fact most of the currently deployed solutions constantly lag behind the requirements and expectations. Similarly, there are no well established flexible, performant and scalable digital library service to service communication standards, besides the basic protocols mostly pertaining to metadata transfers.

Besides the technology, also our understanding of user-centric design approach changes from the initial perception that service consumers are human actors towards seeing them

increasingly as other services processing the available textual or digital data and generating new semantic knowledge and pieces of information. The key to a successful and future-proof digital library system seems to lie in basing it on a standardized, open infrastructure that would be able to adequately expose content for automated machine-processing, much of which remains yet to be seen.

5. References

Alexander Ivanyukovich, Maurizio Marchese, Fausto Giunchiglia, (2008). ScienceTreks: an autonomous digital library system. *Online Information Review*, Vol. 32 Iss: 4, pp. 488-499

Bondi, A.B. (2000). Characteristics of scalability and their impact on performance. *Proceedings of the 2nd international workshop on Software and performance*. Ottawa, Ontario, Canada, 2000, ISBN 1-58113-195-X, pp. 195–203

Bowman, C.M.; Danzig, P.B.; Hardy, D.R.; Manber U., & Michael F. Schwartz M.F. (1995). The Harvest Information Discovery and Access System. Computer Networks and ISDN Systems, Vol 28, Issues 102, pp. 119-125. doi:10.1016/0169-7552(95)00098-5

Candela, L.; Castelli, D.; Ferro, N.; Ioannidis, Y.; Koutrika, G.; Meghini, C.; Pagano, P.; Ross, S.; Soergel, D.; Agosti, M.; Dobreva, M.; Katifori, V. & Schuldt, H. (2007). The DELOS Digital Library Reference Model – 0.98, p. 20

Emtage A. & Deutsch P. (1992). Archie - an electronic directory service for the Internet. *Proceedings of the USENIX Winter Conference*, pp. 93-110, January 1992.

Gonçalves, M.A.; Fox, E.A.; Watson, L.T. & Kipp, N.A. (2004). Streams, Structures, Spaces, Scenarios, and Societies (5S): A Formal Model for Digital Library Framework and Its Applications. ACM Transactions on Information Systems, 22, 2, (April 2004), 270-312, ISSN:1046-8188

Hill, M.D. (1990). What is scalability ? *ACM SIGARCH Computer Architecture News*, Volume 18 Issue 4, pages 18-21, ISSN 0163-5964)

Horton, M. & Adams, R. (1987). Standard for Interchange of USENET Messages, RFC 1038 (December 1987)

Liaw, S. S. & Huang, H. M. (2003). An investigation of users attitudes toward search engines as an information retrieval tool. Computers in Human Behavior, 19, 751–765.

Manghi, P.; Mikulicic, M.; Candela, L.; Castelli, D.; Pagano, P. (2010). Realizing and Maintaining Aggregative Digital Library Systems: D-NET Software Toolkit and OAIster System. D-Lib Magazine, 16 (3/4), March/April 2010, doi:10.1045/march2010-manghi

Marty, P. F. (2008). An introduction to digital convergence: libraries, archives, and museums in the information age. Archival Science. Vol. 8, No. 4 (December 2008), pp. 247-250, ISSN 1389-0166, Springer

Nowitz D. A. & Lesk, M. E. (1978) A Dial-Up Network of UNIX Systems, In: UNIX Programmer's Manual, Seventh Ed., Bell Laboratories, Murray Hill, New Jersey

Pieper, D.; Wolf, S. (2007). BASE - Eine Suchmaschine für OAI-Quellen und wissenschaftliche Webseiten. Information, Wissenschaft & Praxis (IWP), 58(3), 179-182, ISSN 1434-4653

Reis, D.; Freire, N.; Manguinhas, H., Pedrosa, G. (2009). REPOX: a framework for metadata interchange. Lecture Notes In Computer Science. Proceedings of the 13th European conference on Research and advanced technology for digital libraries. pp. 479-480

Salz, R. (1992). InterNetNews: Usenet transport for Internet sites. Proceedings of Summer '92 USENIX, pp. 93-98. June 8-12, 1992 – San Antonio, TX

Schwartz, M. F.; Emtage, A.; Kahle, B & Neuman, B. C. (1992). A Comparison of Internet Resource Discovery Approaches, Computing Systems, pp. 461-493, 5(4), August 1992

Sonnenreich, W. (1998). A history of Search Engines, In: Web Developer Guide to Search Engines, Sonnenreich, W.; Macinta, T., p. 464, Wiley, ISBN 978-0-471-24638-1

Suleman, H.; Fox, E.A. (2001). A framework for building open digital libraries. D-Lib Magazine 7(12), ISSN 1082-9873, available online at http://www.dlib.org/dlib/december01/suleman/12suleman.html

Sylwestrzak, W.; Borbinha, J.; Bouche, T.; Nowiński, A.; Sojka, P. (2010). EuDML—Towards the European Digital Mathematics Library. Proceedings of DML 2010. pp. 11-26, Paris, France (Jul 2010). ISBN: 978-80-210-5242-0

Teregowda, P.B.; Councill, I.G.; Fernández R., J.P.; Kasbha, M.; Zheng, S. and Giles, C.L. (2010). SeerSuite: Developing a Scalable and Reliable Application Framework for Building Digital Libraries by Crawling the Web. 2010 USENIX Conference on Web Application Development, June 23–24, 2010, Boston, MA, USA

Van de Sompel, H.; Jeroen Bekaert, J.; Liu, X.; Balakireva, L.; Schwander, T. (2005). aDORe: A Modular, Standards-Based Digital Object Repository. The Computer Journal. 48(5) pp. 514-535, doi:10.1093/comjnl/bxh114

Wessels, D. (2004). Squid. The definitive guide. O'Reilly and Associates ISBN 0-596-00162-2

Zamłyńska, K.; Bolikowski, Ł.; Rosiek, T. (2008). Migration of the Mathematical Collection of Polish Virtual Library of Science to the YADDA Platform. In: Sojka, Petr (ed.): Towards Digital Mathematics Library. Birmingham, United Kingdom, July 27th, 2008. Masaryk University, Brno, 2008. pp. 127-130

Products and Services; from R&D to Final Solutions

Edited by Igor Fuerstner

ISBN 978-953-307-211-1
Hard cover, 422 pages
Publisher Sciyo
Published online 02, November, 2010
Published in print edition November, 2010

Todayâ€™s global economy offers more opportunities, but is also more complex and competitive than ever before. This fact leads to a wide range of research activity in different fields of interest, especially in the so-called high-tech sectors. This book is a result of widespread research and development activity from many researchers worldwide, covering the aspects of development activities in general, as well as various aspects of the practical application of knowledge.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821