A Reference Architecture for Digital Library Systems:
Principles and Applications

Leonardo Candela Donatella Castelli Pasquale Pagano
Istituto di Scienza e Technologie dell'Informazione "A. Faedo"
Consiglio Nazionale delle Ricerche
Via G. Moruzzi, 1 – 56124 Pisa - Italy
{candela|castelli|pagano}@isti.cnr.it

Abstract

A reference architecture for a domain provides an architectural template which can be used as a starting point for designing the software architecture of a system in that domain. Despite the popularity of tools and systems commonly termed "Digital Library" very few attempts exists to set the foundation governing their development thus making integration of reuse of third party assets and results very difficult. This paper presents a reference architecture for a Digital Library domain characterised by many, multidisciplinary and distributed players, both resource providers and consumers, whose requirements evolve along the time. The paper validates this reference architecture by describing the structure of two current systems, DILIGENT and DRIVER, facing the problem to deliver a large-scale digital library in two different contexts and with diverse technologies.

Categories and Subject Descriptors
H.3 [Information Storage and Retrieval]: H.3.7 Digital Libraries;
General Terms
System Architecture
Keywords
Reference Architecture, Digital Library System, DILIGENT, DRIVER

1 Introduction

Digital Library is a complex area where a large number of heterogeneous disciplines and fields converge. This highly multidisciplinary nature has created several conceptions of what a Digital Library is, each one influenced by the perspective of the primary discipline of the conceiver(s) [10][13][14][15][21][5]. One of the consequences of this heterogeneity is that during the last fifteen years a lot of Digital Library Systems have been pragmatically developed with specialized methodologies obtained by adapting techniques borrowed from other disciplines. This kind of approach produced very many heterogeneous entities and systems, thus rendering the interoperability, reuse, sharing, and cooperative development of digital libraries extremely difficult.

The Digital Library community recognizes these drawbacks and expresses the needs to invest in Architectures for Digital Library Systems (DLS) and to implement generic Digital Library Management Systems (DLMS) having all the key features that appear fundamental in supporting the entire spectrum of digital library functionality as it arises in several possible contexts [14][15]. In particular, a DLMS should incorporate functionality that is related to all generic issues as well as the generic component of all mixed issues. Moreover, its design should allow for easy integration of sub-systems supporting the specialized functionality required by each particular environment.

To speculate on this understanding, the DELOS Network of Excellence on Digital Libraries promotes an activity aiming at setting the foundations for and identifying the cornerstone concepts within the Digital Library universe in order to facilitate the integration of research results and propose better ways of developing appropriate systems. The first outcome of this activity is *The Digital Library Manifesto* [6]. This document offers a new vision of this universe by presenting it as populated by three different "systems" and by carefully establishing their roles and the relationships among them. The three "systems" are the Digital Library (DL), the Digital Library System (DLS), and the Digital Library Management System (DLMS), as depicted in Figure 1. In particular, the DL is the abstract system as perceived by end-users, the DLS emerges as the software system providing all the functionality that is required by a particular DL, while the DLMS is the software system that provides the infrastructure for producing and administering a DLS and integrates additional software offering extra functionality.

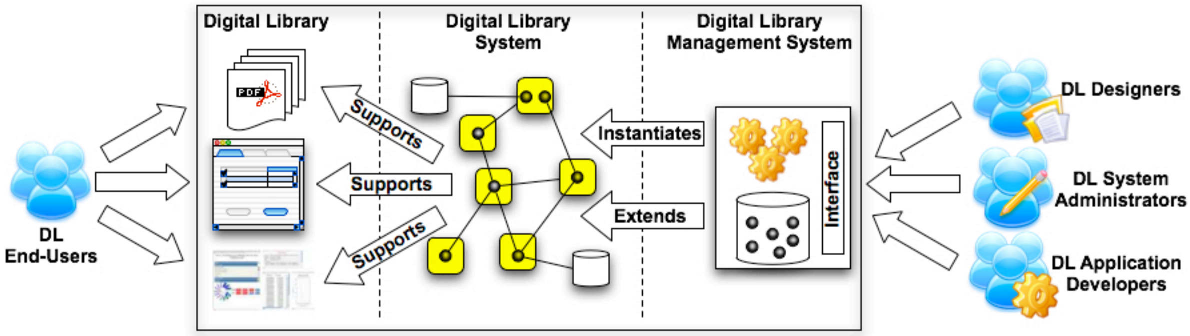

Figure 1. DL, DLS, and DLMS: A Three-Tier Framework

To realise this idea, the Manifesto proposes to implement a series of frameworks that collectively concur to meet the initial goal, i.e., facilitate the implementation of digital library "systems". These frameworks are: (*i*) a *Reference Model*, consisting of "a minimal set of unifying concepts, axioms and relationships within a particular problem domain. It is independent of specific standards, technologies, implementations, or other concrete details" [18]; (*ii*) one or more *Reference Architectures*, containing an architectural design pattern indicating an abstract solution in implementing the concepts and relationships identified in the Reference Model; and (*iii*) one or more *Concrete Architectures*, containing additional elements that make the reference architecture more concrete; e.g. it replaces the mechanisms envisaged in the Reference Architecture with concrete standards and specifications.

This paper presents a first step toward the definition of a fully-fledged Reference Architecture for a Digital Library domain characterised by many, multidisciplinary and distributed players, both providing and consuming resources, whose requirements evolve along the time. Essentially, such first attempt of reference architecture identifies a consistent and comprehensive set of software components necessary for a DLS and the interactions between them. It acts as a blueprint for the creation and management of a DLS.

The remainder of this paper is organised as follows. Section 2 contextualises the work by providing a survey of current and past effort spent in similar tasks. Section 3 presents the reference architecture. Section 4 validates the reference architecture by describing the structure of two existing systems, namely DILIGENT and DRIVER, that rely on the reference architecture. Finally, Section 5 presents the conclusion of this work and the future research issues.

2 Related Work

In addition to the attempt presented in this paper, few other initiatives exist to tackle the definition of Digital Library architectures especially in the framework of *large-scale* DLSs, i.e. DLSs characterised by a huge number of distributed resources aiming at serving one or more distributed communities. One of the first attempt is represented by Dienst [16] in 1995. This DLS, developed at Cornell University, proposes tasks clearly divided and specified by a protocol based on HTTP providing helpful abstractions to its users, e.g. collections uniformly searchable without regard to objects locations. Although technically sound, the approach underlying this system requires an investment in software, methodology, and support that some prospective users were not willing to make. To reach interoperability in distributed DLSs two approaches exist: the federated and the harvesting approaches. In the federated approach a number of organizations agree on a number of service specifications, usually expressed as formal standards. When establishing a federation, the problem to overcome is the effort required by each organization to implement the services and keep them current with all the agreements. For instance, many libraries have standardized on the Z39.50 protocol [2] to meet the needs for record sharing and distributed search. On the other hand, the Open Archive Initiative [17] promotes the harvesting model as the mechanism for building digital library services over archives making their metadata available through a simple and lightweight protocol. Suleman [22] proposes to extend the work of the OAI to support the inter-component interaction within a componentized DLS built by connecting small software components that communicate through a family of lightweight protocols. Gonçalves, Fox, Watson, and Kipp [12] introduced a formal framework based on five fundamental abstractions, i.e. Streams, Structures, Spaces, Scenarios, and Societies, to define digital libraries rigorously and usefully. In such a framework the architecture of DLSs is expressed in terms of services by focusing on their behavioural description (scenarios) and on the possibility to model co-operating services (structures). An exploitation of such a framework to implement DLSs is described in [11]. The JISC Information Environment (JISC IE) technical architecture[1] [20] specifies a set of standards and protocols that support the development and delivery of an integrated set of networked services that allow the end-user to discover, access, use and publish resources as part of their learning and research activities.

In parallel with the above initiatives, three systems architecture paradigms have emerged in the last years: Service Oriented Architectures, Peer-to-Peer Architectures, and Grid Infrastructures. In [1], a survey of these architectural paradigms and their exploitation in current DLSs is reported. This study aims at identifying the similarities among these paradigms as well as their distinguishing peculiarities in fulfilling the various requirements arising in the Digital Library arena. The major outcome of this study consists in the recognition of the complementary nature of these architectures and of the needs to combine them in order to fruitfully exploit the available resources in different contexts.

3 The Digital Library System organization

The concept map presented in Figure 2 introduced in the Reference Model [2] shows the main entities and their relationships in modelling DLS architectures. *Components* and *hosting nodes* play the role of first class citizens. A component is defined as a software module providing a well defined set of functions such that (*i*) it is autonomously configurable, and (*ii*) it is deployable on one or more hosting nodes. A hosting node is defined as a hardware device providing computational and storage capabilities such that (*i*) it must be networked, (*ii*) it must be capable to host components, and (*iii*) its usage must be regulated by policies. Even if the reference model does not constrain the granularity of components (potentially a single component may constitute a DLS), it gives the possibility to decompose DLSs into a certain number of components distributed in multiple hosting nodes so that flexible, better structured, and more manageable systems be created.

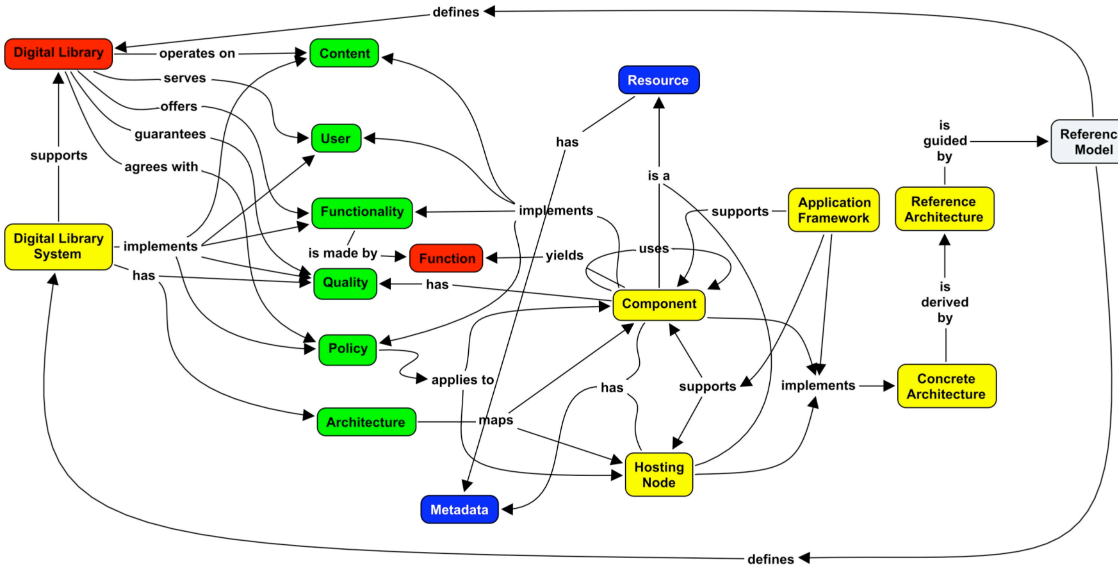

Figure 2. Digital Library System Concept Map

Moreover, the Reference Model identifies the concept of *Application Framework*, i.e., the middleware available at the hosting nodes and providing the run-time environment for the components.

Based on this conceptualization, Figure 3 shows the Digital Library System Reference Architecture that we propose in order to fulfil the needs of large-scale digital libraries, i.e. digital libraries characterised by a large amount of resources and serving one or more communities world-wide spanned, multidisciplinary and whose requirements evolve along the time.

1 JISC Information Environment Architecture http://www.ukoln.ac.uk/distributed-systems/jisc-ie/arch/

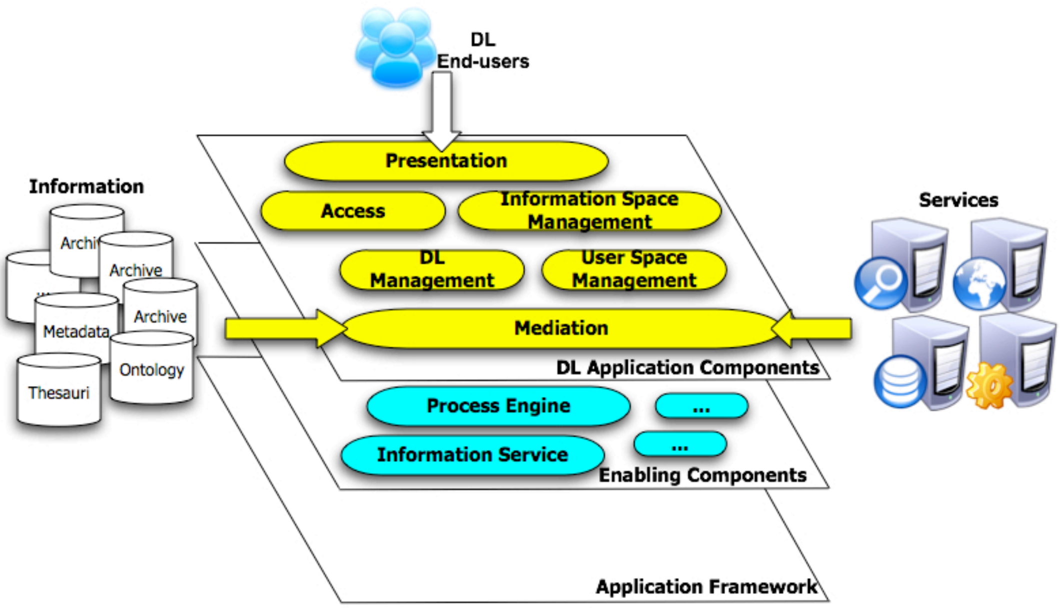

Figure 3. The Digital Library System Reference Architecture

As commonly recognised, DLSs are complex systems whose implementation and management beneficiate by their the decomposition into smaller and more manageable parts. The Reference Architecture applies the decomposition technique and proposes a loosely coupled component oriented approach. This approach is fundamental for the purposes of the reference architecture since it allows for (*i*) easy design of the DLS through component selection and replacement, (*ii*) reuse of the components in different contexts, (*iii*) distributed installation and maintenance since each component can be independently implemented and deployed, and (*iv*) easy supporting of heterogeneity issues by using or providing an appropriate component dealing with the particular problem.

In addition to the component-oriented approach, this Reference Architecture adopts a layered approach. This approach organizes components on three tiers: (*i*) the *Application Framework*, i.e. the set of software libraries and subsystems supporting the operation of the other DLS components; (*ii*) the *Enabling Components*, i.e. the DLS components providing the functionality required to support the communication and co-operation among the components implementing the DL application; (*iii*) the *DL Application Components*, i.e. the DL system components providing the end-user specific DL functionality.

In the context of the Reference Model, a set of functionalities have been identified and grouped in functional areas. DL Application Components are thus described and organized through these functional areas: (*i*) *Mediation* - dealing with and providing access to third party information sources varying in their structure, format, media, and physical representation as well as third party services; (*ii*) *Information Space Management* - implementing and managing the digital library information objects by providing storage and preservation facilities; (*iii*) *Access* – supporting discovery of the DL information objects via search and browse facilities; (*iv*) *User Space Management* - providing support for registration and activities concerning administration of the users; (*v*) *DL Management* - implementing the functionality for the administration of the DL in terms of the librarian activities, e.g. review processes, policies management, preservation activities; and (*vi*) *Presentation* - providing users with friendly access to the DL information and services, namely the graphical user interface.

By relying on the concepts identified in the Reference Model, the components reported in Table 1 have been envisaged as the minimal pool needed to deliver the expected functionality. *Openness*, i.e., the possibility to add a new component or replace an existing one with a new one, is one of the most important characteristics to ensure the sustainability of the system. This characteristic is guaranteed by the presence of the Enabling Components taking care of the management activity, aimed at ensuring the desired quality of service. This is not a trivial task and the pool of services needed to reach it may vary a lot dependently on the concrete application scenario. It involves many functions such as: security, e.g. authorization of the request, encryption and decryption as required, validation, etc.; deployment, allowing the service to be redeployed (moved) around the network for achieving a better performance, and a greater redundancy for improving availability, or other reasons; logging for auditing, metering, etc.; dynamic rerouting for fail over or load balancing and maintenance, i.e. management of new versions of a service or new services to satisfy new users needs.

Table 1. Digital Library System Reference Architecture: Mandatory Components

Mediation

- Document Conformer
- Collection Virtualizer
- Ontology Aligner
- Metadata Schema Mapper
- Content Transformer
- Content Translator
- Service Wrapper

Information Space Management

- Storage Component
- Repository Component
  - Preservation Support Manager, Information Object Management, Ingest, Policy Enforce, Error Recovery, IO Registry, Configuration Manager, Metadata Generator, Storage Manager, Validation
- Collection Manager
- Annotation Manager

Access

- Search
  - Query Parser, Query Optimizer, Query Execution, Query Adapter, Collection Selection, Collection Description, Personalization
- Index
  - Feature Extractor, Trigger
- Data Fusion

User Space Management

- User Registry
  - User Manager, Tracer, Registration, User Profile Config Manager
- Group Registry
  - Group Manager, Subscription, Group Profile Config Manager
- Profile Repository
  - Role Manager, Role Enforce, Role Config Manager, Checker
- Policy Manager

DL Management

- User Administration
- Publication Process Manager
- DL Monitoring
- Recommender
- Preservation Manager

Presentation

- User Interface
- OAI-PMH Publisher
- API Interface

Enabling Components

- Information Service
- Authentication, Authorization, and Auditing (AAA)
- Broker
- Process Engine

The component-oriented approach of the DLS Reference Architecture makes it possible to design and implement DLMSs equipped with a DL Generator. This component is in charge of automatically creating a DLS by appropriately selecting, aggregating (and potentially deploying) a set of available components in order to fulfil the requirements of a certain community. This kind of DLMS allows delivering DLs according to the Business on Demand Model (BoDM), a model created around the Internet rush that became the de facto business model of the 21st century. The goal of this model is to efficiently manage and synchronize resources to match fluctuating requirements and to release the optimal service at an affordable cost via controlled resource sharing.

In the next section, the components and principles envisaged in the Reference Architecture are validated by describing the structure of two current systems, namely DILIGENT and DRIVER.

4 Applying the Reference Architecture

The Reference Architecture proposes an abstract architectural design pattern that can have multiple instances once implemented in a concrete scenario. This section describes two existing large-scale digital library systems using the reference architecture in different application contexts.

4.1 DILIGENT A Digital Library Infrastructure on Grid ENabled Technologies

DILIGENT [8] is an ongoing EU IST project combining Digital Library and Grid technologies in order to deliver an infrastructure supporting scientists to define and operate the supporting environment they are looking for, i.e. the new kind of Digital Library application providing them with an integrated working environment giving access to the data and the facilities needed to perform their daily tasks. In this system the emphasis is on the exploitation of Grid facilities to gather both computing and storage power on demand as well as being capable to organise the resources needed to operate a Digital Library on demand.

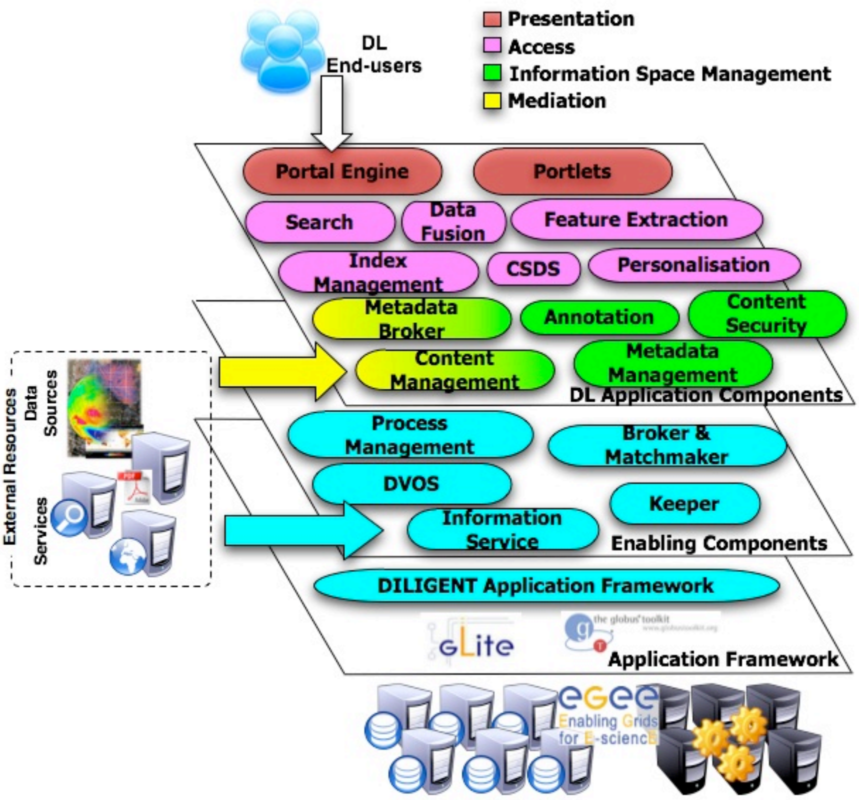

Figure 4. The DILIGENT Architecture

The DILIGENT architecture is depicted in Figure 4.

The Application Layer, being in charge to provide other components with the hosting environment needed to them to operate, must provide Grid facilities in the DILIGENT scenario. Thus it hosts the gLite[2] components needed to interact with the Grid infrastructure released by the EGEE project[3]. Moreover, all the services constituting the system are designed by relying on the facilities provided by the WSRF specifications [4] in terms of dynamic service creation, lifetime management, notification, manageability, naming and discovering of service instances, thus the application framework host the container implementing these functionalities realised by the Globus project[4]. Finally, the Application Framework hosts a pool of software libraries easing the interaction with the services forming the Enabling Components layer.

Many services form the Enabling Components layer. These services are in charge of providing the functionality for (*i*) ensuring that the other DILIGENT services work together securely and correctly, (*ii*) enabling the definition and creation of DLs by dynamically deploying and aggregating the needed resources, and (*iii*) enabling the creation of advanced functionality by combining resources via processes.

The Information Service supports the discovery and real-time monitoring of DILIGENT resources. In particular it (*i*) provides an interface enabling the other services to query for resources having certain characteristics and (*ii*) supports the subscription/notification mechanism thus acting proactively in providing the newly available information to those services interested in it.

The Keeper Service acts as the "orchestrator" of the services which form a DLS whilst assuring the Quality of Service required by the DL's definition criteria. To do so, it monitors the status of services and resources during the lifetime of the DL and accordingly re-designs its topology at runtime. This re-design consists in dynamically deploying new service instances on available hosting nodes provided by the infrastructure. By relying on such facilities, the DILIGENT system is enabled to maintain up and running a series of DLs on a pool of shared resources and promote their optimal use.

The Broker and Matchmaker Service facilitates the efficient usage of the Grid infrastructure through an optimal distribution of services and resources across hosting nodes. The service exploits and extends underlying grid middleware capabilities so as to match the needs of the DLs hosted by DILIGENT.

The Virtual Organisation Support Service is dedicated to the management of Virtual Organizations. A Virtual Organisation (VO) is a dynamic pool of distributed resources shared by a dynamic set of users from one or

---

[2] http://glite.web.cern.ch/glite/

[3] http://www.eu-egee.org/

[4] http://www.globus.org

more organizations in a trusted way. In DILIGENT, Virtual Digital Libraries make use of the VO mechanism to glue together users and resources in the trusted environment of the DL.
Finally Process Management provides functionality supporting the definition and execution of advanced functionality via the definition of processes, i.e. combination of existing services into more complex workflows that, once defined, can be repeatedly executed. Moreover, this area is equipped with an execution engine that performs the steps defined in each process in a Peer-to-Peer fashion in order to take advantage from the presence of distributed and replicated resources.
On top of these infrastructural services there resides the pool of those services devoted to deliver the functionality expected by end-users. Accordingly to the Reference Architecture they are organised in the Information Space Management, Access, and Presentation areas.
Information Space Management provides all the functionality required for storing information objects and their associated metadata, organizing such information objects into collections, and dealing with annotations. In order to take advantage from the presence of the potentially huge storage capacity deriving from the Grid, this area is also equipped with a service providing replication management facilities. These facilities automatically create and distribute replicas of information objects, thereby significantly increasing their availability. Finally, a service taking care of applying watermark and encryption techniques in order to protect the information objects from unauthorized accesses completes this area. The latter aspect is particularly relevant since the storage facilities provided by the Content Management service make use of Grid storage facilities, therefore it might occurs that physical files are stored on third party devices that are not under the direct control of the DILIGENT infrastructure.
Access provides the functionality required to discover the DL information objects. The whole task is orchestrated by the Search service that exploits the capabilities provided by (*i*) the Index Management, that builds and maintains indexes of various types on the information domain; (*ii*) the Feature Extraction, that extracts different types of features from different kinds of media; (*iii*) the Content Source Description & Selection, that supports the discovery of the collections where to search in for a given cross collection query by relying on the similarities between their content description and the query; (*iv*) the Data Fusion, that supports the merging of the result sets resulting from querying the involved collections; and (*v*) the Personalization, that enriches the query with user characteristics for customizing the search results. All of them have been designed to co-operate by relying on the Process Management facilities.
Finally, the Presentation provides the graphical user interface needed to make DILIGENT resources and functionality available in an intuitive and easy to use environment. From a technological point of view, the required functionality is implemented via Portal and Portlets technologies, where portlets are pluggable user interfaces that can be managed and hosted in a portal via appropriate standards. In particular, the DILIGENT infrastructure is equipped with a set of portlets, one for each service having a user interface, and the WebSphere portal engine has been used to host them.
The principles and guidelines proposed in the context of the Reference Architecture proved to be very useful in the context of a complex system like DILIGENT. The layered approach together with the component oriented one as well as the functional areas proposed fits very well with the organization of DILIGENT.

4.2 DRIVER Digital Repository Vision for European Research

DRIVER [9] is an IST project co-funded by the European Commission under the Framework 6 “Research Infrastructure” program. It officially started in June 2006 and has a planned duration spanning across eighteen months. The goal of this project is to develop an organization and a system as the first step towards a pan-European Digital Repository Infrastructure, i.e., an European production-quality infrastructure integrating existing national, regional and thematic repositories in order to provide a virtually centralized view over their distributed content. The project emphasis is therefore on providing cross-repository services in a large-scale scenario. Despite previous attempts have been implemented with a similar goal, e.g. the National Science Digital Library in US [19] and the Dutch network of Digital Academic Repositories (DARE) [23], they potentially suffer of the sustainability problem[5] because in their context, like in the DRIVER one, consumers are world-wide, heterogeneous, multidisciplinary and evolving. To overcome this problem the DRIVER architecture has taken advantage from the Reference Architecture and, in particular, by the loosely coupled component oriented approach giving the possibility to easily replace one piece of the system and thus to adapt it to novel requirements.

---

5 Sustainability is the characteristic of a system to be capable of flexibly and continuously working and adapting their resources and functionality to the requirements of its evolving user community.

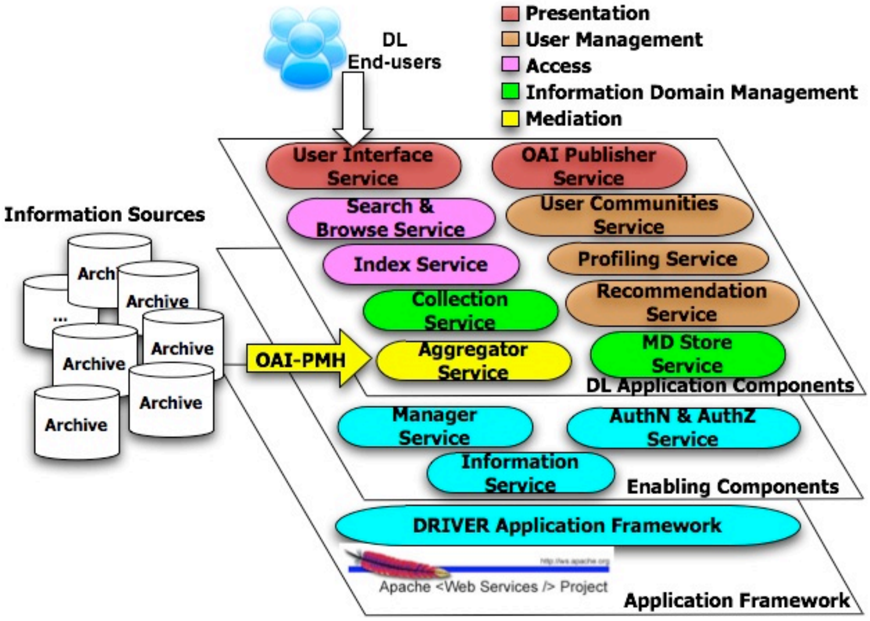

Figure 5. The DRIVER Architecture

The DRIVER architecture is depicted in Figure 5. All the components are realized as Web services. These services represent classes of services that co-exist and interact in multiple running instances in the DRIVER infrastructure. As a consequence the Application Framework is equipped with an hosting environment implementing the Web Services standards. Moreover, a pool of software libraries supporting hosted services in accessing the rest of the services forming the system is implemented and forms the DRIVER Application Framework.

In accordance with the Reference Architecture, the DRIVER Enabling Components layer offers the functionalities required to support the co-operation among the services running instances. These functionalities are: (i) controlled access to the pool of DRIVER resources, (ii) discoverability of the pool of the dynamic pool of DRIVER resources, and (iii) orchestration of family of services to deliver complex functionality.

Authentication and Authorization Service addresses *security*, i.e., access to services must occur according to the established policies thus the system must guarantee that only recognized and authorized consumers "consume" them; the service adopts the eXtensible Access Control Markup Language (XACML) standard[6];

Information Service addresses *discoverability*, i.e., each service that needs to interact with another service must be able to discover it;

Finally, *Manager Service* addresses *orchestration*, i.e., loosely coupled services must be configured and trained in order to behave as a single application.

On top of these infrastructural services there resides the pool of services devoted to deliver the functionality expected by end-users. Accordingly to the Reference Architecture they are organised in the Mediation, Information Domain Management, Access, User Management, and Presentation.

Mediation implements the functionality needed to collect content from external information sources and populate the DRIVER information space. In particular, the *Aggregator Service*, gathers content from external sources and makes it available to the other services [7].

Information Domain Management implements the DRIVER information space. In particular, the *MD Store* stores the collected metadata while the *Collection Service* supports the organization of the gathered content into dynamic collections.

The Access offers functionality for discovery objects forming the Information Space. *Index Service* provides appropriate data structures supporting content retrieval while *Search & Browse Service* provides facilities enabling content discovery.

The User Management implements facilities for dealing with end-users. In particular, *User Communities Service* supports the management of users and groups; *Profiling Service* collects information about registered users that are used to customize the system behaviour; *Recommendation Service* - notifies registered users each time an event the user is interested in occurs.

---

[6] XACML addresses the issues of authorization in distributed, heterogeneous, enterprise scale systems, which very well characterizes DRIVER environment.

Finally, the presentation offers the interfaces needed to interact with the system. In particular, the *User Interface Service* implements the graphical user interface providing users with access to the system functionality while the *OAI-Publisher Service* - implements the OAI-PMH making DRIVER an open archive data provider [17][7].
The approaches proposed by the Reference Architecture have proved their usefulness in a distributed and large-scale system like DRIVER. The component oriented approach as well as the identification of the three layers and the functional areas is demonstrated in a context characterised by many, multidisciplinary and distributed players, both providing and consuming resources, whose requirements evolve along the time.

5 Conclusion and Future Trends

Many Digital Library Systems are still poorly designed and implemented from scratch by focusing on the specific problems and requirements the system have to deal with. We argue for a development process based on general purpose DLMSs capable to deliver classes of DLSs fulfilling end-user requirements. To support such a view we presented a first step toward a Reference Architecture for Digital Library Systems adopting a component oriented approach and establishing the organization of such components in functional areas. Such reference architecture promotes component reuse as well as plug-ability of novel and specialized components to fulfil specific requirements. This development process complies with the Business on Demand Model supporting an efficient management of available resources.
Despites the proposed guidelines and principles proved to be very useful and appropriate with respect to the context they are envisaged for, i.e. Digital Library Systems serving communities whose requirements evolve along the time and characterised by many, multidisciplinary and distributed players, both providing and consuming resources, further effort must be invested in providing additional details and abstract solutions having a fine-grained granularity. As a consequence there is the need to apply the component oriented approach in organising each of the until now envisaged components internally and discover any re-factoring promoting a more fine grained component sharing and reuse. We are well aware of the "distance" that pass between the current Reference Architecture and the form that it should have in must to make easy the realisation on Concrete Architectures based on it. However, the until now proposed guidelines and approaches have proven to be appropriate in organising and describing two concrete and complex systems as DILIGENT and DRIVER convincing to continue in this direction and suggesting additional patterns and solution to be included in the next versions of the framework.

Acknowledgement

This work is partially funded by DELOS the Network of Excellence on Digital Libraries (IST-507618).
Special thanks go to Maria Bruna Baldacci for her valuable help in improving the readability of the paper.

6 References

[1] Agosti, M., Bischofs, L., Candela, L., Castelli, D., Ferro, N., Hasselbring, W., Moumoutzis, N., Schuldt, H., Weikum, G., Wurz, M., and Zezula, P. (2006) Survey on Peer-to-Peer Architectures, Grid Infrastructures, and Service-oriented Architectures for Digital Libraries. DELOS Deliverable 1.1.1.

[2] Agosti, M.; Candela, L.; Castelli, D.; Ferro, N.; Ioannidis, Y.; Koutrika, G.; Meghini, C.; Pagano, P.; Ross, S.; Schek, H.-J.; Schuldt, H. (2007). *The Digital Library Reference Model*. DELOS

[3] ANSI/NISO (2003) Z39.50 Information retrieval: Application Service Definition & Protocol Specification. NISO Press, National Information Standards Organization.

[4] Banks T. (2006). Web Services Resource Framework (WSRF) – Primer v1.2. OASIS Committee Draft http://docs.oasis-open.org/wsrf/wsrf-primer-1.2-primer-cd-02.pdf

[5] Borgman C.L. (1999). What are digital libraries? Competing visions. Information processing and Management, 35(3):227-243.

[6] Candela, L.; Castelli, D.; Ioannidis, Y.; Koutrika, G.; Pagano, P.; Ross, S.; Schek, H.-J.; Schuldt, H. (2006). *The Digital Library Manifesto*. DELOS, ISBN 2-912335-24-8

[7] Candela, L.; Castelli, D.; Manghi, P.; Pagano, P. (2007). *Item-Oriented Aggregator Services*. Third Italian Research Conference on Digital Library Systems, Padova, Italy,

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

[9] DRIVER – Digital Repository Infrastructure Vision for European Research. IST-2006-034047. http://www.driver-repository.eu

[10] Fox, E.A.; Akscyn, R.M.; Furuta, R.; Leggett, J.J. (1995). Digital Libraries. Communications of the ACM, 38(4):23– 28.
[11] Gonçalves, M.A. (2004) Streams, Structures, Spaces, Scenarios, Societies (5S): A Formal Digital Library Framework and Its Applications. PhD Dissertation. Virginia Tech Department of Computer Science.
[12] Gonçalves, M.A., Fox, E.A., Watson, L.T., and Kipp, N.A. (2004). Streams, Structures, Spaces, Scenarios, Societies (5S): A Formal Model for Digital Libraries. ACM Transactions of Information Systems (TOIS), 22(2), 270-312.
[13] Ioannidis Y. (ed.) (2001). Digital Libraries: Future Directions for a European Research Programme. DELOS Brainstorming Report, San Cassiano, Italy.
[14] Ioannidis, Y. (2005). Digital libraries at a crossroads. International Journal of Digital Libraries, 5(4), 255-265.
[15] Ioannidis, Y., Maier, D., Abiteboul, S., Buneman, P., Davidson, S., Fox, E., Halevy, A., Knoblock, C., Rabitti, F., Schek, H., and Weikum, G. (2005). Digital library information-technology infrastructures. International Journal of Digital Libraries, 5(4), 266-274.
[16] Lagoze, C., and Davis, J.R. (1995). Dienst - An Architecture for Distributed Document Libraries. Communications of the ACM. 38(4), 47.
[17] Lagoze, C., and Van de Sompel, H. (2001). The Open Archives Initiative: Building a low-barrier interoperability framework. Proceedings of the ACM-IEEE Joint Conference on Digital Libraries, Roanoke, VA, USA, 24-28 June 2001, pp. 54-62.
[18] MacKenzie, C.M., Laskey, K., McCabe, F., Brown, P.F., and Metz, R. (2006). Reference Model for Service Oriented Architecture 1.0. OASIS Committee Specification. http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=soa-rm
[19] NSDL. The National Science Digital Library. http://nsdl.org
[20] Powell, A. (2005). A 'service oriented' view of the JISC Information Environment. JISC. http://www.ukoln.ac.uk/distributed-systems/jisc-ie/arch/soa/
[21] Soergel, D. A (2002). Framework for Digital Library Research. DLib Magazine, 8(12).
[22] Suleman, H. (2002). Open Digital Libraries. PhD Dissertation. Virginia Tech Department of Computer Science.
[23] van der Kuil, A. and Feijen, M. (2004). The Dawning of the Dutch Network of Digital Academic Repositories (DARE): A shared Experience. Ariadne, 41.