D-Lib Magazine

March/April 2010
Volume 16, Number 3/4

Realizing and Maintaining Aggregative Digital Library Systems: D-NET Software Toolkit and OAIster System[1]

OAIster and D-NET: comparing sustainability of "traditional" and "infrastructural" solutions for Aggregative Digital Library Systems

Paolo Manghi, Marko Mikulicic, Leonardo Candela, Donatella Castelli, Pasquale Pagano
Istituto di Scienza e Tecnologie dell'Informazione "Alessandro Faedo", Consiglio Nazionale delle Ricerche, Italy
{manghi, marko.mikulicic, candela, castelli, pagano}@isti.cnr.it

doi:10.1045/march2010-manghi

Abstract

Aggregative Digital Library Systems (ADLSs) provide end users with web portals to operate over an information space of descriptive metadata records, collected and aggregated from a pool of possibly heterogeneous repositories. Due to the costs of software realization and system maintenance, existing "traditional" ADLS solutions are not easily sustainable over time for the supporting organizations. Recently, the DRIVER EC project proposed a new approach to ADLS construction, based on Service-Oriented Infrastructures. The resulting D-NET software toolkit enables a running, distributed system in which one or multiple organizations can collaboratively build and maintain their service-oriented ADLSs in a sustainable way. In this paper, we advocate that D-NET's "infrastructural" approach to ADLS realization and maintenance proves to be generally more sustainable than "traditional" ones. To demonstrate our thesis, we report on the sustainability of the "traditional" OAIster System ADLS, based on DLXS software (University of Michigan), and those of the "infrastructural" DRIVER ADLS, based on D-NET.

1 Introduction

In the digital library world, organizations responsible for large research communities — e.g., national consortia, research institutions, universities, foundations — are often tempted to supply their researchers, with *production systems* (i.e., high quality on-line services) for cross-operating over the bibliographic metadata records of publications aggregated from a set of institutional repositories. Such systems are powered by software systems, which we refer to here as *Aggregative Digital Library Systems (ADLSs)*, that typically address two main challenges: (1) populating an information space of metadata records by harvesting and normalizing records from several OAI-PMH compatible repositories; and (2) providing portals to deliver the functionalities required by the user community to operate over the aggregated information space, for example, search, annotations, recommendations, collections, user profiling, etc.

Two categories of ADLS scenarios can be identified. In "Static ADLSs", one responsible organization is willing to serve its user community with one information space and one customized portal under stable initial requirements (OAIster [12], BASE [15], DAREnet [11], etc). Recently a new category of ADLSs is emerging, namely "evolving ADLSs". Such systems support one or more organizations at constructing one or more information spaces and portals whose requirements tend to change over time. Examples of evolving ADLSs are the data infrastructures funded by FP6 and FP7 European Commission calls, e.g., DRIVER, DRIVER-II [8], EFG [10], Europeana [20].

In both scenarios, organizations have to face the problem of considerable software realization costs, i.e., design and development, and production system maintenance costs, i.e., hardware, administration and system refinement. Indeed, due to the absence of general-purpose ADLS software, "traditional" ADLS solutions tend to be realized in-house from scratch. To minimize realization costs, they address the specific static requirements for which they were conceived and, as such, are hardly reusable under different scenarios. As to evolving scenarios, any ad-hoc software solution would turn unsustainable in time, due to the inevitable and high refinement costs it would entail.

The *DRIVER project*, financed by the EC from May 2006, had the goal of delivering a production system capable of maintaining the European information space of open access publications [1] and enabling the dynamic deployment of portals over such space. Since traditional ad-hoc solutions do not cope well with evolving ADLSs, the DRIVER Consortium embraced a novel approach to ADLS construction. The resulting D-NET Software Toolkit [19] supports *a service-oriented infrastructure (SOI)* system where *customizability, openness, sharing, reuse and orchestration* of the given services enable sustainable patterns of ADLS realization and maintenance.

In this paper we present D-NET as a general-purpose software system supporting organizations in the construction of static and evolving ADLSs in a sustainable way. In particular, we do that by highlighting the differences with the level of effort entailed by a "traditional" ADLS approach. As an exemplar of traditional solutions we rely on the well-known OAIster System [12], whose technology was realized at the University of Michigan. The analysis will show that constructing static or evolving ADLSs using D-NET can notably reduce software realization costs and that, for evolving requirements, refinement costs for maintenance can be made more sustainable over time.

In section 2 we define static and evolving ADLS scenarios. In section 3 we discuss sustainability of traditional ADLS solutions in static scenarios, by reporting on the costs for realizing and maintaining the OAIster system, and in evolving scenarios, by estimating such costs based on the OAIster experience. In section 4 we present the D-NET Software Toolkit. Finally, in section 5 we discuss sustainability of D-NET infrastructural ADLS solutions in static scenarios, by reporting on the costs of the European Film Gateway system, and in evolving scenarios, by reporting on cost of the DRIVER infrastructure.

2 Aggregative Digital Library Systems

Repositories are defined here as software systems that typically offer functionalities for storing and accessing research publications and relative metadata information. Access usually has the twofold form of search through a web portal and bulk metadata retrieval through OAI-PMH interfaces [2]. In recent years, research institutions, university libraries, and other organizations have been increasingly setting up repository installations (based on technologies such as Fedora [3], ePrints [16], DSpace [4], Greenstone [5], OpenDlib [23], etc) to improve the impact and visibility of their user communities' research outcomes.

As a natural consequence of the diffusion of *institutional repositories*, research communities manifested the need to cross-operate over the metadata records from a set of repositories. Such a demand, which led to the definition of the OAI-PMH protocol in the first place, brought in a novel class of software systems, which we shall call in the following *Aggregative Digital Library Systems (ADLSs)*. Two main classes of ADLSs can be identified in the literature, *static* and *evolving*, described in the following sections.

2.1 Static ADLSs

Static ADLSs are typically under the responsibility of ***one*** organization willing to serve its user community with ***one*** information space and ***one*** portal with stable content and functional requirements:

> *Requirement 1:* harvesting metadata records from a given number or an arbitrary number of institutional repositories;

Requirement 2: forming a uniform information space by mapping (cleaning and enrichment) the harvested records onto records conforming to a target metadata format established by the user community;

Requirement 3: operating over such information space through a web portal, which offers the functionalities required by the user community.

The architecture of static ADLSs (see Figure 1 below, left side) typically consists of two main components: the *aggregation system* component, whose purpose is to form the information space according to the requirements 1 and 2, and the *portal* component matching requirement 3. In particular, portal requirements feature digital library functionalities, such as search, collections, etc. Aggregation system requirements include tools for harvesting metadata records from a set of OAI-PMH compliant repositories, plus tools for supporting the management of aggregation. Examples are: tools for administering workflows of participating repositories; tools for managing storing and indexing of the aggregated records; user interfaces for the construction of mappings from-to metadata formats, otherwise delegated to Perl or XSLT programmers.

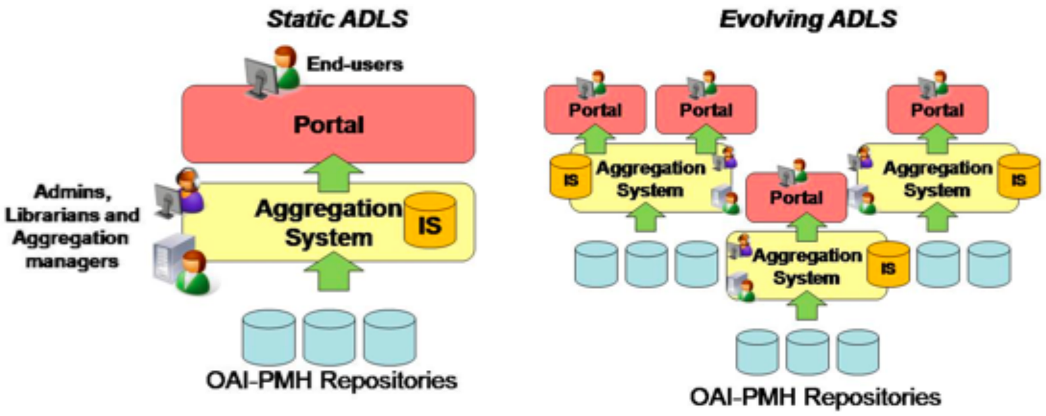


Figure 1: Static and evolving ADLSs

In the last few years, national consortia (e.g., DAREnet, BASE) and subject-based communities (e.g., NEEO project [6], DART Europe [7]) have been investing in the realization of traditional ADLS technologies (software realization) and in the maintenance of the relative production systems (hardware, system administration, software refinement). The causes of expenditure such organizations had to confront with are depicted by the cost model in Table 1, together with the kind of users involved in ADLS life-cycles.

Software realization. Since there is no general purpose ADLS software in the market, organizations tend to invest in ad-hoc solutions. They must cover the cost of *designers* and *developers* or outsource the work to external companies. Cost of software licenses, if needed, must be faced, although open source software is frequently adopted, e.g., Repox [25], DLXS harvester toolkit [24], Lucene and Solr [26].

Hardware installation and deployment. In order to offer robust and 24/7 services, organizations must invest in the purchase and administration of reliable servers and high-speed connections or, alternatively, outsource this work to external providers.

System deployment and maintenance. When the system is in operation a number of administrative tasks should be carried out. Three typical administrative roles can be identified:

- *Librarians:* portal managers, in charge of double-checking the quality of the information space, looking after end-user management (registration, profiling, etc.), and other user-oriented administrative tasks (e.g., updating vocabularies, managing document collections);
- *Aggregation managers:* in charge of keeping the information space up-to-date with the content of repositories. Examples of their duties are: adding or removing repositories from the harvesting

space, initiating harvesting processes over the repositories, and writing and activating XSLT scripts to transform the incoming metadata records into information space-conformant records;

- *Administrators:* in charge of handling hardware and software issues. These include operations such as ADLS software installation and configuration, web server availability, allocating or replicating storage and indexing space so as to host larger data sizes or ensure robustness and optimize query performance and scalability.

Software refinement. End-user portal and information space requirements may evolve in time, due to the latest trends or user interests. Modifying an existing system is generally an expensive software realization operation.

	Designers	***Developers***	***Librarians***	***Aggregation Managers***	***Admins***
Software realization	From scratch	From scratch			
Hardware					Installation, configuration and maintenance
System administration			Portal management	Coding transformation scripts	Installation and configuration of the software, system QoS, storage and indexing management
Software refinement	From scratch	From scratch			

Table 1: Eligible costs for traditional ADLSs

2.2 Evolving ADLSs

In evolving ADLSs, organizations, communities, information spaces and portals are arbitrary in number and their requirements tend to change over time. Typically, they fulfill the requirements of ***one or more organizations*** willing to unify their efforts to populate ***one or more*** information spaces to be operated over by ***one or more*** portals (see Figure 1 above, right side). Examples of such systems are *data infrastructures* such as DRIVER and Europeana. The requirements of evolving ADLSs are:

> *Requirement 4:* several information spaces can be maintained, and each may feature a different target format and aggregate records from an arbitrary number of repositories or other information spaces;
>
> *Requirement 5:* an information space may be under the responsibility of one or more organizations, which in turn may be responsible for one or more information spaces;
>
> *Requirement 6:* due to the participation of different organizations, hardware and content management may be located at physically distributed sites;
>
> *Requirement 7:* organizations may be responsible for one or more portals over one or more information spaces.

3 Realizing and maintaining ADLSs: the "traditional" approach

Typically, organizations willing to set up a static scenario have to face software realization costs rather than simple configuration and installation costs for existing ADLS software. The reason is the absence in the market of a general purpose ADLS. Indeed, existing ADLSs support specific information space or portal requirements, often not matching those peculiar to any other interested organizations.

3.1 The OAIster System: static ADLSs and traditional solutions

As a sample of "traditional" ADLS scenario we report on the OAIster system experience. OAIster was conceived at the University of Michigan in collaboration with the University of Illinois at Urbana-Champaign (UIUC) and funded by a Mellon Foundation grant in 2002. Its main motivation is that of "revealing the hidden web", through an ADLS aggregating the descriptive metadata records of digital resources such as digitized/scanned books and articles, born-digital texts, audio files (e.g., wav, mp3), images (e.g., tiff, gif), movies (e.g., mp4, quicktime), and datasets (e.g., downloadable statistics files). The number of repositories to be aggregated by OAIster is arbitrary and the information space may need to scale up to tens of millions of records.

The OAIster service was implemented by extending, with aggregation system facilities, the *Digital Library eXtension Service* (DLXS [13]), a well-documented open source product originally designed to provide ingest and search facilities for digital libraries. Today, the software includes functionalities to harvest records in DC format using OAI-PMH and convert them through XSLT transformations into an internal and more descriptive metadata format, named *BiBClass*. Robustness and quality of the software are well proven by the production system, whose information space counts today over 23,000,000 records collected from over 1000 repositories and archives worldwide.[2]

OAIster's product quality makes its life-cycle, from realization to maintenance, a reliable representative of the efforts an organization needs to take when building a traditional ADLS from scratch. In particular, the overall costs have been estimated to be:

Software realization. The initial cost, over 3 years of work, was that of 2 full-time employees who played the role of both designers and developers.

Hardware installation and deployment. The OAIster production system runs on the University laboratory infrastructure, of which it uses 33%, for a total of 4-5 servers and 2TB of disk per year.

System deployment and maintenance. Over the years, aggregation managers demanded more push-button tools, for repository management, harvesting and cleaning, in order to reduce the amount of manual programming work. With such tools, the work for aggregation managers, also in charge of the XSLT scripts, is estimated as 1/3 of a full-time employee a year.

Software refinement. The OAIster system has been continuously upgraded with new functionality over the last 6 years. Overall, several designers/developers had to be occasionally hired for a total of 4 years of work of 1 employee.

Such analysis clearly shows that ADLSs are affordable only for strong organizations which can rely on constant funding, and, possibly, on local laboratories and an existing team of programmers. The highest costs are due to design and development skills required in all phases of realization, maintenance and refinement.

3.2 Evolving ADLSs and traditional solutions

Due to their rather "dynamic" requirements, the way to minimize costs when handling evolving ADLSs is to develop software solutions designed to be reusable in different contexts, e.g., cope with any metadata formats and be easily extended with new functionality. Unfortunately, as mentioned in the previous section, existing ADLS solutions adopt a traditional approach, hence feature low degrees of reusability and extensibility. In order to minimize realization costs, their software is designed assuming a set of static requirements and developed to address them as efficiently as possible. Such a design choice, effective for static scenarios, does not pay when applied to evolving ADLSs. In fact, when faced with highly evolving requirements, the costs of refining monolithic software systems can be in the worst case equivalent to the realization of a new aggregation system or portal (see OAIster's software refinement costs). In particular:

> *Issues with requirements 4, 5:* traditional ADLSs are defined to operate one information space, the cost of adapting them to handle multiple information spaces equates to that of building a new system; moreover, when an organization is willing to support a metadata format different from the one in use in the system (e.g., DLXS's BibClass), low-level code refinement operations are needed.
>
> *Issues with requirement 6:* being devised to serve one organization, ADLSs are built to be locally monitored and administered; any adaptation to a distributed scenario would require major system modifications.
>
> *Issues with requirement 7:* each portal implies new and independent software realization costs.

4 D-NET Software Toolkit

The DRIVER project's applicative goal is that of promoting open access [1] business models among researchers and publishers by giving centralized access and visibility to open access publications available in Europe and world-wide. To this end, the project aimed at delivering an ADLS capable of aggregating tens of millions of open access metadata records from OAI-PMH repositories from Europe and beyond into one information space. In addition, the information space had to be "open" to any organization, such as national consortia, willing to operate over it through portals and external consuming applications or with the intention of building further information spaces from it.

To achieve the objective, the DRIVER project created the *D-NET* Software Toolkit. The software, inspired by the service-oriented principles [17] of Service-Oriented Infrastructures (SOIs), offers a run-time distributed environment in which services can be dynamically deployed, shared, and combined into applications. In particular, services are implemented as web services [18] and are designed to capture general ADLS functional patterns that can be instantiated to match the needs of a specific scenario. In a D-NET infrastructure, by *exploiting sharing, distribution, compositionality, autonomicity (orchestration)* and *customizability* of services, multiple organizations can collaboratively construct their ADLSs in a sustainable way. Specifically, one *responsible organization* (RO) administrates the infrastructure and supports a number of *participating organizations* (POs) at realizing and maintaining their ADLSs.

D-NET services are organized into three architectural areas (see Figure 2): *Enabling, Data* and *User Functionality* Areas.

The Enabling Area includes so-called *enabling services* [9], which serve as the fulcrum of the infrastructure, administered by the RO and available 24/7:

- *Information Service*. Services must *register* their *profile* (*unregister* when they leave) with the Information Service, which contains information about their web location, the functionality they expose and their internal status. Through the Information Service, any other service can *discover* if the services it requires are available and where they are located before interacting with them.
- *Manager Service*. Manager Services are workflow engines used to support system autonomic behavior by executing sequences of actions involving a set of services. Typically, they *monitor* the system service map available through the Information Service and react to relevant events by *orchestrating* the services into chains of actions. For example, the actions needed to automate replicas of records could be: whenever records are stored in a Store Service instance A, discover through the Information Service another Store Service instance B that is available to store a replica of the records, then read the records from A and store them in B.
- *Authorization and Authentication Service*. Through such service, organizations can specify which other services are authorized to discover and access their services; for the sake of simplicity, in this

paper we shall assume full service sharing and reusability.

The Data and User Functionality Areas include the services shown in Figure 2 below, used to form aggregation systems and portals respectively. Services in both areas are designed to be generic with respect to the metadata format they can manage, so that they can be instantiated to match the requirements of their communities. In particular, the Generic UI Service can be configured, personalized and instantiated in an infrastructure to set up a portal. Such portals can integrate none or some of the user functionalities available through the running services.

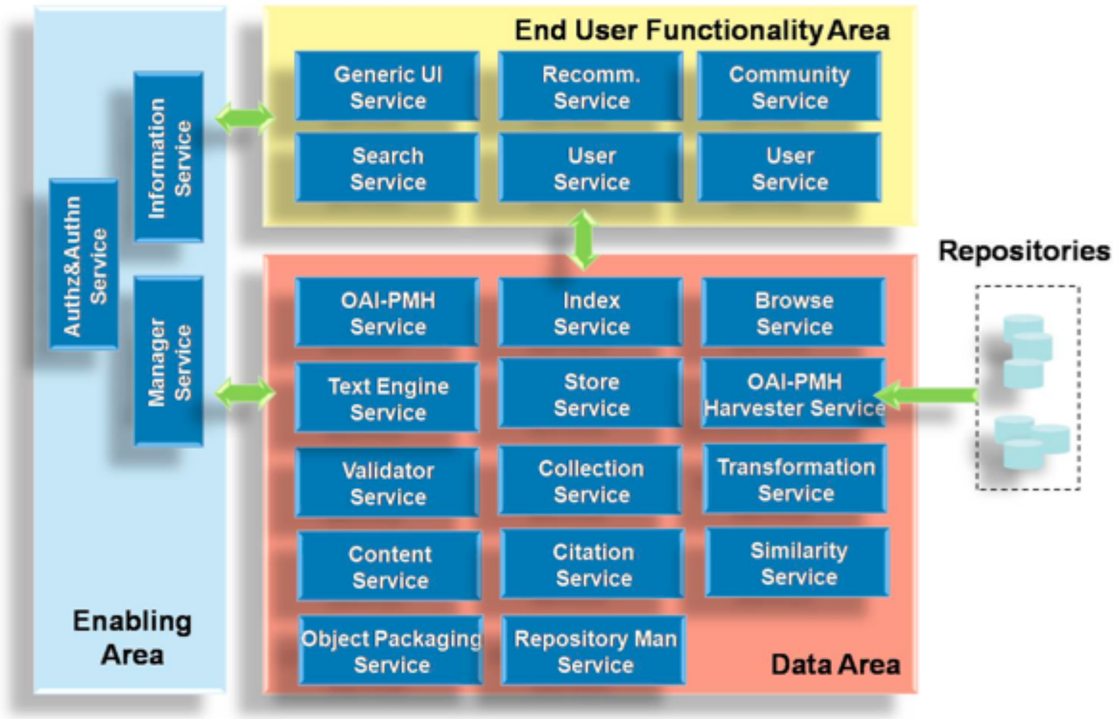


Figure 2: The D-NET service architecture

5 ADLSs in D-NET: the "infrastructural" approach

In D-NET ADLSs are constructed by following a "Lego" approach, that is, by deploying services of data and functionality areas and combining them into workflows, based on their application requirements.

For example (Figure 3), an aggregation system is a D-NET application [22] consisting of data areas services running in possibly multiple instances, so as to handle record replicas or record distribution (vertical and horizontal partitioning) at different sites. Services can be combined through customized workflows, in order to match the requirements of different aggregation systems; e.g., OAI-PMH Harvester Services could deliver records to Transformation Services and then to Store Services (as in Figure 3), or directly to Store Services if no mapping is needed.

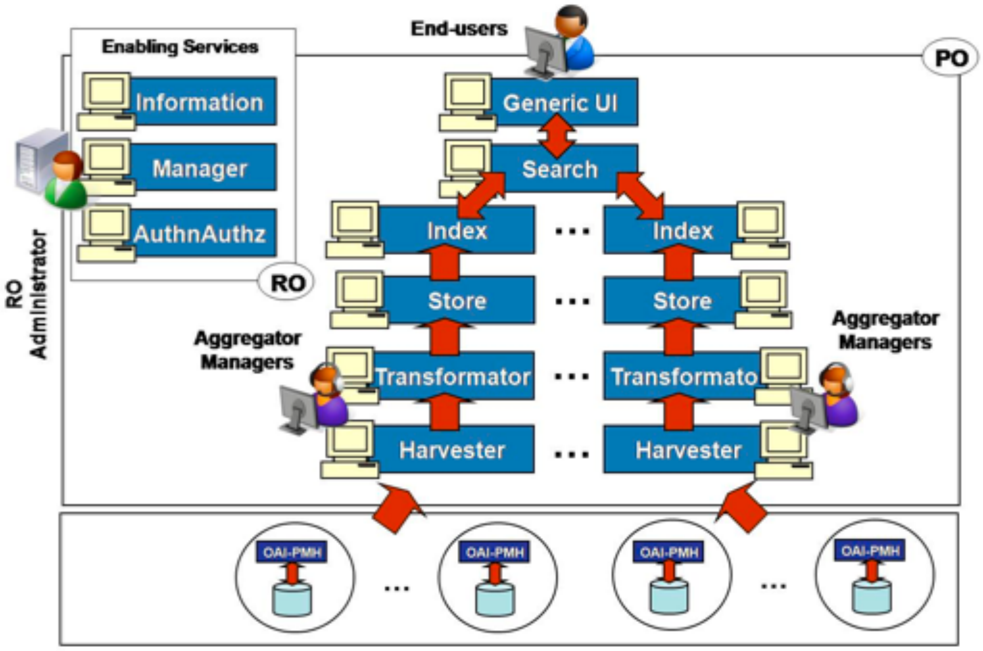


Figure 3 - Data Area Services: example of aggregation system

Similarly, portals consist of a number of Functionality Area services (see Figure 4). Appropriately combined, such services can form a variety of community portals, which can be configured to operate over one of the available information spaces. In particular, portals are modularly independent from the aggregation systems and can automatically adapt to any metadata format supported by the relative information spaces; moreover, they can be configured to focus on a subpart of one information space and to activate only the subset of functionalities of interest.

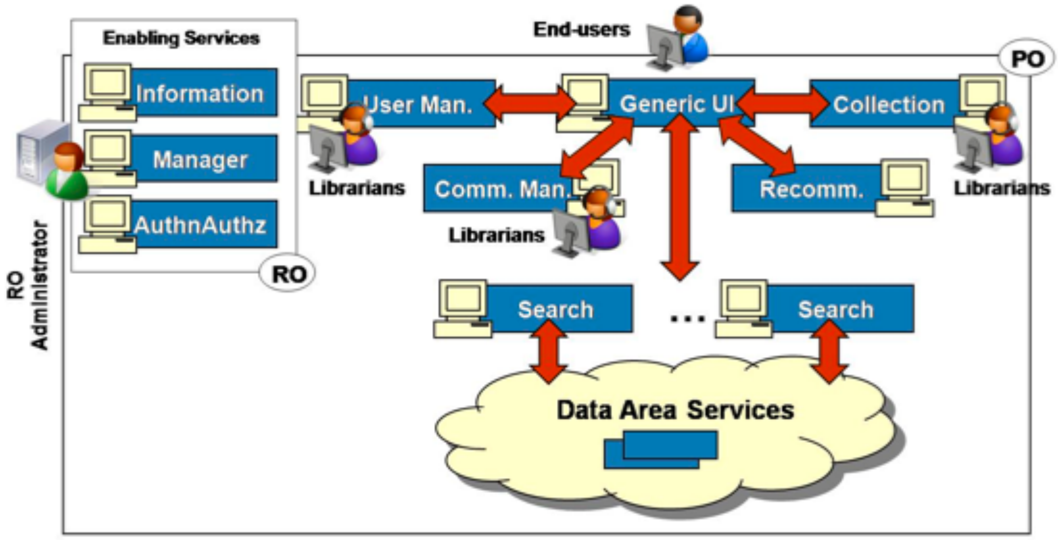


Figure 4 - Functionality Area Services

Most importantly, D-NET's framework is "open", i.e., organizations can extend Data and Functionality Areas with new services so as to introduce functionality not yet supported by the available D-NET software and build the workflows they need.

Today, a number of organizations have successfully built and are maintaining their static and evolving ADLSs by running D-NET. In the following, we shall describe the cost model entailed by the D-NET framework and report on real-world scenarios. This analysis will show that, compared to the traditional approach, adopting D-NET generally implies minor realization costs to the organizations and results in production systems that are more sustainable in the long-term.

5.1 Cost Model

D-NET's framework reduces ADLS realization and maintenance costs for POs as follows:

(i) Software realization costs are limited to installation costs, since service functionalities capture generic ADLS patterns and can be customized to specific scenarios;

(ii) Hardware costs can be reduced by hardware sharing with other organizations (consolidation effect);

(iii) Administrator costs are reduced by Manager Service orchestration mechanisms, e.g. storage allocation, index replicas, robustness, quality of service;

(iv) Aggregation manager and librarian costs can be reduced by sharing information spaces maintained by other organizations;

(v) Software refinement costs are reduced by the loosely-coupled service framework.

The separation between RO and POs leads to a different cost model, where the RO installing and maintaining a new D-NET system instance is conceptually distinct from the PO deploying an ADLS into an existing D-NET instance.

Table 2 reflects this distinction by introducing the role of *RO administrators*. It can be observed that while librarians and aggregator managers perform the same work as they would for traditional ADLSs, all other actors are either freed or facilitated in their tasks by the framework and orchestration mechanisms and software realization and maintenance costs are eliminated or softened.

	PO Designers	***PO Developers***	***PO Librarians***	***PO Aggr. Mgrs.***	***PO Admins***	***RO Admins***
Software realization	Service configuration and portal graphics	Service configuration and portal graphics				
Hardware					Installation, configuration and maintenance: not necessary when sharing	Installation, configuration and maintenance for Enabling Layer
System administration			Portal management	Advanced GUIs	Supported by service monitoring	Installation and configuration of the software. System QoS. Storage and indexing management,

						service QoS supported by service orchestration
Software refinement	Enabled by design: Customizability and Openness	Enabled by design: Customizability and Openness				

Table 2 - Eligible cost for infrastructural ADLSs

Costs for ROs. ROs bear the costs of administrators installing and maintaining the infrastructure and all PO applications.

Infrastructure installation and maintenance. Acquiring an initial set of machines connected to the Internet and installing/deploying the Enabling Services governing the infrastructure. Such machines, being central to the infrastructure activities, must be 24/7 reliable and available, e.g., equivalent in power and cost to those that traditional ADLS servers should have.

Infrastructure administration. Give support to POs that need to deploy services for their ADLSs.

Experience has proven that such costs are similar to that of installing and maintaining traditional ADLSs. In fact, the cost of monitoring the higher number of ADLSs that can be hosted in a D-NET environment is compensated by Manager Service orchestration mechanisms (monitoring availability, performance, workload, etc). Autonomic administration, normally missing in traditional ADLS technologies, frees administrators of monitoring and management tasks, and warns them in the case of major failures.

Costs for Pos. ROs give support to POs who want to keep down the costs of their ADLSs. In particular, given the current D-NET service packages, POs can exploit two main application patterns: *aggregation system* and *portal* applications. PO costs for building ADLS in D-NET are the following:

Software realization. Services in the Data and Functionality Areas can be combined, orchestrated and configured (adapted to a given metadata format) by the PO or the RO to satisfy the given PO's application scenario, i.e., specific information spaces or portals. Realization costs are therefore minimal for the PO involved, which need only deploy the services and supply the appropriate customized portal graphics.

Hardware installation and deployment. Hardware can be partly provided by the PO and partly shared from other POs or the RO itself. The extent of sharing is decided and evaluated by the POs involved and the RO. In some cases, PO applications may fully dependent on other POs machines, with zero hardware costs.

System deployment and maintenance. ADLS administrators are not required, to a large extent, since RO administrators can configure the Manager Services to orchestrate services to satisfy requirements of robustness and scalability, i.e. storage replication, index redundancy, quality of service. When needed, new services can be deployed any time to dynamically empower the infrastructure with new resources. Costs for librarians and aggregation managers are the same as for traditional ADLSs. Aggregator Managers can rely on push-button interfaces, through which they can define format-to-format mappings without specific programming skills. XSLT defined mappings are still possible.

Software refinement. A further level of participation is that of POs willing to include new typologies of services into D-NET or provide better implementations of existing typologies. Design and development cost are necessary, however D-NET's openness eases the integration of the new services. Two aspects further reduce refinement costs: *(1)* any D-NET enhancement can be reused by other PO applications and *(2)* D-NET services can naturally adapt to changes by simple configuration, e.g., change of the metadata formats do not require new development.

5.2 The European Film Gateway project: static ADLSs and infrastructural solutions

The costs of realizing a static ADLS in D-NET are those of an organization playing both the roles of the RO and of the only PO involved. Differently from traditional ADLS approaches, in D-NET realization costs are minimal, often limited to installation and portal graphics customization. Its flexible software can be installed and configured to match the ADLS requirements of the organization involved. Administration costs, as described above, are equivalent. A further benefit is that the resulting production system can be flexibly extended with further functionality or dynamically adapted to change its behavior, e.g., metadata format modification, and, if needed, distributed over a cluster of machines. Maintenance costs are therefore minor with respect to ad-hoc technological solutions.

D-NET is currently running the ADLS of the European Film Gateway (EFG) project [10], started in September 2008 and scheduled to last three years. EFG aims at aggregating metadata relative to filmography resources available from 15 relevant OAI-PMH data providers. Such resources can be movies, persons, corporate bodies, posters, audio-video material, all generally described by heterogeneous metadata formats at the original sites. The EFG Consortium has defined a common EFG metadata format, onto which the metadata records from the data providers must be mapped, and has provided the specification of the portal through which such data will have to be searched and accessed.

D-NET was convenient for two main reasons: the possibility of configuring an aggregation system to handle the EFG information space; and the openness and flexibility of the resulting production system. Realization costs were the installation of a D-NET infrastructure, the configuration and deployment of the aggregation system and portal, for an initial cost of one day of work and three servers in a local network. Aggregator Managers costs for defining format-to-format mappings are limited to the 15 data sources and therefore limited to an initial effort. Librarian costs remain as they would in the OAIster or any other system.

5.3 The DRIVER Infrastructure: evolving ADLSs and infrastructural solutions

The DRIVER Infrastructure is the evolving D-NET-based ADLS production system resulting from the DRIVER and DRIVER-II EU projects. In particular, the DRIVER Consortium plays the role of the RO and of the PO responsible for the main DRIVER ADLS consisting of: the aggregation system maintaining the ***DRIVER** information space* (DIS) of open access publications [22] and the DRIVER community portal, delivering to researchers advanced functionalities over the DIS. Two further POs, i.e., Belgium (University of Ghent) and National and University Library of Ljubljana (Slovenia) representing their respective national repository consortia, are responsible for the community portals and the aggregative systems harvesting and operating over the subsets of the DIS relative to their country.

Today, the DIS counts around 2,400,000 metadata records for open access publications, harvested from more than 250 open access repositories from 33 countries in Europe and beyond. The number of entries is expected to keep growing as repository organizations are attracted to become data provider to increment their visibility, thus recompensing their local efforts, and to get feedback on the quality of their service through special D-NET Validator Services, capable of ranking quality of repository content.

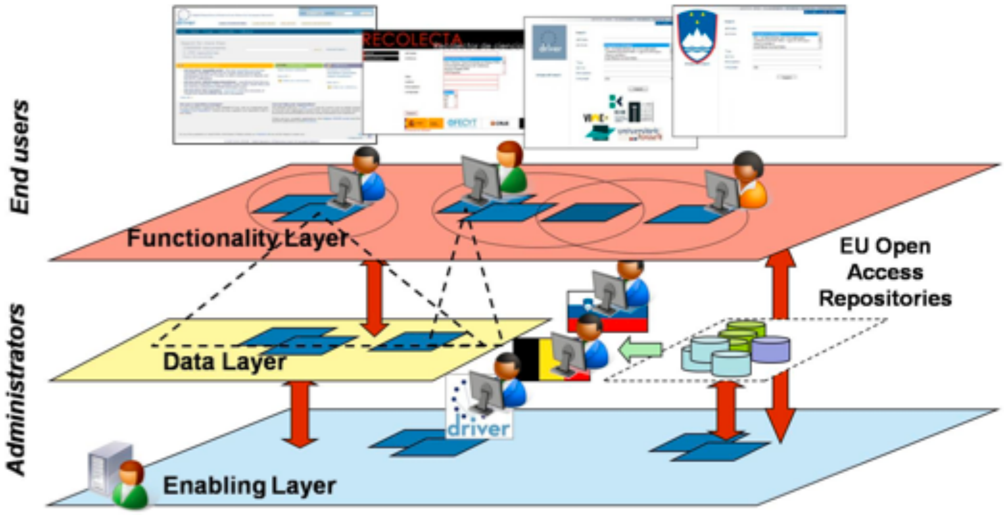


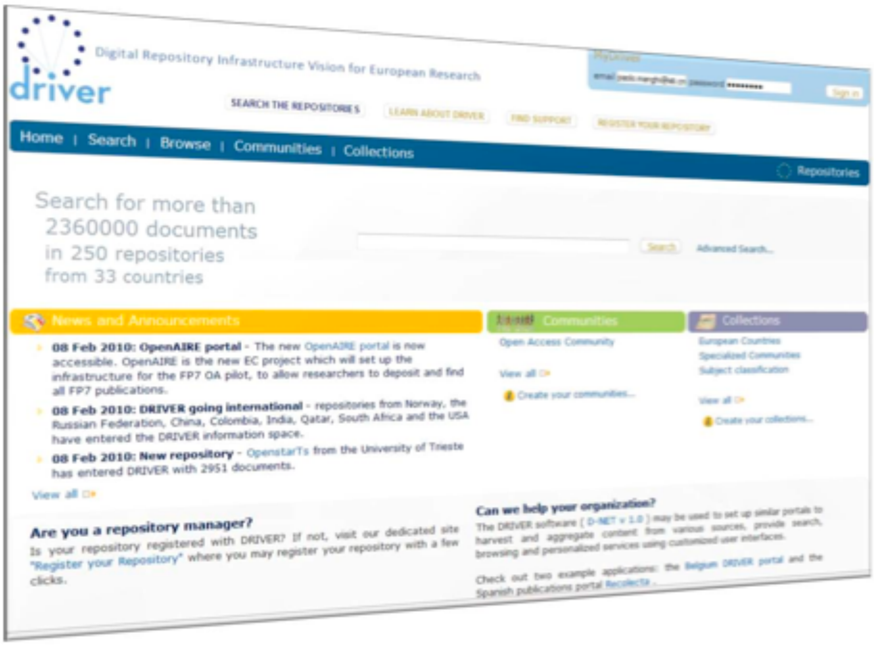
Figure 5 - DRIVER Infrastructure production environment

The DIS contains records in *DRIVER Metadata Format* (DMF). DMF records describe publication resources in terms of their provenance (name of the aggregator that fetched them, of the original repository, of the institution and country) and Dublin Core [21] bibliographical description. They are generated by a set of Transformation Services, whose aggregator managers apply cleaning and mapping rules to the DC records harvested from repositories. Note that the responsibility of such services can be delegated to national organizations so as to distribute the overall aggregative effort. As of 2010, Slovenia and Spain are examples of this methodology, while Belgium, Greece and Bulgaria are on the waiting list for technical support.

Manager Services, given the mapping from DC to DMF for a given repository, are configured to orchestrate the workflow maintaining the DIS, which is: harvesting DC records, storing and transforming them onto DMF records, indexing DMF records, and maintaining three replicas of all stores and indices on different servers (this number can be varied any time).

Three different instances of the Generic User Interface Service are running, all configured to accept DMF queries. The DIS portal (Figure 6) of the DRIVER PO operates over the whole DIS and enables all functionalities, from user profiling and collections to recommendations and advanced search. The portals of Slovenia[3] and Belgium[4] national consortium POs (Figure 7) enable simple keyword-based search over the subset of DIS records harvested from the relative countries.

It is important to note that Consorcio Madroño (Madrid, Spain), formerly a PO in the DRIVER infrastructure, has deployed and today maintains an independent D-NET infrastructure installation to build the Spanish repository infrastructure and serve as national web portal.[5] The D-NET Spanish infrastructure acts as an external aggregator and currently feeds to the DIS around 80,000+ documents from about 30+ repositories.



Search for more than 2360000 documents in 250 repositories from 33 countries

- **08 Feb 2010: OpenAIRE portal** - The new OpenAIRE portal is now accessible. OpenAIRE is the new EC project which will set up the infrastructure for the FP7 OA pilot, to allow researchers to deposit and find all FP7 publications.
- **08 Feb 2010: DRIVER going international** - repositories from Norway, the Russian Federation, China, Colombia, India, Qatar, South Africa and the USA have entered the DRIVER information space.
- **08 Feb 2010: New repository** - OpenstarTs from the University of Trieste has entered DRIVER with 2951 documents.

Are you a repository manager?
Is your repository registered with DRIVER? If not, visit our dedicated site "Register your Repository" where you may register your repository with a few clicks.

Can we help your organization?
The DRIVER software (D-NET v 1.0) may be used to set up similar portals to harvest and aggregate content from various sources, provide search, browsing and personalized services using customized user interfaces.

Check out two example applications: the Belgium DRIVER portal and the Spanish publications portal Recolecta .

Figure 6 - DRIVER Main Portal

driver
Search
All Fields
Archives
Title
Author
Description
Language
Advanced search
Slovenian DNet Portal
Search Open-Access documents
Please enter your search terms:
All Fields
Archives
Title
Author
Description
Language
Search
RECOLECTA
Recolector de ciencia abierta
Inicio | Buscar | Recorrer |
Buscar
Are you a repository manager?
Can we help your organization?

Figure 7 - Belgium, Slovenian, and Spain-Recolecta Portals

The overall costs of the DRIVER ADLS are minor compared to those required to build the same evolving ADLS with traditional systems:

Software realization. Installation and configuration of services and workflows, to handle the desired metadata and maintain the level of robustness and scalability needed. Graphic customization is also required. The cost for setting up an aggregation system is three days of work, while the cost for setting up a portal on an existing information space is one day.

Hardware installation and deployment. Hardware costs are currently those of 8 servers running at different locations in Europe, to be spread across the four supplying organizations.

System deployment and maintenance. These costs are those of librarians, in charge of managing users, communities and collections and that of aggregator managers. Administration at the installation level is required only when alerts from the Enabling Services are launched, hence is less than that of a system like OAIster, where replicas must be managed by humans. The overall cost of Aggregator Managers can be estimated as that required by the OAIster System, but spread across the participating organizations.

Software refinement. POs may add new services any time, to Data and Functionality Areas, so as to offer new data process functionality or end user functionality through the portal. Although such numbers cannot be used for a comparison with other scenarios, it is interesting to know that the experience of integrating a new D-NET Index Service encapsulating Yadda index technology (ICM, Poland) was that of one month of work of one programmer. The component, developed independently according to the D-NET framework specification, was then transparently integrated in the system workflows.

6 Conclusions

As highlighted by the OAIster experience, the realization of traditional solutions to Aggregative Digital Library Systems tend to be expensive and the resulting production systems will be hard to sustain in the long term. This is especially true when applied to "evolving" ADLS scenarios, which assume a multitude of end-users, large amounts of data and evolving functional requirements. This paper proposes the D-NET Software Toolkit as an innovative technological solution to this problem, being capable of supporting organizations in the construction of sustainable ADLSs. D-NET's approach has also proven to be apt for "static" ADLS scenarios, due to its high configurability and easy deployment. Several communities, from national consortia to subject based communities, are now considering building their ADLSs using D-NET.

7 Acknowledgments

This work would have not been possible without the precious support and cooperation of the other members of the D-NET technical team: ICM Research Centre (Warsaw) for Yadda Index Services and authentication and authorization mechanisms in the Enabling Area; University of Bielefeld Library (Bielefeld, Germany) for the Data Management Area; and the Department of Informatics, National and Kapodistrian University of Athens (Greece) for the End User Functionality Area. Furthermore, special gratitude goes to the staff of OAIster and in particular to Kat Hagedorn, OAIster Metadata Harvesting Librarian, for their kindness and availability to show us the back-stage of the OAIster System. We are also thankful to University of Ghent, National and University Library of Ljubljana and Consorcio Madroño, representing the respective repository national consortia, whose cooperation, enthusiasm and trust in our work gave us the chance to improve and refine our software and thoroughly test its effectiveness.

Notes

1 This work is partially supported by the INFRA-2007-1.2.1 Research Infrastructures Program of the European Commission as part of the DRIVER-II project (Grant Agreement no. 212147).

2 OAIster has recently migrated its information space maintenance to OCLC [14].

3 http://search.slovenia.driver.research-infrastructures.eu

4 http://search.belgium.driver.research-infrastructures.eu

5 http://search.recolecta.driver.research-infrastructures.eu

8 References

[1] Peter Suber, *Open Access Overview*, http://www.earlham.edu/~peters/fos/overview.htm, 2007

[2] Lagoze, C., Van de Sompel, H., *The open archives initiative: building a low-barrier interoperability framework*. In: Proceedings of the first ACM/IEEE-CS Joint Conference on Digital Libraries, ACM Press (2001) 54-62

[3] Lagoze, C., Payette, S., Shin, E., Wilper, C., *Fedora: An Architecture for Complex Objects and their Relationships*, International Journal on Digital Libraries 6 (2005) 124 - 138

[4] Tansley, R., Bass, M., Stuve, D., Branschofsky, M., Chudnov, D., McClellan, G., Smith, M., *The DSpace Institutional Digital Repository System: current functionality*. In: ACM/IEEE 2003 Joint Conference on Digital Libraries (JCDL 2003), 27-31 May 2003, Houston, Texas, USA, Proceedings, IEEE Computer Society (2003) 87-97

[5] *Greenstone Digital Library Software*, http://www.greenstone.org

[6] *NEEO Project: Network of European Economists Online*, http://www.nereus4economics.info/neeo.html

[7] *DEEP: The DART-Europe E-theses Portal*, http://www.dart-europe.eu

[8] *DRIVER Project: Digital Repository Infrastructure Vision for European Research*, http://www.driver-community.eu

[9] *Enabling Services in Knowledge Infrastructures: The DRIVER Experience*, Leonardo Candela, Donatella Castelli, Paolo Manghi and Pasquale Pagano, Proceedings of the Third Italian Research Conference on Digital Library Systems (IRCDL), Padua, Italy, 2007

[10] *European Film Gateway Project*, http://www.europeanfilmgateway.eu

[11] *DAREnet: Digital Academic Repositories*, http://www.narcis.info

[12] *OAIster project*, University of Michigan, http://www.oaister.org

[13] *DLXS project*, University of Michigan http://www.dlxs.org

[14] *OCLC Online Computer Library Center*, http://www.oclc.org

[15] *BASE: Bielefeld Academic Search Engine*, http://www.base-search.net

[16] Millington, P., Nixon, W.J., *EPrints 3 Pre-Launch Briefing*, Ariadne 50, 2007

[17] Ali Arsanjani, *Service-oriented modeling and architecture*, IBM developerWorks, http://www-128.ibm.com/developerworks/webservices/library/ws-soa-design1

[18] *W3C Web Services Activity web site*, http://www.w3.org/2002/ws

[19] *D-NET Project*, http://www.D-NET.research-infrastructures.eu, Istituto di Scienza e Tecnologie dell'Informazione, Centro Nazionale delle Ricerche (Pisa, Italy), ICM Research Centre (Warsaw), University of Bielefeld Library (Bielefeld, Germany), and Department of Informatics, National and Kapodistrian University of Athens (Greece)

[20] *Europeana Connecting Cultural Heritage Project*, http://www.europeana.eu

[21] *Dublin Core Metadata Initiative*, http://dublincore.org

[22] Candela L., Castelli D., Manghi P., Pagano P. *Item-Oriented Aggregator Services*, IRCDL2007, Padua, Italy, 2007

[23] Candela L., Castelli D., Pagano P. *OpenDLib: a digital library service system*. In: Handbook of Research on Digital Library Design, Development, and Inpact. pp. 1 - 7. Yin-Leng Theng, Schubert Foo, Dion Goh, Jin-Cheon Na (eds.). Hershey, PA, USA: IGI Global, 2009

[24] University of Michigan, *OAI Toolkit*, http://sourceforge.net/projects/umoaitoolkit

[25] Technical University of Lisbon, Instituto Superior Técnico Repox — A Metadata Space Manager, http://repox.ist.utl.pt

[26] *Lucene Apache Project*, http://lucene.apache.org/

About the Authors

Paolo Manghi is Research Fellow at the NMIS lab of the Istituto di Scienza e Tecnologie dell'Informazione (ISTI), Consiglio Nazionale delle Ricerche, Pisa, Italy. His interests are in the fields of Data Models for Digital Libraries, Types for Compound Objects in Digital Repositories, Digital Library Systems and Services, Service-Oriented Infrastructures for Digital Libraries, Database Systems, Type systems for XML languages, Query languages for XML data, XML P2P database systems. He is a member of the core expert group of Europeana and of the DL.org working group on content interoperability. He is involved in the DRIVER/DRIVER II, OpenAIRE and EFG EC projects and in the Microsoft project R2D2.

Marko Mikulicic is a senior software engineer and developer at CNR-ISTI since 2007. He graduated in Computer Science in 2002 and his main interests cover advanced design (SOAs, WSRF, Web User Interfaces tools), data management systems (RDBMS, triple stores, XML native databases, OAI-PMH and ORE-PMH related technologies and repositories, e.g., Fedora, OpenDlib) and software development environments. He is involved in the DRIVER/DRIVER II, OpenAIRE and EFG EC projects, and in the Microsoft project R2D2.

Leonardo Candela is a researcher at the Networked Multimedia Information Systems (NMIS) Laboratory of the Institute of Information Science and Technologies - Italian National Research Council (ISTI - CNR). Dr. Candela graduated with a degree in Computer Science in 2001 at University of Pisa and completed a Ph.D. in Information Engineering in 2006 at the University of Pisa. He joined the NMIS Laboratory in 2001. Since then he has been involved in the CYCLADES, Open Archives Forum, DELOS, DILIGENT, DRIVER and D4Science projects. He was a member of the DELOS Reference Model Technical Committee and of the OAI-ORE Liaison Group. He is currently involved in the D4Science, D4Science-II and DL.org projects. His research interests include Digital Library [Management] Systems and Architectures, Digital Libraries Models, Distributed Information Retrieval, and Grid Computing.

Donatella Castelli is a Senior Researcher and has worked at the "Information Science and Technologies of the Italian National Research Council" (ISTI-CNR) since 1988. Dr. Castelli graduated in Computer Science from the University of Pisa, and there she was employed as researcher for two years before joining ISTI-CNR Networked Multimedia Information Systems. Since 1996, she has scientifically coordinated several European and Nationally funded projects on digital libraries acquiring considerable experience in this domain. She is currently leading the activity of the DELOS Network of Excellence on Digital Libraries dedicated to the definition of a Reference Model for digital libraries. Her current research interests include digital library architectures and infrastructures, information object modeling and interoperability.

Pasquale Pagano is Senior Researcher at the Networked Multimedia Information Systems Laboratory of the "Istituto di Scienza e Tecnologie della Informazione A. Faedo" (ISTI) of the Italian National Research Council (CNR). He received the M.Sc. in Information Systems Technologies from the Department of Computer Science of the University of Pisa (1998), and the Ph.D. degree in Information Engineering from the Department of Information Engineering: Electronics, Information Theory, Telecommunications of the same university (2006). The aim of his research is the study and experimentation of models, methodologies and techniques for the design and development of distributed virtual research environments (VREs) which require the handling of heterogeneous resources. He has a strong background on digital library distributed architectures. He participated to the design of the most relevant DL systems developed by CNR — ISTI. He is currently the Technical Director of the D4Science-II project and he has been involved in the D4Science, DRIVER II and BELIEF II projects.