Donatella Castelli Yannis Ioannidis Seamus Ross
(Editors)

# Pre-proceedings of the 2nd DL.org Workshop

# Making Digital Libraries Interoperable: Challenges and Approaches

9-10 September 2010, Glasgow, Scotland (UK),
in conjunction with the 14th European Conference on Research and Advanced Technology for Digital Libraries (ECDL 2010)

# Preface

The central theme of the 2nd DL.org Workshop is "Digital Library Interoperability". Interoperability is a multifaceted and very context-specific concept, encompassing different levels along a multi-dimensional spectrum ranging from organizational to technological aspects. These multidimensional aspects make it hard to find generic and fully-comprehensive solutions, thus requiring an approach that investigates interoperability from multiple perspectives.

The workshop builds on the successful outcomes of the 1st DL.org Workshop held in September 2009 during the 13th European Conference on Research and Advanced Technology for Digital Libraries (ECDL2009), which raised many new questions related to interoperability scientific challenges and to its conceptualization. This year's workshop has sought to continue the path initiated during the 1st Workshop by soliciting and gathering research paper submissions that analyze the different aspects involved in achieving interoperability, from conceptualization at a high organizational level to instantiation at process level, as well as to modeling techniques for representing and enabling interoperability between heterogeneous digital libraries, mediation approaches, methods, and supporting systems.

These pre-proceedings contain the papers presented at the workshop. The results illustrated by these papers provide researchers, practitioners and digital library developers with novel solutions for improving digital library interoperability, thus contributing an effective cross-exploitation of the resources managed by these systems.

All these papers have been peer reviewed by at least three reviewers. We would like to thank the members of the international program committee and the authors of the papers submitted for their contribution to the quality and success of the Workshop.

September 2010

Donatella Castelli, Yannis Ioannidis, Seamus Ross
Workshop Organisers
2nd DL.org Workshop

# Organization

The 2nd DL.org Workshop was organised by DL.org (www.dlorg.eu). DL.org is a Coordination and Support Action (FP7 of the European Commission, ICT-2007.4.3, Contract No. 231551) mobilising Digital Library (DL) designers, developers, end-users and researchers from diverse domains in the drive towards interoperability, best practices and modelling foundatons for the enhanced development of next-generation DL systems.

## Organising Committee

| | |
|---|---|
| Donatella Castelli | Senior Researcher, Institute of Information& Technologies, National Research Council ofItaly - DL.org Co-ordinator |
| Yannis Ioannidis | Professor at the Department of Informatics and Telecommunications, University of Athens, Greece - Scientific Chair of the User Working Group |
| Seamus Ross | Dean of the Faculty of Information, University of Toronto, Canada - Member of DL.org's Working Groups on Policy & Quality |

## Program Committee

| | |
|---|---|
| Tiziana Catarci | Professor of Computer Science, University of Rome "LaSapienza", Italy - Member of DL.org's User Working Group |
| Jane Hunter | Professor at the School of Information Technology & Electrical Engineering, University of Queensland, Australia - Member of DL.org's Liaison Group |
| Ronald Larsen | Dean of the School of Information Sciences, University of Pittsburgh, U.S. - Member of DL.org's Liaison Group |
| John Mylopoulos | Professor of Computer Science, University of Toronto, Canada |
| Fabio Simeoni | Research Assistant, University of Strathclyde, Scotland, UK |
| Dagobert Soergel | Dean of Library & Information Studies, University at Buffalo, U.S. - Scientific Chair of DL.org's Functionality Working Group |

# Table of Contents

# A Framework for Digital Library Function Description, Publication, and Discovery: A prerequisite for interoperable digital libraries

G. Athanasopoulos[1], K. El Raheb[1], E. Fox[2], G. Kakaletris[1], N. Manola[1], C. Meghini[3], A. Rauber[4], D. Soergel[5]

[1] Dept. of Informatics and Telecommunications, University of Athens, Greece
{gathanas, k.elraheb, gkakas, natalia}@di.uoa.gr
[2] Dept. of Computer Science, Virginia Tech, Blacksburg, VA, USA fox@vt.edu
[3] Istituto della Scienza e delle Tecnologie della Informazione, Consiglio Nazionale delle Ricerche, Pisa, Italy carlo.meghini@isti.cnr.it
[4] Dept. of Software Technology and Interactive Systems, Vienna University of Technology, Vienna, Austria rauber@ifs.tuwien.ac.at
[5] Dept. of Library and Information Studies, University of Buffalo, New York, USA
Scientific Chair, DL.org Working Group Functionality
dsoergel@buffalo.edu

**Abstract.** Digital Library (DL) interoperability, as it is being reported in the DL.org Digital Library Reference Model, is intimately related to function interoperability. A prerequisite for the later is an appropriate function description, publication and discovery mechanisms. The importance of a framework which accommodates the specification of key DL function characteristics such as interface, behavior, dependencies and semantics, has been highlighted by the DL.org Functionality working group. Such a framework should be used in appropriate registries, which cater for the publication and discovery of DL functionality. In this paper we report the findings of the DL.org Functionality working group in terms of a DL function description framework and a set of contemporary registries that can serve as the basis for the provision of a DL Function interoperability enabling registry.

**Key Words:** Digital Libraries, Function, Interoperability, Registry Repository, Semantic Web

## 1 Introduction

Since the early years of digital library (DL) development, interoperability has been an important concern [13]. Visions of worldwide DLs call for systems that can work together, since issues related to boundaries of the administrative domains, in terms of resource and user control, as well as economic, political, social, and intellectual property concerns force adoption of a distributed system approach. Furthermore, users seek access to information across platforms and sites and interfaces. This is crucial if

scholars and professionals are to address today's interdisciplinary grand challenges; knowledge workers are most effective if rapid response allows them to follow the complex web of citations and concept overlap that is a natural part of the knowledge society.

Interoperability among different DLs and computing systems generally, is crucial to improving efficiency and effectiveness, as well as to avoid segmented systems and duplicated efforts. Based on the DL Reference Model [1] the DL.org project [4] addresses interoperability issues from the perspectives of content, users, functionality, policy, quality and architecture. There are many published definitions for function; the Reference Model defines function as "an action a DL component or a DL user performs" [5]. We define function interoperability later in this paper.

The importance of a suitable framework has been demonstrated in the context of a related approach, leading to the 5S framework [7]. In particular, an ontology has been developed for DL functions, specifying formally the input and output behavior, as well as semantics, including how functions can be composed [8]. Nonetheless, for digital libraries to interoperate beyond the current minimal situation function definition interpretable by machines as well as humans must be commonplace.

Accordingly, in this paper we approach function interoperability by proposing a function specification framework and semantic-based mechanisms, to harmonize function interoperation among DL systems repositories.

## 2 Function Interoperability Concerns

Function interoperability can lead to many benefits. For example, DL functionality extensions can be achieved by the integration of external function implementations. In addition, if DLs are to be rapidly constructed and tailored to specific needs, it is important that they are specified in a general manner and then implemented in a chosen environment using a pool of services satisfying functional requirements.

To achieve function interoperability, functions have to be mutually able to comprehend each other. In particular, a function must be (a) described using a standardized shared format, rich enough to provide all the information about the purpose, the usage and the manner, in which this function performs automated interaction with other systems and (b) implementing standards' compliant interfaces.

The Functionality working group identified a variety of issues concerning function specification, which fall into the following categories:

- *Function Behavior:* refers to the description of supported interactions with actors (systems/users) in terms of supported input/output exchanges, their (logical or temporal) ordering and constraints. The working group investigates the mechanisms to facilitate the description of function behavior properties, comparing syntactic-based and semantic-based behavior description approaches which enable the application of automated compatibility checking algorithms.

- *API/Interface Specification* captures information about operating a function in a manner of specifying Input and Output parameters, using particular data formats and standards.
- *Preconditions and Postconditions* are the specifications of conditions that should be satisfied prior to and after the execution of a function e.g. assumptions that may hold or quality constraints. Including formal methods and tools for the validation of software systems, several approaches have been used up to now to facilitate the specification of such conditions either using programming languages, e.g., Eiffel, Java, C/C++, or declarative standards, e.g., Unified Modeling Language (UML), Object Constraint Language (OCL).
- *Dependencies and Relationships* in the use of functions is another important issue as the operation of a function cannot be described isolated from the operating environment in which it runs; other functions needed, functions that invoke this function, composite functions and workflows should be also considered.

Having specified the necessary components of rich function-descriptions, we must still meet the challenge of publishing these specifications in a manner that can be discovered and understood among diverse systems. This challenge requires the use of registries built on shared standards and mechanisms that can be implemented on diverse systems and technologies.

Several mechanisms and standards exist so far for this purpose, e.g., service registries, however most of them are syntax-based, leading to poor precision and recall due to the lack of appropriate meta-information which would enable the accurate discovery of requested functionality. They also suffer in scalability in terms of published/discovered functions and/or performed transactions, which mainly accrues from the centralized architectural style used by most of the existing registries.

## 3 A DL function specification framework

Addressing the challenge of describing and discovering existing functionality has been intimately related to the activities of the DL.org Functionality WG. As illustrated in Figure 1, a function specification framework along with an enabling function repository can support the interaction and interoperation of DLs. The specification of such as framework has been one of the key objectives of the DL.org Functionality WG. The key idea is that once a function description framework has been specified, many collaborators can create a database with consistent function descriptions

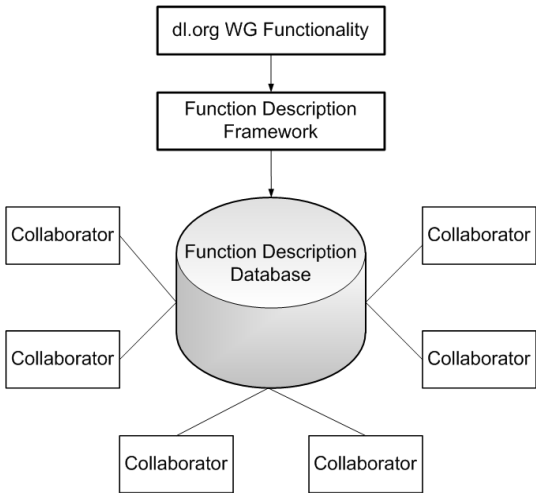


**Figure 1: Framework usage model**

We first revisit the concept of function and define function interoperability. Accordingly a function is "an action a DL component or a DL user performs (abstract function, emphasis on semantics, what does the function do)". In the following ***function*** is used broadly to mean either

- abstract function (such as *annotate* or *browse*) or
- software component implementing a function

Some functions illustrating particular interoperability issues are the following:

| **Behind the scene** | **For users** |
|---|---|
| Feature extraction | Federated search |
| Classification / clustering | Incorporating distributed content on the fly |
| Sharing authority files | Display and visualization |
| Log file analysis | Timelines |
| Sharing user profiles | Maps |
| Harvesting , aggregating | Playing videos |
| Shared storage and backup | Same look-and-feel browse |

Function interoperability has three main purposes:

1 **Functionality sharing:** Find desired functions, and modules that implement them, that will work in a given DL environment
2 Enable **content sharing and federated search**
3 Make **switching from one DL to another** easy for the user

The following scenarios illustrate types on interoperability and reuse of functions:

- The developer of a *browse* module looks for an *automatic clustering* module to incorporate browsing by cluster
- A DL administrator wants to make available a better image search system

- A user found 30 documents in a DL and wants to invoke a *Web service* to create a *multi-document summary*

These considerations lead to a deeper analysis of how functions can be interoperable (interoperability types):

**1 Interoperability from a system perspective, focus on software components**

1.1 Composability (f2 can work with f1)

1.2 Replaceability / interchangeability (f2 can replace f1)
(f1 and f2 serve same purpose)

Definitions 1.1 and 1.2 can be based on process or on data (exchanging data or using same data)

2 **Cross-function** (cross-product) **compatibility: user perspective**
Similar detailed functionality and user interface

A *function description (specification, profile)* tells what a function does and how a system or a human may interact with it. For ease of retrieval and subsequent use of functions, function descriptions should follow a standards laid out in *function description framework.* Such a framework has three components

- An *Entity –Relationship Schema* that defines all the kinds of statements that can be made about a function
- A *function ontology* or hierarchy of functions and more specific functions
- A *function description template* that provides a standard way to organize all the statements about a function.

In the following we discuss these components.

**Entity-relationship schema**

The boxes give some examples of entity types and relationship types respectively.

Entity types

| | |
|---|---|
| **Resource (**Function, DataSet, or DataFormat**)**<br>**Function**<br>**SoftwareComponent**<br>(a software system or module, or a code snippet) | **DesignPattern**<br>**InteropType**<br>**DataSet**<br>**DataFormat** |

Relationship types

| | |
|---|---|
| Resource ***<hasComponent>*** Resource<br>Function ***<isKindOf>*** Function<br>Function ***<implementedBy>*** SoftwareCo. | Function ***<representeBy>***DesignPattern<br>Resource ***<interoperableWith>*** (Resource, InteropType) |

**Function ontology examples**

discover
- navigate
- browse
- search
  - quick search
  - advanced search

Component functions of search

| **Quick Search** | **Advanced Search** |
|---|---|
| Enter a query and click search<br>Enter keywords/phrases for selected fields<br>Limit results to<br>Search subscribed titels<br>Clear | Enter a query and click search<br>Enter keywords or phrases for selected fields<br>Select keyword from a list<br>Select Boolean operator (explicit)<br>Define phrase match (explicit)<br>Clear<br>Search within results<br>Limit results to (preselection)<br>Sort by (preselection)<br>Select display options<br>Display X results per page<br>Display search history |

Component functions of annotate

| |
|---|
| Select object to be annotated (need to indicate selection method)<br>Mark region in the object (many different methods depending on the object)<br>Select type of annotation (highlight, mark with special meaning, text, image, sound)<br>If text, image, sound<br>Specify relationship to object to be annotated<br>Select or create the annotating object (possibly specifying a region<br>Annotating within one system<br>Annotating across systems |

Another example of a function hierarchy (from 5S) is shown in Figure 2.

The function ontology provides a vocabulary which of the component function of an abstract function such as *annotate* are available in a specific module that implements *annotate*. The Reference Model includes beginnings (the upper level) of a function ontology, but this must be much expanded by extension to deeper levels of specificity

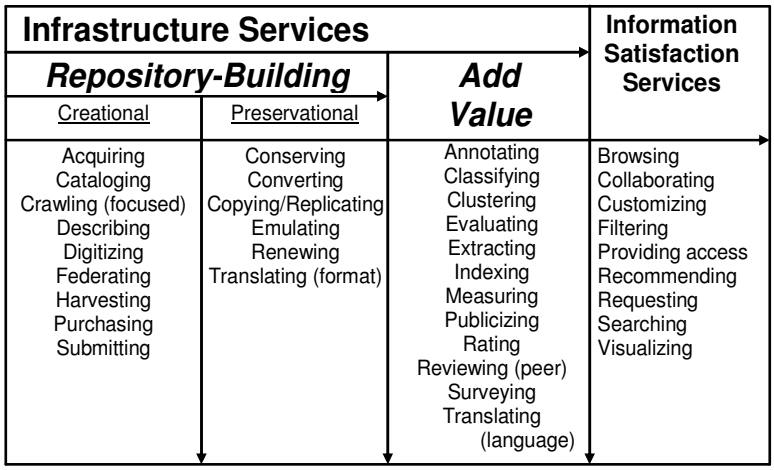


**Figure 2. Function hierarchy example(from 5S)**

A preliminary Function description template is shown in Figure 3. The description of all this information may be based on existing formats and languages used in other domains, such as languages used in the Service Oriented Computing domains, i.e. WSDL, SAWSDL, etc.

It becomes apparent that a repository catering for the description and discovery of functions should accommodate entities and relationships, which support the specification of the template information. For example such a schema should provide prime entities such as that of a "*Function"* or that of "*Data Format"*, and relationships such as "*Interoperable With"* and "*Has Component"*.

Nevertheless, considering that registries have already appeared for the description and discovery of artifacts in other domains, e.g. services in the service-oriented computing domain, it is prudent to consider existing mechanisms and approaches. A list of contemporary registries that have been used mainly in the service domain is presented in the following. These registries can serve as a basis for the provision of a specialized DL functionality registry, which will support the objectives of the function interoperability framework.

## 4 Function Specification & Discovery Mechanisms

Clearly, there are generic concerns regarding interoperability of functions that cut across all types of software systems, whilst others are particular to DLs. Currently, there exist a number of standards that can be used to describe functions and manage their publication and discovery mechanisms.

**Function Behavior**

**Description**: What does the function do. This requires the vocabulary of the function ontology

**Interaction with Actors** (Systems/Users)
Is the function invoked by the user or the system
What actions does the user take
What actions does the system take
Special user groups /roles; user characteristics
Can the function be applied to different contexts

**API/Interface Specification**
Input: Data and parameters, data formats / standards
Output: Data and parameters, data formats / standards

**Preconditions and Postconditions**
Preconditions that should hold prior to the execution of the function
Postconditions that should hold once function execution is completed

**Dependencies/Relationships/Use**
Operating environment in which the function runs.
Other functions it needs
Other functions that invoke this function
Other functions invoked. Composite functions
Work flow

**Interoperability Concerns**
What is required for interoperability (distinguish type of interoperability, for example product compatibility)
How does a specific implementation meet these requirements

**Assessment. Performance. Advice for use**
For specific types of functions, such as search, the template should include quality parameters (for example from the Reference Model to assist with assessment and facilitate comparison of assessments – software evaluation criteria

**Usage conditions**
Rights, type of license, costs, etc.

**Figure 3. Function description template**

The close affiliation of a function and a service renders the Service-Oriented Computing (SOC) domain particularly interesting. The description of many of the properties defined in the framework can be based on existing widely used SOC standards such as WSDL, SAWSDL, OWL-S, WSMO, or even BPEL4WS.

With respect to registry repositories, the contemporary service registries fall within the following categories:

- Syntax based, e.g., ebXML[5], UDDI[3]
- Semantics based, e.g., METEOR-S [11], SpiDeR[15], DIRE [2], PYRAMID-S [14], Ockham [12], e-Framework [6]

The Universal Description, Discovery, and Integration (UDDI) [3] is the best-known industry standard for the description and discovery of web services. UDDI supports an XML-based schema, which defines four key data structures: business entities, business services, binding templates, and tModels. Along with WSDL, it

includes the main tools providing syntactic information on existing services and facilitating the provision of white, green, and yellow page information collections.

Apart from UDDI another syntax-based registry is the ebXML [5]**.** It is a global electronic business standard that is sponsored by UN/CEFACT and standardized by OASIS. It defines a framework that allows businesses to find each other and conduct business based on well-defined XML messages within the context of standard business processes that are governed by standard or mutually-negotiated partner agreements.

Recently, substantial progress has occurred in this area thanks to research and industrial efforts including UDDI registries, similarity search, query languages and indexing efforts, peer-to-peer (P2P) discovery techniques, semantic web approaches, behavioral and ontological matching. Nevertheless, these solutions are typically limited because they are centralized, which means that they may have a limited scalability and low resiliency to faults..

The METEOR-S project consists of a number of subprojects aiming to extend existing syntactic-based standards, e.g., WSDL, UDDI, and BPEL with Semantic Web technologies to achieve greater dynamism and scalability. WSDL-S [17] and BPEL4WS [16] are part of the METEOR-S achievements [11].

PYRAMID-S [10] is a scalable framework for unified publication and discovery of semantically enhanced services over heterogeneous registries. It is based on syntactic, semantic, and Quality of Service (QoS) information, improving precision and recall; it uses a scalable infrastructure which organizes registries based on domains.

SPiDeR [15] is a peer-to-peer (P2P)-based framework that supports a variety of Web service discovery operations. SPiDeR organizes the service providers into a structured P2P overlay and allows them to advertise and lookup services in a completely decentralized and dynamic manner. For advertising and locating services, it supports three different kinds of search operations based on (1) keywords extracted from service descriptions (*keyword- based search*), (2) categories from a global ontology (*ontology-based search*), and (3) paths from the service automaton (*behavior-based search*) using process flow languages, e.g., BPEL. All of these leverage the basic exact key lookup functionality provided by the super-peer overlay, but they incorporate different semantic meanings and so give SPiDeR a richer set of querying capabilities. Consider, e.g., the functionality and process behavior of services during discovery and support for quality rating lookups.

The Ockham [12] effort promotes a scalable, decentralized framework for DL, underlining the need for useful systems which lack unnecessary complexity. The Ockham initiative proposes lightweight protocols e.g. OAI-PMH, SRU and Perl for they are quicker to implement than full-featured interoperability protocols, usually modular and minimal in scope, but can fit clearly into a larger environment, and imply the pre-existence of a reference model guiding the specification.

The e-Framework [6], an international initiative for education and research, proposes different patterns of distributed search, such as keyword, ontology-based searching, hypothesis-based searching, query-by-example, and query-by-content searching, browsing/navigation matching and ranking, data mining knowledge extraction, and portal services.

In summary, from the above listing one may easily conclude that most of the existing registries have mainly focused on the accommodation of syntactic and semantic functional features. But supporting the needs of the function interoperability framework clearly necessitates the provision of additional properties such as behavioral, evaluation/assessment and interoperability information. Thus, using an existing registry as a basis for the provision of the functionality registry envisaged in this paper would also include the extension of the supported information model.

Among the list of presented registries it is apparent that some of them are closely related to the needs of our framework. Specifically the one that seems to be the most appropriate to be uses as a basis is SPiDeR [15]. SPiDeR accommodates the description of behavioral apart from syntactic and semantic information. Moreover, it also provides a distributed (peer-to-peer) platform that can easily scale to the needs of large user communities such as the one related to the DL domain.

## 5 Conclusions

The DL.org initiative has extended the discussion of DL frameworks, especially with regard to interoperability challenges. It has aimed to expand the impact of formal approaches to advance the state-of-the-art and application of DL theories, systems, and implementations. A key part of that effort relates to DL functionality.

In this paper we have illustrated one of the outcomes of the Dl.org Functionality Working Group in that arena. To this end, we call for broader use of registries of DL functions, building upon a function interoperability framework. The proposed approach supports searching and discovering of functions, assessing their interoperability and thus enabling their reusability. This is achieved through a common definition of the notion of function, a template for function description and a function registry that is based on an appropriate entity-relationship schema. Function descriptions are semantically enhanced to cover all different levels of interoperability using function ontologies.

The merits expected to be acquired by the application of the proposed framework are substantial. Additional features supporting DL management and development activities can be placed upon it. For example, DL mangers will be able to better assert and steer the extensibility of existing DLs based on evaluations of the functionality that can be added to existing DLs. Developers and system integrators will be able to identify and reuse appropriate implementations based on their compatibility with existing function implementations. In addition, more advanced approaches towards the on-demand integration of additional functionality, i.e. to satisfy on-the fly the needs of end-users can be better supported. For example, appropriate software integration (composition) algorithms and mechanisms, e.g. AI planning techniques, along with the provided function descriptions and registries can facilitate the automated integration of existing function implementations.

In conclusion, we need to state here again that the presented outcomes are still under investigation. Considering that this is one of the domains investigated by the DL.org project i.e. the functionality domain, additional effort should be spent on

merging and aligning the presented framework with the properties of the rest of the domains, e.g. content, user, architecture. We hope that further research will allow the testing and validation of these ideas, leading to more collaboration and re-use of DL software worldwide.

### Acknowledgments

This work is partially supported by the European Commission under project "DL.org: Digital Library Interoperability, Best Practices and Modelling Foundations" Contract num: 231551. Thanks also go to NSF for support through grants IIS-0916733 and DUE-0840719.

## 6 References

[1] Athanasopoulos G., Candela L., Castelli D., Innocenti P., Ioannidis Y., Katifori A., Nika A., Vullo G., Ross S.: The Digital Library Reference Model Ver 1.0., DL.org Coordination Action on Digital Library Interoperability, Best Practices and Modelling Foundations - Project no. 231551, (Jan. 2010)

[2] Baresi L., Miraz M., A Distributed Approach for the Federation of Heterogenous Registries, Service Oriented Computing ICSOC 2006, 4th International Conference, Chicago, IL , USA, December 2006 Proceedings, Springer

[3] Clement L., Hately A., von Riegen C., Rogers T., Universal Description, Discovery and Integration v3 Standard, Available online at http://uddi.org/pubs/uddi_v3.htm

[4] DL.org: Digital Library Interoperability, Best Practices and Modelling Foundations. EU funded project, Contract no. 231551, Available at: http://www.dlorg.eu

[5] Dubray, J.-J., Amand St. S., Martin J. M., ebXML Business Process Specification Schema Technical Specification v2.0.4, OASIS Standard, Dec.2006

[6] **e-**Framework Technical Model, eFramework initiative for education and research partners UK's Joint Information Systems Committee (JISC) , Australia's Department of Education, Employment and Workplace Relations (DEEWR), July 2009. http://www.e-framework.org/

[7] Goncalves,M., Fox, E., et al.: Streams, Structures, Spaces, Scenarios, Societies (5S): A Formal Model for Digital Libraries. ACM Transactions on Information Systems, April 2004, 22(2): 270-312.

[8] Goncalves, M.., Fox, E., Watson, L.: Towards a Digital Library Theory: A Formal Digital Library Ontology. Int. J. Digital Libraries 8(2): 91-114, April 2008, DOI: 10.1007/s00799-008-0033-1

[9] Goncalves, M., Moreira, B., et al.: What is a good digital library? - A quality model for digital libraries. Information Processing and Management, 43(5): 1416-1437, September 2007, http://dx.doi.org/10.1016/j.ipm.2006.11.010

[10] Kelapure, R., Goncalves, M. Fox, E.: Scenario-Based Generation of Digital Library Services. Proc. 7th European Conference on Digital Libraries (ECDL 2003), 17-22 August, Trondheim, Norway, Springer-Verlag GmbH, Lecture Notes in Computer Science, vol. 2769, 2004, 263-275

[11] Meteor-S, http://lsdis.cs.uga.edu/projects/meteor-s/

[12] Morgan, E.L Frumkin, J., Fox, E.A.: "The OCKHAM Initiative - Building Component-based Digital Library Services and Collections", D-Lib Magazine, Nov. 2004, 10(11) http://dlib.org/dlib/november04/11inbrief.html#FOX

[13] Paepcke, A., Chen-Chuan Chang,K., et al.: Interoperability for Digital Libraries Worldwide. CACM 41(4): 33-43, April 1998

[14] Pilioura T., Tsalgatidou A.: Unified Publication and Discovery of Semantic Web Services *ACM Transactions on the Web*, 3(3), June 2009

[15] Sahin, O. D., Gerede, C. E., et al.: SPiDeR: P2P-Based Web Service Discovery, In Proceedings of the Third International Conference Service-Oriented Computing (ICSOC 2005) (December 2005), pp. 157-169.

[16] Verma, K., Akkiraju, R., et al.:On Accommodating Inter Service Dependencies in Web Process Flow Composition, Proceedings of the AAAI Spring Symposium on Semantic Web Services, March, 2004

[17] WSDL-S, http://lsdis.cs.uga.edu/projects/meteor-s/wsdl-s/

# Quality interoperability within digital libraries: the DL.org perspective

Giuseppina Vullo[1], Genevieve Clavel[2], Nicola Ferro[3], Sarah Higgins[4], René van Horik[5], Wolfram Horstmann[6], Sarantos Kapidakis[7], Seamus Ross[8]

[1] HATII, University of Glasgow, [2] Swiss National Library, [3] University of Padua, [4] Aberystwyth University, [5] Data Archiving and Networked Services (DANS), [6] University of Bielefeld, [7] Ionian University, [8] University of Toronto

**Abstract.** Quality is the most dynamic aspect of DLs, and becomes even more complex with respect to interoperability. This paper formalizes the research motivations and hypotheses on quality interoperability conducted by the Quality Working Group within the EU-funded project DL.org (http://www.dlorg.eu/). After providing a multi-level interoperability framework – adopted by DL.org - the authors illustrate key-research points and approaches on the way to the interoperability of DLs quality, grounding them in the DELOS Reference Model. By applying the DELOS Reference Model Quality Concept Map to their interoperability motivating scenario, the authors subsequently present the two main research outcomes of their investigation - the Quality Core Model and the Quality Interoperability Survey.

**Keywords:** Interoperability; Quality; Digital Libraries; Digital Repositories; Quality Core Model; DELOS Reference Model; DL.org

## 1 Introduction

Among the conclusions of a pioneering paper on DLs interoperability emerged “an urgent need to solve the problems hindering true interoperability on national and international scales” [1, p. 43], and the necessity to investigate this complex issue from cross-domain perspectives. Twelve years after that paper, these two needs are still crucial, and represent the research motivations of the EU-funded DL.org project (http://www.dlorg.eu).

DL.org is aiming to identify requirements, solutions and future challenges for achieving DL interoperability by adopting a cross-domain and multi-layered approach investigating the six core domains (Content, Functionality, Policy, Quality, User, Architecture) captured by the DELOS Digital Library Reference Model [2], which correspond to six dedicated working groups.

This paper focuses on the research analysis on quality interoperability developed within the DL.org Quality Working Group, and illustrates the two main research outcomes of its investigation - the Quality Core Model and the Quality Interoperability Survey.

## 2 A multi-level approach to interoperability

Digital libraries are complex systems, intrinsically interdisciplinary. They involve collaboration support, digital preservation, digital rights management, distributed data management, hypertext, information retrieval, human-computer interaction, library automation, publishing [3, 4].

The most crucial issue involved in the integration of heterogeneous DLs is interoperability. The IEEE defines interoperability as "the ability of two or more systems or components to exchange information and to use the information that has been exchanged" [5, p. 114]; the ISO/IEC 2382-2001 Information Technology Vocabulary, Fundamental Terms defines interoperability as "the capability to communicate, execute programs, or transfer data among various functional units in a manner that requires minimal knowledge of the unique characteristics of those units" [6].

As you can note, the ISO definition contains all the main features needed to characterize interoperability from a general point of view but, as a consequence, it lacks the contextualization necessary to apply it to a specific domain, as the DLs one can be. On the other hand, the IEEE definition takes into account a more functional perspective and it is mainly focused on the exchange of information resources, which represents only one of the facets of interoperability.

In order to achieve interoperability, in fact, DLs need to cooperate and agree at three different levels. Technical agreements cover formats, protocols, security systems, so that messages can be exchanged; content agreements cover the data and metadata, and include semantic agreements on the interpretation of the information; organisational agreements cover the ground rules for access, preservation of collections and services, payments, authentication, etc. [7, 8].

This three-tier interoperability classification (*organizational*, *semantic*, *technical*) has been used in 2004 within the European Commission by the Interoperable Delivery of European eGovernment Services to public Administrations, Businesses and Citizens (IDABC), which developed a European Interoperability Framework for eGovernment services [9], and has been adopted by DL.org in order to address the interoperability issue exhaustively. *Organisational interoperability* – concerned with defining business goals, modelling business processes – involves in particular the role of policy makers. Their part in allowing interoperability has been stressed in the last years [10, 11], to the extent that the European Interoperability Framework 2.0 will also include a *political context* (cooperating partners having compatible visions, and focusing on the same things) and a *legal interoperability* (appropriate synchronization of the legislations) level [11].

## 3 Towards quality interoperability: context and key-issues

A small fraction of works on DLs is dedicated to quality: those that do often focus on the establishment, adoption and measurement of quality requirements and performance indicators. However, the manner in which these quality indicators can interoperate is still under-researched.

The investigation of the DL.org Quality Working Group aims to gain insight into this area, underpinning work on other aspects of interoperability addressed by DL.org (Content, Architecture, Policy, Quality, Functionality, User), according to the DELOS Reference Model [2].

Quality is the degree that the DL conforms to the specified policy that expresses what the goal of a DL is. The policy can cover from very general guidelines to very technical issues, like the maximum response time of a system.

The ISO standard 8402-1994 defines quality as "the totality of characteristics of an entity that bear on its ability to satisfy stated and implied needs" [12]. This definition has been further refined in the ISO standards about quality "the degree to which a set of inherent characteristics fulfils requirements" [13], where requirements are needs or expectations that are stated, generally implied or obligatory while characteristics are distinguishing features of a product, process, or system.

Both definitions highlight how quality can be applied to either overall or single aspects of any products, services and processes, and is "usually defined in relation to a set of guidelines or criteria" [14, p. 33].

A quality model for DLs was elaborated in 2007 within the 5S (Streams, Structures, Spaces, Scenarios, and Societies) theoretical framework [15, 16]: the model was addressed to digital library managers, designers and system developers, and defined a number of dimensions which were illustrated with real case studies.

Within the DELOS Digital Library Reference Model [2], quality is described as one of the six core domains of the Digital Library Universe as follows: "The Quality concept represents the parameters that can be used to characterize and evaluate the content and behavior of a Digital Library. Quality can be associated not only with each class of content or functionality but also with specific information objects or services" [2, p. 20] In Section II of the DELOS Reference Model, a further elaboration is given: "The Quality Domain represents the aspects that permit considering digital library systems from a quality point of view, with the goal of judging and evaluating them with respect to specific facets [2, p. 48].

Overall, the DELOS Reference Model embraces the ISO 9000:2005 definition of quality, discussed above, and defines the *Quality parameter* as a resource that indicates, or is linked to, performance or fulfilment of requirements by another resource. A quality parameter is evaluated by a measure and expresses the assessment of a user. With respect to the ISO definition, we can note that: the "set of inherent characteristics" corresponds to the pair (resource, quality parameter); the "degree of … fulfilment" fits in with the concept of measure; finally, the "requirements" are taken into consideration by the assessment expressed by a user.

Moreover, the representation of the quality parameter provided by the DELOS Reference Model is extensible with respect to the several quality dimensions each institution would like to model.

The relationships and the interdependencies among quality and interoperability can be extremely complex. Quality and interoperability can highly affect each other: offering high quality services can require a high degree of interoperability among the different components of a system; similarly, poorly designed or low quality services can affect the degree of interoperability among different components that can be achieved, thus preventing the successful cooperation among different systems.

The previous considerations mainly concern a functional perspective but the distributed nature and the composition of different services in a user-centered perspective impacts also different dimensions of the quality of a digital library. Consider, for example, the possibility of adding user generated content to the information resources managed by a digital library: this basically breaks the traditional curatorial and selection process that, for example, distinguishes digital libraries from the Web, ensures the quality and reliability of the managed information resources, and keeps a digital library updated and fitting to the needs of one or more user communities. Indeed, the quality of the content added by users may be varying and it may not match the level and the requirements adopted when selecting the information resources to be managed by the digital library. This impacts not only the overall perceived quality of the digital library but also the policies adopted and enforced by the digital library: for example, a moderation step could be envisioned to review users' content before accepting and publishing it in a digital library, but this requires to have specific policies concerning the staff responsible for moderating annotations, the rules of which define when an annotation can be accepted or not, the procedures and functionalities for the ingestion of new content and so on. As a consequence, the quality of the policies themselves adopted by the digital library is concerned in this scenario, since they need to prove to be exhaustive, flexible, and powerful enough to be able to deal with the creation and the addition of new content by users [17].

Quality is still a low-priority issue with regards to DLs interoperability. Quality is not on the same level with the other interoperability issues. There are specific metrics for estimating content quality, functionality quality, architecture quality, user interface quality, etc. For example, content quality could be expressed by the completeness and the accuracy of the content. The overall quality of a digital library – which is a most challenging issue - could deal with the combined quality of all the issues involved, and the effects of the individual quality factors to it. For example, how the timeouts (from the system architecture - that make some of the results inaccessible), the content quality, and the sources functionality affect the quality of the search results.

The DL.org Quality Working Group defined *quality interoperability* as "the possibility for digital libraries to share a common quality framework", and is investigating both the research areas and the real-world cases in which quality issues have been developed.

Quality interoperability is a decentralised paradigm that poses the question of how to link very heterogeneous and dispersed resources from all around the world keeping the reliability of services and data precision. When building systems and operating on data in a distributed infrastructure, for example, each system needs to rely on every part and considerable effort is needed to arrange all the filters to ensure the end user has an homogeneous experience in working with such diverse sources. Quality must thus be provided in a decentralised manner, which requires standards.

One of the main obstacles towards the identification of quality interoperability solutions within the DL field is that often quality is not formally described but implied or "hidden" as a background degree of excellence, compliance to standards, effectiveness, performance, etc. which is not anyhow formally specified. That's why

quality aspects can be found e.g. within content, policy or functionality interoperability solutions.

Upon the agreement to adopt the DELOS Reference Model as the conceptual framework, the Quality Working Group analysed its "Three-tier Framework" [2, p. 17] and suggested to consider an additional level termed "Organisation", over-arching the existing levels of Digital Library (DL), Digital Library System (DLS) and Digital Library Management System (DLMS). The underlying rationale of this extension is that the concept "Digital Library" on its own may not be sufficient to address all interoperability issues that are under investigation in DL.org, in particular the organisational interoperability issues. It is considered that there is an organisation beyond a DL which defines the policy of the overall system in which the DL is operating. As an example, this organisation might be a subject community, a university, or a library steering committee that does not consider the DL itself the primary objective of a policy and might not even be termed 'library' at all.

## 4 The DL.org Quality Core Model

Upon the agreement that - from a system perspective - the core business of DLs resides in the management of their collections, the Quality Working Group identified a quality pattern that is thought to be most characteristic for DLs and that shall help DLs to interoperate in the quality domain. This pattern is grounded on the DELOS Reference Model Quality Concept Map [2, p. 191], where *Generic parameter*, *Content parameter* and *Policy parameter* express three of the six different facets (including also *Functionality*, *User* and *Architecture* parameters) of the *Quality parameter*. The pattern includes the three *Quality parameter* facets which have been considered crucial to allow interoperability, and has been thus called the "Quality Core Model" (Fig. 1).

The Quality Core Model's motivating scenario considers that representatives of two (or more) DLs have a round table to negotiate a service level agreement (SLA) defining their interoperability requirements and for this establish a quality threshold that each individual DL has to meet or exceed; in this case, "Quality" would provide transparent qualitative or quantitative parameters for defining the threshold.

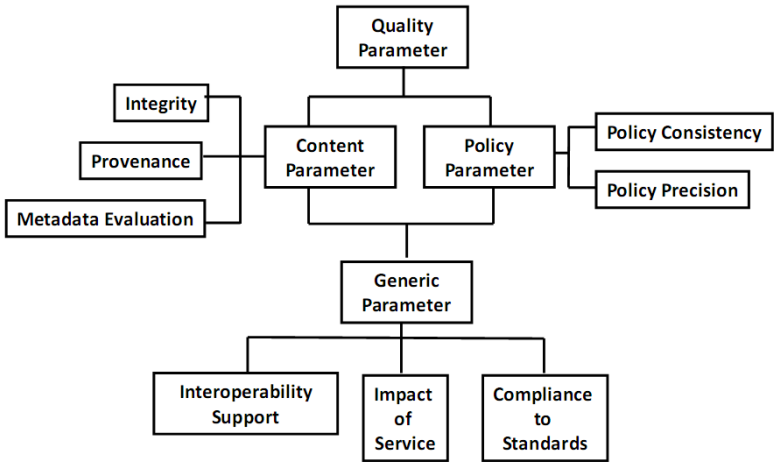


**Fig. 1.** The DL.org Quality Core Model

As facets of the *Quality parameter*, the *Generic*, *Content* and *Policy* parameters includes specific sub-parameters. The DELOS Reference Model Concept Map [2, p. 191] comprehensively lists forty-two sub-parameters, distributed within the six *Quality parameter*'s facets.

The sub-parameters that are currently included in the Quality Core Model are:

- *Compliance to standards*: the degree to which standards have been adopted in developing, managing and delivering a digital library service [2]
- *Impact of service*: the influence that a digital library service has on the users' knowledge and behaviour [2]
- *Interoperability support*: the capability of a digital library to interoperate with other digital libraries as well as the ability to integrate with legacy systems and solutions
- *Integrity*: the quality of being whole and unaltered through loss, tampering, or corruption [18]
- *Metadata evaluation*: the measurements of metadata schemas and their individual fields to support the collection, management, discovery and preservation of digital library content [2]
- *Provenance*: information regarding the origins, custody, and ownership of an item or collection [18]
- *Policy consistency*: the extent to which a policy or a set of policies are free of contradictions [2]

– *Policy precision*: the extent to which a set of policies have defined impacts and do not have unintended consequences [2]

The Quality Working Group investigated the Quality Core Model parameters' definitions and relationships – referring to The Society of American Archivists' definitions [18] when it was felt the DELOS Reference Model ones would still need to be enhanced, and producing related real user scenarios. As an example, we present here a user scenario from DRIVER (http://www.driver-community.eu/) on *Policy consistency*:

– *Check consistency between the DRIVER primate of fulltext exposure (content policy: DRIVER Guidelines) and DRIVER repository registration policy.*
The DRIVER repository network has guidelines for content providers that define how to expose fulltexts with OAI-PMH. This is to make clear that DRIVER expects repositories to expose fulltexts rather than catalogue entries. At the same time DRIVER has registration policies for including repositories in the network. Consistency can be checked by whether or not the content policy is reflected in the registration policy. During registration DRIVER offers repositories a validator tool to check their compliance with the DRIVER- Guidelines. However, for logical and technical reasons a binary decision for or against compliance cannot be made and repositories (and therefore also DRIVER) may still offer records to users that do not lead to a full text. As a consequence, an inconsistency between content policy and registration policy could be stated. However, DRIVER applies a quantitative compliance rate. This simplified example makes clear that an actual application of the DELOS Reference Model to a real user scenario may pose numerous challenges in the modelling relations provided by the DELOS RM, e.g. the relation between Policy by compliance and Policy consistency.

The selection of quality parameters for the "Quality Core Model" is not intended to alter the DELOS Reference Model Quality Concept Map, or to ignore quality aspects such as the functionality or the user ones; it arose, instead, from the application of the Quality Concept Map to a specific interoperability scenario, and from the need to identify - with a practical approach - core quality aspects that real-world DLs should take into account and measure in view of interoperability.

## 5 The Quality Interoperability Survey

The Quality Working Group is currently working on implementing the "Quality Core Model" by creating and running a Quality Interoperability Survey.

Digital repositories are included in the Quality Working Group's survey in the same way as DLs because they can be considered as the most dynamic example of information systems [19].

The results will help to understand what the professional community understands by quality interoperability issues, how it responds to them and from this to identify best practices in this area.

The Quality Working Group successfully completed the survey pilot and recently produced and distributed its official online version. The survey was organised in order to gather information on quality requirements regarding the different quality facets and asked specific questions on quality interoperability, focusing on the Quality Core Model parameters.

One of the main results of the survey pilot was that "quality" is considered as a subjective and dynamic entity, and that a common understanding even on the basic terms used is needed. In response, the Quality Working Group has prepared a glossary of terms that has been integrated with the survey's official version. In addition, comments from the pilot participants enabled the Group to simplify and improve the structure of the questionnaire (see Appendix, Table 1).

Among the specific best practices and recommendations, it is expected that a key-role will be played by certifications, checklists, validators and standards. These are typically quality areas in which digital libraries seeking to interoperate and negotiating a service level agreement (as postulated in the Quality Working group's motivation scenario) will need to define their approaches as a means to set their interoperability requirements and establish the quality threshold for their respective services.

A high level set of practical recommendations based on the Quality Core Model parameters and the Quality Interoperability Survey results will be then produced and presented as a checklist. It is hoped that the checklist will enable institutions to prepare the ground for interoperability discussions on quality but also that it may suggest areas in which institutions may / should be checking the quality of their data or services.

The checklist will first enable institutions to list areas that may be checked for quality in their own individual repositories / digital libraries taking into account that within one institution there may be a number of these with different responses. For each element examples will be provided based on the responses from the survey. Areas covered are:

- Formats
- Format compliance checking tools (and results)
- Metadata standards
- Metadata compliance checking tools (and results)
- Communication protocols
- Communication protocol compliance checking tools (and results)
- Web guidelines / standards in the areas of accessibility, usability, multilingualism
- Legal obligations *e.g.* for web standards

A second level will enable institutions to indicate whether and with what results they have already followed multi-level guidelines such as the DRIVER ones [20], taken part in certification processes such as DINI [21], monitored user satisfaction, and to check current policy for interoperation if such exists.

Together these first two areas will enable evaluation of the DL concerned according to the *generic quality* parameters of the Quality Core Model (*Interoperability support*, *Impact of service*, and *Compliance to standards*).

The central part of the checklist covers the parameters identified by the group as the most crucial for interoperability between digital collections/libraries. These points cover the following areas:

- Content
  - identifiers
  - metadata (type, compulsory elements, checking, use of ontologies etc., completeness)
  - identity authentication
  - provenance
  - tracking /recording changes
  - preservation

These areas cover the integrity, provenance and metadata parameters of the model.

Finally, a checklist of areas in which an institution may/should have policy guidelines (*e.g* for user access, preservation, metadata, networks, authentication, and service level agreements) recaps the areas above and covers the *policy quality* parameter section of the model.

An institution that completes the list and brings together the different documents pertaining to these parameters will not only be in an excellent position to analyse its own quality of system and service but also be well placed to compare and adjust in negotiation with other institutions as hypothesised in the group's motivating scenario.

## 6 Conclusions and Future Work

Quality is the most dynamic aspect of DLs, and becomes even more complex with respect to interoperability. By grounding its research on the DELOS Reference Model, analysing its Quality Concept Map, providing additional definitions and real user scenarios, the DL.org Quality Working Group identified a core selection of parameters that are considered to be essential to achieve interoperability.

The simplified pattern – called the "Quality Core Model" - has been implemented by developing and running an online survey on quality interoperability of current DLs and digital repositories. It is expected that the survey results will give an overview of best practices and adopted solutions towards DLs quality interoperability, by testing the feasibility of the Quality Core Model.

The survey will also lead DLs and digital repositories managers to identify core quality aspects with regards to interoperability, providing them with a first quality interoperability checklist.

In parallel, the DL.org Quality Working Group will elaborate further the definition of *quality interoperability*, by instantiating it at a technical, semantic and organisational level, providing examples of how it can be achieved.

**Acknowledgments.** The work reported has been partially supported by the DL.org Coordination and Support Action, within FP7 of the European Commission, ICT-2007.4.3, Contract No. 2315515

## References


1. Paepcke, A., Chang, C. K., Winograd T., García-Molina H.: Interoperability for digital libraries worldwide. Commun. ACM 41(4), 33–42 (1998)
2. 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.: The DELOS digital library reference model. Foundations for digital libraries, version 0.98. Tech. rep., DELOS, A Network of Excellence on Digital Libraries (2007). http://www.delos.info/files/pdf/ReferenceModel/DELOS_DLReferenceModel_0.98.pdf
3. Fox E.A., Marchionini G.: Toward a worldwide Digital Library. Commun. ACM 41(4), 29–32 (1998)
4. Borgman, C.: What are digital libraries? Competing visions. Inf. Process. Manage. 35, 277–243 (1999)
5. IEEE: IEEE Standard Computer Dictionary. A Compilation of IEEE Standard Computer Glossaries : 610. IEEE, New York (1991)
6. ISO/IEC 2382:2001. Information Technology Vocabulary – Fundamental Terms
7. Shen, R., Vemuri, N. S., Fan, W., Fox, E. A.: Integration of complex archaeology digital libraries: An ETANA-DL experience. Information Systems, 33(7–8), 699–723 (2008)
8. Arms, W., Hillmann, D., Lagoze, C., Krafft, D., Marisa, R., Saylor, J., Terrizzi, C., Sompel, H. van de: A Spectrum of Interoperability: The Site for Science Prototype for the NDSL. D-Lib Mag. 8(2) (2002). http://www.dlib.org/dlib/january02/arms/01arms.html
9. IDABC: European Interoperability Framework for pan-European eGovernment Services. European Commission, Luxembourg (2004). http://ec.europa.eu/idabc/en/document/2319/5644
10. QualiPSo Consortium: Organizational Interoperability: Important issues and requirements for organizational interoperability and state-of-the-art of corresponding methods, languages, and notations. QualiPSo, Quality Platform for Open Source Software (2008). http://www.qualipso.org/sites/default/files/WD3.3.1A-V2.0.pdf
11. IDABC: Draft document as basis for EIF 2.0. European Commission, Luxembourg (2008). http://ec.europa.eu/idabc/servlets/Doc?id=31597
12. ISO 8402:1994. Quality management and quality assurance – Vocabulary
13. ISO 9000:2005. Quality management systems – Fundamentals and vocabulary
14. UKOLN Metadata Group: Selection Criteria for Quality Controlled Information Gateways, DESIRE, Development of a European Service for Information on Research and Education, Project Deliverable (1996). http://www.ukoln.ac.uk/metadata/desire/quality/quality.pdf
15. Goncalves M.A., Fox E., Kipp N. and Watson L.: Streams, structures, spaces, scenarios, societies (5S): a formal model for digital libraries. ACM Trans. Inf. Syst. 22: 270–312 (2004)
16. Goncalves, M.A., Moreira, B.L., Fox, E.A., Watson, L.T.: What is a good digital library? - A quality model for digital libraries. Inf. Process. Manage. 43(5), 1416–1437 (2007)
17. Ferro, N.: Quality and Interoperability: The Quest for the Optimal Balance. In: Iglezakis, I., Kapidakis, S., Synodinou, T. editors, E- Publishing and Digital Libraries: Legal and Organizational Issues. IGI Global, USA (2010) (in print)
18. Pearce-Moses, R.: A Glossary of Archival and Records Terminology. The Society of American Archivists, Chicago (2005). http://www.archivists.org/glossary/
19. Weenink K., Leo Waaijers L., Godtsenhoven K. van: A DRIVER's Guide to European Repositories: Five studies of important Digital Repository related issues and good Practices. Amsterdam University Press, Amsterdam (2007). http://dare.uva.nl/document/93898
20. DRIVER: DRIVER Guidelines 2.0: Guidelines for content providers - Exposing textual resources with OAI-PMH. DRIVER (2008). http://www.driver-support.eu/documents/DRIVER_Guidelines_v2_Final_2008-11-13.pdf

21. Dobratz S., Scholze F.: DINI institutional repository certification and beyond. Library Hi Tech 24(4), 583–594 (2006)

## Appendix

**Table 1.** The Quality Interoperability Survey's questionnaire*

| A. QUALITY | |
|---|---|
| 1 | Which type of digital objects are included in/collected by the DL/digital collection eg texts, images, audio, etc. (please separate names by commas)? |
| 2 | Do you have any guidelines on formats for these objects? |
| 2a | If yes, which? |
| 3 | Do you use any validation tools to check the format compliance? |
| 3a | If yes, which? |
| 4 | Which are the metadata standards in place? |
| 5 | Do you use any validation tools to check the metadata compliance? |
| 5a | If yes, which? |
| 6 | Which are the communication protocols standards in place? |
| 7 | Do you use any validation tools to check the compliance to the communication protocols standards? |
| 7a | If yes, which? |
| 8 | For which aspect(s) do you have guidelines or standards for the Web interface?<br>[ ] Accessibility<br>[ ] Usability<br>[ ] Multilingualism |
| 8a | Please specify which guidelines |
| 9 | Do you have any specific legal obligations on the Web interface? |
| 9a | If yes, which? |
| 10 | Do you follow multi-level guidelines eg DRIVER 2.0, national association or institutional guidelines? |
| 10a | If yes, which? |
| 11 | Have you ever been involved in a certification process eg with TRAC, DRAMBORA, DINI? |
| 11a | If yes, please provide details |
| 12 | Do you monitor user satisfaction? |
| 12a | If yes, by which method(s)? |
| 13 | Do you have collection(s) that need to interoperate with collection(s) from other institutions? |
| 13a | If yes, please check the appropriate box(es)<br>[ ] Academic institutions<br>[ ] Private institutions<br>[ ] Public institutions<br>[ ] Research institutions<br>[ ] Other |

| | |
|---|---|
| 13b | If Other, please specify |
| 13c | Please indicate any written/publicly available policy on each interoperation |
| B. QUALITY AND INTEROPERABILITY | |
| 14 | Is the DL/ digital collection interoperating as part of a network eg DRIVER, TEL, etc.? |
| 14a | If yes, which? |
| 15 | Are persistent identifiers mandatory for the collection? |
| 15a | If not, what percentage has them? |
| 16 | What percentage of those resources that do have a persistent identifier still resolve correctly? |
| 17 | Which standard(s) are used for the persistent identifiers? |
| 18 | To what extent do you use Dublin Core metadata? |
| 19 | Does the DL system define authorization for identities that have been authenticated by identity federations? |
| 19a | If yes, please specify |
| 20 | Do you measure the impact of your DL services? |
| 20a | If yes, please detail |
| 21 | Do you record/track changes to data items? |
| 21a | If yes, please detail |
| 22 | Do you modify the content for preservation purposes? |
| 22a | If yes, please detail |
| 23 | Please describe any actions you take concerning the tracking of provenance at a collection and/or an item level |
| 24 | Is there a minimum set of metadata fields which are compulsory when a new item is submitted? |
| 25 | How do you ensure consistent metadata values eg data values, subject terms, etc.? |
| 26 | Do you use thesauri, word lists, ontologies or authority files |
| 26a | If yes, please detail |
| 27 | Do you use automation tools for technical metadata creation? |
| 27a | If yes, please detail |
| 28 | Do you monitor updates, additions and changes to community practice for any standards you use? |
| 29 | On a scale 1-5 [1 very incomplete; 2 incomplete; 3 sufficient; 4 complete; 5 very complete], how complete is your metadata? |
| 30 | In your opinion, what is the single greatest barrier to metadata creation? |
| 31 | Please indicate if your organisation or the DL itself follows written policies or some other statement(s) that guide its development and maintenance<br>[ ] User access<br>[ ] Preservation<br>[ ] Metadata<br>[ ] Networks<br>[ ] Online collections and services<br>[ ] Intellectual property<br>[ ] Authentication |

| | |
|---|---|
| | [ ] Service Level Agreements<br>[ ] Other |
| 31a | If Other, please specify |
| 32 | Please provide the URL of any publicly available policies according to the following areas<br>Please indicate if your organisation or the DL itself follows written policies or some other statement(s) that guide its development and maintenance<br>*Same categories of 31* |
| 33 | Do you know of any inconsistencies between the above policies? |
| 33a | If yes, please detail |
| 34 | Are there any procedures in place to check how well a policy is implemented? |
| 34a | If yes, please specify |
| 35 | In your opinion, are there any crucial quality aspects for interoperability that are not covered by part B of this survey (14-36)? |
| 35a | If yes, please specify |
| 36 | Please tick the appropriate box(es)<br>*1. Successful interoperability is largely a technical issue*<br>( ) Strongly disagree<br>( ) Disagree<br>( ) Neutral<br>( ) Agree<br>( ) Strongly agree<br><br>*2. Quality aspects are crucial for successful interoperability*<br><br>*3. We considered quality aspects for improving our interoperability within our organization* |
| C. FINAL QUESTIONS | |
| 37 | What do you consider to be a “good quality” Digital Library (DL)? |
| 38 | Are you familiar with the DELOS Reference Model? |
| 38a | If yes, to what extent the models plays/played a role in the design and operation of your DL? |

** Questions 2, 3, 5, 7, 9, 10, 11, 12 13, 14, 15, 19, 20, 21, 22, 24, 26, 27, 28, 33, 34 allow Yes/No/Don’t know answers, while questions 35 and 38 allow Yes/No answers only. Answer options for sentence 1 in question 36 are repeated for sentence 2 and 3.*

# Modeling Users and Context in Digital Libraries: Interoperability Issues

**Anna Nika[1], Tiziana Catarci[2], Akrivi Katifori[1], Georgia Koutrika[3], Natalia Manola[1], Andreas Nürnberger[4], Manfred Thaller[5]**

[1] Department of Informatics and Telecommunications, University of Athens, Greece
{a.nika, vivi, natalia}@di.uoa.gr
[2] Department of Computer and System Sciences, Sapienza University of Rome, Italy
catarci@dis.uniroma1.it
[3] Department of Computer Science, Stanford University, California, USA
koutrika@stanford.edu
[4] Faculty of Computer Science, Otto-von-Guericke University Magdeburg, Germany
andreas.nuernberger@ovgu.de
[5] Humanities Computer Science, University of Cologne, Germany
manfred.thaller@uni-koeln.de

**Abstract.** The modeling of user characteristics is an important mechanism for the support and enhancement of personalization in Digital Libraries (DLs). Contextual information is sometimes considered in the literature as part of the user model, although it in fact influences user-DL interaction. However, context modeling and user modeling are strongly interrelated. Similar methods are employed for the representation of user context and user model and the use of ontologies constitutes a promising approach for such a representation because they facilitate the sharing of information and reinforce interoperability. In this paper, we define the user model and user context for Digital Libraries, we examine identified context dimensions as well as context-independent and context-dependent user model attributes, we proposes the use of an ontology for user representation in DLs, and we discuss the advantages of such an approach for augmenting personalization and achieving user interoperability.

**Keywords:** user context dimensions, user model attributes, ontology, user interoperability

## 1 Introduction

Digital Libraries (DLs) are heterogeneous systems that provide different functionalities. So far, there is no accepted definition of what a Digital Library is. According to the Digital Library Reference Model [1], a DL is *an organization, which might be virtual, that comprehensively collects, manages and preserves for the long term rich digital content, and offers to its user communities specialized functionality on that content, of measurable quality and according to codified policies*. This

general definition affects also the user attributes that need to be captured by the DL in order to offer a personalized experience to different users. Until now, there is no generally accepted user model for DLs and the Reference Model does not specifies what user characteristics are significant to be reflected in such a user model. Furthermore, contextual information is of particular importance for Digital Libraries because it influences user interaction with the DL.

In the last years, much research has been devoted to the understanding and the subsequent definition and modeling of the notion of context. Context has been examined for general purpose systems as well as specific systems such as mobile and sensor networks where context changes rapidly. As far as the authors are aware, there has been no comprehensive study of the notion of user context in the DL field.

In this paper, our aim is to propose definitions for the user model and the user context for DLs as well as to provide a clear distinction between user model attributes and user context dimensions. Context-aware DLs will support personalized access to users by providing the right information and services. It is evident that by identifying and obtaining relevant context, DLs may adapt themselves to better suit users needs. Nevertheless, user context modeling is conceptually different from modeling of user attributes. Some context dimensions can hardly be considered information about the user, but, in the literature, they are often captured and included in user models. Similarly, user attributes are frequently regarded as part of different context models. However, context modeling and user modeling are strongly interrelated. Not only similar methods are used for the representation of user context and user model, but also context dimensions influence specific user attributes such as preferences and interests that will be named context-dependent attributes in following sections. For this reason, we investigate and propose the use of an ontology for the representation of the user model and the contextual information that affect the user's behavior because ontologies facilitate the sharing of information and enhance interoperability.

The remainder of this paper is structured as follows. Section 2 provides related work on the definition and modeling of context as well as a review of the current user model representation approaches. Section 3 introduces our definitions for user model and user context and the identified context dimensions as well as context-independent and context-dependent user model attributes. Section 4 proposes the use of an ontology for the user representation in DLs and discusses the advantages of such an approach. Section 5 presents our thoughts for achieving user model and context interoperability with the use of the proposed ontology. Finally, Section 5 concludes the paper and introduces future research directions.

## 2 Related work

### 2.1 Context Dimensions and Modeling

In recent years, researchers have attempted to better understand context by creating different context definitions. Schilit et al. [20] were among the first that focused their attention on the definition of context. They described is as location, identities of nearby people, objects, and changes to those objects. Benerecetti et al. [2] introduced

two dimensions of context: physical context and cultural context. Physical context is a set of environmental characteristics while cultural context includes user information, user preferences, background beliefs, etc. Lieberman and Selker [13] defined context as the state of the user, of the physical environment, and of the computational environment as well as the history of user-computer-environment interaction. The authors introduced information about the computational environment as a separate context dimension because they believed that such information could be of interest to the user and related computing devices. Lucas [14] distinguished three dimensions of contextual information: physical context, device context, and information context. Device context includes characteristics of the device itself while information context is related to the knowledge about information objects whose existence and identity are distinct from those of the devices that process them. Gross and Specht [10] considered four dimensions of context: identity, location, time, and environment. Brown et al. [3] defined context as location, identities of the people around the user, the time of day, season, temperature, etc. Ryan et al. [19] included in their work four user context dimensions: user's location, environment, identity, and time. Nowadays, Dey and Abowd's definition of context [7] is considered as the most accepted one. They defined context as follows: "*Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves*."

Apart from the aforementioned attempts to define context, much research has been devoted to the area of context modeling. A well designed context model constitutes a critical step for the development of context-aware systems. Several approaches have been proposed for the representation of context. Strang and Linnhoff-Popien [21] performed a comparative study of different context modeling approaches. The authors identified six categories for context representation: a) Key-value models, b) Markup scheme models, c) Graphical models, d) Object oriented models, e) Logic based models, and f) Ontology based models. They arrived at the conclusion that the most promising context modeling method can be found in the ontology category [22] because it provides a good sharing of information with common semantics.

### 2.2 User Model Attributes and Representation Approaches

User models and the attributes captured in these models have been studied extensively. The number of features represented in a user model depends to a large extent on the kind of adaptation that the respective system wishes to provide. Rich [18] proposed a three-dimensional space of user models: 1) canonical vs. collection of individual user models, 2) explicit vs. implicit user models, and 3) long-term vs. short-term user models. Brusilovsky and Millán [4] collected the five most popular and useful user characteristics: the user's knowledge, interests, goals, background, and individual traits. Preferences [12] and interests [24] are considered as very important attributes for most systems that incorporate user profiles. Tazari et al. [23] defined the user model as a set of parameters such as personal information, general characteristics, education, occupation, interaction-related info, and user state. General

characteristics consist of physical factors such as weight and height, physical disabilities, and abilities like reading, speaking and writing. Personal information includes the user's name, birthday, address, bank account, and credit card information. The state of the user is described by a set of parameters like current activity, current terminal, location, and orientation.

There are several approaches related to the structure and representation of information in user models. Overlay models [27] are used for modeling user's knowledge as a subset of the domain model. The overlay model stores an estimation of the user's knowledge level for each part of the domain. For the representation of user interests a common model is the weighed vector of keywords [15]. Weights are related to keywords and express numerical description of user's interests. A user model may be represented by a weighted semantic network [8] in which each node expresses a concept. The use of ontologies constitutes a promising approach for the representation of user models because they facilitate the sharing of information and enhance interoperability. Significant proposed ontologies produced for user and context modeling are the General User Model Ontology [11], the Unified User Context Model [16], and the Ontology based User Model [17]. Finally, Golemati et al. [9] created a user profile ontology that includes mainly static information without modeling dynamic characteristics like the current user position.

## 3 Identified User Model Attributes and User Context Dimensions

After having examined the various definitions of user model and user context, we propose in this section our definitions for these two important notions of the Digital Library field.

**Definition 1**: A *user model* for Digital Libraries is a collection of the most important user attributes, either context-independent or context-dependent, that are captured in order for DLs to behave differently to different users. Context-independent are the attributes that do not change if the context is different. Context-dependent attributes are those that are affected by context variations.

**Definition 2**: *User context* in a Digital Library is a set of dimensions that affect user interaction with the DL. Specifically, context dimensions influence context-dependent user model attributes that the DL captures. This results in different personalization information provided by the DL for different contexts.

A thorough study of the different context definitions presented in Section 2.1 [2, 3, 7, 10, 13, 14, 19, 20] led to the identification of the following **user context dimensions** that are considered important for Digital Libraries.

- **Role**. The Role is a set of allowed actions that a user is eligible to perform. The Digital Library Reference Model [1] defines four basic Roles that a user can play in a DL: End-user, DL Designer, DL System Administrator, and DL Application Developer.

- **Goal**. The user's goal represents the current purpose for a user's work within a DL and it is the most variable context dimension as it can often change several times within a session. The goal is an answer to the question "What does the user actually want to achieve?" [4].
- **Mood**. Mood represents the user's current emotional state, e.g., happy, sad. It can be acquired by explicitly asking the user to set his mood in the respective field of the DL interface.
- **Location**. Location information includes the user's absolute or relative address. Location can be acquired by the IP address if the user uses a PC or by using GPS information that provides geographic coordinates of the user. The geographic coordinates can be used to understand where the user is, e.g., street, at home, in the car.
- **Time**. Time is an important context dimension. Besides the specification of time in UTC format, a different representation can be used to group various time periods, e.g., working hours, weekend.
- **Device**. Device information is important in order for the DL to provide general services. A device can be a PC, a laptop, or a mobile phone. This dimension also includes information such as screen size, operating system, and memory.

The general characteristic of the aforementioned context dimensions is that they influence context-dependent user model attributes. The selection of some dimensions, e.g., Role and Goal, to be part of the context and not the user model was based on the criterion that these features influence user attributes. The identification of context dimensions is considered difficult and context-acquisition mechanisms are of particular importance. Chen [6] introduced three different approaches for context acquisition: direct sensor access, middleware infrastructure and context server.

Apart from the above context dimensions, our investigation [4, 9, 12, 23, 24] revealed the following specific attributes for the two user model categories.

**Context-independent user model attributes**:

- **Personal Information**. Personal information includes basic user information like name, birthday, country, address, and email.
- **Physical Characteristics**. These characteristics consist of gender, eye color, height, weight, etc.
- **Ability**. Ability contains information about user abilities such as writing, reading as well as user disabilities such as blindness, deafness, and other physical disabilities.
- **Education**. Education embodies information about user's diplomas and foreign languages that the user knows.
- **Profession**. This attribute includes information about user's profession such as position, type, and company.
- **Expertise**. This attribute contains all kind of expertise like computer expertise [9].

**Context-dependent user model attributes**:

- **Credentials**. This attribute includes the user name and the password that the user uses to login to the DL. Credentials depend on the Role that the user plays

in a DL, e.g., a user uses other credentials to connect to a DL as an End-user and other as an Administrator.

- **Access Rights**. Access rights embody all the related information about user's authority in certain areas and resources of the DL. This attribute depends on the Role that the user plays, e.g., a user has other access rights as an Administrator and other as an End-user.
- **Preference**. This attribute is related to the liking of something or the favoring of one thing over another, e.g., "prefer chocolate, not vanilla", "like green color". Preference may be influenced by all context dimensions**.** For example, when Kate is happy she prefers to hear pop songs, whereas when she is sad she prefers to hear gothic songs.
- **Interest**. Interest can be something that concerns or is important for a user, e.g., interest in music, interest in reading. This attribute may be influenced by all context dimensions. For example, John is more interested in sports than reading during the weekend.
- **Activity**. This attribute contains user's current activity, e.g., search for a book, read an article. Also, Activity can be regarded as the history of the user-DL interaction. It is apparent that Activity may be influenced by all context dimensions. For example, if Ian performs a DL maintenance activity, this is related to the DL Administrator Role that he plays. Similarly, if Mary is in Rome, this influences her activity of searching for the book "Rough Guide to Rome".

## 4 Proposed Ontology for User Representation in Digital Libraries

As we have already identified, ontologies constitute a widely accepted approach for the representation of user models and context information. Ontology is a set of concepts with "is-a" relationships between them. Each concept is a class that may have one or more parent classes. Also, a class has properties describing various features of a class, as well as restrictions on the properties. Each property has a type and could have a restricted number of allowed values, which may be of simple types (strings, integers, Boolean, etc.) or instances of other classes. An instance corresponds to an individual object and has a concrete value for each property of the class it belongs to.

In our work, we propose an ontology for the formalization of the user representation in Digital Libraries. A class hierarchy of the ontology, as displayed in Protégé, is presented in Figure 1. The central classes in the ontology are the "DLUserEntry" abstract class and the concrete class "User". The "DLUserEntry" is used to provide the hierarchy of the user model attributes and context dimensions that a DL should capture for a user. This class has as abstract subclasses the "UserContextDimesnion" class and the "UserModelAttribute" class that represent user context and user model respectively. The "UserModelAttribute" class is further divided into "Context-DependentAttribute" class and "Context-IndependentAttribute" class. Each class contains as subclasses all the corresponding attributes identified in Section 3. Finally, the class User is used to describe the user of a DL and contains as

properties instances of all the context-independent and context-dependent user model attributes.

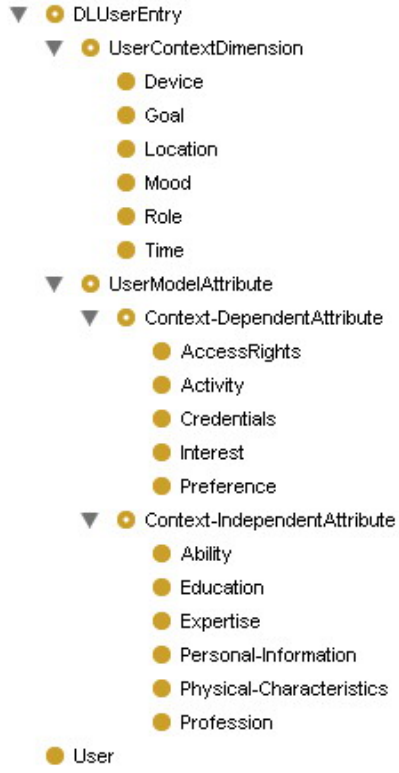


**Fig. 1.** DL Ontology as displayed in Protégé.

The important advantage of using this ontology is that we can connect context-dependent user model attributes with context dimensions, i.e., each attribute may be accompanied by some or all context dimensions. These relations for context-dependent attributes were identified in Section 3. For each context dimension that influences an attribute, a property is created in the specific context-dependent attribute that is an instance of the respective context dimension. For example, we identified that the context-dependent attribute "Preference" may be influenced by all context dimensions. Particularly, the relation between the preference and the user's mood was examined in Section 3. To express this, the class Preference should have as property an instance of the class Mood. In this way, information about preferences can be always accompanied by the mood information. Such a property expressed in OWL for the class Preference and the class Mood is presented below. The same applies for all the other context dimensions and context-dependent user model attributes.

```
ObjectProperty(pp:accompaniedBy_mood
domain(pp:Preference) range(pp:Mood))
```

The significance of accompanying each context-dependent user model attribute with context comes from the need to have better personalization. User's preferences, interests, and other attributes are different according to the context around the user. By enabling this context to be part of each attribute we increase the potentials for the DL to offer better personalization services. In this way, the DL may not only retrace the preferences and interest of the user, but also the context present when these attributes were captured to offer tailored recommendations based on the current context. For example, if John was searching for rock music songs when he was "happy" during the "weekend" and "at home", it is possible the DL system to recommend rock songs when John is "at home", or during the "weekend", or when he is "happy", or for whatever combination of the previous context dimensions.

## 5 User Interoperability for the Proposed Ontology

Nowadays, users interact with different Digital Libraries and their information is scattered throughout them. This raises the need for personalization not to be limited to one system but to be applied across DLs. Cross-Digital Library personalization means sharing and combining user information across different DL systems so that a DL system may take advantage of data from others. This user information, however, is not easily transferable from one system to another. To achieve cross-Digital Library personalization, DL systems should take into account the precise nature of user profiles and proceed with a new theory for handling user model and context interoperability. Interoperability in terms of user modeling refers to the ability of DL systems to support compliant and interoperable user models that enable the propagation of user information across different DLs. The user modeling community focuses on ontology based approaches to achieve user model and context interoperability. As Carmagnola [5], van der Sluijs and Houben [25] proposed, a challenging approach to achieve user model interoperability is to use a sharable user model that contains the most used concepts within a domain and then to provide a mapping of the user data from one system to another. Typically, this sharable user model could be based on an ontology. By adopting the aforementioned approach, our proposed ontology could be used as a shared ontology among DL systems and mediation methods could be applied to achieve syntactic and semantic interoperability. These methods should be able to achieve syntactic and semantic mapping of user model attributes and user context dimensions from one DL to another. Finally, important is the use of reconciliation and/or concatenation rules for different values of user model attributes and context dimensions, like those proposed in the reference [26].

## 6 Conclusion and Future Work

In this paper we addressed the issue of user context and user model for Digital Libraries by providing definitions for these two important notions. Then, we distinguished context dimensions from user model attributes. These user model

attributes were divided into two categories: context-independent attributes and context-dependent attributes. Inspired by the use of ontologies as a tool for information sharing, we created an ontology for user representation in DLs. The important characteristic of this ontology is the use of properties that allow context-dependent user model attributes to be accompanied by context dimensions. In this way, along with preferences, interest, and other attributes the context present, when the previous features were captured, is retraced in order for the DL to provide better personalization services. Finally, the sharing of this ontology among different DL systems could be the starting point for accomplishing user interoperability. An additional step is needed, that is the use of mediation methods for achieving semantic and syntactic interoperability.

Currently, our work continues on several directions. We evaluate the possibility of amending and enhancing the proposed ontology with additional user model attributes and context dimensions. Furthermore, in order to achieve interoperability among DL systems we investigate the use of different methods for performing semantic mapping of user model attributes and context dimensions. Finally, we examine the applicability of various reconciliation rules for different values of user attributes that are captured in different DLs.

**Acknowledgments.** This work has been partially supported by the European Commission under FP7 Contract #231551 “DL.org: Digital Library Interoperability, Best Practices, and Modelling Foundations”.

## References

1. Athanasopoulos G., Candela L., Castelli D., Innocenti P., Ioannidis Y, Katifori V, Nika A., Vullo G., Ross S., The Digital Library Reference Model, D3.2a. DL.org Project Deliverable (2010)
2. Benerecetti M., Bouquet P., and Bonifacio M., Distributed Context-Aware Systems, Human-Computer Interaction, Special issue on Context-aware computing, pp. 213-228 (2001)
3. Brown P.J., Bovey J.D. Chen X., Context-Aware Applications: From the Laboratory to the Marketplace, IEEE Personal Communications, pp.58-64, (1997)
4. Brusilovsky P., Millán E., User Models for Adaptive Hypermedia and Adaptive Educational Systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) The Adaptive Web, LNCS, vol. 4321, pp. 3-53. Springer, Heidelberg, (2007)
5. Carmagnola F., Handling Semantic Heterogeneity in Interoperable Distributed User Models, Advances in Ubiquitous User Modelling, pp. 20-36 (2009)
6. Chen H., An Intelligent Broker Architecture for Pervasive Context-Aware Systems, PhD Thesis, University of Maryland, Baltimore County (2004)
7. Dey A. K., and Abowd G. D., Towards a better understanding of context and context-awareness, In Proc. of the Workshop on the What, Who, Where, When and How of Context-Awareness (2000)
8. Gentili G., Micarelli A., Sciarrone F., Infoweb: An Adaptive Information Filtering System for the Cultural Heritage Domain, Applied Artificial Intelligence 17(8-9), pp.715-744, (2003)

9. Golemati M., Katifori A., Vassilakis C., Lepouras G., Halatsis C. Creating an Ontology for the User Profile:Method and Applications, In Proc. Of the First RCIS Conference, Ouarzazate, Morocco, April 23-26 (2007)
10. Gross T., Specht M., Awareness in Context-Aware Information Systems, In Mensch&Computer – 1. Fachübergreifende Konferenz, Bad Honnef, Germany, Oberquelle, Oppermann and Krause (Eds.) Teubner, pp. 173–182 (2001)
11. Heckmann, D., Schwartz, T., Brandherm, B., Schmitz, M., and Wilamowitz-Moellendorff, M.: Gumo - the general user model ontology. In: Ardissono, L., Brna, P.,Mitrovi´c, A. (eds.) UM 2005. LNCS (LNAI), vol. 3538, pp. 428–432. Springer, Heidelberg, (2005)
12. Kobsa A., User Modelling: Recent work, prospects and hazards, Adaptive User Interfaces: Principles and Practices, Schneider-Hufschmidt, T. Khme, U. Malinowski, eds., (1993)
13. Lieberman H., Selker T., Out of context: Computer Systems that Adapt to, and Learn from, Context, IBM Systems Journal, pp. 617-632 (2000)
14. Lucas P., Mobile Devices and Mobile Data-Issues of Identity and Reference, Human-Computer Interaction, pp.323-336 (2001)
15. Moukas, A., Amalthaea: Information Discovery And Filtering Using A Multiagent Evolving Ecosystem, In: Applied Artificial Intelligence 11(5), pp.437-457 (1997)
16. Niederee, C.J., Stewart, A., Mehta, B., and Hemmje, M.: A multi-dimensional, unified user model for cross-system personalization. In: Proc. of Workshop on Environments for Personalized Information Access, in Advanced Visual Interfaces International Working Conference, Gallipoli, Italy, pp. 34–54 (2004)
17. Razmerita, L., Angehrn, A., and Maedche, A. Ontology-based user modeling for knowledge management systems. In 9th Int. Conf. on User Modeling, UM'03, pp. 213-217, Johnstown, PA, USA. Springer, LNCS 2702, (2003)
18. Rich E., Users are individuals: individualizing user models, International Journal of Man-machine Studies 18(3), 199-214, (1983)
19. Ryan N., Pascoe J., Morse D., Enhanced Reality Fieldwork: the Context-Aware Archaeological Assistant, Computer Applications in Archaeology, (1997)
20. Schilit B., and Theimer M., Disseminating active map information to mobile hosts, IEEE Network, 8(5): pp. 22-32, (1994)
21. Strang T. and Linnhoff-Popien C., A Context Modeling Survey, In the first International Workshop on Advanced context modeling, Reasoning and management, UbiComp (2004)
22. Strang T., Linnhoff-Popien C, and Frank K., CoOL: A Context Ontology Language to enable Contextual Interoperability. In LNCS 2893: Proceedings of 4th IFIP WG 6.1 International Conference on Distributed Applications and Interoperable Systems, pp. 236–247, Paris/France, November (2003)
23. Tazari M., Grimm M., Finke M., Modeling User Context, In Proceedings of the 10th International Conference on Human-Computer Interaction, Crete (Greece), June (2003)
24. Teevan J., Dumais S., and Horvitz E., Personalizing Search via Automated Analysis of Interests and Activities, Proceedings of SIGIR 2005, ACM Press, August (2005)
25. van der Sluijs K. and Houben G., A generic component for exchanging user models between web-based systems. International Journal of Continuing Engineering Education and Life-Long Learning, 16(1/2):64-76 (2006)
26. van der Sluijs, K. and Houben, G.J.: Towards a generic user model component. In: Proc. Of Workshop on Decentralized, Agent Based and Special Approaches to User Modelling, In International Conference on User Modelling, Edinburgh, Scotland, pp. 43–52 (2005)
27. VanLehn K., Student models, In Polson, M.C., Richardson, J.J. (eds.): Foundations of intelligent tutoring systems. Lawrence Erlbaum Associates, Hillsdale, pp.55-78 (1988)

# The gCube Interoperability Framework

Leonardo Candela[1], George Kakaletris[2], Pasquale Pagano[1],
Giorgos Papanikos[2], and Fabio Simeoni[3]

[1] Istituto di Scienza e Tecnologie dell'Informazione "Alessandro Faedo" – CNR, Pisa - Italy
{leonardo.candela | pasquale.pagano}@isti.cnr.it
[2] Computer Science Department, University of Athens – Athens, Greece
{g.kakaletris | g.papanikos}@di.uoa.gr
[3] Department of Computer and Information Sciences, University of Strathclyde, Glasgow, UK
fabio.simeoni@cis.strath.ac.uk

**Abstract.** Interoperability is one of the most challenging issues in the design and operation of large-scale computing infrastructures for multidisciplinary research. It requires solutions that can "embrace" heterogeneity, i.e. accommodate multiple incarnations of similar resources, as well as "hide" it, i.e. provide consumers with a homogeneous view of diverse resources. This paper overviews the measures put in place in the D4Science infrastructure to address a range of interoperability problems.

## 1 Introduction

*eScience* scenarios encompass research investigations that span multiple institutions and disciplines. This calls for innovative environments where scientists can seamlessly access data, software, and processing resources managed by diverse systems in separate administration domains. Dependently on context, these environments are commonly referred to as *Virtual Research Environments* [18,10], *collaboratories* [35,15], *digital libraries* [17,8], *e-Infrastructures* [5] and *cyberinfrastructure* [4]. A variety of systems fall within the scope of these definitions, from ad-hoc portals with minimal access services to content resources held in external repositories (low integration) to general-purpose management systems with advanced services defined over a wide range of resources (high integration). In some cases, motivations and design align with the principles of *grid computing* and its ecology of *virtual organisations* [16]. In all cases, the systems aggregate autonomous components and their ability to interoperate emerges as a core design requirement.

This paper describes the interoperability solutions that have been developed in the context of the D4Science-II EU Project [12], a continuation of the work carried out in the GÉANT, EGEE, DILIGENT and D4Science projects towards the deployment of networking, grid-based and data-centric e-Infrastructures. The distinguishing feature of D4Science-II is the aim to provide *on demand* Virtual Research Environments (VREs) for the creation and dissemination of scientific and technical knowledge. To this end, the project bridges a number of well-established e-Infrastructures from various domains, including high-energy physics, biodiversity, fishery and aquaculture resources management. The result is an infrastructure that exemplifies the concept of *e-Infrastructures ecosystem*.

The remainder of the paper is organised as follows. Section 2 outlines the scope of the project and overviews *gCube*, i.e. the software system that supports the operation of the D4Science-II infrastructure. Section 3 illustrates the project's perspective on interoperability and the solutions that have emerged from it. The goal of these solutions is to seamlessly

manage, and consume services and resources which are "external" to the infrastructure. Section 4 presents the current status of the work, its success stories, and its main achievements to date. Finally, Section 5 concludes the paper and reports on future activities.

## 2 D4Science-II and its Enabling Technology in a Nutshell

D4Science-II [12] is a project co-funded by the European Commission and started in October 2009. Its goal is to build a knowledge ecosystem as a set of interoperable data e-Infrastructures, repositories, and scientific communities (cf. Figure 1). The project builds on the results of other projects and initiatives, including: DILIGENT and D4Science, which have built an e-Infrastructure for *on demand* VREs [9,3]; GÉANT, EGEE, DRIVER, GENESI-DR, INSPIRE, AquaMaps and other, which have built e-Infrastructures for storing and processing collections of documents and data, including statistical, satellite and species distribution data. Figure 1 illustrates the pivotal role of the D4Science e-Infrastructure in the target ecosystem: as a *virtual aggregator* of resources that originate *from* other e-Infrastructures, and as a *provider* of aggregated resources *to* other e-Infrastructures. Resource aggregation and provision occur in the context of complex VREs for multidisciplinary scientific communities.

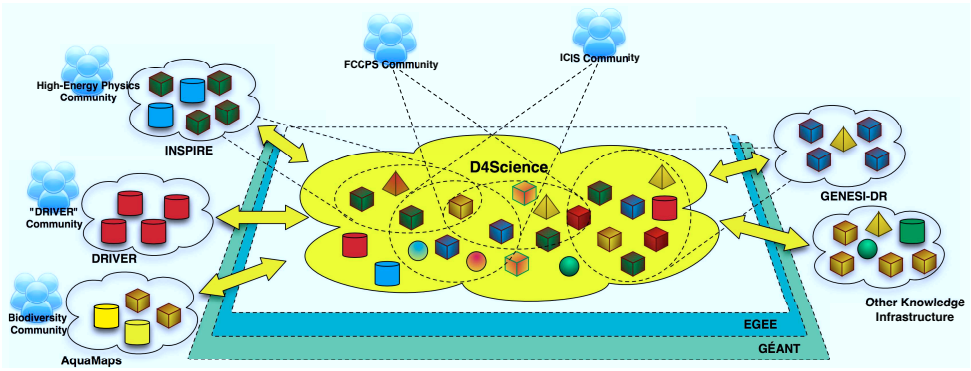


**Fig. 1.** The D4Science-II knowledge ecosystem

The ecosystem is enabled by an innovative software system, *gCube*[4]. gCube has been designed from the ground up to support the full lifecycle of modern scientific enquiry, with particular emphasis on application-level requirements of information and knowledge management. To this end, it interfaces pan-European Grid middleware for shared access to high-end computational and storage resources [2], but it complements this with a rich array of services that collate, describe, annotate, merge, transform, index, search, and present information for a variety of multidisciplinary and international communities. Services, data collections, and machines are infrastructural resources that communities select, share, and consume in the scope of collaborative VREs [10].

[4] `www.gcube-system.org`

gCube services are grouped in functional subsystems and are arranged in a Service Oriented Architecture. These include subsystems guaranteeing the operation of the infrastructure and its VREs; subsystems supporting the management (storage, organisation, description and annotation) of information represented via a rich and flexible model; subsystems guaranteeing information discovery; subsystems enabling the processing and manipulation of information via dedicated services (e.g. species distribution maps estimation) to produce new artifacts; subsystems providing a seamless access to the services and resources forming the infrastructure for human as well as programmatic consumption.

The architecture of the overall system adopts the following principles:

- the services are WSRF-compliant Web Services;
- the services discover each other dynamically, by mediation of the Information System;
- the services can be orchestrated by the system so as to execute workflows.

An application framework for developing gCube services and their clients has also been developed, the *gCore Framework* [25]. gCore allows gCube developers to abstract over functionalities that are offered at lower layers of the Web Services stack (WSRF, WS Notification, WS Addressing, etc.), and to make use of advanced features for the management of state, scope, events, security, configuration, fault, service lifetime, and publication and discovery.

In D4Science-II, this technology is consolidated and enhanced to address the interoperability requirements that emerge from the role of the D4Science infrastructure within the target ecosystem. The resulting services are described in the rest of the paper.

## 3 Interoperability Perception and Approaches in gCube

Despite the critical role given to interoperability in modern Information Systems, there is little theoretical guidance on how to address interoperability issues. There is in fact no definition of interoperability which is universally accepted [11]. The IEEE Glossary defines it as "*the ability of two or more systems or components to exchange information and to use the information that has been exchanged*" [19]. Two conditions must thus be met for a producer and a consumer to interoperate: (*i*) they must be able to exchange information; (*ii*) the consumer must be able to make effective use of the exchanged information, i.e. it perform the envisaged tasks by relying only on the exchanged information.

In the context of gCube, we define interoperability from a more general perspective, as *the ability of an arbitrary number of information systems to collaborate into achieving a common cause*. Typically, the common cause is the provision of functionality that is outside the scope of individual systems.

Here, we do not insist on consumer and producer roles and allow for different degrees of "autonomicity", from cases where interoperability is achieved in full automation to cases where human mediation is required. Most existing interoperability solutions fall within these extremes.

Analysing the problem in a top-down manner, the producer-consumer scheme is useful. Complex interoperability issues can be decomposed in fundamental concepts so as to allow for more fine-grained control and evaluation of interoperability solutions, whether these deal with message exchange protocols (manifestations, transports, etc.), message semantics, policies and so on.

As recognized by Paepcke et al. [24], over the years systems designers have developed different approaches and solutions to achieve interoperability. They have followed pragmatic approaches and started to implement solutions that blend into each other by combining various ways of dealing with the issues, including standards and mediators. Too often these solutions remain confined to the systems for which they have been designed. This leads in turn to "from-scratch" development scenarios and much duplication of effort when similar interoperability scenarios recur in different contexts.

In D4Science-II, we have adopted a two-step approach to effectively meet the challenges of developing interoperability solutions, while maintaining the maximum sustainability and visibility of the results. First, we have performed a per-case analysis in all the domains in which multiple infrastructures must interoperate, starting from the functional objective to then identify a set of individual actors that exchange messages. In the second step, we analyzed individual actors from the perspective of specifications, so as to conform to widely adopted ones, i.e. standards, rather than cope with message exchanges in an ad-hoc manner.

This approach has led to a number of interesting, widely visible and well-accepted results, which are enumerated later as success stories and adopted standards.

### 3.1 Resource Discovery

Resource discovery is a critical service in any distributed system. Its primary goal is to support the coordination of decentralised resources, using general-purpose, standard protocols as well as interfaces that fully characterise the available resources (*resource profile*) and that allow their identification against given requirements. In service-oriented infrastructures such as those enabled by gCube, the problem of resource discovery is complicated further by the plethora of available resource types, and by the fact that resources do change very frequently. Resource discovery raises also management issues, including workload balancing, performance monitoring, and problem diagnosis.

To support the role of the D4Science infrastructure in the knowledge ecosystem, the resource discovery facilities in gCube have been enhanced in two directions. From the modelling perspective, the set of supported resource types has been enlarged and the profiling capabilities have been improved. In addition to gCube resources, the new resource model supports other types of resources such as "external" data sources and "external" services. From the discovery perspective, the architecture of the information services has been revised so as to cope with the increased workload, i.e. the number of resource profiles and discovery requests. New strategies for dynamic data partitioning and replication have been conceived along with new facilities for asynchronous production and paged consumption of discovery results.

### 3.2 Data Access

Data access services deal with the provision of content available in a given scope. These services complement data discovery services (cf. Sec. 3.3), as in standard interaction patterns discovery is preliminary to access.

To cope with the scenarios envisaged in Section 2, a new content management system has been conceived. The *Open Content Management Architecture* (*OCMA*) acknowledges that gCube is concerned with content that may: (*i*) be hosted inside or outside a gCube infrastructure; (*ii*) be described with a variety of models, for different media, and with different degrees of structure; and (*iii*) be accessed with a variety of protocols. This service

provides its clients with seamless access to content that resides in multiple repositories and is designed to be open and easily adaptable to a variety of repositories. In terms of data model an OCMA service assumes that: (*i*) content is created, accessed, and distributed in units called *documents*; (*ii*) documents are grouped in *collections*; and (*iii*) collections are hosted in local management systems called *repositories*. As far as openness is concerned, OCMA services rely on plug-in based adaptivity facilities offered by gCube [28], i.e. specific plug-ins are developed to manage the interaction with external data providers.

### 3.3 Data Discovery

The gCube platform offers a rich set of Information Retrieval (IR) capabilities built on top of a stack of services that operate within the platform itself, taking advantage of the constructs, tools and methodologies that comprise it. While extending the interoperability of the gCube platform, however, Information Retrieval components have been enriched to accommodate data providers that live outside the gCube boundaries.

Traditionally, gCube Search has exploited external data providers via a mediator-based approach, i.e. the inclusion of custom operators in its search plans [27]. These operators are invoked to reply to an information discovery enquiry on such data sources. In essence, they can forward whichever query excerpt is needed to an external provider, retrieve the results, and integrate them with data hosted in the platform. These operators should (*i*) identify instances of the external engine, (*ii*) profile the external engine in the same way as internal gCube search providers are profiled and (*iii*) possibly wrap the retrieved content with the same constructs the internal services utilize for data representation and transport.

Although this approach can handle all cases, it involves the generation of custom operators for the adaptation of each new external engine type, thus forcing the system into an endless extension of the code base. The OpenSearch [23] specification is of great assistance to interoperability in the Information Retrieval domain. It finds its way in gCube as a new search operator implemented as a brokered OpenSearch client. The new facilities added include source discovery, template parsing, information retrieval and result paging and transformation in one servicing entity. The full range of OpenSearch criteria is exploited, including the geo-spatial / temporal criteria that are of major interest to the communities targeted by major data infrastructures.

Furthermore, the gCube Application Support Layer – the area where external access is granted to gCube facilities over common HTTP API, i.e. REST web services – has been extended to offer an OpenSearch provider interface that enables gCube's Information Retrieval capacities to be integrated into external search engines and OpenSearch consumers (e.g. web browsers).

### 3.4 Process Execution

One common need of data-oriented infrastructures is large data processing over owned or shared content. The gCube infrastructure goes beyond the limits of such a paradigm, however. Not only does it need and consume computational resources, it also manages and exposes resources offered by neighbouring Grid and Cloud infrastructures. Exploiting these resources and exposing them to other infrastructures is challenging from a functional point of view, and it requires addressing interoperability issues beyond a strictly systematic point of view.

By definition, a gCube infrastructure is comprised of a distributed set of nodes that can service computation under a wide range of technologies. These technologies can be Web

Services of several paradigms [14,32,6], technology-specific binaries, script executables, and they all raise a variety of requirements in terms of storage, deployed dependencies, communication and control, each representing permutations in terms of characteristics and behaviour. Coping with such a diverse environment calls for a component of largely unbounded expressiveness, which enforces even greater demands on the upper layers.

In order to facilitate the exploitation of these capacities whilst hiding the complexity of the landscape, at the core of the gCube platform lies a mechanism that is able to orchestrate flows of invocations on this large and diverse set of targets. This mechanism, named *Process Execution Engine* (*PE2ng*), builds on modern principles of data flow processing [31] appropriately expanded in the direction of interoperability. These include the plan (flow of execution), the operators (i.e. executable logic), the transport and control abstraction (implemented by the gCube Result Set – gRS), the containers (areas of execution), the *state* holders (e.g. storage), the *resource profiles* (definitions of resources characteristics for exploitation in a plan).

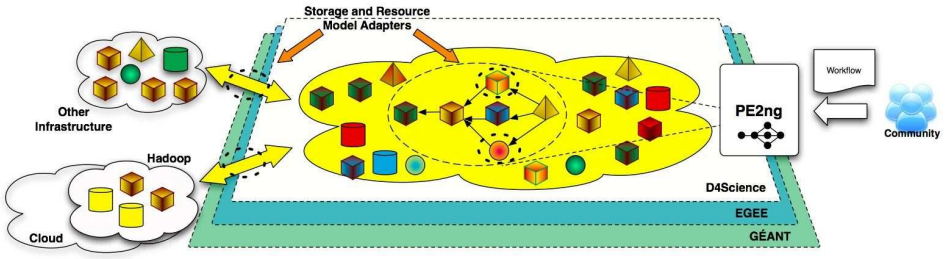


**Fig. 2.** PE2ng Architecture

PE2ng is suitable for modern Cloud computing infrastructures and it is capable of near-optimal exploitation of standalone and distributed computing resources, as it can employ selection and communication strategies transparently to its operators.

The PE2ng is not only able to execute flows consisting of executables of several technologies but it can operate as a gateway to other computational and storage infrastructures. The overall goal is to bridge gCube with heterogeneous platforms such as the underlying EGEE gLite grid [13], the Condor [29], the Hadoop [33] and more. By employing an architecture (cf. Fig. 2) that involves a number of abstractions (infrastructure adapters and storage providers) and a number of standards such as JSDL [1] for the specifications of the operations, PE2ng can serve effectively the aforementioned cause.

The *gCube Data Transformation Services* (gDTS) offer functionality to transform content and metadata into different formats and specifications. This functionality is essential to enable interoperability among Digital Libraries, as a number of facilities offered by Digital Libraries actually rely on the manifestations of the data they contain.

gDTS operates on top of the PE2ng, constructing on demand flows that are capable to transform content among two manifestations. Based on MIME-Type specifications that include “micro-format” refinements (e.g. qualities, sizes, encodings), gDTS is capable to locate paths that manage to migrate one content type to another and consequently employs PE2ng for carrying out the operation over the infrastructure.

A common use case of gDTS is its involvement in the preparation of content to be imported or exported to the Data Access and Retrieval mechanisms of gCube. The engine can also be exploited to apply transformations to objects of external infrastructures. In that case, the input data will not be imported from gCube services. Rather, they are supplied by external sources and made accessible over a set of common standards (e.g. ftp, http, grid ftp, a file system mount point). Depending on the use case the transformed content (and its metadata if such is the use) can be stored within the gCube infrastructure or sent back to the clients.

## 4 Evaluation

The new features of the gCube system, which are currently under development, are producing quite encouraging results, and success stories are growing by the day.

The new architecture for content management proved to fit the scope. An OAI-PMH [20] plug in has been developed and various OAI-PMH based data providers, including DRIVER, have been easily managed. In addition, the same protocol is used to further expose content previously aggregated in the D4Science infrastructure.

*Information Retrieval interoperability* via OpenSearch protocol is exploited in a number of cases. In the DRIVER / OpenAIRE [22] interoperability case, the external infrastructure IR services are exploited via basic OpenSearch compliant invocations. However, the GENESI-DR infrastructure[5] is more complex and a brokered approach is taken, leading initially to the discovery of data providers and subsequently to the results themselves. Finally, the gCube Information Retrieval is in itself exploited via OpenSearch and the reverse interoperability cases are also supported.

*Data Transformation* has allowed gCube to interoperate with numerous external sources provided by the communities of the aforementioned infrastructures. After being "imported", the content is consumed by the gDTS, which takes care of transforming it into homogeneous formats that populate additional collections of the infrastructure (e.g. indexing). A more complex case is the provision on-demand of alternative representations of content that resides in external e-Infrastructures (e.g. thumbnails generation). In this case, large external datasets are temporarily "fed" to the D4Science e-Infrastructure and consumed by the gDTS service. The transformed results are then returned to the original e-Infrastructure.

*Process Execution capacities* are exploited in several scenarios, the most interesting of which from the interoperability is the INSPIRE - gCube scenario. Here, processing *Optical Character Recognition* (*OCR*) over large data volumes can be outsourced to the D4Science infrastructure, where it is made efficient by the underlying grid infrastructure. On the other side, daily reconstruction of inverted indices over an extended database requires the resource allocation strategies of the Cloud and constructs of the kind offered by a Map-Reduce Hadoop infrastructure. Somewhere in the middle, one could position the *Author Identification* case, which is less demanding in terms of data but requires parallel computation. This kind of process can be fruitfully executed on gCube worker nodes. In the AquaMaps scenario, model-based, large-scale predictions of marine species occurrences have to be generated. Predictions are generated by matching habitat usage of species, termed environmental envelopes, against local environmental conditions to determine the relative suitability of specific geographic areas for a given species. Data and algorithms are scattered along a

[5] `www.genesi-dr.eu`

number of gCube worker nodes to build up information-rich data sets. These are then further processed to construct biodiversity maps using the underlying Grid resources.

The major success of the Interoperability approach of gCube is the adoption of standards rather than proprietary protocols. As a result of this work a number of specifications are today exploited in a number of services and access points to the gCube platform, including OAI-PMH for data access, OpenSearch for data discovery, JDL for job submission, FTP / GridFTP / HTTP for storage access.

## 5 Conclusions and Future Work

Interoperability is one of the most critical issues that arise when building systems as "collections" of independently developed systems that should cooperate and rely on each other to accomplish larger tasks. An e-Infrastructure Ecosystem falls in this category of challenging systems and its realisation demands for the development of a rich array of interoperability approaches. In this paper, we have described the approaches put in place in the context of the D4Science-II. In particular, we have presented our interpretation of interoperability and the solutions put in place for resource discovery, data access, data discovery and process execution.

Current interoperability solutions in gCube are driven by the requirements and opportunities raised by the infrastructures addressed by the D4Science project, once these are seen under the perspective of generalization and standards adoption. We expect that this approach will maximise the usefulness of gCube in similar e-Infrastructure ecosystems. This assumption will be tested in other domains, for other e-Infrastructures and data providers have been already selected to enrich the D4Science knowledge ecosystem (e.g. 4D4Life[6], BioFresh[7]). The current solutions will be exploited to address the requirements that arise in these additional interoperability cases. The mechanisms for exposing gCube resources will be reinforced by supporting other standards and protocols. Mechanisms like the Linked Data [7] approach, the OAI-ORE [21] protocol and the SRU discovery protocol [30] are under investigation.

**Acknowledgments** The work reported has been partially supported by the D4Science project (FP7 of the European Commission, INFRA-2007-1.2.2, Contract No. 212488), the D4Science-II project (FP7 of the European Commission, INFRA-2008-1.2.2, Contract No. 239019) and the DL.org Coordination and Support Action (FP7 of the European Commission, ICT-2007.4.3, Contract No. 231551).

## References

1. A. Anjomshoaa, F. Brisard, M. Drescher, D. Fellows, A. Ly, S. McGough, D. Pulsipher, and A. Savva. Job Submission Description Language (JSDL) Specification, Version 1.0. Technical Report GFD-R.056, Open Grid Forum, 2005.
2. O. Appleton, B. Jones, D. Kranzmuller, and E. Laure. The EGEE-II project: Evolution towards a permanent european grid inititative. 16:424–435, Mar. 2008.

[6] www.4d4life.eu
[7] http://www.freshwaterbiodiversity.eu/

3. M. Assante, L. Candela, D. Castelli, L. Frosini, L. Lelii, P. Manghi, A. Manzi, P. Pagano, and M. Simi. An Extensible Virtual Digital Libraries Generator. In B. Christensen-Dalsgaard, D. Castelli, B. A. Jurik, and J. Lippincott, editors, *12th European Conference on Research and Advanced Technology for Digital Libraries, ECDL 2008, Aarhus, Denmark, September 14-19*, volume 5173 of *Lecture Notes in Computer Science*, pages 122–134. Springer, 2008.
4. D. E. Atkins, K. K. Droegemeier, S. I. Feldman, H. García-Molina, M. L. Klein, D. G. Messerschmitt, P. Messina, J. P. Ostriker, and M. H. Wright. Revolutionizing science and engineering through cyberinfrastructure. Report of the National Science Foundation Blue-Ribbon Advisory Panel on Cyberinfrastructure, January 2003.
5. M. Atkinson, J. Crowcroft, C. Goble, J. Gurd, T. Rodden, N. Shadbolt, M. Sloman, I. Sommerville, and T. Storey. Computer Challenges to emerge from eScience. e-Science vision document., August 2002.
6. T. Banks. Web Services Resource Framework (WSRF) - Primer. Committee draft 01, OASIS, December 2005. `http://docs.oasis-open.org/wsrf/wsrf-primer-1.2-primer-cd-01.pdf`.
7. C. Bizer, T. Heath, and T. Berners-Lee. Linked Data - The Story So Far. *International Journal on Semantic Web and Information Systems*, 5(3):1–22, 2009.
8. L. Candela. *Virtual Digital Libraries*. PhD thesis, Information Engineering Department, University of Pisa, 2006.
9. L. Candela, F. Akal, H. Avancini, D. Castelli, L. Fusco, V. Guidetti, C. Langguth, A. Manzi, P. Pagano, H. Schuldt, M. Simi, M. Springmann, and L. Voicu. DILIGENT: integrating Digital Library and Grid Technologies for a new Earth Observation Research Infrastructure. *International Journal on Digital Libraries*, 7(1-2):59–80, October 2007.
10. L. Candela, D. Castelli, and P. Pagano. On-demand Virtual Research Environments and the Changing Roles of Librarians. *Library Hi Tech*, 27(2):239–251, 2009.
11. L. Candela, D. Castelli, and C. Thanos. Making Digital Library Content Interoperable. In *Sixth Italian Research Conference on Digital Libraries (IRCDL 2010)*, 2010.
12. D. Castelli. D4Science-II - An e-Infrastructure Ecosystem for Science. *ERCIM News*, 79:9, October 2009.
13. EGEE. gLite: Lightweight Middleware for Grid Computing. `http://glite.web.cern.ch/glite/`.
14. R. Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. Phd dissertation, University of California, Irvine, 2000.
15. T. A. Finholt. Collaboratories. In *Annual Review of Information Science and Technology*, volume 36, pages 73–107, 2005.
16. I. Foster and C. Kesselman. *The Grid: Blueprint for a Future Computing Infrastructure*. Morgan-Kaufmann, 2004.
17. E. A. Fox, R. M. Akscyn, R. Furuta, and J. J. Leggett. Digital Libraries. *Communications of the ACM*, 38(4):23–28, April 1995.
18. M. Fraser. Virtual Research Environments: Overview and Activity. *Ariadne*, 44, 2005.
19. A. Geraci. *IEEE Standard Computer Dictionary: Compilation of IEEE Standard Computer Glossaries*. IEEE Press, 1991.
20. C. Lagoze and H. Van de Sompel. The open archives initiative: building a low-barrier interoperability framework. In *Proceedings of the first ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 54–62. ACM Press, 2001.
21. C. Lagoze, H. Van de Sompel, P. Johnston, M. Nelson, R. Sanderson, and S. Warner. Open Archives Initiative Object Reuse and Exchange (OAI-ORE) User Guide - Primer. Technical report, Open Archives Initiative, 2008.
22. P. Manghi, M. Mikulicic, L. Candela, D. Castelli, and P. Pagano. Realizing and Maintaining Aggregative Digital Library Systems: D-NET Software Toolkit and OAIster System. *D-Lib Magazine*, 16(3/4), March/April 2010.
23. OpenSearch Community. OpenSearch. Project website.
24. A. Paepcke, C.-C. K. Chang, T. Winograd, and H. García-Molina. Interoperability for Digital Libraries Worldwide. *Communications of the ACM*, 41(4):33–42, 1998.

25. P. Pagano, F. Simeoni, M. Simi, and L. Candela. Taming development complexity in service-oriented e-Infrastructures: the gCore application framework and distribution for gCube. *Zero-In e-Infrastructure News Magazine*, 1(1):19 – 21, 2009.
26. E. Rahm and P. A. Bernstein. A survey of approaches to automatic schema matching. *The VLDB Journal*, 10(4):334–350, 2001.
27. F. Simeoni, L. Candela, G. Kakaletris, M. Sibeko, P. Pagano, G. Papanikos, P. Polydoras, Y. E. Ioannidis, D. Aarvaag, and F. Crestani. A Grid-Based Infrastructure for Distributed Retrieval. In L. Kovács, N. Fuhr, and C. Meghini, editors, *Research and Advanced Technology for Digital Libraries, 11th European Conference, ECDL 2007, Budapest, Hungary, September 16-21, 2007, Proceedings*, volume 4675 of *Lecture Notes in Computer Science*, pages 161–173. Springer-Verlag, 2007.
28. F. Simeoni, L. Candela, D. Lievens, P. Pagano, and M. Simi. Functional adaptivity for Digital Library Services in e-Infrastructures: the gCube Approach. In M. Agosti, J. Borbinha, S. Kapidakis, C. Papatheodorou, and G. Tsakonas, editors, *13th European Conference on Research and Advanced Technology for Digital Libraries, ECDL 2009*, 2009.
29. D. Thain, T. Tannenbaum, and M. Livny. Distributed computing in practice: the condor experience. *Concurrency - Practice and Experience*, 17(2-4):323–356, 2005.
30. The Library of Congress. Sru: Search/retrieval via url. Website.
31. M. Tsangaris, G. Kakaletris, H. Kllapi, G. Papanikos, F. Pentaris, P. Polydoras, E. Sitaridi, V. Stoumpos, and Y. Ioannidis. Dataflow Processing and Optimization on Grid and Cloud Infrastructures. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 32(1):67–74, March 2009.
32. W3C. Simple Object Access Protocol (SOAP).
33. T. White. *Hadoop: The Definitive Guide*. O'Reilly Media, 2009.
34. G. Wiederhold. Mediators in the Architecture of Future Information Systems. *Computer*, 25(3):38–49, 1992.
35. W. A. Wulf. The collaboratory opportunity. *Science*, 261:854–855, August 1993.

# Handling Repository-Related Interoperability Issues: the SONEX Workgroup

Peter Burnhill[1], Pablo de Castro[2] *, Jim Downing[3], Richard Jones[4], and Mogens Sandfær[5]

[1] EDINA National Data Centre, University of Edinburgh, Edinburgh, UK
[2] Carlos III University Madrid, Leganés (Madrid), Spain
[3] University of Cambridge, Cambridge, UK
[4] Symplectic Ltd, London, UK
[5] Technical University of Denmark (DTU), Copenhagen, Denmark
http://sonexworkgroup.blogspot.com/

**Abstract.** Sharing of scholarly content through a network of Open Access repositories is becoming commonplace but there is still need for systematic attention into ways to increase the rate of deposit into, and transfer of content across, the OA repository space. This is a report of the work of a small international group, supported by JISC, with remit to describe, analyse and make recommendations on deposit opportunities and use cases that might provide a framework for project activity geared to the ingest of research papers and other scholarly works. The multi-authored, multi-institutional work is put forward as the default, and nine use case actors are listed, as deposit agents, with four main use case scenarios. There is also some comment and pointers to projects in Europe which address some of these use case scenarios.

**Keywords:** repositories, interoperability, deposit, research output, CRIS, metadata

## 1 Introduction

The SONEX Group [1] has its origins in a workshop held in Amsterdam in March 2009 that was held to "identify essential components of international repository infrastructure". SONEX is an acronym for Scholarly Output Notification and Exchange, and reflects a focus on interoperability between repositories of all sorts and across many different countries. SONEX has remit to analyze opportunities for deposit of new content into repositories, as well as ways of assisting transfer of content across repositories. The graphic in figure 1 illustrates the variety of repository into which authors deposit/issue their content.

Our purpose is to share comment and recommendations about the implied use case actors (stakeholders) and hence some use case scenarios that might be

* Correspondence: Pablo de Castro, Servicio de Recursos Electrónicos, Universidad Carlos III de Madrid, Avda. de la Universidad 30; E-28911 Leganés (Madrid), Spain, tel: +34 916249081, e-mail: pcastro@db.uc3m.es

addressed and tested by other initiatives or projects in their implementation of technology and solutions. We report on work of others whom we know are doing just that.

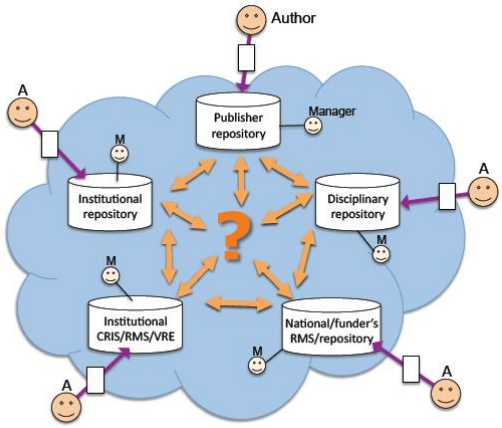


**Fig. 1.** Interoperability needs within repository workspace

By publishing our analysis and recommendations, via blog, presentations at meetings and through this article, our intention is to assist project managers and developers in their investigation and implementation of enabling (repository) facilities, as well as the Joint Information Systems Committee (JISC) and related funding agencies in implementation of their strategic objectives.

## 2 Origin and objectives

The International Repositories Workshop in Amsterdam on March 19-20th, 2009 was organised by JISC, the SURF Foundation and the DRIVER Project [2]. The workshop brought together representatives from many of the main projects and initiatives involved worldwide in developing repositories and repository-related services. A series of requirements for building repository infrastructure had been grouped into four main strands to be discussed within seminar groups. There were 'brainstorming' sessions on repository interoperability, citation services, identifiers for authors and institutions, and international repository organisation. Action Plans for workgroups were subsequently established following each of those four sessions and published on a wiki [3].

There were representatives from Denmark, India, Ireland, Israel, Japan, Netherlands, Spain, UK, USA and beyond among the twenty people who took part in the 'Repository Handshake' (RH) session. The consensus reached was the need

to work to a strategy that addressed the low level of content, such as research papers, in most Institutional Repositories (IRs), rather than have detailed technical debate on protocols, although there was general support for using the SWORD (Simple Web-service Offering Repository Deposit) protocol [4]. Key to success in populating repositories was an understanding of the full range of possible opportunities for the deposit of content into repositories, and to characterise what needed to be done at these 'deposit opportunities', by individuals and by machine-as-user, including automation of content ingest.

Some of the basic ideas upon which the SONEX group would later build its analysis are summarized below:

- A focus on automatic mechanism for populating IRs with researchers' scholarly output: PUT (rather than KEEP and GET) functionality through machine-to-machine (m2m) interoperability, whenever possible.
- Consensus is needed on what sufficiently-good-metadata is required at ingest and for onward transfer of object+metadata into repositories; attention should be paid to leveraging existing accurate metadata.
- The technological means for interoperability largely existed: the SWORD protocol was the accepted means to support negotiation between depositing agents and target repositories.
- Dissatisfaction that projects and IRs had their focus on single-author, single-institution research papers; regret that IRs and subject repositories were often viewed as competing alternatives.
- Recognition that CRIS (Current Research Information Systems) and desktop software were sources of content but, as with bibliographic authority files, were presently disconnected from repositories.

The Action Plan [5] drawn up at the end of the workshop sessions called for real-life use cases as exemplars: their analysis on what was common or specific to each; a gap analysis of tools and mechanisms needed; outreach to willing and able partners to test one or two preferred use cases. JISC decided to convene a small group to work on this Action Plan. Intent on retaining the international character of the workshop session in the group, the four individuals[1] invited came from the Danish Technical University (DTU), the Spanish National Research Council (CSIC), and the Universities of Cambridge and Edinburgh (EDINA) [6]. No specific timeline was set for the Action Plan, other than recommending an initial and brief gap analysis in order to prevent risk of conclusions becoming rapidly outdated.

## 3 SONEX analysis

The main objective of the 'Repository Handshake' workgroup after the Amsterdam workshop was to meet and agree upon a conceptual model and vocabulary.

[1] As of Nov'09 the workgroup coordinator changed affiliation from CSIC to Carlos III University Madrid, and a new member with affiliation Symplectic Ltd became part of the group in Apr'10 [7].

We noted that the initial focus in the workshop session on journal articles was fine but concluded that models and infrastructure enabling repositories should be tested for the deposit and sharing of research data and learning materials (and perhaps also open software geared for academic purpose). The acronym SONEX was chosen to reflect that: *Scholarly Output Notification and Exchange*.

We were free from obligation to carry out project work ourselves, although we could prompt and facilitate others to do project activity. Sharing the burden of travel, the SONEX Group held their meetings since 2009 in Cambridge, Copenhagen, Edinburgh, Geneva and Madrid, taking economic advantage of being able to tag onto other repository meetings.

Having defined our problem space as 'metadata+digital object deposit, notification and transfer', we set about examination of system verbs/operations, followed by some further study on the possibilities of communication between repositories. We then reviewed the initial RH use-case scenarios. There were a growing number of projects, funded by JISC or other partners, focusing on populating repositories with some overlap of scope, see table 1 below.

**Table 1.** Some repository-related projects running at the start of SONEX

| *Project* | *Institution* |
|---|---|
| SWORD | UKOLN-JISC |
| Open Access Repository Junction (OA-RJ) | EDINA-JISC |
| PEER Project | STM, ESF, Göttingen Univ. |
| CRIS/OAR Interoperability Project | DTU-KE |
| JournalTOCsAPI project | Heriot-Watt University-JISC |
| The Deposit Plait | Aberystwyth University-JISC |
| EM-Loader | EDINA-JISC |
| EIDeR | King's College London-JISC |

An updated version of this listing is available on the SONEX Blog [8], from which the URLs may be actioned.

## 4 Identifying use-cases from deposit opportunities

The significance of the author is evident in figure 1 above in which one or more authors (A) deposit into an environment of repositories. This array of different kinds of interacting repository systems includes institutional (IRs) and subject-specific and disciplinary repositories (SRs). The managers of each different types of repository have potential interest in any given submission; a given author might be required to make multiple submissions for the same work.
Repositories managed as part of research information systems (CRIS) are alternative sources for ingest or transfer of content into institutional and subject repositories. The inclusion of publishers in the picture is recognition of their significance for peer-reviewed journal articles and for authors, and both necessary

for some types of research output, and also special with respect to interests and rights for related versions of a work, typically the author's final copy and the publisher's version.

**Table 2.** Deposit agents (SONEX use-case scenarios)

| |
|---|
| *Default content: multi-authored, multi-institutional work (*eg *journal article)* |
| **Use-case Actor 1**: Individual (co-)author/researcher<br>*Variant 1a*: where the author making the deposit is the PI of funded research project (need for compliance with mandate from funder to deposit)<br>*Variant 1b*: the author making the deposit is not the PI of funded research project but the work being deposited is associated with one or more funded research projects (and one or more PIs) |
| **Use-case Actor 2**: Depositor (not author) - Delegated deposit<br>*Variant 2a*: Mediated by an actor directly reporting to the author<br>*Variant 2b*: Mediated by another institutional agent - *eg* library. |
| **Use case Actor 3**: Institutional/Departmental Research Support Systems (CRIS/RMS/VRE systems) |
| **Use case Actor 4**: Publisher<br>a) OA deposit of the author's final copy<br>b) Supply of authoritative metadata and identifiers (DOIs and pointer to published copy) |
| **Use case Actor 5**: Funding bodies/Policy bodies |
| **Use case Actor 6**: Repository Manager (RM) of an IR, with co-authored work wishing to notify RM(s) of the other IR(s); RM of subject (SR) wishing to do similarly; RMs of IRs wishing to know of and obtain copy of work by author (now) at institution. |
| **Use case Actor 7**: Developer/vendor of authoring software |
| **Use case Actor 8**: Repository software developer/vendor |
| **Use case Actor 9**: Libraries<br>Potential depositors of their own resources and document collections |

From the start we wished to promote the SONEX default of the 'multi-authored, multi-institution' work. We would urge projects and systems not to plan on the basis that research papers are generally from a single author, nor that a given research paper is relevant to only a single institution. By default any deposit might be of interest to more than one institution. Whatever a repository

manager holds is potentially of interest to another - researchers and authors move, and for assessment purposes past publications count. One the other side of the coin, there is prospect of multiple issue of the same work by its several co-authors.

All this provides real motivation for exchange of information, and content, across repository space: across institutional and national boundaries as well as across repositories of different types.

Our task was to complete the list of actor-based use-cases produced during the Repository Handshake workshop sessions. The most important persons in discussions about Open Access, and also for plans of repository infrastructure and enabling projects, are the researchers/authors and the readers: both wish for prompt and ready release or the author's work, for attention by her/his peers and the world beyond the academy. It is for all intermediaries to assist that goal. The extended list, based on the deposit opportunities illustrated in figure 1, is given above in table 2.

Reflection on these use cases helped identify four principal use case scenarios for further analysis. Use case scenarios would also serve as basis on which to review which projects could test them through implementation:

I **CRIS/RMS/VRE systems** (at institution or national levels) [use case actor 3] where transfer of objects plus agreed metadata into all relevant IRs should be automated. [Associated Projects: Trinity College Dublin (TCD), Technical University of Denmark (DTU)]

II **Publisher** [use case actor 4] as publisher wishes to deliver service to author, by depositing the (co-)author's final copy in appropriate IR, complete with DOI and pointer to the published version. [Associated Projects: Nature Publishing Group (NPG) + Repository Junction (OA-RJ) Project; many publishers in PEER Project]

III **Funder-mandated deposit** [use case actor 5] wishes assurance of compliance that output from funded projects is deposited under OA with relevant SR and/or IRs. [May also involve use case actors 1, 2 and 3] [Associated Projects: Repository Junction (OA-RJ) Project with UKPMC].

IV **Deposit via personal software**, by researcher/co-author [use case actor 1] e.g. desktop software [use case actor 7] or bibliography (web or desktop).

## 5 Use case scenario I: CRIS systems as source of content

An increasing number of universities in Europe are investing in some kind of Current Research Information System (CRIS) in order to support research groups and to keep track of their research output. These systems can hold information

on a variety of research-related activity such as projects, grants, PhDs, patents, often relating to obligation to funders and (national) systems of research assessment [9]. They include metadata on research publication but rarely the full text of the published work.

There was first-hand knowledge of the *CRIS/OAR Interoperability Project* within the SONEX Group, developed by Knowledge Exchange and the Danish Technical University (DTU). This project had the objective to "increase the practical interoperability between Current Research Information Systems (CRIS) and Open Access Repository (OAR) systems by defining and proposing a metadata exchange format for publication information with an associated common vocabulary". A CRIS/OAR interoperability workshop was held last June at the CRIS2010 conference in Aalborg, Denmark, in which the KE group presented its results [10]. Several CERIF-based initiatives, both institutional and national, were also presented on metadata exchange and CRIS/IR integration [11]. The high number of projects and a growing commercial engagement in the area indicates that the CRIS/IR interoperability use-case scenario is being successfully addressed.

## 6 Use case scenario II: Publishers as use communities

Publishers are the recipients of the author's final copy, prior to the publication of the publisher's version, a principal source for deposit of that copy into the institutional repository. Nominally, there may appear to be a conflict of business with publishers, which are seen as wishing to have paid-for attention to the copy that they publish. However, publishers rely on the free supply of content from authors, and wish to do service to authors by exploring ways in which they can assist authors comply with mandates from funders that the author's final copy should be deposited in Open Access repositories. They can act as an appointed proxy for the co-authors of an article. Several projects exist, two of which are reported here: the EU-funded PEER Project and JISC-funded Open Access Repository Junction (OA-RJ) Project.

The **PEER Project** (Publishing and the Ecology of European Research) [12], began in October 2008 with support from the European Commission eContent+ Programme. PEER aims to investigate the effects of the large-scale, systematic deposit of authors' final peer-reviewed manuscripts (also known as postprint versions) upon reader access, author visibility, and journal viability, as well as on the broader ecology of European research. The project is a collaboration between publishers in the International Association of Scientific, Technical and Medical Publishers (STM) and a number of repositories belonging to partner academic and research institutions (Göttingen State and University Library, the Max Planck Society, University of Bielefeld, INRIA, Kaunas University of Technology, University Library of Debrecen and Koninklijke Bibliotheek Amsterdam). In carrying out the publication transfer, PEER intends to unify the ingestion services based either on format used or protocols such as OAI-PMH or SWORD.

The PEER workflow is represented in figure 2 below.

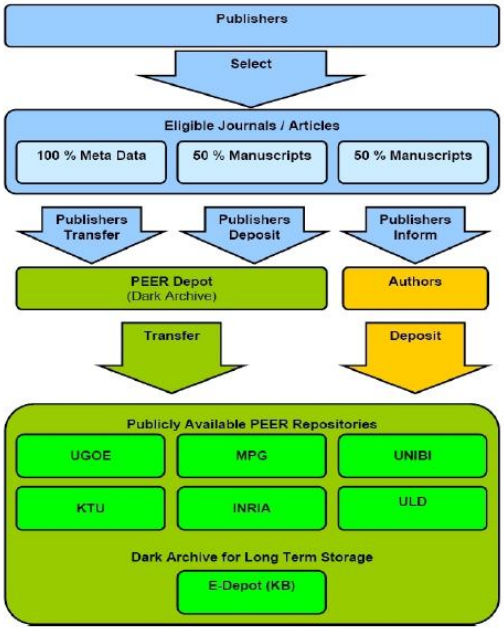


**Fig. 2.** PEER workflow for journal articles, from publisher to OA repository

The JISC-supported **Open Access Repository Junction** (OA-RJ) Project, which started in August 2009 has a generic approach to automatic ingest into a broker which then manages notification and transfer of metadata, with or without content, to target repositories. This generic deposit broker, which deploys SWORD and makes use of OpenDOAR and ROAR, is designed to work for all types of content and in one strand is being applied to research papers in work with Nature Publishing Group and a group of test target repositories as part of a test implementation for use case scenario II. By interacting with the business needs of the publisher, an understanding and a mechanism has been achieved for a two-stage workflow, as shown in figure 3 below.

This shows a two-stage engagement which is sympathetic to the opportunity that publishers have to enhance the author's own copy with additional metadata derived from the publisher's own version (e.g. DOI and URL to the publisher's copy), and to do so with bibliographic accuracy. The intention is to generalise this so that any journal publisher might initiate deposit of the co-authors' final copy into each of the respective institutional repositories. More about this is reported on the OA-RJ Blog [13].

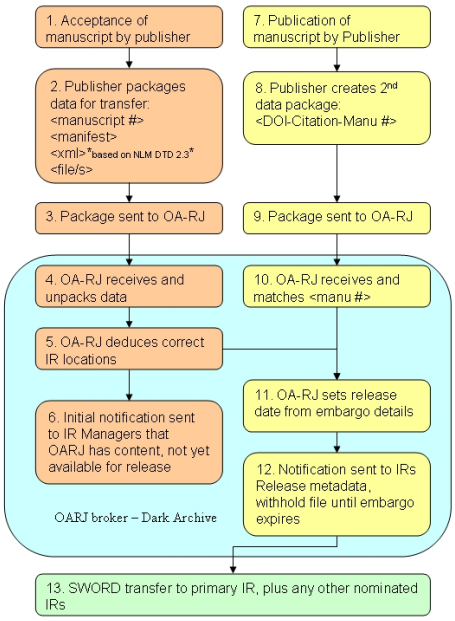


**Fig. 3.** OA-RJ workflow for journal articles, from publisher to OA repository

We have also noted other more recent initiatives, such as the BioMed Central partnership with MIT Libraries to deposit open access articles using Sword [14], via a paid for service. These follow similar deposit patterns and workflow for research output transfer into repositories, so some common guidelines for publisher-driven deposit arising from PEER and OA-RJ could be very useful to the repository community. This is one of the lines for SONEX current work.

## 7 Use case scenario III: Funders and subject repositories as use communities

As research councils and other research funders turn to mandates for the issue of research work into Open Access repositories, so there is need for facility such as that being developed as the OA-RJ broker[2]. This is being tested by OA-RJ with UKPubMed Central, which is a subject repository supported by multiple funders in the bio-medical area. This will engage with both the repository manager for a subject repository, use case actor 6, but also the funder, use case actor 5, together with routing of notification to repository managers of the respective IRs.

[2] An earlier version of the OA-RJ broker is used for the Depot [15], which helps authors self-deposit by re-directing the author to her/his institutional repository; it will be used to support OpenDepot.org [16] as a global facility for both affiliated and non-affiliated researchers who wish to release their research work as Open Access.

The OA-RJ broker architecture lends itself very well to networking and SONEX is investigating how this may be tested in practice. Scalability seems well served by a distributed approach (see figure 4 below) where national/regional brokers on the one hand are responsible for a select group of publishers/subject repositories and on the other hand for their national/regional network of IRs. Whenever such a networked OA-RJ broker receives information on publications relevant to IRs of another nation/region it transfers the information to the respective broker. Thus the networked OA-RJ brokers may share the work of establishing relationships and technical interfaces with the world's publishers and subject repositories - and these will only have to deal with one broker to the world of institutional repositories. Likewise each broker only needs to maintain detailed information on and relations with its own national or regional group of IRs.

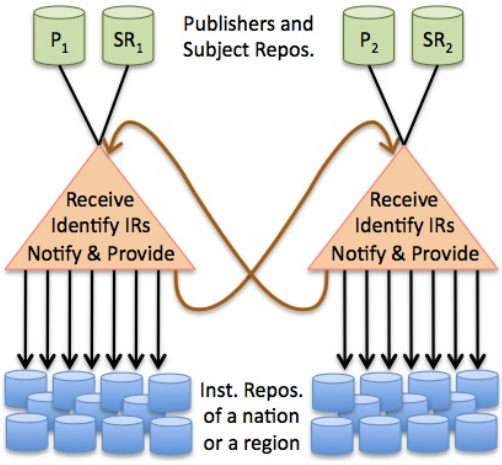


**Fig. 4.** An array of networked national/regional brokers

## 8 Looking to the future: JISC deposit call

We have so far presented much of what we set out to do as deposit use-cases review of project activity. We now look to outreach and to helping create an international space for exchange of information about interoperability issues with regard to repositories.

Strand A of a recent funding call by JISC (Funding Call 2/10, March 2010) had the specific objective of "ensuring take-up of solutions that enable and encourage author deposit of Open Access research outputs into repositories by embedding deposit into research or related practice" [17]. The outcome of this

funding call represents an opportunity for the SONEX Group to assist the selected projects by (i) providing a use case scenario framework and (ii) providing webspace/poster-space for deposit initiatives[3].

Three projects [18] were announced in July 2010 at the Open Repositories Conference held in Madrid, and representatives of each attended a SONEX Deposit meeting along OR10 to give a briefing and examine how they relate to the SONEX use case scenarios.

- **DepositMO: Modus Operandi for Repository Deposits** [19]. Led by the University of Southampton, in close liaison with Microsoft and the University of Edinburgh, the DepositMO project aims to create a workflow connecting the user's computer desktop: part of use case scenario IV. This involves interoperability between MS Office and the EPrints and DSpace repository software [use case actor 7]. The DepositMO project further tests the Sonex use-case analysis with content such as datasets and software, as well as a real-life exemplar for software vendor-driven deposit.

- **RePosit: positing a new kind of deposit** [20]. The RePosit Project seeks to increase uptake of a web-based repository deposit tool embedded in a researcher-facing publications management system. Led by the University of Leeds, it involves several additional universities and Symplectic Ltd as commercial partner. This project further tests SONEX use case scenario I, with a variety of starting points and institutional strategies for research information system.

- **DURA: Direct User Repository Access** [21]. The DURA project, lead by the University of Cambridge with Mendeley Ltd and Symplectic Ltd as consultant firms, aims to embed institutional deposit into the academic workflow by using Mendeley and Symplectic tools to allow researchers to synchronise their personal research collections with institutional systems. This also addresses SONEX use case scenario IV, which had previously been represented as *Our Bibliography tools* as presented on the Repositories Infrastructure wiki [22].

It is time now to re-visit and update our use case framework in the light of these new projects and with more than just research papers in mind. For this we need feedback and criticism, and would welcome both.

## Acknowledgements

Members of the SONEX workgroup are grateful to the Joint Information Systems Commitee (JISC) for support. In particular, thanks are due to Neil Jacobs and Andrew McGregor for their cooperation and guidance.

[3] Suggestion for providing a SONEX webspace on deposit initiatives was followed by the creation of the SONEX blog in Apr'10 for publication of the workgroup files.

## References


1. SONEX workgroup for Scholarly Output Notification and Exchange blog, http://sonexworkgroup.blogspot.com/
2. International Repositories Workshop (Amsterdam, The Netherlands, Mar 16-17, 2009), http://www.ukoln.ac.uk/events/ir-workshop-2009/
3. International Repositories Infrastructure wiki, http://repinf.pbworks.com/
4. SWORD: Simple Web-service Offering Repository Deposit. SWORD2 Project Final Report, http://www.jisc.ac.uk/media/documents/programmes/reppres/sword2finalreport.pdf
5. Action plan for Repository Handshake at the International Repositories Infrastructure wiki, http://repinf.pbworks.com/Repository-handshake
6. SONEX workgroup members and affiliations, http://sonexworkgroup.blogspot.com/2010/04/workgroup-participants-and-affiliation.html
7. "Richard Jones joins SONEX workgroup", http://sonexworkgroup.blogspot.com/2010/04/richard-jones-joins-sonex-workgroup.html
8. "A summary of ongoing deposit-related projects", http://sonexworkgroup.blogspot.com/2010/05/summary-of-ongoing-deposit-related_9901.html
9. Grace, S., Gartner, R.: Modelling national research assessments in CERIF. In: CRIS2010, Jun 2-5, Aalborg, Denmark, http://www.cris2010.org/index.php?SID=ceacd8d1ab7c5e930f435b7991ca8fa5&action=showPaper&from=0&id=27
10. Elbæk, M.K., Sandfær, M., Simmons, E.: CRIS/OAR interoperability workshop. In: CRIS2010, Jun 2-5, Aalborg, Denmark, http://www.cris2010.org/index.php?SID=ceacd8d1ab7c5e930f435b7991ca8fa5&action=showPaper&from=0&id=27
11. "CRIS2010 Aalborg: a brief report", http://sonexworkgroup.blogspot.com/2010/06/cris2010-aalborg-brief-report.html
12. The PEER Project, http://www.peerproject.eu/
13. Open Access Repository Junction (OA-RJ) Project blog, http://oarepojunction.wordpress.com/
14. "BioMed Central partners with Massachusetts Institute of Technology Libraries to deposit open access articles automatically using SWORD protocol", BMC Press Release, http://www.biomedcentral.com/info/presscenter/pressreleases?pr=20100429
15. The Depot, http://depot.edina.ac.uk/about/
16. OpenDepot.org, http://opendepot.org/
17. JISC Grant Funding Call 2/10: Deposit of research outputs and Exposing digital content for education and research, http://www.jisc.ac.uk/fundingopportunities/funding_calls/2010/03/210depositexpose.aspx
18. First projects selected at JISC Deposit Call, http://sonexworkgroup.blogspot.com/2010/07/first-projects-selected-at-jiscdepo.html
19. DepositMO: Modus Operandi for Repository Deposits, http://blogs.ecs.soton.ac.uk/depositmo/
20. RePosit: positing a new kind of deposit, http://jiscreposit.blogspot.com/
21. DURA: Direct User Repository Access, http://jisc-dura.blogspot.com/
22. SONEX Deposit Opportunities: a Poster, https://repinf.pbworks.com/f/poster_SONEX_deposit_opportunities.pdf

# Epidemiology Experiment and Simulation Management through Schema-Based Digital Libraries

Jonathan Leidig[1], Edward A. Fox[2], Madhav Marathe[1], and Henning Mortveit[1]

[1] NDSSL, Virginia Tech, Research Building XV (0477), 1880 Pratt Drive, Blacksburg, VA 24061, USA
[2] Dept. of Computer Science, 114 McBryde Hall, Mail Code 0106, Virginia Tech, Blacksburg, VA 24061, USA

**Abstract.** Digital libraries that maintain inputs, result datasets, and documentation of analyses would be beneficial for users of simulation and experimentation systems. A schema describing the simulation model may be used by a generated digital library for semantic integration with a simulation infrastructure. Based on a provided schema, the digital library may generate a generic user interface layout, database schema, and processes for launching new simulations. User roles are supported through a generic interface capable of performing a variety of user tasks. Formal descriptions of simulation content enable provenance investigations and domain-specific search. Simulation infrastructures with digital libraries are provided interoperability through managed datasets and interconnected software components. We present a generic, component-based simulation supporting a digital library that is customized through provided schemas and extensible through the addition of components.

**Keywords:** Automated Generation, Experiment Management, Formalisms

## 1 Introduction

Computational epidemiology supports study of the spread of diseases. Recent collaborations between computational epidemiology institutes have identified the need for digital libraries to support simulation infrastructure. Digital libraries (DLs) may be leveraged to support data management, user interface generation, and computation submission functions for modeling and simulation software. Simulations within the computational epidemiology domain allow public health officials to experiment and analyze potential policies for various contexts. This experimentation process makes use of a sequence of digital content: simulation schemas describing a model, input parameter configurations, raw datasets, result summaries, analyses and plots, documentation, publications, and annotations. Content may be captured at each stage of the scientific process with different users interested in generating and accessing content to support specific user roles and tasks. Computational epidemiology studies require collaboration by model builders, computational scientists, analysts, locals to an area who are aware of

a region's population structure and characteristics, public health officials, and funding providers. These groups typically lack DL building expertise and would benefit from a system which produces a DL and infrastructure based on an epidemiological model.

Many standalone simulation applications and online repositories interact with users through a customized, static web or command line interface for providing input parameters and returning results. Grid, cloud, and volunteer computational resources form the backend of computational epidemiology simulation and analysis applications. DLs may be incorporated into the simulation infrastructure to enable access to content and to provide seamless interoperability between system components. A DL paired with a simulation model may automate: user input value harvesting; launching a study; harvesting results; conducting analyses; generating plots; organizing and managing simulation-related content; disseminating studies and findings; and providing a mechanism for collaboration between simulation researchers, study designers, analysts, and policy setters.

Here we describe our current efforts in developing a framework to formalize and generate this class of digital library systems. We present research on SIMulation supporting Digital Library (SimDL). Domain information is encapsulated inside schemas to allow extensions into other simulation fields or non-simulation experimentation processes. SimDL is similar to efforts in generating e-Science and e-Infrastructure digital libraries. Infrastructures for e-Science DLs typically combine grid computational and storage networks with collections of shared content. e-Science DLs provide seamless access to content and service resources to support communication and collaboration for a research community [2]. Research communities may not necessarily have experience in DL design and development but desire available services and discoverable content through a portal. The DILIGENT system produces DLs that manage data and metadata to support the entire knowledge production and consumption life-cycle for a virtual organization [2]. DILIGENT allows a community to construct a specialized DL combining services and workflows. D4Science and D4Science-II are a continuation of the efforts to develop DILIGENT and promote interaction and interoperability across multiple e-Infrastructures [1].

These projects and SimDL provide services for scientific content generation, curation, representation, and management. However, SimDL focuses on structured simulation processes. A generated SimDL instance provides an interface, request submission, analysis submission, and data management for simulation models and software. Additional workflows are integrated through added DL components, or plug-ins, that make use of staged content (e.g., additional analysis or data mining may make use of existing datasets). These parallel e-Science DL efforts aim to bring together interoperable e-Infrastructures to generate worldwide virtual research environments. The focus of SimDL is to provide DL services related to computational technologies for individual researchers, institutions, and collaborators, not semantic integration between resources. Semantic integration of epidemiological models is an unsolved problem though an ontology and DL of models and content will be of future use in integration efforts.

Digital libraries are important for managing and curating scientific content. DLs provide user interoperability for collaboration and cooperation on scientific endeavors. Multiple collections of content in different formats from specific stages of a simulation study may be managed in a DL. DLs within a larger infrastructure manage content and connect systems into a workflow.

## 2 User Interoperability

Societies are defined as a set of communities and a set of relationships among communities through activities [4]. Simulation DLs support a wide range of users interested in different aspects of experiment design and management. Collaboration between users is crucial in conducting quality simulation studies that guide public policies. SimDL provides a medium for exchanging information, streamlining the progression of a study from design to result validation, and communication between participants. As an example, public health workers in Bandafassi, Senegal should be able to upload information on population demographics and bed net surveys to allow simulations in areas at high risk of malaria.

### 2.1 User Society Modeling

The following communities maintain inter-community relationships through activities as described through a set of community-specific digital library scenarios.

1. *Tool builder:* formally define a proposed DL; locate, reuse, and assemble existing components; generate novel components; deploy a DL instance; integrate with a simulation or digitized experimentation infrastructure
2. *System and DL administrators:* set data management policies; clean datasets or metadata; curate content; manage accounts; evaluate existing components
3. *Related systems:* submission and retrieval with experimentation applications and high-performance computing architectures; analysis of submissions and retrieval with analyst oriented software; validation of inputs
4. *Study designer:* generate new model schemas and transform to updated versions; enter or load input configurations; query, browse, and load sub-configurations; save configurations and sub-configurations; validate configurations; submit configurations for execution; monitor experiment progress
5. *Analyst:* submit new analysis requests; view automated or requested analyses
6. *Annotators:* mark annotations on streams of content or individual documents
7. *Explorer:* query and browse collections or experiment streams of documents

### 2.2 User Collaboration and Cooperation

Collaborations between computational epidemiology institutes based in the US and Switzerland, each with years of experience in simulation studies, have identified the need for SimDL. SimDL aims to provide a generic, extensible component-based digital library. The reliance on schemas allows SimDL instances to support simulation applications for an unrestricted set of domains and contexts that

have structured input requirements and a waiting simulation launching broker. Non-simulation scientific content also may be handled by SimDL instances sans the simulation submission component (e.g., wet labs or instruments producing digital data points along with environmental input conditions).

SimDL was designed out of a need to automate the management of information produced by multiple simulation applications at each institute. Tool builders desired fully automated deployment of a basic DL instance and low amounts of continued maintenance. The management of scientific data requires policy decisions for data preservation, curation, and distribution mechanisms within the digital library. Shared access to a digital library hosted by one institution allows privileged experts at multiple locations to directly and indirectly collaborate, communicate, cooperate, and coordinate research efforts through the DL. The automation of conducting simulation studies is accomplished by the integration of SimDL into an existing simulation infrastructure. Until scientific digital libraries become mainstream in the simulation community, DLs will likely continue to be integrated into a pre-established experimentation process and existing software system. Customization of the simulation-launching component may be required to transmit inputs to simulation software (e.g., transmitting XML input files to a waiting computation request broker). The generic user interface, shown in Fig. 1 and implemented with the Google Web Toolkit[1], is constructed by displaying a user selected model. The parameters to the model are then displayed along with tabs to load existing inputs or switch roles (e.g., analyst).

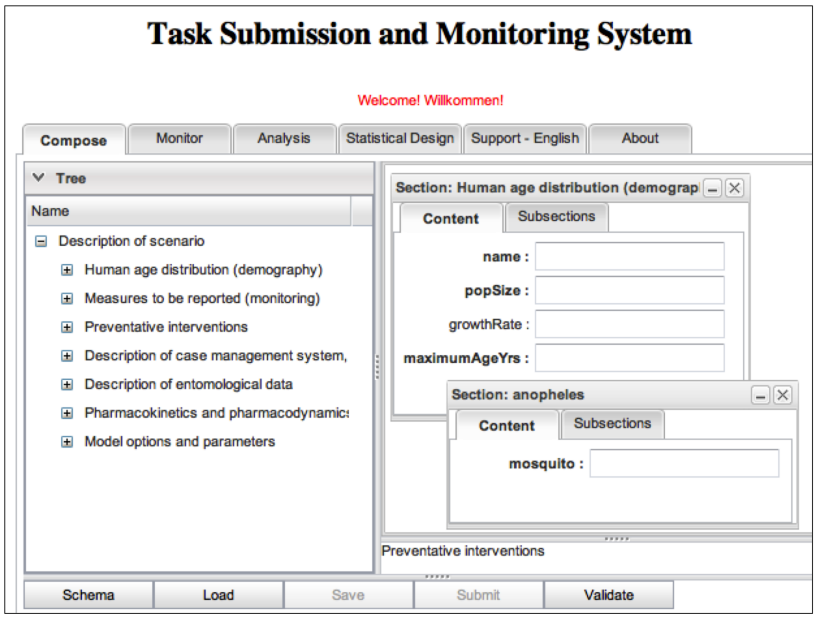


**Fig. 1.** SimDL's automated, generic web UI snapshot displaying a model's schema.

[1] http://code.google.com/webtoolkit

Through SimDL, various types of users can interact in one place. Multiple experts are required to conduct complex simulation-based research. Similarly, multiple user roles exist for simulation-based digital libraries which support collaboration, participation, and privacy as defined in [5]. In SimDL, collaboration is assisted by capturing and exchanging information produced from complex tasks performed by users at different stages of the experimentation process. The collaboration requires multiple tools to answer research questions by translating a query or simulation request into successive units of work. Although the submission of simulation results to an analysis system may be automated, participation from users is required to launch new simulations, perform customized analysis, draw conclusions, and annotate streams of content. Users with similar and differing roles work on the same content, further existing work between and within streams of content, and switch between consuming digital objects from previous experimental stages to providing content for users in successive stages. Practical consideration may restrict the number of users with a simulation launching role to reduce the stress on computational and data storage resources. Users under most roles interact with a digital library's content and are presented with the ability to interact with multiple schemas and versions of schemas across all available simulation applications, domains, and contexts within a single, generic interface. By selecting tabs in the UI, users with a particular role may transition seamlessly to another role (e.g., study designers switching to analysts when viewing the results from a launched simulation).

Tool builders are provided with reusable, formally defined components. The use of a schema automates the UI generation, database mapping, simulation launching, and collection management processes. Human involvement consists of developing a schema for a model and starting the generation process. For study designers, the digital library hides the complexity of launching and analyzing simulations. Experts in an field (e.g., mosquito vectors or population demographics) may upload high-quality sets of input parameters to be reused by users with less knowledge of a portion of an epidemiological model. The designer may reuse and modify existing sub-configurations, submit simulation requests, make use of data management services, and interact with the simulation infrastructure. Analysts are able to retrieve datasets and summaries of datasets along with the input conditions from which the results are derived. Automated tasks for analysts include the generation of plots and summary statistics. Explorers are able to query and retrieve content across the entire provenance trail for findings without having to interact with each simulation system component. Users with this role may discover existing content or identify a lack of previous simulations for a queried segment of the simulation system's multi-dimensional input space.

## 3 Content Interoperability

Digital libraries have been non-uniform in the provision and implementation of services. The DL community would benefit from formal definitions of existing and proposed digital libraries along with the services provided. Formal definitions

of content promotes services for discovery, reuse, and provenance investigations of scientific data. Simulation related content includes a chain of staged data produced by successive steps in the experimental process. A simulation model's schema provides the domain context required to automate the generation of a model-specific DL interface with data management support.

### 3.1 Theoretical Foundations

Previous efforts to define digital libraries have progressed towards a digital library reference model. Two such efforts include the 5S framework [4] and the DELOS Reference Model [3]. The initially proposed 5S framework consisted of formal descriptions of core DL functionality. The framework has since been extended through the addition of subsequent definitions tailored to describe aspects of digital libraries within a particular scope (e.g., content based image retrieval) [7]. Common services required by many DLs involve indexing, searching, and browsing content as well as query and annotation processes [4]. Definitions of new services or aspects to DLs may build on top of existing definitions, and producing formal descriptions of digital libraries is facilitated through reuse of the existing set of definitions. Assumptions 1-3 below underlie the reasoning for expending efforts in formally describing the foundation for SimDL.

**Assumption 1:** *Simulation-supporting DLs can provide interoperability through UI generation, infrastructure functionality, and managing experiments.*

**Assumption 2:** *Formal descriptions of SimDL content, services, and users exist and can fully characterize this class of DLs.*

**Assumption 3:** *Formal descriptions of DL components and functionality may be leveraged to produce and deploy DL instances with the stated capabilities.*

Extending the 5S framework concentrates new definitions to describing functionality required by experiment supporting DL systems and not included in previous DL descriptions. The following set of 5S extending definitions describe the digital content supported by the informally described functionality and services of a scientific experiment supporting DL instance.

### 3.2 Digital and Complex Objects

These definitions build upon the set of previous 5S definitions (i.e., handles, streams, structures, digital objects, complex objects, and annotations).

*Definition 1:* A *schema* is a digital object of tuple $sch = (h, sm, S)$ where

1. $h \in H$, where $H$ is a set of universally unique handles (labels);
2. $sm$ is a stream;
3. $S$ is a structure that composes the schema into a specific format (e.g., XSD structure of elements and attributes of restricted values [8]).

*Definition 2:* An *input configuration* specification matching an XSD schema is a tuple $icfg = (h, sm, ELE, ATT)$ where

1. $ELE$ is a set of XSD elements;

2. $ATT$ is a set of XSD attribute values for an element $ELE_i$.

*Definition 3:* A *sub-configuration*, a subset of an input configuration, is a tuple $sub - icfg = (h, sm, icfg, ELE, ATT)$ where

1. $icfg$ conforms to an XSD schema and $icfg \supseteq sub - icfg$;
2. $ELE$ is a set of XSD elements where $ELE \subseteq icfg$'s $ELE$;
3. $ATT$ is a set of XSD attribute values for an element in $ELE$.

*Definition 4: Analysis* of a set of experiments is a complex object consisting of textual or numeric documents and images (e.g., plots and graphs) and is defined as a tuple $ana = (h, SCDO = DO \cup SM, S, icfg)$ where

1. $DO = do_1, do_2, ..., do_n$, where $do_i$ is a digital object;
2. $SM = sm_1, sm_2, ..., sm_n$ is a set of streams;
3. $S$ is a structure that composes the complex object *cdo* into its parts in $SCDO$;
4. $icfg$ is the input configuration and one-to-one mapping to a raw dataset.

*Definition 5:* An *experiment* is a complex object consisting of the full range of information constituting an experiment within a domain and is defined as a tuple $exp = (h, SM, sch, icfg, ana, D, A)$ where

1. $SM = sm_1, sm_2, ..., sm_n$ is a set of streams;
2. $D = d_1, d_2, ..., d_n$ a set of additional documents, e.g., summary or publication;
3. $A = an_1, an_2, ..., an_n$ a set of annotations describing the overall experiment and individual digital documents.

Support for provenance investigations follows directly from these definitions of the structured workflow of scientific studies and simulation experiments. Allowing a digital library's users access to an entire provenance stream allows claims to be supported, organizes the existing body of previous work, and allows data mining of collections related to a simulation system. See Fig. 2 for an Open Provenance Model representation overview of the preceding definitions [6].

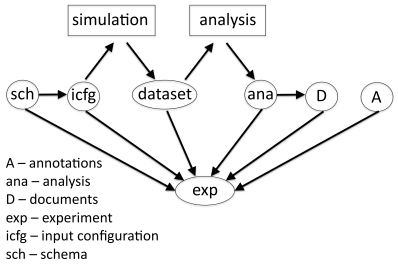


**Fig. 2.** Context-specific simulation-based digital object provenance as represented by the Open Provenance Model.

Federation of multiple heterogeneous collections is provided by leveraging the general one-to-one mapping between sets of items produced by each experimentation stage. The use of model schemas by SimDL allows digital library services to make use of contextual information, e.g., allow for model parameters to be queried in search functions. In computational epidemiology, connecting non-standardized models requires human-intensive collaboration between model builders aided by schemas and ontologies, falling outside of SimDL's initial scope of automated services. Users could conceivably test hypotheses across semantically linked systems, models, and datasets from different groups.

## 4 System Interoperability

Managing simulation studies involves an interface to harvest input parameters, a mechanism to launch a model's software with parameters, storage of output results, a mechanism to launch analysis software, storage of analyses, and storage of annotations and related documents. Digital libraries are an important component in infrastructures which manage these types of scientific content and may assist in providing interoperable services between stages of the content generation workflow. SimDL attempts to provide interoperability for a user interface, model software, analysis software, and backend databases.

### 4.1 Simulation and Experiment Supported Infrastructure

Many institutions and research groups across a broad spectrum of domains make use of simulation applications to produce experimental data. A model's schema, including required input parameters, may be used to generate a specialized DL, interface, and data management process. With an input schema, a generic user interface (UI) may be constructed by the digital library to allow study designers to provide the necessary parameters for an experiment. Input parameters may then be harvested by the DL, validated, and sent to the simulation system.

The raw simulation results, result summaries, and annotations may be stored along with an input configuration. Connecting a DL with analysis software automates submitting analysis requests and receiving analyses or plots. A model's schema provides domain information that guides the indexing, searching, ranking, and retrieval of the multiple stages of content produced by the experimentation process. Databases and tables may be automatically designed and built based on a schema. By maintaining a set of schemas and simulation connections, a single DL instance may support multiple applications and backend computational resources with data management services. Figure 3 details the structure of SimDL which provides a 'front end' for one or more simulation models and software. DL designers may elect to customize SimDL by including a subset of the pre-existing components; add new components as plug-ins; customize the generic interface; and connect data mining software, visualization tools, or connections to other digital libraries through the external systems connection broker.

### 4.2 Infrastructure Interoperability

SimDL acts as a services provider when integrated within a larger infrastructure. The generic interface provides a single point of access to multiple collections and eases the effort required to manage users, support roles, and allow users to switch roles. The automated interface provides a browser with a consistent look and feel across user roles. A simulation-launching component submits simulation requests and input configurations to a backend model's software. Simulation brokers monitoring the simulation requests then can process the request and launch the experiment on a set of optimized computational resources. Similarly, submitting analysis requests can be automated as a digital library service. Through

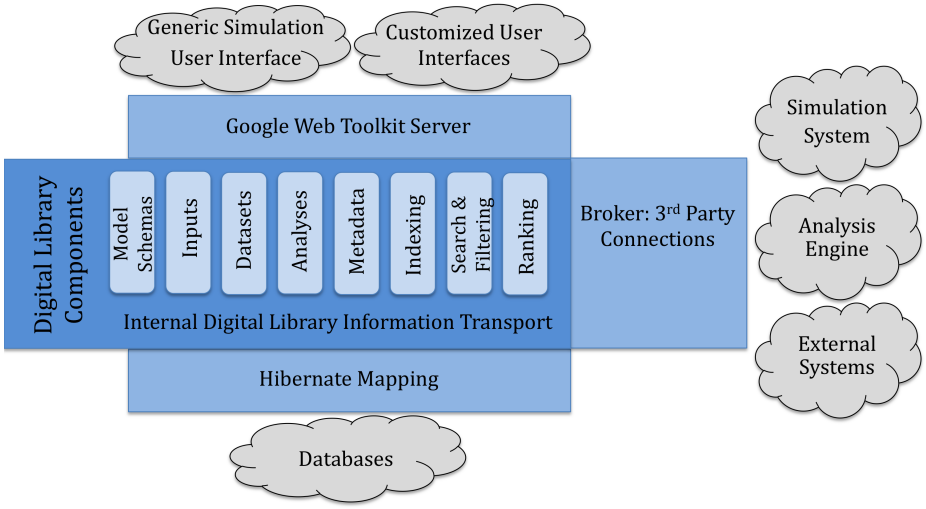


**Fig. 3.** The internal organization of SimDL's system components.

the use of an analysis request staging space, analysis procedures and software can be invoked for a recently completed simulation's datasets. The storage system for the digital library need not be set up manually. SimDL automates the process of creating, inserting into, or querying a database after processing the logic contained by a selected schema. An intuitive, simplistic querying function over a model's parameters and user annotations provides access to the stream of digital objects generated by each of the composite functions.

Automatically generated storage components are customizable to directly support a schema (e.g., constructing a database and access methods by parsing a schema's XSD structure of elements and attributes). Hibernate's [2] automation functionality allows database tables to be constructed to match a recently uploaded schema and map records in each table with objects SimDL may handle. This automation produces the SimDL storage layer for a specific schema and allows for multiple DL components to access a schema's related collections. Hibernate is used throughout SimDL to connect to databases, and only requires that a SimDL database account be setup by an institution's database administrators.

## 5 Future Work

SimDL currently supports input configurations as desired by one institution. Functional extensions and research efforts to improve SimDL are in continual, iterative development. While the intention is to provide SimDL as an extensible platform for deploying digital libraries to manage scientific content, several components still in development are required to promote adoption within the epidemiology, simulation, and larger scientific communities. Plug-in components are

[2] http://www.hibernate.org

in development to annotate and recommend digital objects; ask high level semantic search queries; query across SimDL instances; federate SimDL instances with other digital libraries through semantic schema mapping and domain ontologies; provide interoperability for accessing metadata and collections; and provide a method for communication within the DL system (e.g., message boards). Visualization components may be added to the generated UI to provide intuitive input configuration navigation; dragging and dropping of sub-configurations; fisheye overviews of a schema; and treemap views of datasets, analyses, documents, and annotations collections. Maintaining user sessions across SimDL instances also may provide more coherent use of user credentials, demographics, access rights, preferences, tailored views, expertise, classes of rights (access to content), and roles (access to system functionalities). The foremost expectations for a SimDL instance are to holistically provide DL services for interested simulation groups within a domain using multiple models as contexts.

## Acknowledgements

This work has been partially supported by NIH MIDAS project 2U01GM070694-7, DTRA CNIMS Grant HDTRA1-07-C-0113 and R&D Grant HDTRA1-0901-0017, and NSF HSD Grant SES-0729441 and PetaApps Grant OCI-0904844.

## References

1. Candela L., Castelli D., Pagano P. D4Science: an e-infrastructure for supporting virtual research. In: IRCDL 2009 - 5th Italian Research Conference on Digital Libraries, (2009) 166 - 169.
2. Candela, L., Castelli, D., Langguth, C., Pagano, P., Schuldt, H., Simi, M., Voicu, L.: On-Demand Service Deployment and Process Support in e-Science DLs: the DILIGENT Experience. In: ECDL Workshop DLSci06: Digital Library Goes e-Science, (2006) 37-51.
3. Castelli, D., Ioannidis, Y., Ross, S., Schek, H., Schuldt, H. DELOS Network of Excellence on Digital Libraries - Reference Model for DLMS (2006).
4. Goncalves, M., Fox, E., Watson, L. Towards a digital library theory: a formal digital library ontology. Int. J. Digit. Libr. **8** (2008) 91-114.
5. Ioannidis, Y. User Working Group: Towards User Interoperability. First DL.org Workshop presentation (2009).
6. Moreau (Editor), L., Plale, B., Miles, S., Goble, C., Missier, P., Barga, R., Simmhan, Y., Futrelle, J., McGrath, R., Myers, J., Paulson, P., Bowers, S., Ludaescher, B., Kwasnikowska, N., Van den Bussche, J., Ellkvist, T., Freire, J. and Groth, P. The Open Provenance Model (v1.01). Technical Report, Electronics and Computer Science, University of Southampton (2008).
7. Murthy, U., Gorton, D., Torres, R., Goncalves, M., Fox, E., Delcambre, L. Extending the 5S Digital Library (DL) Framework: From a Minimal DL towards a DL Reference Model. JCDL - First Digital Library Foundations Workshop, (2007).
8. XSD. XML Schema, W3C recommendation in May 2001.

# Interoperability Patterns in Digital Library Systems Federations

Paolo Manghi, Leonardo Candela, and Pasquale Pagano

Istituto di Scienza e Tecnologie dell'Informazione "Alessandro Faedo" – CNR, Pisa - Italy
{paolo.manghi | leonardo.candela | pasquale.pagano}@isti.cnr.it

**Abstract.** Service providers willing to offer functionality over an aggregation of information objects are becoming the key actors in the Digital Library world. This aggregation is usually performed by collecting and processing objects from a federation of data providers – e.g., institutional repositories, data archives – through well-known standard protocols for data-exchange. Higher-level interoperability problems affect these scenarios and determine the quality of the service offered by and the success of these providers initiatives. Interoperability problems are mainly due to the impedance mismatch occurring between data models of the objects to be exported by data providers and the data model of the service provider and can be measured in terms of "structural heterogeneity", "semantic heterogeneity", and "granularity-of-representation heterogeneity". In this paper, we define a basic architecture through which we describe the interoperability patterns arising when constructing such a federations and present the D-NET Software Toolkit as a general-purpose practical solution to these issues.

## 1 Introduction

In the last decade, research communities increasingly adopted Digital Library Systems (DLSs) to preserve, disseminate and exchange their research outcome, from articles and books to images, videos and research data. The multidisciplinary character of modern research, combined with the urgency of having immediate access to the latest results, moved research communities towards the realization of service providers aggregating content from federations of DLSs. Service providers, which act here as *data consumers*, offer functionality over an aggregation of information objects [1] obtained by manipulating objects collected from a set of DLSs, which act here as *data providers*. Data providers expose *content resources* while the service provider is willing to consume such resources to accomplish its *tasks*.[1] In order to interoperate, the two interlocutors have to face the following challenges: (*i*) "how to exchange objects", i.e., identifying common data-exchange practices, and (*ii*) "how to harmonize objects", i.e., resolve data impedance mismatch problems arising from distinct data models assumed by the two. Typically, data access across different platforms is overcome by adopting XML as lingua-franca and standard data-exchange protocols, such

[1] By data providers we mean DLSs whose collection of objects are useful to a service provider for accomplishing its tasks. In other words, a DLS cannot be a data provider for a service provider that is not interested in its content resources, i.e., these are not useful for achieving its tasks. In this sense, being or not being interoperable is an exclusive problem of a data provider and a service provider, hence of two interlocutors willing to interact to accomplish a task together.

as OAI-PMH [3] and OAI-ORE [4]. If the adoption of these standards represents an important step towards interoperability, it still leaves open core interoperability challenges. These challenges have mainly to do with impedance mismatch issues that can be classified as:

**Data model impedance mismatch** – the mismatch between the service provider data model (i.e., the XML schema capturing structure and semantics of the information objects to be generated) and the data providers data model (i.e., the XML schema capturing structure and semantics of the information objects to be collected and elaborated).

**Granularity impedance mismatch** – the mismatch between XML encodings of information objects at the service provider and data providers sites, which may consider different levels of granularity. For example, one DIDLXML file may represent (i.e., package) a set of the information objects, together with relationships between them, namely a "compound object"; a MARCXML file, instead, typically represents the descriptive metadata of one information object.

When service providers and data providers feature different data models or granularity of representation, specific solutions to achieve interoperability must be devised and implemented.

In this paper we shall describe the interoperability patterns which generally surface when constructing DLS federations featuring data model and granularity impedance mismatch, and present the D-NET Software Toolkit, today used in several real-case scenarios, as a practical general-purpose solution to those. In particular, Section 2 introduces the terminology and defines a basic architecture for DLS federations. Section 3 depicts DLS federation interoperability issues and sketches ideas for solutions. Section 4 presents the D-NET Software Toolkit as an implementation of them. Finally, Section 5 concludes the paper.

## 2 Digital Library System Federations

Figure 1 shows the basic architecture of a DLS federation. Data providers manage a collection of information objects that match a given data model. In particular, information objects may be publications, audio and video material, compound objects (i.e., sets of interlinked information objects), or "surrogates" of all these, namely metadata descriptions of information objects. In general, we can assume that data providers handle a "graph" of information objects, whose structural and semantic complexity depends on the data model to which it conforms. Typically, data providers expose a "view" of their objects collection, by identifying the subset of objects to be exported and, for those, the structural aspects to be revealed to the world.

Similarly, the service provider operates a collection of information objects matching a local data model. Such a collection is obtained by bulk-fetching, manipulating and then aggregating information objects exported by the individual data providers.[2]

As highlighted in Figure 1, in the Digital Library world the basic interoperability issues occurring between a service provider and data provider to exchange information objects are typically overcome by adopting XML as lingua-franca in combination with standard data-exchange protocols (e.g., OAI-PMH). XML files have a labelled tree structure and can

[2] Other federative approaches are possible, for example adopting distributed search as interaction mechanisms, but these are out of the scope of this paper.

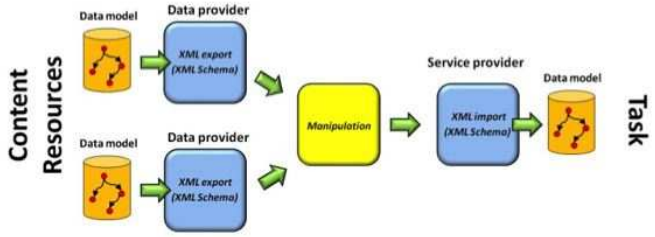


**Fig. 1.** DLS federation: basic architecture

therefore represent any kind of information objects; in principle XML files can also contain the payload of another file (e.g., a PDF) instead of a reference to a payload.

On the data provider side, information objects are associated with an XML schema that captures the essence of their data model and special "exporting components" are developed to generate and return the relative XML representations in response to service provider's requests. On the service provider side, a similar but inverted situation occurs. Information objects have a correspondent XML schema and a special "importing component" is constructed, capable of converting an XML file onto an information object instance.[3] Note that having the same data model does not mean to have the same XML schema representation.

## 3 Interoperability Patterns

Observing Figure 1, it is clear that if the XML schema and the intended semantics of the element values of all data providers and the service provider match, no interoperability issue occurs. The service provider can smoothly collect from data providers information object XML representations and directly aggregate them locally into one collection. When this happens, it is because common agreements have been established or an interoperability solution has been already realized. When this is not the case, data and service providers cannot interoperate due to *data model* or *granularity* impedance mismatches and solutions must be devised.

### 3.1 Data Model Impedance Mismatch

Data model mismatches occur at the level of the data provider and the service provider data models when the relative XML schema views have paths of XML elements (paths in the following) and/or the relative value domains (leaves in the following) that do not exactly match. In this case interoperability issues arise because, due to either structural or semantic heterogeneity, the service provider cannot directly aggregate the information object XML representations of the data providers. Typically, depending on the kind of mismatch, the

[3] Note that, in some cases, data provider or service provider implementations may manage their information objects directly as XML files, onto native XML databases or full-text indices.

solution consists of software components capable of overcoming such differences by applying appropriate XML file transformations (see Figure 2). Implicitly, such transformations convert information objects from one data model onto another.

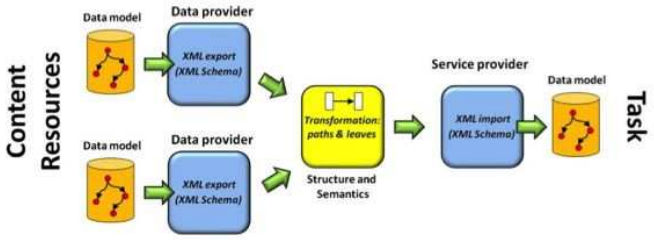


**Fig. 2.** Interoperability issue: data model mismatch

In particular, we can identify two kinds of mismatch, strictly related with each other:

**Structural heterogeneity** occurs when the XML schema of data provider and service provider are not equal, i.e., when their paths do not match. Typical examples are: the service provider paths are a subset of the data provider paths, service provider has paths that are different but correspond to data providers paths (i.e., using different element names or hierarchies to describe the same concept), service provider has paths that do not have a data provider path correspondent (i.e., service provider data model is richer than data provider's one).

**Semantic heterogeneity** occurs at the level of leaves, under two circumstances:

1. service provider and data provider have corresponding leaves in the relative XML schemas (i.e., the schema are equal or are not equal but a one-to-one mapping between their paths can be identified), but do not share the same formats (e.g., date/time formats, person names) and vocabularies;
2. the service provider has leaves in the XML schema that do not find a direct correspondent in the data provider XML schema; such leaves must be derived by elaborating (i.e., computing over) leaves of the data providers XML files.

Interoperability solutions to data model mismatches consist in the realization of *transformation components*, capable of converting XML files conformant to data providers schema onto XML files conforming to the service provider schema. The logic of the component maps paths in the original schema onto paths of the target schema. In principle, source and target paths may have nothing in common, either the element names or hierarchical structure of the elements. Similarly, the values of the leaves of the output XML files may completely differ from the ones in the input XML files, in domain, format and meaning.

Depending on the application scenario the implementation of transformation components may largely differ. We can identify the following cases, together with possible categories of solutions, where, in some cases, the cardinality of data providers in the federation may impact on cost and sustainability.

**All data providers have the same XML schema** The transformation component should generate XML files conforming to the XML schema of the service provider from the data provider XML files, whose paths are the same. To this aim all leaves of service provider XML files are generated by processing the leaves of the incoming XML files through transformation functions ($F$). The complexity of the $F$ can be arbitrary: "feature extraction" functions taking a URL, downloading the file (e.g., HTML, PDF, JPG) and returning content extracted from it; "conversion" functions applying a translation from one vocabulary term to another vocabulary term; "transcoding" functions transforming a leaf from one representation format to another (e.g., date format conversions); "regular expression" functions generating one leaf from a set of leaves (e.g., generating a person name leaf by concatenating name and surname originally kept in two distinct leaves). Since only one source XML schema is involved, the component can be developed around the only one mapping necessary to identify which input leaves must be used to generate an output leaf through a given $F$.

**Data providers have different XML schemas** The transformation component should generate XML files conforming to the XML schema of the service provider assuming the incoming XML files have different paths, depending on the data provider's XML schema. In principle, the solution could be that of realizing one component as the one described for the previous scenario for each set of data providers sharing the same schema. However, if an arbitrary number of data providers is expected, possibly showing different structure and semantics and therefore requiring different transformation functions, this "from-scratch" approach is not generally sustainable. One solution is that of providing general-purpose tools, capable of managing a set of "mappings" (e.g., Repox[4], D-NET [5]). These consist in named lists (*input paths*, $F$, *output path*), where the output path (which may be a leaf) is obtained by applying $F$ to the input paths. Mappings can be saved, modified or removed, and be reused while collecting XML files from data providers sharing the same structure and semantics. Similarly, the component should allow for the addition of new $F$ to match unexpected requirements in the future.

### 3.2 Granularity Impedance Mismatch

In designing an interoperability solution between a data provider and a service provider, the "granularity" of the objects exported by a data provider may not coincide with that intended by the service provider. For example, data providers may export XML files that represent "compound objects", which are rooted sub-graphs of the local object collection. The service provider might be interested in the compound object as a whole, thus adopting the same granularity, or only in some of the objects that are part of it, thus adopting a finer granularity. Hence, in addition to the data model mismatch, a granularity impedance mismatch may arise.

The following scenarios typically occur (see Figure 3):

**(1:N)** each XML file of the data provider translates onto more XML files of the service provider, e.g., un-packaging of compound object XML representations. The solution

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

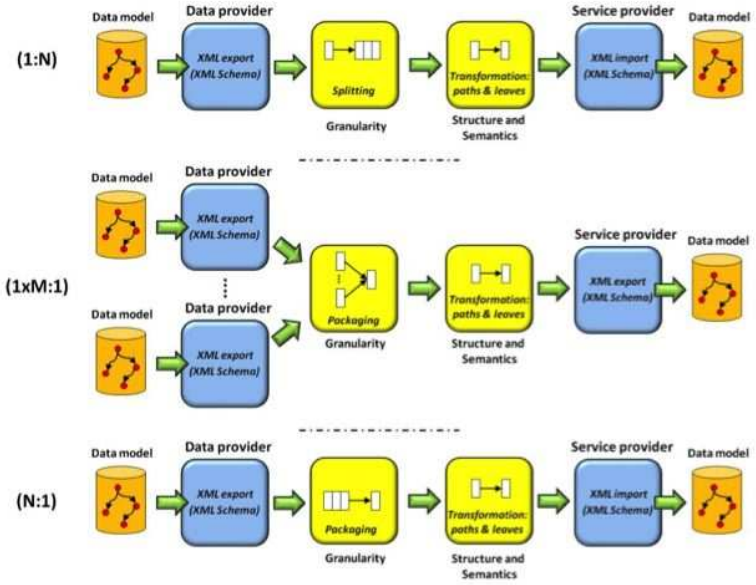


**Fig. 3.** Interoperability issue: granularity mismatch

requires the realization of a *splitting component*, capable of obtaining a list of XML files from each XML file exported by the data provider. The operation may occur before or after structural and semantic interoperability issues have been tackled.

**(N:1)** more XML files from one data provider correspond to one XML file of the service provider. The solution requires the realization of a *packaging component* capable of identifying the set of XML files exported by the data provider which have to be combined onto one XML file of the service provider. The logic of such a combination may vary across different application domains, but is often based on shared identifiers and/or external references to those.

**(1xM:1)** the combination of one XML file for each data providers in the federation corresponds to one XML file of the service provider. The solution is similar to the case (N:1), where the *packaging component* has to be able of managing the XML files exported by a set of data providers and identify those that have to be combined onto one XML file of the service provider.

### 3.3 Architecture of interoperability solutions

The reader may have noticed that no assumption has been made on which entity is in charge of the realization of the components. In fact, given an interoperability scenario like the one we described, the transformation, splitting and packaging components may be fully or partly realized by the data providers or by the service providers, depending on the level of engagement and agreements established by the federation.

**"Bottom-up" federations** Some federations are realized by organizations who have control over the set of participating data providers. All interoperability issues are to be solved at the data providers sites and the service provider is a simple object collector and aggregator. Although quite rare in practice, due to the difficulties of autonomous and independent organizations of respecting external guidelines, this is the case for example for DAREnet-NARCIS[5], the service provider of the research and academic institutions of the Netherlands. The relative institutional repositories agreed on exporting their bibliographic metadata records according to Dublin Core XML format and to a precise semantics of the elements (e.g., given vocabularies and formats for dates and creators).

**"Open" federations** Some federations are attractive to data providers, which are available to adhere to given "data model" specifications in order to join the aggregation. However, to not discourage participation, service providers define "data provider guidelines" that are often limited to the adoption of simple XML schemas and to light-weight best practices on usage of leaves. Therefore, in most of the cases, the realization of leaf transformation components is left to the service provider (e.g., the DRIVER repository infrastructure)[6].

**"Community-oriented" federations** A community of data providers handling the same typology of content, but in different ways, finds an agreement on a common data model and together invests on the realization of a service provider capable of enabling a federation by solving all interoperability issues that may arise (e.g., the European Film Gateway project[7]). In such scenarios, packaging, if needed, typically occurs at the service provider side, while part of the transformation may also occur at the data provider side, before XML export takes place. If this is not the case, data providers are directly involved in the definition of the paths and leaves transformation specification, while the service provider limits the intervention to the relative implementation.

**"Top-down" federations** Federations may be the result of the interest of a service provider to offer functionality over data providers whose content is openly reachable according to some declared XML schema (e.g., OAIster-OCLC project[8], BASE search engine[9]). In such cases, it is the service providers that has to deal with interoperability issues.

## 4 D-NET Software Toolkit

The D-NET Software Toolkit was designed and developed within the DRIVER and DRIVER-II EC projects and is being used and extended in EFG and OpenAIRE projects. Its software

[5] DAREnet: Digital Academic Repositories, `http://www.narcis.nl`
[6] The DRIVER Infrastructure, `http://search.driver.research-infrastructures.eu`
[7] The European Film Gateway project, `http://www.eureopanfilmgateway.eu`
[8] OAIster-OCLC project, `http://www.oclc.org/oaister/`
[9] BASE: Bielefeld Academic Search Engine, `http://www.base-search.net`

is open source, developed in Java and available for download[10]. D-NET implements a Service Oriented Architecture (SOA), based on the Web Service framework, capable of operating run-time environments where multiple organizations can share data sources and services to collaboratively construct DLS federations [5]. To this aim, D-NET provides several services with which developers can "assemble" applications for the collection and manipulation of information objects from a pool of data sources. Specifically, a D-NET infrastructure is made of two main logical layers: the system core, called enabling layer, whose function is to support the operation of the application layer, which consists of the services forming the applications.

### 4.1 Enabling Services

The enabling services support a framework where developers can combine services to build a running application. The ResultSet Enabling Service, for example, can create, delete, operate a set of ResultSets, i.e., "containers" for transferring lists of objects between a "provider" service and a "consumer" service. Technically, a ResultSet is an ordered list of files identified by an End Point Reference (EPR), which can be accessed by a consumer through paging mechanisms, while being fed by a provider, in order to reduce response delays and limit the objects to be transferred. D-NET services are designed to be organized into workflows by relying on ResultSet EPRS, i.e., they are conceived to accept as input parameters and return as results of their invocations ResultSet EPRs. Other enabling services offer functionalities for service orchestration and autonomicity, safe communication, and discovery [2].

### 4.2 Application Services

D-NET services involved in the construction of Digital Library System federations are *mediation services*, i.e., information object collection components, and *manipulation services*, i.e., splitting, packaging and transforming components. We shall see that D-NET also offers service kits for the realization of advanced service providers, capable of managing information objects of arbitrary data models. All D-NET services return and accept objects encoded as XML files through a ResultSet.

**Mediation services** D-NET offers *Object Collection Services* for accessing data providers exposing content through OAI-PMH interfaces (e.g., repositories, archives) or through remote file systems with folders containing XML files. Moreover, special *Data Provider Management Services* allow data provider administrators to "register" their DLSs to the federation, and D-NET administrators to automatically fetch objects from the registered DLSs through *Object Collection Services*.

**Manipulation services** Services in the manipulation area are capable of handling objects to transform them from one XML schema to another. Bulk transfer of XML files from a service to another is performed by sending a ResultSet EPR to the service, which will then actively pull the XML files from it for local usage. Manipulation services implement splitting, packaging and transformation components and include:

[10] D-NET Project, `www.d-net.research-infrastructures.eu`

**MDStore Service** An MDStore Service manages a set of *MDStore units*, each capable of storing XML files of a given XML schema. Consumers can create and delete units, and add, remove, update, fetch, get statistics on XML files from-to a given unit.

**Feature Extractor Service** A Feature Extractor Service generates a ResultSet of XML files from a ResultSet of input XML files by applying a given extraction algorithm to given leaves of the input file. Examples are: extracting the histograms of image payload objects; extracting full-text of PDF payload objects; generating payload objects from payload objects (DOC to PDF). Algorithms can be plugged-in as special software modules, whose invocation becomes available to consumers.

**Transformator Service** A Transformator Service is capable of transforming XML files of one schema into XML files of one output schema. The logic of the transformation, called mapping, is expressed in terms of a rule language offering rules for: (*i*) path removal and addition, leaf concatenation and switch, (*ii*) regular expressions over leaves, and (*iii*) invocation of an algorithm through a Feature Extractor Service on a leaf. User interfaces support administrators at defining, updating and testing a set of mappings, which they can save, refine and reuse in the future. The idea is to offer tools for systematizing the process of definition of XML mappings, traditionally consisting in programming XSLT scripts or other sophisticated and ad-hoc code. The tools capture a very common scenario, in which paths to the leaves of the input and output XML schemas can be mapped onto a flat list of labels ("flat record"). The rules are then applied to input label values in order to obtain output label values and automatically generate the corresponding XML files. For all other cases, administrators can opt for a "traditional" solution by uploading mappings defined as arbitrary complex XSLT transformations or adding new algorithms in the Feature Extractor Service, capable of processing the whole XML file as desired.

**Object Packaging Service** An Object Packaging Service can perform both splitting and packaging operations over XML files provided through ResultSets. Splitting is performed by applying a given xpath query to each of the XML files in an input ResultSet; the XML pieces returned by each query are returned as individual XML files through a ResultSet. Packaging is performed by accepting a set of Result Sets and generating one valid XML file that includes several XML files, one for each of the input ResultSets. The operation accesses the ResultSets incrementally, hence it is important to properly order the ResultSets to obtain the expected XML packages.

**"Service provider" services** D-NET offers a multitude of data management services, used by developers to build service providers capable of managing objects of various data models in the most appropriate way. These include: Store Services (for payloads of any MIME type), Index Services (for full-text indexing and faceted browsing over XML files of any schemas), Compound Object Services (for the creation of typed-graphs of metadata, payload and relationship objects), Database Services (for managing objects as relational records), OAI-PMH Publisher Services (for exporting XML files through the known protocol), Validation Services (for checking the "quality" of a ResultSet of XML files according to a set of validation rules), Search Services (for querying and browsing XML files into Index Services according to SRW/CQL interfaces).

## 5 Conclusions

Since 2006, the D-NET software toolkit was constantly refined and extended to accommodate the interoperability issues arising in the construction of DLS infrastructures across different application domains. In this paper we identified the interoperability patterns recurring in these applicative scenarios and schematized the reasoning behind the realization of possible solutions. Finally we presented the D-NET services that were developed to offer general-purpose tools capable of addressing and tackling the requirements of the identified patterns.

**Acknowledgments** Research 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) and by the ICT-2007.4.3 Information and Communication Technologies Program as part of the DL.org project (Grant Agreement no. 231551).

## References

1. L. Candela, D. Castelli, N. Ferro, Y. Ioannidis, G. Koutrika, C. Meghini, P. Pagano, S. Ross, D. Soergel, M. Agosti, M. Dobreva, V. Katifori, and H. Schuldt. *The DELOS Digital Library Reference Model - Foundations for Digital Libraries*. DELOS: a Network of Excellence on Digital Libraries, February 2008. ISSN 1818-8044 ISBN 2-912335-37-X.
2. L. Candela, D. Castelli, P. Manghi, and P. Pagano. Enabling Services in Knowledge Infrastructures: The DRIVER Experience. In F. E. M. Agosti and C. Thanos, editors, *Post-proceedings of the Third Italian Research Conference on Digital Library Systems (IRCDL 2007)*, pages 71–77. DELOS: a Network of Excellence on Digital Libraries, 2007.
3. C. Lagoze and H. Van de Sompel. The OAI Protocol for Metadata Harvesting. `http://www.openarchives.org/OAI/openarchivesprotocol.html`.
4. C. Lagoze, H. Van de Sompel, P. Johnston, M. Nelson, R. Sanderson, and S. Warner. Open Archives Initiative Object Reuse and Exchange (OAI-ORE) User Guide - Primer. Technical report, Open Archives Initiative, 2008.
5. P. Manghi, M. Mikulicic, L. Candela, D. Castelli, and P. Pagano. Realizing and Maintaining Aggregative Digital Library Systems: D-NET Software Toolkit and OAIster System. *D-Lib Magazine*, 16(3/4), March/April 2010.
6. E. Rahm and P. A. Bernstein. A survey of approaches to automatic schema matching. *The VLDB Journal*, 10(4):334–350, 2001.
7. G. Wiederhold. Mediators in the Architecture of Future Information Systems. *Computer*, 25(3):38–49, 1992.

# Author Index