DL.org: Coordination Action on Digital Library Interoperability, Best Practices and Modelling Foundations

Funded under the Seventh Framework Programme, ICT Programme – “Cultural Heritage and Technology Enhanced Learning”

Project Number: 231551

Deliverable Title: D3.4 Digital Library Technology and Methodology Cookbook

Submission Due Date: February 2011

Actual Submission Date: April 2011

Work Package: WP3

Responsible Partner: CNR

Deliverable Status: Final Version

European Commission
Information Society and Media

SEVENTH FRAMEWORK
PROGRAMME

Document Information

Project	
Project acronym:	DL.org
Project full title:	Coordination Action on Digital Library Interoperability, Best Practices & Modelling Foundations
Project start:	1 December 2008
Project duration:	27 months
Call:	ICT CALL 3, FP7-ICT-2007-3
Grant agreement no.:	231551

Document	
Deliverable number:	D3.4
Deliverable title:	Digital Library Technology and Methodology Cookbook
Editor(s):	L. Candela (CNR), A. Nardi (CNR)
Author(s):	G. Athanasopoulos (NKUA), L. Candela (CNR), D. Castelli (CNR), K. El Raheb (NKUA), P. Innocenti (UG), Y. Ioannidis (NKUA), V. Katifori (NKUA), A. Nika (NKUA), S. Ross (University of Toronto), A. Tani (CNR), C. Thanos (CNR), E. Toli (NKUA), G. Vullo (UG)
Reviewer(s):	C. Thanos (CNR)
Contributor(s):	K. Ashley (DCC), P. Burnhill (University of Edinburgh), T. Catarci (University of Rome "La Sapienza"), G. Clavel-Merrin (Swiss National Library), P. De Castro (Carlos III University Madrid), A. De Robbio (University of Padua), J. Faundeen (USGS), N. Ferro (University of Padua), E. Fox (Virginia Tech), S. Higgins (DCC), R. van Horik (DANS), W. Horstmann (Bielefeld University Library), R. Jones (Symplectic Ltd), G. Kakaletris (NKUA), S. Kapidakis (Ionian University of Corfu), G. Koutrika (Stanford University), P. Manghi (CNR), N. Manola (NKUA), C. Meghini (CNR), R. W. Moore (University of North Carolina at Chapel Hill), L. Moreau (University of Southampton), A. Nürnberger (University Magdeburg), P. Pagano (CNR), H. Pfeiffenberger (Alfred Wegener Institute), A. Rauber (TU-Wien), M. Smith (MIT), D. Soergel (University of Buffalo), M. Thaller (University of Cologne)
Participant(s):	CNR, NKUA, UG
Work package no.:	WP3
Work package title:	Digital Library Models and Patterns
Work package leader:	CNR
Work package participants:	CNR, NKUA, UG
Est. Person-months:	6
Distribution:	Public

Nature:	Report
Version/Revision:	1.0
Draft/Final	Final
Total number of pages: *(including cover)*	125
Keywords:	Digital Library; Digital Library System; Interoperability; Pattern; Interoperability Approach; Best Practice;

Disclaimer

This document contains information on the core activities, findings, and outcomes of the EC-funded project, DL.org, and in some instances, distinguished experts forming part of the project's Liaison Group, six Thematic Working Groups and External Advisory Board. The document may contain references to content in the DELOS Digital Library Reference Model, which is under copyright. Any references to content herein should clearly indicate the authors, source, organisation and date of publication.

This publication has been produced with the funding of the European Commission. The content of this publication is the sole responsibility of the DL.org consortium and cannot be considered to reflect the views of the European Commission.

European Commission
Information Society and Media

The European Union was established in accordance with the Treaty on European Union (Maastricht). There are currently 27 Member States of the Union. It is based on the European Communities and member states cooperation in the fields of Common Foreign and Security Policy and Justice and Home Affairs. The five main institutions of the European Union are the European Parliament, the Council of Ministers, the European Commission, the Court of Justice and the Court of Auditors. (http://europa.eu.int/)

DL.org is funded by the European Commission under the 7th Framework Programme (FP7).

Table of Contents

List of Figures

Summary

The demand for powerful and rich Digital Libraries able to support a large variety of interdisciplinary activities as well as the data deluge the information society is nowadays confronted with have increased the need for '*building by re-use*' and '*sharing*'. Interoperability at a technical, semantic and organisational level is a central issue to satisfy these needs. Despite its importance, and the many attempts to resolve this problem in the past, existing solutions are still very limited. The main reasons for this slow progress are lack of any systematic approach for addressing the issue and scarce knowledge of the adopted solutions. Too often these remain confined to the systems they have been designed for. In order to overcome this gap, DL.org promotes the production of this document with the goal to collect and describe a portfolio of best practices and pattern solutions to common issues faced when developing large-scale interoperable Digital Library systems. This document represents the final version of the Digital Library Technology and Methodology Cookbook.

1 Introduction

Digital libraries represent the confluence of many interdisciplinary fields, from data management, information retrieval, library sciences, document management to web services, information systems, image processing, artificial intelligence, human-computer interaction, and digital curation. Its multi-faceted nature has led researchers to offer a variety of definitions as to what a digital library is, often reflecting on different disciplinary perspectives (Borgman, 2000), (Fox, Akscyn, Furuta, & Leggett, 1995), (Fox & Marchionini, 1998), (Bertino, et al., 2001), (Ioannidis Y. , 2005), (Ioannidis, et al., 2005), (Lagoze C. , 2010). As Gonçalves et al. have explained (Gonçalves, Fox, Watson, & Kipp, 2004), the lack of well defined and agreed boundaries of the term "digital library" arises because digital libraries are essentially complex multi-dimensional applications. Ross pinpointed those aspects by characterizing a digital library as "*the infrastructure, policies and procedures, and organisational, political and economic mechanisms necessary to enable access to and preservation of digital content*" (Ross S. , 2003)(p. 5).

Among the current digital library implementations, there is a variety in character and type of content. Some are homogeneous collections on particular topics or media whereas others have a heterogeneous character (Ross S. , 2003). In addition to that, there is a variety also in services offered over digital library content and audience served. All digital libraries are information systems, and they instantiate particular software systems and information architectures. The lack of agreement on the best design of digital library systems reflects, in part, a lack of agreement on the nature, functionality, and architecture of such information systems.

DELOS[1], the Network of Excellence on Digital Libraries, has contributed to address this issue by launching a long-term process aimed at introducing a foundational framework for the area. The result of this activity have been the 'Digital Library Manifesto' (Candela L. , et al., 2006) and the 'DELOS Digital Library Reference Model' (Candela L. , et al., 2008).[2] The *Manifesto* is a document motivating and declaring an organised characterisation of the Digital Library field and setting an agenda leading to a foundational theory for Digital Libraries. This characterization captures Digital Libraries in terms of six orthogonal yet interrelated core domains, i.e., *content*, *user*, *functionality*, *policy*, *quality* and *architecture*. Moreover, it introduces three notions of 'system' having a key role in this domain: *Digital Library*, *Digital Library System* and *Digital Library Management System*, with the goal to clarify the distinguishing features that are perceived while using, operating and developing 'digital libraries'. The *Manifesto* also introduces three actors playing a key role in this domain, i.e., *DL End-users*, *DL Managers* and *DL Software Developers*, highlighting their link with one of the above systems and discusses how modern Librarians might be requested to assume one or more of such roles. Finally, it describes a development framework that from an abstract conceptualisation of the digital library domain leads to the implementation of concrete systems via different artefacts. Each artefact specifically shapes aspects captured by the previous one. The *Digital Library Reference Model* is the abstract conceptualisation of the domain. It captures the main concepts, axioms and relationships needed to appropriately

[1] www.delos.info

[2] A revised version of these two documents is contained in the DL.org D3.2b project deliverable Candela, L., Athanasopoulos, G., Castelli, D., El Raheb, K., Innocenti, P., Ioannidis, Y., et al. (2011). *The Digital Library Reference Model.* D3.2b DL.org Project Deliverable..

represent the various aspects characterizing the Digital Library universe independently of specific standards, technologies, implementations, or other concrete details. The envisaged artefacts are (*i*) the *Reference Architecture*, which indicates abstract solutions implementing the concepts and relationships identified in the Reference Model; (*ii*) the *Concrete Architecture*, which enriches the Reference Architecture with concrete standards and specifications; and (*iii*) the *Implementation*, which realises the Concrete Architecture in terms of software systems.

Starting with the DELOS Digital Library Reference Model as its conceptual framework, the EU funded project DL.org – Digital Library Interoperability, Best Practices and Modeling Foundations – networked an outstanding group of Digital Library leading researchers and practitioners to investigate and address one of the most challenging issues affecting nowadays Digital Libraries: interoperability.

1.1 Interoperability levels and digital libraries

Interoperability is among the most critical issues to be faced when building systems as "collections" of independently developed constituents (systems on its own) that should co-operate and rely on each other to accomplish larger tasks. There is no single interoperability solution or approach that is generic and powerful enough to serve all the needs of digital library organisations and digital library systems.

Actually, there is no single definition of interoperability which is accepted in the Digital Library community or by other communities facing this kind of problem.

But, as it has been pointed out, '*while full interoperability may have a "plug and play" flavour (connect it and it works), interoperation can be thought about in terms of different levels of technical and conceptual agreement, such as agreements at syntactic, protocol levels, or conceptual and semantic modeling levels, or overall process level. Even though agreement at conceptual levels may not provide "plug and play", it can greatly facilitate the configuration of information systems to make components work together*' (Gridwise Architecture Council, 2005).

The "Digital Agenda for Europe" (European Commission, 2010), one of the seven flagship initiatives of the Europe 2020 Strategy, has recently recognised the following facts on interoperability: (*i*) the lack of interoperability is among the most significant obstacles undermining the usage of Information and Communication Technologies (ICT) and (*ii*) interoperability is much more than the exchange of data between ICT systems but it includes the ability of disparate organisations to work together. In particular, the European Commission adopts the *European Interoperability Framework* (EIF) (IDABC, 2004) which defines interoperability as follows "*Interoperability is the ability of disparate and diverse organisations to interact towards mutually beneficial and agreed common goals, involving the sharing of information and knowledge between the organizations via the business processes they support, by means of the exchange of data between their respective information and communication technology (ICT) systems.*".

The DL.org project is addressing the multiple digital library interoperability levels, along the classification of the *European Interoperability Framework* (EIF):

- ***Organisational interoperability*** is concerned with defining business goals, modelling business processes and bringing about the collaboration of digital library (and their underlying systems) institutions that wish to exchange resources[3] and may

[3] According to the Reference Model (Candela, L., Athanasopoulos, G., Castelli, D., El Raheb, K., Innocenti, P., Ioannidis, Y., et al. (2011). *The Digital Library Reference Model.* D3.2b DL.org Project

have different internal structures and processes. Moreover, organisational interoperability aims at addressing the requirements of the user community by making resources available, easily identifiable, accessible and user-oriented.

- ***Semantic interoperability*** is concerned with ensuring that the precise meaning of exchanged digital library resources is understandable by any other digital library "system" that was not initially developed to deal with it. Semantic interoperability enables systems to combine received resources with other resources and to process (exploit) it in a meaningful manner;
- ***Technical interoperability*** is concerned with the technical issues of linking computer systems and services implementing the digital libraries and their resources.

At the organisational level, interoperability is a property referring to the ability of diverse organisations to work together. Today organisational interoperability is considered a key step to move from isolated digital archives and digital libraries towards a common information space that allow users to browse through different resources within a single integrated environment (Fox, Akscyn, Furuta, & Leggett, 1995; Borgman, 2000; Miller, 2000; Ross S. , 2008; Lagoze C. , 2010).

Organisation interoperability for digital libraries is a challenging and almost uncharted research area. Some studies have been addressing organisational interoperability in fields as diverse as are engineering, military defence, GIS, data grids, open source software, public administration, e-learning – e.g., (Bishr, 1998; Clark & Jones, 1999; Tolk, 2003; Tolk & Muguira, 2003; IDABC, 2004; Gridwise Architecture Council, 2005; Assche, 2006; Tolk, Diallo, & Turnitsa, 2007). In the digital library domain there are some activities, e.g., (Dekkers, 2007; Bygstad, Ghinea, & Klaebo, 2008), some of them related to addressing digital preservation from a holistic point of view (Innocenti, Ross, Maceciuvite, Wilson, Ludwig, & Pempe, 2009).

As for semantic and technical levels, Wegner (Wegner, 1996) defines interoperability as "*the ability of two or more software components to cooperate despite differences in language, interface, and execution platform. It is a scalable form of reusability, being concerned with the reuse of server resources by clients whose accessing mechanisms may be plug-incompatible with sockets of the server*". He also identifies in *interface standardization* and *interface bridging* two of the major mechanisms for interoperation. Heiler (Heiler, 1995) defines interoperability as "*the ability to exchange services and data with one another. It is based on agreements between requesters and providers on, for example, message passing protocols, procedure names, error codes, and argument types*". He also defines *semantic interoperability* as ensuring "*that these exchanges make sense – that the requester and the provider have a common understanding of the 'meanings' of the requested services and data. Semantic interoperability is based on agreements on, for example, algorithms for computing requested values, the expected side effects of a requested procedure, or the source or accuracy of requested data elements*". Park and Ram (Park & Ram, 2004) define syntactic interoperability as "*the knowledge-level interoperability that provides cooperating businesses with the ability to bridge semantic conflicts arising from differences in implicit meanings, perspectives, and assumptions, thus creating a semantically compatible information environment based on the agreed concepts between different business entities*". They also

Deliverable.) a Resource is any entity managed in the Digital Library universe. Instances of the concept of *Resource* are *Information Objects* in all their forms (e.g., documents, images, videos, multimedia compound objects, annotations and metadata packets, streams, databases, collections, queries and their result sets), *Actors (*both humans and inanimate entities), *Functions*, *Policies*, *Quality Parameters* and *Architectural Components*.

define semantic interoperability as "*the application-level interoperability that allows multiple software components to cooperate even though their implementation languages, interfaces, and execution platforms are different*" (Ram, Park, & Lee, 1999). In addition to that they state that emerging[4] standards, such as XML and Web Services based on SOAP (Simple Object Access Protocol), UDDI (Universal, Description, Discovery, and Integration), and WSDL (Web Service Description Language), might resolve many application-level interoperability problems.

As recognized by Paepcke et al. (Paepcke, Chang, Winograd, & García-Molina, 1998) more than ten years ago, over the years systems designers have developed different approaches and solutions to achieve interoperability. They have put in place a pragmatic approach and started to implement solutions blending into each other by combining various ways of dealing with the issues including *standards* and *mediators*. Too often these remain confined to the systems they have been designed for and lead to '*from-scratch*' development and duplication of effort whenever similar interoperability scenarios occur in different contexts.

The aim of this document is to provide its readers with an organised framework to capture common interoperability issues and related solutions. The document collects, documents and assesses a portfolio of best practices and pattern solutions to common issues faced when developing interoperable Digital Library systems.

1.2 Overview of this document

The remainder of the document is organised as follows. Section 2 describes a model/framework that has been conceived to characterise interoperability scenarios and solutions. Section 3 documents a list of approaches, best practices and solutions that proved to be effective to resolve well identified interoperability issues. Section 4 discusses a number of common and challenging interoperability scenarios faced when building digital libraries and the concrete approaches put in place to resolve them. Finally, Section 5 concludes the document by summarising its content and reporting closing remarks.

Four appendixes complete the document. Appendix A includes a glossary of terms related to interoperability and digital libraries. Appendix B is an index of the various interoperability solutions discussed in the document for simplifying their discovery. Appendix C reports some of the comments that have been raised during the RFC period. Appendix D includes acknowledgments.

[4] The identified standards were emerging at the time they prepared the paper.

2 Digital Library Interoperability Model / Framework

One of the main difficulties affecting the interoperability domain is the lack of a common model that can be used to characterise – in a systematic way – the problem facets as well as the existing and forthcoming solutions and approaches. In this section, it is presented the interoperability model underlying this Digital Library Technology and Methodology Cookbook. Interoperability approaches, methodologies, best practices and solutions reported in Section 3 are described with this model as blueprint.

2.1 Digital Library Interoperability Characterisation

The IEEE Glossary defines interoperability as "*the ability of two or more systems or components to exchange information and to use the information that has been exchanged*" (Geraci, 1991). This definition highlights that to achieve interoperability between two entities (provider, consumer) two conditions must be satisfied: *(i)* the two entities must be able to exchange information and *(ii)* the consumer entity must be able to effectively use the exchanged information, i.e., the consumer must be able to perform the tasks it is willing to do by relying on the exchanged information.

By having this definition as a firm starting point, we identify the following three concepts:

- *interoperability scenario*, i.e., the settings where interoperability takes place;
- *interoperability issue*, i.e., a problem hindering an interoperability scenario;
- *interoperability solution*, i.e., an approach aiming at removing an interoperability issue to achieve an interoperability scenario.

An ***interoperability scenario*** occurs whenever the following conditions manifest:

- there are *at least two entities* that have to cooperate in the context of the scenario. One of the entities is playing the role of ***Provider*** while the other one is playing the role of ***Consumer***;
- the cooperation consists in a *Consumer* willing to exploit a certain ***Resource***[5] – owned by the *Provider* – to perform a certain ***Task*** – the work the *Consumer* is willing to do by relying on that third party *Resource*;
- to make the scenario feasible the two entities should be able to *exchange "meaningful" information*. There can be no exchange of information without a ***communication channel*** and a *protocol* regulating the channel functioning, i.e., a medium enabling information exchange and some rules governing its effective use to pass information among entities. There can be no information without some form of ***representation***, i.e., information is "carried by" or "arises from" a representation (Devlin, 1991). The *meaningfulness* of the

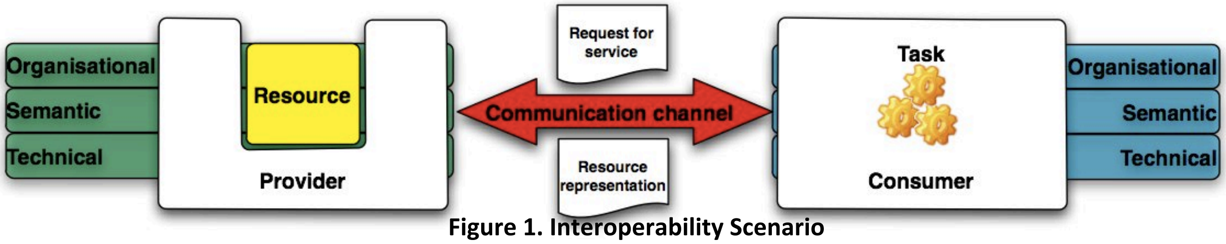

Figure 1. Interoperability Scenario

[5] According to the Reference Model it is an identifiable entity in the *Digital Library* universe. It includes Information Objects, Collections, Resource Sets, Functions, Architectural Components.

information depends on the *Resource* and the *Task* characterising the scenario, i.e., the *Resource* should satisfy the *Consumer* needs and the *Consumer* should acquire the information on the *Resource* that is required to perform the *Task* (***Task preconditions***);

- the operation of each entity, either *Provider* or *Consumer*, depends on ***Organisational***, ***Semantic*** and ***Technical*** aspects. *Organisational aspects* capture characteristics of business goals and processes of the institution operating the entity. Examples of organisational aspects are the type of policies governing Information Objects consumption, the type of functionality to be exposed to Consumers, the quality of service to be supported with respect to a specific functionality. *Semantic aspects* capture characteristics of the meaning of the exchanged digital library resource as well as of the rest of information exchanged through the communication channel. Examples of semantic aspects are the meaning assigned to a certain policy, the meaning assigned to a certain quality parameter, the meaning assigned to a certain value in a metadata record. *Technical aspects* capture characteristics of the technology supporting the operation of the entity as well as of the communication channel and the information exchanged through it. Examples of technical aspects are the Digital Library Management Systems (DLMS) used to implement the Digital Library, the protocol used to expose a certain function, the encoding format of an Information Object. It is important to notice that these three levels influence each other in a top-down fashion, i.e., organisational aspects set the scene of the entire domain characterising its scope and its overall functioning, semantic aspects define the meaning of the entities involved in the domain according to the organisational aspects, technical aspects have to put in place / implement the organisational and semantic aspects.

Note on Provider-Consumer model

The above characterisation of interoperability scenarios in terms of bilateral interaction(s) between a *Provider* and a *Consumer* need to be analysed and motivated to not cause misunderstanding and sceptical reactions on the effectiveness of the proposed model. This note is dedicated to this and to advocate on the need and effectiveness of a simple model as the one presented above.

Concrete interoperability scenarios are complex problems that fall very quickly in settings involving multiple “actors” and “resources”. Solutions and approaches aiming at resolving such a kind of problem are complex themselves because they have to accommodate multiple heterogeneities. These “complex” problems and solutions can be actually seen as “compound” problems and solutions. Thus a “*divide and conquer*”-like approach will help in identifying sub-problems a complex interoperability problem consists of until the identified sub-problems become “simple” enough to be solved directly. The solutions to sub-problems are then combined to give solution to the original interoperability problem. The above framework is mainly intended for capturing the “simple problems”, actually to capture the minimal heterogeneity settings an existing solution removes. It is minimal as to capture the simplest interaction among multiple entities because (*a*) the number of involved entities is 2, (*b*) the subject of the interaction is the resource and (*c*) it subsumes a directional flow, i.e., the resource should conceptually flow from the provider to the consumer. Because of this, it is suitable for capturing with an appropriate level of detail the interoperability issues an existing solution resolves. The integration of multiple solutions toward the definition of a compound solution capable to resolve compound problems is out of the scope of this simple framework. However, the definition of such a compound solution results to be simplified thanks to the description of "simple" solutions via the proposed schema since it provides for detailed descriptions highlighting the distinguishing features of the described approach.

The simple provider-consumer schema is also that underlying the OAI-PMH protocol (cf. Section 3.1.1.1), i.e., one of the most famous interoperability solutions in building digital library services by aggregating "content" from multiple repositories. In describing the interoperability problem and the related solution, the multiplicity in terms of the involved repositories become very quickly a secondary aspect to take care of, thus the "N service providers – M data providers" scenario is actually managed by identifying a solution involving "1 service provider – 1 data provider".

Note on Organisational, Semantic and Technical aspects of Interoperability

The Organisational, Semantic and Technical aspects envisaged in the interoperability scenario characterisation represent a useful mechanism to explicitly report details that usually are overlooked, hidden or mixed each other. Any working interoperability scenario accommodate the need of organisational, semantic and technical aspects, none of them exist otherwise it is the case of an interoperability issue that needs a solution. To be an effective and comprehensive characterisation of the discussed items for the Digital Library community in the large, it is crucial that interoperability scenarios, issues and solutions described in this Cookbook contain the entire bunch of information resulting from the analysis of all the three aspects, i.e. organisational, semantic and technical.

Note on Interoperability Levels

Fully fledged interoperability subsumes *exchangeability*, *compatibility* and *usability*.

Exchangeability is the capability of the two entities involved in an interoperability scenario to actually exchange information about the target Resource.

Compatibility is the capability of the two entities involved in an interoperability scenario to acquire information about the target Resource that is 'logically consistent'.

Usability is the capability of the two entities involved in an interoperability scenario to actually use the information that is acquired about the Resource for a certain purpose (the Task).

All of these three aspects are captured by the previously discussed framework. Exchangeability is guaranteed by the communication channel. Compatibility and Usability deals with the three aspects (Organisational, Semantic, and Technical) that have been discussed and represents the pre-requisite for the Consumer to perform the Task.

An ***interoperability issue*** occurs whenever the *Task preconditions* are not satisfied. Task preconditions are not satisfied whenever *Consumers'* expectations about the *Provider Resource* in the context of the *Task* to be performed are not met by the settings of the scenario, i.e., the technical, semantic and/or organisational aspects characterising the *Provider* and the *Consumer* regarding the *Resource* and the *Task* are not compatible. Exemplars of interoperability issues include: the format used by the *Provider* to represent an Information Object differs from the format expected by the *Consumer* to support a processing activity; the interface through which the Information Object access function is supported by the Provider differs from the one the *Consumer* is expected to use for content fetching; the semantic of the search function implemented by the *Provider* is different from the semantic the *Consumer* aims at relying on to support a cross system search; the Policy governing Information Object consumption supported by the *Provider* is different from the Policy expected by the *Consumer*.

An ***interoperability solution*** is an approach reconciling the differences captured by an interoperability issue. It is based on a generic ***transformation function*** that conceptually acts at any of the levels characterising *Provider* and *Consumer* interaction – organisational, semantic and technical – to make *Provider* characteristics and *Consumer* needs uniform. Such transformation function may act on Provider characteristics or on Consumer needs as well as on both. Exemplars of interoperability solutions include: the transformation and exposure of metadata objects through the harvesting protocol and format expected by the Consumer, the implementation of a search client based on a search interface specification implemented by the Provider, the implementation of policies client-side and server-side to guarantee the agreed quality of service on a distributed search operation.

2.2 Digital Library Interoperability Patterns

All of the heterogeneous interoperability scenarios and related issues existing in the Digital Library domain can be resolved by relying on two classes of solutions independently of their distinguishing characteristics: ‘*Agreement-based’* approaches and ‘*Mediator-based’* approaches. In practice, interoperability scenarios and issues are complex and require the combination of multiple solutions to be resolved. Even in this case, the constituent solutions are either agreement-based or mediator-based. In some cases agreement-based and mediator-based approaches blend into each other, e.g., a mediator-service is actually implementing part of its mediation function according to the agreement settings and rules.

2.2.1 Agreement-based Approaches

Agreement-based approaches are the traditional way to achieve interoperability, i.e., agreeing on a set of principles that achieves a limited amount of homogeneity among heterogeneous entities is one of the most effective approaches to reach interoperability. Standards belong to this category and the value of standards is clearly demonstrable. The major drawbacks of these solutions reside in the fact that standards and agreements are challenging to agree between different organisations and digital libraries. They often end up being complex combinations of features reflecting the interests of many disparate parties. Moreover, by nature they infringe autonomy of the entities adopting them.

In the rest of this Cookbook we will include in this kind of solution both de facto and de jure standards.

The use of standard(s) is a first step to achieve ‘digital library’ interoperability. Standards are codified rules and guidelines for the creation, description, and management of digital resources. The critical importance of standards is widely recognized, as there is considerable movement to develop specifications to communicate between ‘digital library’ systems.

Standards provide the common medium, serving as the ‘glue’ for ‘digital library’ systems. They offer the following benefits (The Standards Committee of the National Defense Industry Association (NDIA) Robotics Division, 2007):

- *reduce life cycle costs* – the cost to develop, integrate, and support systems is reduced by eliminating custom implementations;
- *reduce development and integration time* – common communications prevent the reinvention of the wheel and allow speed integration since proven technology is being employed;
- *provide a framework for technology insertion* – as new technologies are created, those technologies can be easily integrated with minor to no modification to existing systems.

While the ability to communicate between systems is a pre-requisite for interoperability, it is also necessary to have common ‘dialects’ by

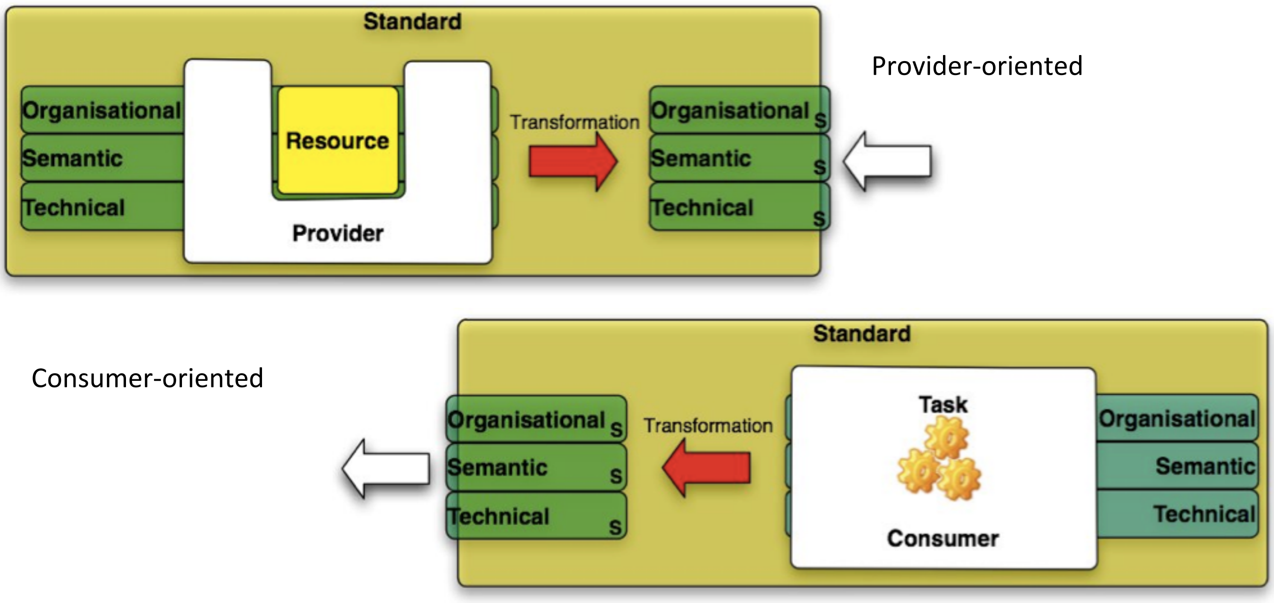

Figure 2. Provider-oriented and Consumer-oriented Agreement-based Approaches

which to share actual information. Some existing standards that are used in the direction of supporting interoperability are listed below.

XML – the eXtensible Markup Language (XML) was developed by the World Wide Web Consortium (W3C) to provide a common language for developing systems and communicating between them. It should be noted that while XML has been implemented in many systems, there is no agreed vocabulary (schema) between vendors. This effectively makes the storage of content proprietary and limits the value of XML in achieving interoperability.

Web Services – Web services is a name given to a collection of specifications for communication between systems (as well as information storage and retrieval) using XML and web technologies. Development in this area is being conducted by the W3C and by many proprietary software companies. Specifications such as SOAP, WSDL, and UDDI form the core of web services, although there are too many other specifications to list here;

Metadata Standards – The goal of metadata consistency has been promoted by the Dublin Core Metadata Initiative (DCMI), which established a base set of metadata elements for all content. This has been implemented widely and has been included as part of the core HTML standard.

Taxonomies and Ontologies – There are a number of active standards related to the structuring and classification of content, including: Resource Description Framework (RDF); Topic maps (XTM); eXchangeable Faceted Metadata Language (XFML); Outline Markup Language (OML); Web Ontology Language (OWL). These provide a range of ways to structure information and are valuable tools for interchange of information between systems.

Unfortunately, there is still a lack of consensus on standards in the Digital Library area and, as Gill and Miller point out (Gill & Miller, 2002), there is still a tendency to either develop completely new standards frameworks from scratch, or to adopt a 'mix and match' approach, using portions from existing domains, and 'adapting' them for specific applications. Although local or adapted standards are certainly better than no standards, this approach can significantly diminish the value of a digital resource by limiting its interoperability with the wider networked world. There is a massive duplication of effort taking place in the realm of standards for digital resources and the sub-optimal nature of this situation is obvious to anyone involved in developing these frameworks.

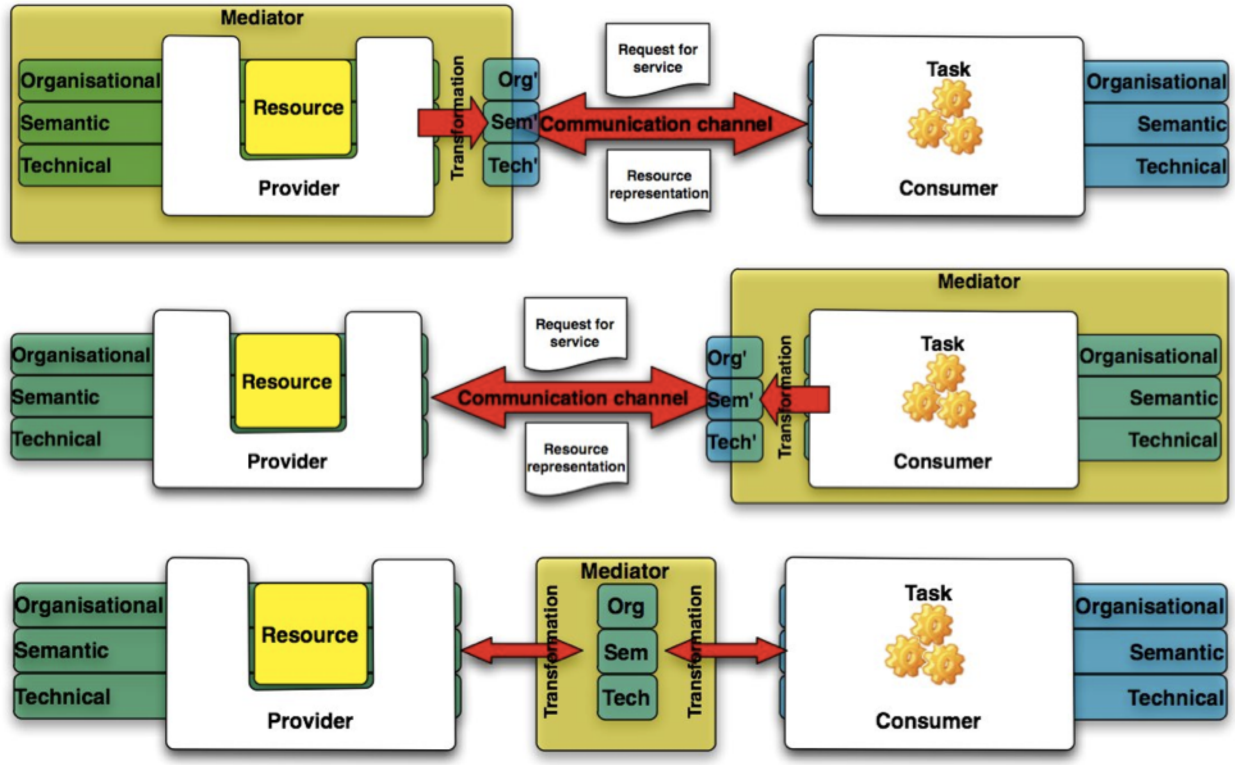

Figure 3. Provider-side, Consumer-side and Third-party Mediator Approaches

As depicted in **Figure 2**, this approach can be implemented *'Provider side'* or *'Consumer side'*. However, to reach interoperability via this approach both Provider and Consumer have to rely on it.

If it is implemented Provider side, it is the Provider entity that makes available its Resource by implementing the guidelines and requirements imposed by the agreement. As a consequence of this, the Provider is willing to serve the needs of any Consumer that relies on such a standard / agreement to exploit a third party resource.

If it is implemented Consumer side, it is the Consumer entity that decides to implement the guidelines and requirements imposed by the standard. As a consequence of this, the Consumer is potentially capable to interoperate with any Provider supporting this standard/agreement.

2.2.2 Mediator-based Approaches

Mediators-based approaches have been proposed to resolve scenarios where there is the need to guarantee an high level of autonomy among the partaking entities. These approaches consist in isolating the interoperability machinery and implementing it in components specifically conceived to link the entities partaking to the scenario. These solutions have been initially conceived in the Information Systems domain (Wiederhold & Genesereth, 1997) and are nowadays used in many cases and realised in many ways (cf. Section 3.6.3).

The most important part of such kind of approaches is represented by the 'mediation function', i.e., the interoperability machinery they implement. Primary functions are transformation of data formats and interaction modes. In the majority of cases, developing a mediation function is very demanding and time consuming (e.g., in the case of non collaborative scenarios, it is the developer of the mediation function that should take care of acquiring the knowledge needed to link Provider and Consumer and implement it) while in others it might be semi-automatic (e.g., in the case of collaborative scenarios, the entities involved expose data characterising them according to certain rules and the developer of the mediation function might rely on these characterisations to link them).

With respect to Standards, Mediators are strong in supporting the criteria of autonomy. However, their effectiveness depends from the dynamicity of the parties they are going to mediate, i.e., every time changes occur in the interacting entities there is the need for changes in the interoperability machinery implemented by the mediator.

As depicted in Figure 3, although the interoperability machinery is implemented by the mediator component, three possible configurations are possible, i.e., the mediator can be Provider-side, Consumer-side or Third-party.

In the case of Provider-side, the interoperability machinery is conceptually close to the Provider, thus it is the Provider that adapts its behaviour and the Resource it offers to the Consumer characteristics. This kind of setting is particularly demanding for the Provider. In order to enable interoperability, the Provider has to revise its behaviour whenever a new Consumer arrives or an existing Consumer changes.

In the case of Consumer-side, the interoperability machinery is conceptually close to the Consumer, thus it is the Consumer that adapts its behaviour and the Tasks it is willing to perform on the characteristics of the Provider(s). This kind of setting is the expected one[6], however it is a costly approach if considered in the large scale because of the potential replication of interoperability machinery implementation effort for every Consumer.

In the case of Third-party, the interoperability machinery is hosted by another entity that provides both Provider and Consumer with a 'linking service', i.e., the components take care of mapping the Provider and Consumer models. This kind of approach is the one that potentially guarantees the minimal effort in case of *n* Provider – *m* Consumers since it supports the sharing of part of the interoperability machinery.

2.2.3 Blending Approaches

The two approaches below are not mutually exclusive, they can be combined each other in concrete interoperability scenarios. The need to combine them arises because of the peculiarities that each scenario or partaking entity has. Thus it may happen that agreements or standards are not sufficient to satisfy the interoperability needs and they have to be complemented with specific interoperability machinery implemented by a mediator or that a mediator relies on one or more standards to regulate the interaction with either the Provider or the Consumer.

2.3 The Interoperability Model in Action

Each interoperability solution is described as follows:

- ***Overview***: a description of the context of the proposed item including a characterisation in terms of the Interoperability Model / Framework and

[6] It is quite straightforward to expect that the Consumer adapts to the characteristics of the Provider in order to reach its own goal

providing the reader with pointers to extensive descriptions of it;

- ***Requirements***: a description of which settings for *Organisational*, *Semantic* and/or *Technical* aspects should occur in order to make it possible to use the solution;
- ***Results***: a description of the changes resulting from the exploitation of the solution in *Organisational*, *Semantic* and/or *Technical* aspects;
- ***Implementation guidelines***: a description of how the solution has to be implemented;
- ***Assessment***: an evaluation of the quality of the proposed approach including an estimation of its implementation cost and effectiveness.

3 Organisational, semantic and technical interoperability: Best practices and solutions

This section represents the central part of this document since it presents an organised list of approaches, best practices and solutions that proved to be effective to resolve well identified interoperability issues. The interoperability solutions discussed are organised according to the Reference Model domains to which the Resource they refer to belongs, i.e., content-oriented (cf. Section 3.1), user-oriented (cf. Section 3.2), functionality-oriented (cf. Section 3.3), policy-oriented (cf. Section 3.4), quality-oriented (cf. Section 3.5), and architecture-oriented (cf. Section 3.6) solutions. In addition to domain oriented solutions, there are some involving concepts that are cross-domain and are gaining a lot of importance in the digital library domain like provenance (cf. Section 3.7). All these solutions are documented by relying on the interoperability framework discussed in Section 2.

3.1 Content Domain Interoperability Best practices and Solutions

Content Domain Interoperability is the problem arising whenever two or more Digital Library "systems" are willing to interoperate by exploiting each other's content resources. In the remainder of this section the following content-oriented interoperability cases are discussed and best practices and solutions for each of them are given: Information Object Description Publishing/Presentation (cf. Section 3.1.1), i.e., approaches dedicated to expose a characterisation of a Provider's Information Object to allow Consumers to realise services by relying on such characterisation; Standards for Information Objects / Metadata (cf. Section 3.1.2), i.e., agreement oriented approaches dedicated to reach a common understanding on Information Object characterisations; Application Profiles (cf. Section 3.1.3), i.e., agreement oriented approaches dedicated to reach a common understanding on schemas for Information Object characterisation; Metadata Mapping / Crosswalks (cf. Section 3.1.4), i.e., approaches dedicated to mediate among different Information Object characterisations; Information Object (Resource) Identifiers (cf. Section 3.1.5), i.e., approaches dedicated to reach a common understanding on tokens allowing different resources to be distinguished.

3.1.1 Information Object Description Publishing/Presentation

Publishing systems are designed to expose metadata of information objects, including compound information objects, so that they can be shared by other systems. Solutions to shareability of information object descriptions among systems have been achieved by the use of protocols or best practices.

Concrete exemplars of this kind of interoperability solution are: *OAI-PMH* (cf. Section 3.1.1.1) – a lightweight protocol for metadata harvesting; *OAI-ORE* (cf. Section 3.1.1.2) – an approach for describing and publishing compound objects in terms of Web resources; *Linked Data* (cf. Section 3.1.1.3) – a set of best practices for publishing and connecting structured data on the Web; *Open Data Protocol* (cf. Section 3.1.1.4) – a data oriented web-based protocol.

3.1.1.1 OAI-PMH

The Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) (Open Archives Initiative, 2002; Lagoze & Van de Sompel, 2001) provides an application-independent interoperability framework for metadata sharing. There are two kinds of actors involved in the framework: *Data Providers* and *Service Providers*. A Data Provider manages a metadata repository and implements the OAI-PMH as a means to expose metadata to harvesters. A harvester is a client application operated by a *Service Provider* to issue OAI-PMH requests to a repository managed by a *Data Provider*.

According to the interoperability framework (cf. Section 2):

- the ***Provider*** is the *Data Provider* while the ***Consumer*** is the *Service Provider*;
- the ***Resource*** the two entities are willing to share is any kind of *metadata record* referring to a repository item and obeying to a metadata formats. The same repository item might be exposed through multiple metadata records of different formats. All the repository items must be exposed via the Dublin Core metadata format;
- the ***Task*** is the service the Service Provider is planning to support. The task poses requirements in terms of the metadata record that has to be exposed, however this is beyond the solution scope, i.e., the solution is *open* with respect to metadata records that can be exchanged. Typical services are cross-repository tasks including search and browse facilities;
- the solution belongs to the **agreement-based approaches** (cf. Section 2.2.1).

Requirements

From the ***Organisational*** point of view, the *Provider* agrees to expose the metadata records of its items in the Dublin Core format and, possibly, in a number of other selected formats. Moreover, it agrees to expose these records via a service residing in a known location that is commonly known as the "base URL". The *Consumer* agrees to acquire metadata records of *Provider's* items by interacting with a service hosted at a known location, i.e., the base URL.

From the ***Semantic*** point of view, the *Provider* and the *Consumer* should share a common understanding on the notions of *repository item*, *metadata record,* and *metadata format*. In particular, the semantic of the metadata format should be shared to reach an effective exchange of the metadata records. This can be achieved by complementing the OAI-PMH solution with others approaches either agreement-based (e.g., shared metadata formats – cf. Section 3.1.2 – and application profiles – cf. Section 3.1.3) or mediator-based (e.g., metadata mappings – cf. Section 3.1.4).

From the ***Technical*** point of view, the *Provider* and the *Consumer* rely on a communication channel based on HTTP and XML.

Results

From the ***Organisational*** point of view, the OAI-PMH approach guarantees that the *Provider* exposes metadata records of its items and other information characterising its service (e.g., the metadata formats supported) to any client sending proper requests. From the *Consumer* perspective, the OAI-PMH approach guarantees that the Consumer can acquire metadata records and other information characterising the service from any Provider implementing it. However, this solution subsumes a sort of service level agreement, i.e., a *Provider* should serve the well defined set of incoming requests envisaged by the OAI-PMH protocol.

From the ***Semantic*** point of view, the OAI-PMH approach guarantees that the *Provider* and the *Consumer* share a common understanding of the model subsumed by the protocol, i.e., the notion of item, the notion of metadata record, and the notion of metadata format. In addition to that, the approach guarantees that the *Provider* and the *Consumer* share a common way to publish/retrieve (i) information on the Provider service (the 'Identify' verb); (ii) the metadata formats made available (the 'ListMetadataFormats' verb); (iii) the sets (groups of items) the *Provider* is offering (the 'ListSets' verb); (iv) the records a *Provider* is offering (the 'ListRecords' verb); (v) the identifiers of the records a *Provider* is offering (the 'ListIdentifiers' verb); (vi) a single metadata record from a *Provider* (the 'GetRecord' verb). Moreover, the solution guarantees that every item is represented through a Metadata Record obeying to the Dublin Core and identified via the 'oai_dc' metadata prefix. The solution does not provide for repository item fetching and metadata format. It is based on metadata format specification advertisement, i.e., every metadata record should declare the format it complies with.

From the ***Technical*** point of view, the *Provider* exposes metadata records and service-related information (*e.g.,* the available metadata formats) through a well defined set of HTTP requests and responses (*i.e.,* the six protocol requests and responses). The *Consumer* can issue the well defined set of HTTP requests and responses to gather the expected information (namely the metadata records) from any OAI-PMH *Provider*. Metadata records are exposed/gathered through their XML serialisation that complies with a metadata format.

Implementation guidelines

The OAI-PMH protocol has to be implemented in both *Provider* and *Consumer*. The *Provider* has to support the requests envisaged by the protocol while the *Consumer* has to issue proper requests and consume the responses. A set of implementation guidelines[7] has been produced ranging from guidelines for minimal implementations to customisation and openness (optional containers), datestamps and granularity, resumption tokens, and error handling. A lot of tools[8] have been implemented and made available by communities like OAI-PMH harvesters and OAI-PMH publishers.

For what is concerned with the metadata records associated to repository items, they should either pre-exist in all the formats that the Provider is willing to expose (one of them must be the Dublin Core) or be produced via mappings (cf. Section 3.1.4). Moreover, the metadata formats might pre-exist or be defined for the scope of a specific application domain. In the second case, a best practice is that of application profiles (cf. Section 3.1.3).

Assessment

OAI-PMH has been conceived to be a lightweight solution to interoperability. Because of this, it is probably one of the most famous interoperability approaches used in the Digital Library domain.

For what is concerned with Provider's ***implementation cost***, this essentially corresponds to the implementation cost of the six verbs envisaged by the protocol and the production of metadata compliant with the Dublin Core format. In addition to that, there might be the cost needed to transform/produce the metadata records in other formats the Provider is willing to support. *Consumer's implementation cost* essentially corresponds to the implementation cost of the six verbs and the consumption of the gathered information. Another implementation cost might be related with the solutions put in place to reach interoperability at the level of metadata formats, i.e., the Consumer should consume metadata records with the same understanding exploited by the Provider to produce them.

For what is concerned with ***effectiveness***, being an agreement based approach it is by definition highly effective for what is captured by the agreement/standard. For what is concerned with aspects that go beyond the standard (e.g., the metadata format) the effectiveness depends on solutions and approaches that are put in place to resolve interoperability with respect to these aspects.

3.1.1.2 OAI-ORE

The *Open Archives Initiative Object Reuse and Exchange* (OAI-ORE) (Open Archives Initiative, 2008; Lagoze & Van de Sompel, Open Archives Initiative Object Reuse and Exchange User Guide - Primer, 2008) defines standards for the description and exchange of aggregations of Web resources, i.e., compound objects consisting of a set of related resources. These aggregations may combine distributed resources with multiple media types including text, images, data, and video. The goal of the standards is to expose the rich content in these

[7] http://www.openarchives.org/OAI/2.0/guidelines.htm

[8] http://www.openarchives.org/pmh/tools/tools.php

aggregations to applications that support authoring, deposit, exchange, visualization, reuse, and preservation.

According to the interoperability framework (cf. Section 2):

- a ***Provider*** exposes compound information objects on the Web (i.e., by relying on Web architecture and services); such objects can be consumed by any ***Consumer*** that is able to comply with the object representation and access facilities envisaged by the OAI-ORE specification;
- the ***Resource*** the two entities are willing to share is any kind of resource that might be represented via a compound information object;
- the ***Task*** is the service the Consumer is planning to support; such a service implements a functionality that requires to be aware of the information objects (and their constituents) exposed by the Provider. Typical services are those that can be built atop one or more repositories and their content including object discovery and object manipulation;
- the solution belongs to the **agreement-based approaches** (cf. Section 2.2.1).

Currently, OAI-ORE manifests in two concrete solutions that share the basic technical protocol, i.e., HTTP, but differ in the representation models and in the semantic tools, which may be *ATOM-based* or *RDF-based*.

Requirements

From the ***Organisational*** point of view, the *Provider* agrees to expose compound information objects on the Web in conformity with the ORE data model, the Web architecture, and services; the *Consumer* agrees to comply with such object representation as well as with the Web architecture and services.

From the ***Semantic*** point of view, the *Provider* and the *Consumer* should share a common understanding of the semantic of the entities forming the ORE model, i.e., *Aggregation*, *Resource Map*, and *Proxies*, and their relationships. Moreover, they should agree on the semantic of the additional vocabularies and terms characterizing the shared model , e.g., Dublin Core and RDF terms, or Atom profiles.

From the ***Technical*** point of view, the *Provider* and the *Consumer* should rely on a communication channel based on HTTP. In particular, each entity involved in the model, such as Aggregation, Resource Map and Proxy, must be identifiable through an HTTP URI. Moreover, *Provider* and *Consumer* should primarily find an agreement on the representation format to be employed to implement the ORE abstract data model, i.e., *Atom-based* or *RDF-based*, as well as on the additional vocabularies to reuse in order to enrich the ORE data model.

Results

From the ***Organisational*** point of view, the ORE approach guarantees that *Provider* and *Consumer* describe and exchange compound resources in conformity with a shared data model and a known communication medium.

From the ***Semantic*** point of view, the ORE solution enables to express the composite nature of an object and to indicate which parts the object is composed of. Moreover, in compliance with the concrete solution adopted and its related representation model (*Atom* or *RDF*), *Provider* and *Consumer* can also share other important properties and metadata belonging to the exchanged resources in addition to their composite nature. Thus, the *Provider* has the possibility to expose meaningful aggregations of Web resources and make them available to multiple *Consumers*, which can properly interpret and consume them. These resources may or may not have a digital representation. In particular, in order to unambiguously refer to a particular aggregation of Web resources, the model introduces a new resource, namely an *Aggregation*. This is an abstract resource that has no representation and indicates a set of related resources that can be treated as a single resource. An Aggregation is described by a *Resource Map*, which is a concrete document that enumerates all the *Aggregated Resources* composing the Aggregation. ORE also introduces an abstract resource, namely a *Proxy*, to indicate an Aggregated Resource in the context of a specific Aggregation.

From the ***Technical*** point of view, a *Resource Map* and the *Aggregation* it describes are identifiable with distinct HTTP URIs. An HTTP redirection can be used by the *Consumer* to obtain the HTTP URI of the Resource Map given the HTTP URI of the corresponding Aggregation. Alternatively, when the support of HTTP redirection is not available, the URI of an Aggregation can be constructed by appending a fragment identifier to the Resource Map URI. Such an identifier should be stripped off before the *Consumer* issues an HTTP request to the *Provider*, so that the request actually regards the Resource Map. A Resource Map is a concrete document, i.e., it has a machine-readable representation that makes the Aggregation description available to clients and agents. All the assertions made by the Resource Map about an Aggregation and its Aggregated Resources must be expressed in the form of triples "subject-predicate-object" that altogether create an RDF graph. Actually, a Resource Map can be expressed in different formats; for example, it currently supports XML serialization in RDF and Atom.

Implementation guidelines

In order to use OAI-ORE, implementers are highly encouraged to use the available tools and libraries to avoid duplication of effort and to reduce errors. In particular, a lot of tools[9] are made available, ranging from validation tools and services to libraries and toolkits for constructing, parsing, manipulating, and serializing OAI-ORE Resource Maps, and for conversion from one format to another.

The ORE abstract data model can be implemented in different serialization formats. In particular, since a Resource Map is an RDF Graph, it can be serialized using any RDF syntax. Currently, a set of guidelines is at one's disposal for the implementation of a Resource Map in RDF/XML[10] and Atom/XML[11], as well as for HTTP implementation[12] and Resource Map discovery[13]. In addition, it is strongly recommended the use of additional terms from existing vocabularies to enrich the ORE data model in order to properly describe a Resource Map, and, thus, the Aggregation and its Aggregated Resources, e.g., by reusing terms from Dublin Core and FOAF.

Assessment

OAI-ORE has been conceived to be a solution to interoperability that strongly relies on Web architecture and its services.

For what is concerned with *Provider's **implementation cost***, it essentially corresponds to the implementation cost of the production of the Aggregation and its Resource Map. In addition to that, there might be the costs needed to transform/produce pieces of the Aggregation the Provider is willing to support. The *Consumer's implementation cost* essentially corresponds to the cost needed to acquire and consume the Aggregations. Another implementation cost might be related with the solutions put in place to reach interoperability at the level of metadata schemas and vocabularies, i.e., the Consumer should consume metadata and other information encoded in the Resource Map with the same understanding exploited by the Provider to produce them.

For what is concerned with ***effectiveness***, being an agreement based approach it is by definition

[9] http://www.openarchives.org/ore/1.0/tools.html

[10] http://www.openarchives.org/ore/1.0/rdfxml.html

[11] http://www.openarchives.org/ore/1.0/atom.html

[12] http://www.openarchives.org/ore/1.0/http.html

[13] http://www.openarchives.org/ore/1.0/discovery.html

highly effective for what is captured by the agreement/standard. For what is concerned with aspects that go beyond the standard (e.g., the vocabularies) the effectiveness depends on the solutions and approaches that are put in place to resolve interoperability with respect to these aspects.

3.1.1.3 Linked Data

Linked Data is a set of best practices for publishing and connecting structured data on the Web (Bizer, Heath, & Berners-Lee, 2009) with the main purpose of allowing people to share structured data on the Web in the same way they can share traditional documents. The following rules/principles are the foundational ones and are known as the Linked Data Principles:

- use URIs (Berners-Lee, Fielding, & Masinter) as names for things;
- use HTTP URIs so that people can look up those names;
- when someone looks up a URI, provide useful information using the standards, namely RDF (Klyne & Carroll) and SPARQL (Prud'hommeaux & Seaborne);
- include links to other URIs, so that they can let additional things be discovered.

According to the interoperability framework (cf. Section 2):

- a ***Provider*** publishes structured data on the Web in compliance with the Linked Data principles; such data can be consumed by any ***Consumer*** that is able to comply with such principles;
- the ***Resource*** the two entities are willing to share is any kind of information about an arbitrary resource, *e.g.,* the description of the resource structure, metadata or contextual relationships with other resources;
- the ***Task*** the Consumer is planning to support is any functionality that requires the availability of information about arbitrary resources, such as structure, metadata, and context information;
- the solution belongs to the **agreement-based approaches** (cf. Section 2.2.1).

The Linking Open Data Project[14], which began in early 2007, is a community project supported by W3C aiming at extending the Web with a data commons by making existing open data sets available in RDF and interlinking them with other data sets.

Requirements

From the ***Organisational*** point of view, the *Provider* agrees to publish structured data and related information on the Web in conformity with the Linked Data principles; the *Consumer* agrees to comply with such principles.

From the ***Semantic*** point of view, *Provider* and *Consumer* should share a common understanding of the semantics of the RDF terms, as well as the meaning of the additional vocabularies, e.g., Dublin Core and FOAF, and ontologies, e.g., RDFS and OWL, they could use to characterize the data descriptions.

From the ***Technical*** point of view, Linked Data principles state that any described resource should be identified by an HTTP URI, so *Provider* and *Consumer* should rely on a communication channel based on HTTP. Moreover, Linked Data principles state that data should be described according to RDF, so *Provider* and *Consumer* should have knowledge of RDF. In addition, they should find an agreement on the additional vocabularies to be used to represent data, e.g., Dublin Core and FOAF, and the ontologies to be used, e.g., OWL and RDFS, as well as the serialization format to be employed in order to implement the RDF descriptions, e.g., RDF/XML.

Results

From the ***Organisational*** point of view, Linked Data guarantees that any *Provider* can publish descriptions of structured data on the Web and provide links to related resources in compliance with a set of well-defined principles. Such

[14] http://esw.w3.org/SweoIG/TaskForces/CommunityProjects/LinkingOpenData

information can be accessed and explored by any Consumer willing to exploit it.

From the ***Semantic*** point of view, Linked Data enables to express useful and meaningful information about any kind of resource. Such information includes descriptions of the resource's structure and metadata, as well as relationships with other resources within the same data source or coming from different data sources.

Information about resources, as well as relationships between different resources, are expressed according to the RDF data model, that allows to describe any resource and define typed links between related resources by merging terms coming from different schema and vocabularies into a single model. In particular, information can be expressed in RDF by using additional terms coming from well-known vocabularies, such as Dublin Core and FOAF, as well as new terms and properties defined by users. In addition, RDF may be combined with ontologies, such as OWL and RDFS, in order to enrich data descriptions with explicit semantics.

Thus, the Web of Data or Semantic Web defined by Linked Data may be seen as a graph composed of meaningful *things* interconnected by *typed* links.

From the ***Technical*** point of view, Linked Data principles state that every described resource must be identifiable with a dereferenceable HTTP URI. URIs of information resources, i.e., resources that have a digital representation, can be de-referenced directly. With regard to non-information resources, HTTP redirection can be used to obtain the HTTP URI of the information resource describing a non-information resource given the HTTP URI of the non-information resource. Alternatively, the URI of the non-information resource can be constructed by appending a fragment identifier to the URI of the related information resource. HTTP content negotiation mechanism can be used to obtain different representations of the same resource description, e.g., HTML or RDF/XML.

Things and their interrelations are described according to the RDF data format. Reuse of terms coming from existing and well-known vocabularies, such as Dublin Core, FOAF and OWL, is strongly recommended to represent information encoded in RDF; however, *Provider* and *Consumer* are allowed to define their own terms. The choice of the serialization format for RDF descriptions, e.g., RDF/XML, Turtle or others, is left to the *Provider* and the *Consumer*.

Implementation guidelines

A lot of guidelines[15] and tutorials are available, ranging from methods for publishing different types of information as Linked Data on the Web to various tools for testing and debugging (Bizer, Cyganiak, & Heath, 2008).

A set of toolkits is made available for clients[16], as well.

Assessment

Even if Linked Data is recognised a as a good practice, it has some drawbacks as discussed in by Mike Bergman[17].

3.1.1.4 Open Data Protocol

The *Open Data Protocol* (OData)[18] is an open Web protocol for data sharing and modification, aiming to unlock and release data from silos of specific applications and formats, and allowing information from different sources, ranging from relational databases and file systems to content management systems, traditional Web sites and more, to be exposed and accessed. A deep commitment to core Web principles allows OData to enable data integration and

[15] http://linkeddata.org/guides-and-tutorials

[16] http://esw.w3.org/TaskForces/CommunityProjects/LinkingOpenData/SemWebClients

[17] http://www.mkbergman.com/902/i-have-yet-to-metadata-i-didnt-like/

[18] http://www.odata.org

interoperability across a broad range of clients, servers, services, and tools.

According to the interoperability framework (cf. Section 2):

- the ***Provider*** is any service that exposes its data in conformity with the OData protocol while the ***Consumer*** is any application that consumes data exposed in conformity with the OData protocol;
- the ***Resource*** the two entities are willing to share is any kind of structured and unstructured data, including associated metadata and available operations;
- the ***Task*** is any operation or set of operations the *Consumer* is planning to execute on the shared resource;
- the solution belongs to the **agreement-based approaches** (cf. Section 2.2.1).

Requirements

From the ***Organisational*** point of view, the Provider agrees to publish its data and related information, such as metadata and supported operations, on the Web in conformity with the OData protocol while the Consumer agrees to access and edit the data exposed by the Provider in compliance with the OData protocol.

From the ***Semantic*** point of view, Provider and Consumer should have a common knowledge of the semantics of the elements forming the abstract data model, i.e., Entity Data Model (EDM), used to define resources and metadata in OData terms, e.g., they should know the meaning underlying the notions of entity type, property, and association, as well as the notions of feed, entry, and link. Moreover, they should know the semantics of the URIs used to identify the resources and metadata exposed, as well as the semantics of the operations allowed by the protocol.

From the ***Technical*** point of view, Provider and Consumer should find an agreement on the representation format to be used in order to describe the exchanged data, e.g., Atom format or JSON format. Moreover, they should rely on a communication channel based on HTTP and they should have a common knowledge of the rules for constructing URIs that will identify the exchanged data and metadata.

Results

From the ***Organisational*** point of view, the OData protocol guarantees that the *Provider* exposes its data on the Web, together with other information characterizing its service (e.g., a service metadata document describing the structure and organization of all the exposed data), in compliance with a well-defined protocol. The exposed data and related information can be accessed and edited by any *Consumer* that is able to comply with such protocol.

From the ***Semantic*** point of view, the OData solution offers *Provider* and *Consumer* a common way to represent resources, metadata, and allowed operations.

An OData *Provider* exposes one or more *feeds*, which are *Collections* of typed *Entries*. Each entry is a structured record consisting of a key and a list of *Properties*, each one of them can have a primitive or complex type. Entries can be organized into a type hierarchy and may have related entries and related feeds through *Links*. In addition to feeds and entries, an OData *Provider* may expose *Service Operations*, namely simple service-specific functions that accept input parameters and return entries or values.

A *Provider* can publish metadata documents to describe itself: a *Service Document*, enumerating all the feeds it exposes (for discovery purpose), and a *Service Metadata Document*, describing the structure and organization of all the exposed resources in terms of an abstract data model, i.e., the EDM. An OData feed is described in the EDM by an *Entity Set*. Each Entry is modelled by an *Entity Type*, properties of an entry correspond to primitive or complex Entity Type properties, and each link between entries is described by a *Navigation Property*, i.e., a property that represents an association between two or more Entity Types.

The OData protocol guarantees that *Provider* and *Consumer* identify resources, as well as retrieve and perform simple operations on such resources, by using well-defined URIs. The OData service interface has a fixed number of operations that have uniform meaning across all resources (retrieve, create, update, and delete). In addition, OData allows servers to expose customized operations (Service Operations).

From the ***Technical*** point of view, the OData protocol offers a uniform service interface that is independent from the data exposed by any individual service. An OData *Provider* exposes all its resources (feeds, entries, properties, links, service documents, and metadata documents) by identifying them with URIs that have to respect well-defined syntax rules. A *Consumer* can access such resources by using URIs and perform operations on them by using standard HTTP requests (GET, POST, PUT, MERGE, DELETE).

OData supports two formats to represent the resources it exposes: the XML-based Atom format and the JSON (Javascript Object Notation) format. HTTP content negotiation can be used by a *Consumer* to indicate its preference for resource representation. An OData Service Metadata Document is formatted according to an XML language for describing models called the Conceptual Schema Definition Language (CSDL).

Implementation guidelines

The Open Data Protocol specification is currently available under the Microsoft Open Specification Promise (OSP)[19] in order to allow anyone, including open source projects, to freely interoperate with OData implementations. OData is designed to be modular, so that any OData implementation can implement only the parts of the OData specification that are required from its target scenario.

The simplest OData service can be implemented as a static file that follows the OData ATOM or JSON payload conventions. For more complex scenarios that go beyond static content, frameworks[20] are available for help in creating OData services.

From a client point of view, all the interactions with an OData service are done using URIs and standard HTTP verbs, thus, any platform with a reasonably complete HTTP stack may be enough for communication. However, many client libraries[21] are available for different platforms and languages, including .NET, Silverlight, PHP, Java, and the iPhone.

Assessment

OData is consistent with the way the Web works – it makes a deep commitment to URIs for resource identification and commits to an HTTP-based uniform interface for interacting with those resources (just like the Web). This should guarantee a new level of data integration and interoperability across a broad range of clients, servers, services, and tools.

3.1.2 Standards for Information Objects / Metadata

Concrete exemplars of this kind of interoperability solution are: Dublin Core (cf. Section 3.1.2.1) – among the most famous metadata element set in the Digital Library domain; the Europeana Data Model (cf. Section 3.1.2.2) – a new proposal for compound objects developed in the context of the Europeana initiative; CERIF (cf. Section 3.1.2.3) – a model for representing Research Information Systems and support their interoperability.

[19] http://www.microsoft.com/interop/osp/default.mspx

[20] http://www.odata.org/developers

[21] http://www.odata.org/developers/odata-sdk

3.1.2.1 Dublin Core

Dublin Core[22] is a metadata standard providing a simple set of elements for the description of any kind of resource, including physical objects like books, digital objects like videos, sounds, images or text files, and compound objects like web pages. Apart from enabling the creation of resources' descriptive records with the aim of enabling effective information retrieval, Dublin Core may be extended and combined with terms coming from other compatible vocabularies for the definition of *application profiles* (cf. Section 3.1.3).

According to the interoperability framework (cf. Section 2):

- a ***Provider*** exposes metadata describing arbitrary resources in conformity with the Dublin Core metadata scheme; such metadata can be exploited by any ***Consumer*** that is able to comply with such metadata scheme;
- the ***Resource*** the two entities are willing to share is any kind of Dublin Core metadata describing an arbitrary resource;
- the ***Task*** is the functionality that any *Consumer* is planning to support by relying on a Dublin Core record; such a functionality requires the availability of information describing a resource, i.e., metadata about a resource;
- the solution belongs to the **agreement-based approaches** (cf. Section 2.2.1).

The semantic of Dublin Core has been established by an international, cross-disciplinary group of professionals ranging from librarianship, computer science and text encoding to the museum community and other related fields of scholarship and practice. The Dublin Core Metadata Initiative (DCMI)[23] is an open organization engaged in the development of interoperable metadata standards supporting a broad range of purposes and business models. The *Dublin Core Metadata Element Set (DCMES)*[24], together with a larger set of metadata vocabulary, namely the *DCMI Metadata Terms*[25], is maintained by the DCMI.

Requirements

From the ***Organisational*** point of view, the *Provider* agrees to expose metadata in conformity with the Dublin Core metadata scheme while the *Consumer* agrees to acquire metadata in compliance with such metadata scheme.

From the ***Semantic*** point of view, *Provider* and *Consumer* should have a common knowledge of the semantic associated with the Dublin Core elements, element refinements, and resource classes. Moreover, they should have a common understanding of the meanings underlying the vocabulary encoding schemes and syntax encoding schemes they could use.

From the ***Technical*** point of view, *Provider* and *Consumer* should agree on the specific element qualifiers, vocabulary encoding schemes, and syntax encoding schemes to be used. An agreement should also be found on the metadata representation format and the communication protocol to be used for exchanging the metadata.

Results

From the ***Organisational*** point of view, the Dublin Core standard guarantees that *Provider* and *Consumer* represent metadata describing arbitrary resources in compliance with a common metadata scheme.

From the ***Semantic*** point of view, *Provider* and *Consumer* have at their disposal a common set of metadata elements associated with semantic that should be universally understood and supported.

The Dublin Core Metadata Element Set (DCMES) comprises 15 basic elements, each of which is

[22] http://dublincore.org

[23] http://dublincore.org/about-us/

[24] http://dublincore.org/documents/dces/

[25] http://dublincore.org/documents/dcmi-terms/

optional and repeatable: contributor, coverage, creator, date, description, format, identifier, language, publisher, relation, rights, source, subject, title, and type.

A large set of additional terms or properties, namely *qualifiers*, has been developed by the DCMI in order to refine the basic elements. A refined element has the same meaning of the corresponding unqualified element, but it has a more specific scope. Element refinement is conceived on the basis of a principle, known as "dumb-down principle", stating that if a client does not understand a specific element refinement it will be able to ignore the qualifier and use the metadata value like the original unqualified element. Thus, qualification is supposed only to refine, and not to extend the semantic of the elements, because eliminating the qualifier still produces a correct and meaningful element.

In addition to element refinements, the DCMI has identified a set of recommended encoding schemes, that have the purpose of improving the interpretation of a term value. These schemes include *controlled vocabularies*, i.e., limited sets of consistently used and carefully defined values, and *formal syntax encoding schemes*. If an encoding scheme is not understood by an application, the value may still be useful to a human reader. A set of resource classes and a type vocabulary have been defined, as well. In particular, formal domains and ranges for properties have been defined in order to specify what kind of resources and values may be assigned to a given DC term, by using a form that explicitly expresses the meanings implicit in natural-language definitions.

Apart from using the recommended qualifiers, *Provider* and *Consumer* are allowed to define and develop their context-specific properties, syntax encoding schemes, and controlled vocabularies.

From the ***Technical*** point of view, it is important that Dublin Core concepts and semantic are designed to be syntax independent. The DCMI provides a rich set of recommended syntax encoding schemes and controlled vocabularies for DC properties, classes and values, as well as an abstract reference model that *Provider* and *Consumer* may use as a guide whether they want to implement Dublin Core or define their own context-specific syntax encoding schemes. The choice of the transmission protocol, as well as the metadata representation format, is left to the *Provider* and *Consumer* specific implementation. Dublin Core currently supports implementations in HTML, XHTML, and RDF/XML formats.

Implementation guidelines

Dublin Core is formally defined by ISO Standard 15836 and NISO Standard Z39.85-2007.

The DCMI makes available a large set of semantic recommendations and user guidelines[26]. Apart from a formal description of the DC concepts, they include guidelines for the creation of application profiles based on Dublin Core[27] and an Abstract Model[28], defining a syntax-independent information model that offers a better understanding of the DC components and constructs to developers of applications supporting Dublin Core metadata and to people interested in developing new syntax encoding guidelines for Dublin Core metadata.

A set of syntax guidelines is proposed as well, including recommendations for expressing Dublin Core metadata in DC-Text format[29], XHTML and HTML[30], XML[31], and RDF[32].

[26] http://dublincore.org/specifications/

[27] http://dublincore.org/documents/profile-guidelines/

[28] http://dublincore.org/documents/abstract-model/

[29] http://dublincore.org/documents/dc-text/

[30] http://dublincore.org/documents/dc-html/

[31] http://dublincore.org/documents/dc-ds-xml/

[32] http://dublincore.org/documents/dc-rdf/

Assessment

Dublin Core is one of the most famous metadata schemes used in the Digital Library domain. It has been conceived to be as simple as possible (15 basic elements only, all of them optional and repeatable). These two reasons are among the factors promoting its diffusions.

3.1.2.2 Europeana Data Model

The *Europeana Data Model* **(EDM)** is a new proposal for structuring objects and metadata coming from Europeana, the European Digital Library, Museum and Archive. EDM is a major improvement on the Europeana Semantic Elements (cf. Section 3.1.3.1), the basic data model with which Europeana was born. Unlike its precursor, EDM does not convert the different metadata standards used by Europeana data providers and aggregators to a common denominator, e.g., Dublin Core-like standard. Instead, it adopts an open, cross-domain Semantic Web-based framework, aiming at providing an anchor to which various models can be attached, in order to support the integration of the various models used in cultural heritage data, retaining the original data while still allowing for semantic interoperability.

According to the interoperability framework (cf. Section 2):

- the ***Provider*** is any data provider willing to expose digital objects and related metadata according to the EDM; such objects can be exploited by any ***Consumer*** that is able to comply with such a model;
- the ***Resource*** the two entities are willing to share is any kind of metadata related to digital content;
- the ***Task*** is the functionality that any *Consumer* is planning to support; such a functionality requires the availability of information about digital objects, such as structure and metadata;
- the solution belongs to the **agreement-based approaches** (cf. Section 2.2.1).

Requirements

From the ***Organisational*** point of view, *Provider* and *Consumer* agree to share digital objects and metadata in accord with EDM and its requirements and principles.

From the ***Semantic*** point of view, the *Provider* and the *Consumer* should share a common understanding of the semantics of the EDM classes and properties, as well as EDM design principles and requirements. A common knowledge of the semantics of existing standards should be shared as well, i.e., Dublin Core, OAI-ORE, and SKOS, as well as the RDF and the RDF Schema.

From the ***Technical*** point of view, it is suggested that all resources are provided with URIs. Moreover, an agreement should be found on the common models to be used in order to enrich data descriptions; Dublin Core (cf. Section 3.1.2.1), SKOS, and OAI-ORE (cf. Section 3.1.1.2) are basic, as well as RDF and RDF Schema.

Results

From the ***Organisational*** point of view, EDM allows any *Provider* to expose objects together with their different digital representations and associated metadata to any *Consumer* willing to exploit them.

From the ***Semantic*** point of view, EDM makes a distinction between the *object* being described, e.g., a painting, a book, a movie, etc, and a set of *digital representations* of such an object which can be accessed through the Web. Similarly, it distinguishes between *object* and *metadata record* describing the object.

EDM follows the typical scheme of application profiles (cf. Section 3.1.3): a set of well-defined classes and properties is introduced together with elements taken from other namespaces, such as OAI-ORE, Dublin Core, RDF, and SKOS, in order to describe objects and associated metadata. Anything in EDM is a Resource as defined in the RDF Schema. EDM classifies such resources as either (a) *Information Resources*, i.e., resources that can have a representation and some realization, or (b) *Non Information*

Resources, e.g., agents, places, events and physical things. An Information Realization is a physical resource that materializes an Information Resource.

EDM provides classes and properties that allow the representation of descriptive metadata for an object following both an object-centric approach, i.e., the described object is directly connected to its features, and an event-centric approach, i.e., the focus is on the various events the described object has been involved in. In addition, EDM allows for advanced modelling by providing properties to express relationships between different objects, such as part-whole links for complex objects as well as derivation and versioning relations.

The OAI-ORE Aggregation is used to organize the data of a Provider by aggregating the resource standing for an object together with one or more resources standing for its digital representations. Descriptive metadata records for an object are attached to ORE Proxies connected with that object. The Europeana Aggregation is a specialization of the general ORE Aggregation and consists of the set of resources related to a single Cultural Heritage Object that collectively represent that object in Europeana.

From the ***Technical*** point of view, a suggestion is given stating that URIs for all objects should be created in order to implement a publication strategy that relies on HTTP services. EDM provides a standard metadata format, i.e., Dublin Core, and a standard vocabulary format, i.e., SKOS, both of which can be specialized. Moreover, OAI-ORE is used for organization of metadata while RDF is used to represent EDM entities and relationships. EDM should be based on the reuse of existing standard models, such as Dublin Core, SKOS, and ORE, but others could be applicable, e.g., provider-customized models.

Implementation guidelines

At the time of writing, EDM is still under development. It will continue to be refined until the end of 2010 and it will be implemented during 2011. Before, during, and after the implementation of EDM, data compliant with ESE will still be accepted, since ESE is compatible with EDM and no data will need to be replaced, although a convertor will be made available for any provider willing to resubmit data in the new model.

Assessment

To be defined.

3.1.2.3 CERIF (the Common European Research Information Format)

The *Common European Research Information Format* (CERIF) is a formal model to setup Research Information Systems and to enable their interoperation. Research Information is information about research entities such as *People*, *Projects*, *Organisations*, *Publications*, *Patents*, *Products*, *Funding*, or *Equipment* and the relationships between them.

The CERIF standard was developed in the late 1980's by the European Union. Since 2002 care and custody of CERIF has been handed by the EC to euroCRIS, a not-for-profit organisation dedicated to the promotion of Current Research Information System (CRIS). CERIF is neutral as to architecture; the data model can be implemented as a relational, object-oriented, RDF/OWL XML database, or as an information retrieval (including Web) system. It was intended for use by CRIS systems to allow them to store and transfer CRIS data among databases and information systems. The current version of the CERIF standard is available through membership of the euroCRIS organisation. Several CERIF compliant CRISs exist in Europe and the standard has been used in the European Union *IST World*.

The purposes of CERIF are the following: (i) to enable storage and interchange of information between CRISs; (ii) to enable information access to CRISs through the Web; (iii) to provide a standard data model, best practices and tools for CRIS developers.

According to the interoperability framework (cf. Section 2):

- the ***Provider*** is any data provider willing to expose digital objects and related metadata according to the CERIF; such objects can be exploited by any ***Consumer*** that is able to comply with such a model;
- the ***Resource*** the two entities are willing to share is any kind of metadata related to digital content;
- the ***Task*** is the functionality that any Consumer is planning to support; such a functionality requires the availability of information about digital objects, such as structure and metadata;
- the solution belongs to the ***agreement-based approaches*** (cf. Section 2.2.1).

Requirements

From the ***Organisational*** point of view, *Provider* and *Consumer* agree to share digital objects and metadata in accord with CERIF and its requirements and principles.

From the ***Semantic*** point of view, the *Provider* and the *Consumer* should share a common understanding of the semantics of the CERIF classes and properties, as well as CERIF design principles and requirements. A common knowledge of the semantics of existing standards should be shared as well, i.e., Dublin Core, OAI-ORE, and SKOS, as well as the RDF and the RDF Schema.

From the ***Technical*** point of view, it is suggested that all resources are provided with URIs. Moreover, an agreement should be found on the common models to be used in order to enrich data descriptions. Dublin Core, SKOS, and OAI-ORE are basic, as well as RDF and RDF Schema.

Results

From the ***Organisational*** point of view, CERIF allows any *Provider* to expose information about research entities, such as People, Projects, Organisations, Publications, Patents, Products, Funding, or Equipment and the relationships between them, to any *Consumer* willing to exploit it.

From the ***Semantic*** point of view, CERIF provides classes and properties that allow the representation of entities in a research information system, structured as *Core Entities, Result Entities, and 2nd Level Entities.* The CERIF Core Entities are ‘Person’, ‘OrganisationUnit’, and ‘Project’. Each core entity recursively links to itself and maintains relationships with other core entities. The Core Entities allow for a representation of scientific actors and their different kinds of interactions. The CERIF Result Entities are ‘ResultPublication’, ‘ResultPatent’, and ‘ResultProduct’. The ‘ResultPublication’ entity like a Core Entity recursively links to itself. The Result Entities represent research output. The 2nd Level Entities allow for the representation of the research context by linking to them from Core and Result entities. Each 2nd Level Entity supplies some basic attributes; at least an ID and an URI attribute. The linkage mechanism and the multilingual features of 2nd Level Entities are equal to the mechanism and features presented with core and result entities. Link entities are considered a major strength of the CERIF model. A link entity always connects two entities, either Core, Result, or 2nd Level entities.

From the ***Technical*** point of view, interoperability through metadata exchange between CRISs and other types of systems has been focused on the exchange of publication data. This can be based on existing standards and knowledge as to create a comprehensive and unbiased carrier of data, enabling an exchange of data without loss for either the data provider or the receiver and no matter what granularity they operate with. To gather information from CRISs, the OAI-PMH protocol can be used.

Implementation guidelines

Guidelines are disseminated via the euroCRIS website[33].

[33] http://www.eurocris.org/

Assessment

Although CERIF compliant CRIs are being operated in some European Universities, interoperation between them has scarcely been documented. Various projects have worked, and still are working, on linking CRISs with Open Access Repositories and on assuring interoperability between them. Documents reporting on and commenting these experiences have been illustrated in the recent Workshop on CRIS[34], CERIF and Institutional Repositories (CNR, Rome, 10-11 May 2010).

3.1.3 Application Profiles

The idea of "Application Profiles," was introduced by Heery & Patel (Heery & Patel, 2000), who defined it as "a type of metadata schema which consists of data elements drawn from one or more namespaces, combined together by implementers, and optimised for a particular local application". Application profiles provide the means to express principles of modularity and extensibility. The purpose of an application profile is to adapt or combine existing schemas into a package tailored to the functional requirements of a particular application, while retaining interoperability with the original base schemas (Duval, Hodgins, Sutton, & Weibel, 2002).

Application Profiles have extensively been discussed in the context of Dublin Core. The stage for technical specification is discussed by two important documents[35, 36] sponsored by the European Committee on Standardization (CEN). In March 2005, published as a DCMI Recommendation, the DCMI Abstract Model defined a metadata model which covered the requirements to formalize a notion of machine-processable application profiles. In September 2007, at the International Conference on Dublin Core and Metadata Applications in Singapore, Mikael Nilsson presented a framework for the definition of Dublin Core Application Profiles, dubbed the "Singapore Framework" (Nilsson, 2008). This framework defines a set of components which are necessary or useful for documenting an Application Profile, i.e., Functional requirements (mandatory), Doman model (mandatory), Description Set Profile (Mandatory), Usage Guidelines (optional), and Encoding syntax guidelines (optional). It also describes how these documentary standards relate to standard domain models and Semantic Web foundation standards. The framework forms a basis for reviewing Application Profiles with respect to documentary completeness and conformance with Web-architectural principles.

In this rapidly changing context, several communities created extensive APs for their communities. Presently specific Task Groups are active at DCMI, namely Dublin Core Collection Description Application Profile Task Group, DCMI Government Application Profile Task Group, DC-Ed Application Profile Task Group, DCMI Metadata Provenance Task Group[37].

[34] http://www.irpps.cnr.it/eventi/OAworkshop/programme.php

[35] CEN - European Committee for Standardization (2003). ECWA14855 - Dublin Core application profile guidelines. available at www.cenorm.be

[36] CEN Workshop Agreement 15248: Guidelines for machine-processable representation of Dublin Core Application Profiles. (2005). – available at www.cenorm.be/

[37] The Dublin Core Metadata Initiative (DCMI) has recently started a task group to address the issue of "metadata provenance". The group aims to define an application profile that allows for making assertions about description statements or description sets, creating a shared model of the data elements required to describe an aggregation of metadata statements in order to collectively import, access, use and publish facts about the quality, rights, timeliness, data source type, trust situation, etc. of the described statements. The Task Group is led by Kai Eckert of the University of Mannheim and Michael Panzer of OCLC who have become members of the DCMI Advisory Board.

According to the interoperability framework (cf. Section 2):

- a ***Provider*** exposes metadata describing arbitrary resources in conformity with an agreed Application Profile. Such metadata can be exploited by any ***Consumer*** that is able to comply with such an Application Profile;
- the ***Resource*** the two entities are willing to share is any kind of metadata describing an arbitrary resource according to an agreed Application Profile;
- the ***Task*** is the functionality that any *Consumer* is planning to support by relying on metadata based on an agreed Application Profile; such a functionality requires the availability of information describing a resource, i.e., metadata about a resource;
- the solution belongs to the **agreement-based approaches** (cf. Section 2.2.1).

There are a lot of concrete exemplars of this kind of interoperability solution from different domains and for different purposes, e.g., the Europeana Semantic Elements (ESE) (cf. Section 3.1.3.1); the Scholarly Works Application Profile (SWAP)(cf. Section 3.1.3.2); the Education Application Profile (cf. Section 3.1.3.2), the Dublin Core Collections Application Profile (cf. Section 3.1.3.4); the DC-Library Application Profile (cf. Section 3.1.3.5); The AGRIS Application Profile (cf. Section 3.1.3.6); the Biological Data Profile (cf. Section 3.1.3.7); the Darwin Core (cf. Section 3.1.3.8); the DCMI Government Application Profile (DC-Gov) (cf. Section 3.1.3.9).

Overall, the application profile approach has the following characterisation.

Requirements

From the ***Organisational*** point of view, the *Provider* agrees to expose metadata in conformity with the agreed Application Profile; the *Consumer* agrees to acquire metadata in compliance with such metadata scheme.

From the ***Semantic*** point of view, *Provider* and *Consumer* should have a common knowledge of the semantic associated to the elements of the Application Profile, according to how they are defined in the different schemes they are selected from; moreover, they should have a common understanding of the meanings underlying the vocabulary encoding schemes and syntax encoding schemes related to those metadata schemes.

From the ***Technical*** point of view, *Provider* and *Consumer* should agree on the specific Application Profile, vocabulary encoding schemes and syntax encoding schemes to be used; an agreement should be found also on the metadata representation format and the communication protocol to be used for exchanging the metadata.

Results

From the ***Organisational*** point of view, by agreeing on an Application Profile, *Provider* and *Consumer* guarantee that metadata describing arbitrary resources are represented according to the agreed schema as well as that they decided to count on a set of pre-existing metadata schemas for their cooperation.

From the ***Semantic*** point of view, *Provider* and *Consumer* have at their disposal a common set of metadata elements associated with semantics that should be universally understood and supported.

From the Technical point of view, each schema element contains a link to the schema where the elements come from.

Implementation guidelines

While defining Application Profiles there are some best practices that should be followed, namely (i) a proper selection of the schema(s) from which Application Profile elements are taken and (ii) the publishing of the selected schema(s) in the Application Profile schema. This process has many similarities with XML schema definitions and namespaces usage.

For instance, guidelines for Dublin Core Application Profiles have been issued in 2009 as a DCMI Recommended Resource[38]. The document explains the key components of a Dublin Core Application Profile (see the Singapore Framework above) and walks through the process of developing a profile. The document is aimed at designers of Application Profiles — people who will bring together metadata terms for use in a specific context. It does not address the creation of machine-readable implementations of an application profile nor the design of metadata applications in a broader sense.

Assessment

As noted by Heery and Patel (Heery & Patel, 2000), implementation and experience are the teachers that best move metadata management techniques forward.

Accordingly, Bruce & Hillmann (Bruce & Hillmann, 2004) provide important criteria for Application Profile quality, namely, that "*application profiles should in general contain those elements that the community would reasonably expect to find*" and that "*they should not contain elements that are not likely to be used because they are superfluous, irrelevant or impossible to implement*". For example, Hillmann & Phipps (Hillmann & Phipps, 2007) state that "*Provenance is difficult to determine with most metadata unless there is a data wrapper (such as provided by OAI-PMH) which contains provenance information, and that information is maintained properly. Provenance is to some extent administrative in nature, and its presence and reliability depends on the policies of the data provider, and potentially a whole chain of data providers that may have touched the metadata in past transactions. At one level, the presence of provenance information is a good beginning point, but without better tracking of where metadata has been and how it has been modified (not really possible using the methods provided within OAI-PMH) there are significant limits to what can be assumed about the quality and integrity of data that has been shared widely*".

This has been recently confirmed by the feedback from users of the AGRIS Application Profile, as reported by Baker & Keizer (Baker & Keiser, 2010).

As concluded by Hillmann & Phipps (Hillmann & Phipps, 2007), the most important initial value of Application Profiles for implementers is that they concur to create community consensus and serve as a spur to discussion of metadata quality. But a machine-assisted way forward requires better rates of registration of the component parts of Application Profiles (metadata schemas and controller vocabularies) as well as registration and change management for Application Profiles themselves. How this infrastructure will be built, sustained and extended is perhaps the most pressing question for implementers, and the lack of reliable answers the biggest impediment to true progress.

3.1.3.1 Europeana Semantic Elements (ESE)

Europeana Semantic Elements (ESE) is a Dublin Core-based application profile developed in the context of the Europeana. It identifies a generic set of DC elements and some locally coined terms, which have been added specifically to support Europeana's functionalities.

3.1.3.2 Scholarly Works Application Profile (SWAP)

The work for SWAP was undertaken within the JISC Digital Repositories programme and coordinated by Julie Allinson (UKOLN, University of Bath) and Andy Powell (Eduserv Foundation) during 2006. The profile was originally called the 'Eprints Application Profile', but this name has now been superseded by 'Scholarly Works Application Profile' (SWAP) - the two profiles are synonymous.

3.1.3.3 Education Application Profile

The DCMI-Education Working Group has designed the Education Application Profile to serve as an interchange format within and

[38] http://dublincore.org/documents/2009/05/18/profile-guidelines/

outside of the education and training domain. It largely relies on Dublin Core elements.

3.1.3.4 Dublin Core Collections Application Profile

The DC Collections Application Profile[39] identifies elements for describing a collection as well as a catalogue or index (i.e., an aggregation of metadata that describes a collection). The term “collection” can be applied to any aggregation of physical and/or digital resources. Those resources may be of any type, so examples might include aggregations of natural objects, created objects, "born-digital" items, digital surrogates of physical items, and the catalogues of such collections (as aggregations of metadata records). The criteria for aggregation may vary: e.g., by location, by type or form of the items, by provenance of the items, by source or ownership, and so on. Collections may contain any number of items and may have varying levels of permanence.

3.1.3.5 DC-Library Application Profile

This application profile[40] clarifies the use of the Dublin Core Metadata Element Set in libraries and library-related applications and projects. It was originally prepared by the DCMI-Libraries Application Profile drafting committee, a subset of the DCMI-Libraries Working Group.

3.1.3.6 The AGRIS Application Profile

The AGRIS Application Profile (AGRIS AP) is an Application Profile specifically conceived to enhance the description, exchange and subsequent retrieval of agricultural document-like Information Objects. It is a metadata schema which draws elements from well known Metadata standards such as Dublin Core (cf. Section 3.1.2.1), Australian Government Locator Service Metadata (AGLS)[41] and Agricultural Metadata Element Set AgMES[42]. It allows sharing of information across dispersed bibliographic systems and provides guidelines on recommended best practices for cataloguing and subject indexing. The AGRIS AP is considered a major step towards exchanging high-quality and medium-complex metadata in an application independent format in the Agricultural domain. The goal is to facilitate interoperability of metadata formats currently in use to enable linking of various types of agricultural information, therefore allowing users to perform cross-searches and other value added services. This approach would also facilitate the harvesting of data from participating countries; with the application of the AGRIS AP model, this harvesting process could be automated.

The FAO Agricultural Information Management Standards team is currently exploring ways to leverage AGRIS in the WEB environment by publishing the entire repository in form of RDF “triples” - the fundamental unit of linked data (cf. Section 3.1.1.3).

3.1.3.7 Biological Data Profile

The Biological Data Profile (BDP)[43] is an approved profile to the FGDC-Content Standard for Digital Geospatial Metadata (CSDGM), meaning it provides additional fields to the FGDC-CSDGM standard. These fields allow biological information such as taxonomy, methodology, and analytical tools to be added to a metadata record. Since biological data sets can be either geospatial or non-geospatial in their nature, the Biological Data Profile is designed to be used to document both geospatial and non-geospatial data sets. As a profile, all the requirements of the Content

[39] http://dublincore.org/groups/collections/collection-application-profile/

[40] http://dublincore.org/documents/library-application-profile/

[41] http://www.naa.gov.au/recordkeeping/gov_online/agls/summary.html

[42] http://www.fao.org/agris/agmes/

[43] http://www.nbii.gov/portal/server.pt/community/fgdc_metadata/255/standards

Standard for Digital Geospatial Metadata must be met for any geospatial biological data set. The Biological Data Profile extends the use of the Content Standard for Digital Geospatial Metadata into documenting non-geospatial data sets, when biological in nature.

3.1.3.8 Darwin Core

The Darwin Core[44] is a body of standards. It includes a glossary of terms (in other contexts these might be called properties, elements, fields, columns, attributes, or concepts) intended to facilitate the sharing of information about biological diversity by providing reference definitions, examples, and commentaries. The Darwin Core is primarily based on taxa, their occurrence in nature as documented by observations, specimens, and samples, and related information. Included are documents describing how these terms are managed, how the set of terms can be extended for new purposes, and how the terms can be used. The *Simple Darwin Core* is a specification for one particular way to use the terms - to share data about taxa and their occurrences in a simply structured way - and is probably what is meant if someone suggests to "format your data according to the Darwin Core".

The Darwin Core standard was originally conceived to facilitate the discovery, retrieval, and integration of information about modern biological specimens, their spatiotemporal occurrence, and their supporting evidence housed in collections (physical or digital). The Darwin Core today is broader in scope and more versatile. It is meant to provide a stable standard reference for sharing information on biological diversity. As a glossary of terms, the Darwin Core is meant to provide stable semantic definitions with the goal of being maximally reusable in a variety of contexts.

3.1.3.9 DCMI Government Application Profile (DC-Gov)

The DC-Government Application Profile (DC-GAP)[45] defines how to describe metadata for governmental resources using the Dublin Core Metadata Element Set. The task force, also known as the editorial board, was formed during the DC-conference in Madrid to elaborate the profile. The DC-Government community is a forum for individuals and organisations involved in implementing Dublin Core metadata in a context of government agencies and International Governmental Organisations (IGO's), with the objective to promote the application of Dublin Core metadata in that context.

3.1.4 Metadata Mapping / Crosswalks

A crosswalk is “a mapping of the elements, semantics, and syntax from one metadata scheme to those of another” (NISO, 2004). Crosswalks are commonly used to enable interoperability between and among different metadata schemes/formats. They are usually based on a chart or table that represents the semantic mapping of data elements in one data standard (source) onto those data elements in another standard (target) which have similar meaning or similar function (Gill, Gilliland, Whalen, & Woodley, 2008). It is mainly intended to enable heterogeneous collections to be searched simultaneously with a single query as if they were a single database. In the recent past, most work in metadata mapping with crosswalks was related to popular schemes, e.g., DC, MARC, LOM, and consisted in directly mapping or establishing equivalency among elements in different schemes. One of their problems is the different degrees of equivalency: *one-to-one*, *one-to-many*, *many-to-one*, and *one-to-none* (Zeng & Xiao, 2001). However, while crosswalking works well when the number of schemes involved is small,

[44] http://rs.tdwg.org/dwc/

[45] http://www.dublincore.org/dcgapwiki

mapping among multiple schemes is not only extremely tedious and labor intensive but also requires enormous intellectual effort. For example, a four-schema crosswalk would require twelve (or six pairs of) mapping processes. To overcome this problem, a **switching schema** (a.k.a. pivot schema) is used to channel crosswalking among multiple schemas. In this approach, one of the schemes is used as the switching mechanism among multiple schemes. Thus, instead of mapping between every pair in the group, each of the individual metadata schemes is mapped to the switching schema only.

If no existing schema is found to be suitable for use as a switching schema, an alternative is the use of a **Lingua Franca**. A lingua franca acts as a superstructure, but is not a "schema" in itself. In this approach, multiple existing metadata schemes are treated as satellites of a superstructure (lingua franca) which consists of elements common or most widely used by individual metadata schemes. This approach facilitates cross-domain searching, but it is not necessarily helpful in data conversion or data exchange. However, the lingua franca approach allows the retention of the richness and granularity of individual schemes.

Concrete exemplars of this kind of interoperability solution are reported below.

3.1.4.1 DSpace to OAI-ORE

In the DSpace data model, an *item* is a grouping of files and descriptive metadata. The files are called *bitstreams* and are combined into abstract sets called *bundles*. Items are grouped into larger sets called *collections*, which are then further grouped into nestable containers called *communities*.

Establishing a mapping between the DSpace architecture and the OAI-ORE data model means implementing a mediator capable of translating a DSpace item into an ORE Aggregation, and back.

According to the interoperability framework (cf. Section 2):

- the ***Provider*** is any system that manages a DSpace repository of compound information objects; the ***Consumer*** is any system willing to consume such objects in compliance with the OAI-ORE data model;
- the ***Resource*** the two entities are willing to share is any kind of compound information object;
- the ***Task*** is the functionality the *Consumer* is willing to support; such a functionality relies on the availability of the shared resource in order to correctly work;
- the solution belongs to the **mediator-based approaches**.

Requirements

From the ***Organisational*** point of view, the *Provider* and the *Consumer* agree to use a DSpace-ORE mediator to exchange structural information about DSpace information objects. It should be found an agreement on which side the mediator is to be implemented, i.e., Provider side or Consumer side or both.

From the ***Semantic*** point of view, the mediator should have an understanding of the semantics of the OAI-ORE data model, as well as the semantics of the DSpace data model. Moreover, it must have knowledge of the mapping rules between the two data models.

No ***Technical*** information is available.

Results

From the ***Organisational*** point of view, the *Provider* exposes its DSpace compound information objects and the *Consumer* can exploit them in compliance with the OAI-ORE data model.

From the ***Semantic*** point of view, *Provider* and *Consumer* have a common way to interpret the structural information of a DSpace information object, which is given by an effective mapping through a shared data model, i.e., OAI-ORE. In particular, the mapping between the DSpace architecture and the ORE data model is the following. Each DSpace item corresponds to an ORE Aggregation, and its component bitstreams are the Aggregated Resources composing the

Aggregation. Furthermore, each DSpace collection is an aggregation of items, and each community is an aggregation of collections. A Resource Map is associated to each DSpace item, collection or community. Any descriptive metadata is encoded outside the ORE model.

No ***Technical*** information is available.

Implementation guidelines

Not available.

Assessment

Not available.

3.1.4.2 Fedora to OAI-ORE

In the Fedora data model, a *digital object* consists of one or more byte streams, called data*streams*. A *datastream* can represent a payload or metadata.

Establishing a mapping between the Fedora architecture and the OAI-ORE data model means implementing a mediator capable of translating a Fedora digital object into an ORE Aggregation, and back.

According to the interoperability framework:

- the ***Provider*** is any system that manages a Fedora repository of compound information objects; the ***Consumer*** is any system willing to consume such objects in compliance with the OAI-ORE data model;
- the ***Resource*** the two entities are willing to share is any kind of compound information object;
- the ***Task*** is the functionality the *Consumer* is willing to support; such a functionality relies on the availability of the shared resource in order to correctly work;
- the solution belongs to the **mediator-based approaches** (cf. Section 2.2.2).

Requirements

From the ***Organisational*** point of view, the *Provider* and the *Consumer* agree to use a Fedora-ORE mediator to exchange structural information about Fedora digital objects. It should be found an agreement on which side the mediator is to be implemented, i.e., Provider side or Consumer side or both.

From the ***Semantic*** point of view, the mediator should have an understanding of the semantics of the OAI-ORE data model, as well as the semantics of the Fedora data model. Moreover, it must have knowledge of the mapping rules between the two data models.

No ***Technical*** information is available.

Results

From the ***Organisational*** point of view, the *Provider* exposes its Fedora digital objects and the *Consumer* can exploit them in compliance with the OAI-ORE data model.

From the ***Semantic*** point of view, *Provider* and *Consumer* have a common way to interpret the structural information of a Fedora digital object, given by an effective mapping with a shared data model, i.e., OAI-ORE. The mapping between the Fedora architecture and the ORE data model should be arranged together by the *Provider* and the *Consumer*. A standard approach for mapping is the following: each Fedora digital object corresponds to an ORE Aggregation, and its component datastreams are the Aggregated Resources composing the Aggregation. A Resource Map is generated and associated to each Fedora digital object. Other customized approaches are possible.

No ***Technical*** information is available.

Implementation guidelines

Fedora does not offer any native interface for supporting ORE dissemination and harvesting. Nevertheless, a Java web open source application that allows creating Resource Maps for Fedora digital objects is available. Two different modalities are provided: a customised approach, that allows one to specify which datastream and digital objects compose an Aggregation, and a standard approach.

The mapping rules between the Fedora architecture and the OAI-ORE data model should be arranged together by the *Provider* and the *Consumer*.

Assessment

Not available.

3.1.5 Information Object (Resource) Identifiers

Concrete exemplars of this kind of interoperability solution are: Uniform Resource Identifier (URI) (cf. Section 3.1.5.1) and the Handle System (cf. Section 3.1.5.2).

3.1.5.1 Uniform Resource Identifier (URI)

A *Uniform Resource Identifier (URI)* provides a simple and extensible means to identify a resource of any kind or nature in a globally unique manner. In the World Wide Web, such identification may enable to interact with different representation of the resource, by using specific protocols and access services.

According to the interoperability framework:

- a ***Provider*** exposes URIs representing specific resources on the Web; the ***Consumer*** is any client that wishes to identify the resource associated to an URI and understands its specific URI scheme;
- the ***Resource*** the two entities are willing to share is a URI associated to any kind of resource;
- the ***Task*** is the functionality that the *Consumer* is planning to support; such a functionality is any kind of process requiring a resource to be identified in a unique and possibly persistent way;
- the solution belongs to the **agreement-based approaches** (cf. Section 2.2.1).

A URI can be further classified as a name (URN), a locator (URL), or both. A URN identifies a resource by name in a particular namespace, while a URL is a URI that, apart from providing for identification, gives a means to retrieve the location of a resource by describing its primary access mechanism. Differently from a URL, a URN is required to remain persistent and globally unique, even when the resource it represents is no longer available.

Requirements

From the ***Organisational*** point of view, *Provider* and *Consumer* agree to adopt the URI standard to identify specific resources in a globally unique and possibly persistent way.

From the ***Semantic*** point of view, *Provider* and *Consumer* should share a common understanding of the notion of *URI*, as well as the semantics of the specific URI scheme they intend to use.

From the ***Technical*** point of view, *Provider* and *Consumer* should have a common knowledge of the syntax of the specific URI scheme they would adopt; moreover, an agreement should be found on the representation format and the protocol to be used for producing and sharing the URIs.

Results

From the ***Organisational*** point of view, the URI standard guarantees that *Provider* and *Consumer* have a common way to identify any kind of resource, either abstract or physical. It is important to underly that URIs are not intended to be used for identifying only accessible resources: a URI provides identification for a resource regardless of whether such a resource is accessible or not. Access, as well as other operations on the resource identified by the URI, is defined by the protocols that make use of URIs, and are not directly guaranteed by the mere presence of the URI itself.

From the ***Semantic*** point of view, *Provider* and *Consumer* have a common way to define and interpret an identifier, namely a *URI*, referencing a specific resource.

The URI syntax consists of a scheme name followed by a scheme-specific part. The semantic interpretation of a URI is determined by the specifications associated with its scheme name, although a generic URI syntax is defined as well, specifying the elements that all independent schemes are required to have in order to promote interoperability. In particular, a generic URI is a sequence of components organised hierarchically which contains different data that together contribute to identify a resource: the *authority* component is optional and consists of user information, a host name and a port number; the *path* component contains hierarchical data that serve to identify a resource within the scope of a URI scheme and a naming authority; the optional *query* component contains additional non-hierarchical identification information; the *fragment identifier* component is optional and allows indirect identification of a secondary resource.

From the ***Technical*** point of view, *Provider* and *Consumer* share a generic URI syntax which is a superset of the syntax of all valid URIs; this allows any URI parser to identify the common components of a URI without knowing the scheme-specific details, and to perform further scheme-specific parsing once the scheme is determined. URI syntax specification does not mandate any particular character encoding for mapping the sequence of URI characters to the sequence of octets used to store and transmit those characters; the choice of the transmission protocol, and consequently of the character encoding, is left to the *Provider* and the *Consumer*, although it is usually suggested by the specific URI scheme, e.g., http scheme suggests the use of HTTP transmission protocol.

Implementation guidelines

A set of specifications and guidelines have been published over the years, until the publication of RFC 3986 in January 2005, which defines the current URI generic syntax, as well as a process for URI resolution and a set of guidelines and security considerations for the use of URIs in a network. Guidelines for the definition (RFC 2718) and registration (RFC 2717) of new URL schemes have been published, as well.

Assessment

Not available.

3.1.5.2 The Handle System

The Handle System[46] is a worldwide distributed system that provides efficient, extensible and secure identifier and resolution services.

The system enables to store identifiers, namely *handles*, of any arbitrary web resource, such as URLs, XML and binary data. A handle for a resource is unique, thus avoiding collisions and ambiguity in name references. Moreover, handles are persistent identifiers, i.e., the name associated to a resource is immutable and not influenced by changes of location and other related state information.

According to the interoperability framework (cf. Section 2):

- the ***Provider*** is the Handle System itself, i.e., any service, either local or global, implementing the Handle protocol; the ***Consumer*** is any client that wishes to identify the web resource associated to a handle and understands the Handle protocol;
- the ***Resource*** the two entities are willing to share is any handle, i.e., a persistent identifier, associated to any kind of web resource;
- the ***Task*** is the functionality that the *Consumer* is planning to support by relying on the shared handle; such a functionality is any kind of process requiring to identify an object in a persistent and unique way;
- the solution belongs to the **agreement-based approaches** (cf. Section 2.2.1).

The largest and best-known implementation of the Handle System is that of the International DOI Foundation (IDF)[47], which handles identifiers for the international publishing sector.

[46] http://www.handle.net/

[47] http://www.doi.org/index.html

Requirements

From the ***Organisational*** point of view, *Provider* and *Consumer* agree to adopt the Handle System to identify web resources in a unique and persistent way. The *Provider* agrees to maintain the association between handles and related resources, while the *Consumer* agrees to query the *Provider* for handle resolution.

From the ***Semantic*** point of view, *Provider* and *Consumer* should share a common understanding of the notion of *handle*, as well as the semantic of the handle protocol.

From the ***Technical*** point of view, *Provider* and *Consume*r should have a common knowledge of the handle protocol to be implemented. At least, they should rely on a communication protocol based on HTTP.

Results

From the ***Organisational*** point of view, the Handle System guarantees that *Provider* and *Consumer* have a common way to associate any resource with a unique and persistent identifier. It is worth underlying that the Handle System is a pure resolution system, so it carries no assumptions on what the client will or will not do with the resolution information: the *Provider* simply returns the value(s) associated to the handle(s) requested by the *Consumer*.

From the ***Semantic*** point of view, *Provider* and *Consumer* have a common way to interpret and resolve a persistent identifier, namely a handle. Handles are identifiers with a very simple syntax: “prefix/suffix”, where the prefix is a naming authority registered by an organization with administrative responsibility for creating and managing identifiers, while the suffix is a unique name defined under the associated prefix.

From the ***Technical*** point of view, the Handle System has a two-level hierarchical service model: a single top level global service, known as the Global Handle Registry, and other lower level handle services, known as Local Handle Services. Local Handle Services are responsible for the creation and management of all identifiers under their associated naming authority, providing clients with resolution and administration services. All identifiers under a given prefix must be maintained in one service. A Local Handle Service can be responsible for more than one local handle namespace, each one corresponding to a unique prefix. The Global Handle Registry is a handle service like any other, so it can be used to manage any handle namespace. In addition, it maintains information about the Local Handle Service responsible for resolving identifiers with a given prefix. In order to resolve a specific identifier, a client should first consult the Global Registry to obtain the necessary information about the Local Service in charge for managing that identifier, and then the Local Service itself for resolution. Communication with the Handle System is carried out using Handle System protocols (RFC 3652[48]), each one of them having a formal specification and some specific implementations. In all cases, all handles can be resolved through an HTTP proxy server[49] that understands the handle protocol and to which any client may be directed for handle resolution.

Implementation guidelines

In order to run a Local Handle Service, a *Provider* should primarily install and configure a Local Handle Server. A typical Local Handle Service has one site and one handle server, but a more scalable solution is possible for a service, by adding more servers. Another way to scale up is to assign more sites to a Local Handle Service, in order to provide redundancy via replication or mirroring of identifiers.

A set of implementation guidelines and requirements is described in order to help users to understand the Handle Protocol and to install and run their own handle servers. In particular, an interface specification[50] is defined, and a

[48] http://www.handle.net/rfc/rfc3652.html

[49] http://www.handle.net/proxy.html

[50] http://www.handle.net/rfcs.html

description of the System fundamentals[51] is available together with a Technical Manual[52] for installing, configuring and managing a handle server, and administering one's own identifiers. A set of libraries[53], tools and plug-ins[54] is available for clients as well.

Assessment

The DOI System utilises the Handle System as one component in building an added value application, for the persistent and semantically interoperable identification of intellectual property entities.

The Handle System provides no means for declaring the semantics of the resource associated to a handle. The DOI System adds this facility, by associating metadata with resources: a kernel of common metadata is provided, which can be extended with other relevant information, in order to properly specify the semantics of the resource.

In the DOI System, Local Handle Services correspond to Registration Agencies (RAs), in charge of allocating DOI prefixes, registering DOI names, and providing the necessary infrastructure to declare and maintain metadata and state information. Registration agencies generally charge a fee to assign a new DOI name, and part of these fees is used to support the IDF. A list of current RAs is maintained by the International DOI Foundation.

3.2 User Domain Interoperability Best practices and Solutions

User interoperability is a particular category of Digital Library interoperability that has not been broadly studied and explored. Nowadays, users interact with different Digital Libraries and other personalised systems on a regular basis and update their profiles stored at these systems. These distributed and heterogeneous user profiles provide a valuable source of information in order for systems to acquire wider knowledge about users and use it to achieve personalization and better adaptation. User interoperability constitutes an essential requirement for these profiles to be shared effectively among different systems.

Interoperability of DLs over the user domain is the ability of two or more DLs to exchange information about the same user and to use the information that has been exchanged meaningfully and accurately in order to produce useful results as defined by the users of these systems. User-level interoperability of DLs arises with respect to issues such as ***user modelling***, ***user profiling***, and ***user management***.

User modelling is the process of capturing all the fundamental information about DL users in order for the system to be able to behave differently to different users, whereas **user profiling** is the process of collecting information about a user in order to generate the user's profile[55], depending on the current user model. Information about the user that may be captured in a DL is user credentials, demographics, access rights, preferences, interests, etc. In general, a user model should be rich enough as to capture the aforementioned characteristics in order to accommodate different user needs for accessing the content and the functionalities provided by the system, while maintaining the explicit or implicit preferences affecting the results of the user operations and differentiating based on the context of the user. Up to now, however, there is no generally accepted user model that may be used in every DL application and ensure that a profile created

[51] http://www.handle.net/documentation.html

[52] http://www.handle.net/tech_manual.html

[53] http://www.handle.net/client_download.html

[54] http://www.handle.net/other_software.html

[55] This is known as the *Actor Profile* in the Digital Library Reference Model.

within a certain DL may be moved effortlessly to another. Thus, **interoperability in terms of user modelling** refers to the ability of DL systems to support compliant and interoperable user models that enable the propagation of user information across different DLs. Furthermore, **interoperability in terms of user profiling** refers to the ability of DL systems to support mechanisms of reconciliation of user profile characteristics. These two issues are strongly associated and achieving user model interoperability constitutes a prerequisite for user profile interoperability.

Recent advances in user model interoperability reveal two basic approaches that focus on achieving syntactic and semantic interoperability of user models: a ***shared format approach*** (cf. Section 3.2.1) and ***a conversion approach*** (cf. Section 3.2.2). The shared format approach enforces the use of a shared syntax and semantics to represent user models. On the other hand, the conversion approach, not using any shared representation for user models, employs appropriate methods to transform the syntax and semantics of the user model used in one system into those of another system.

Finally, **interoperability in terms of user management** (cf. Section 3.2.3) refers to the ability of heterogeneous DL systems to work in synergy on issues that are strongly associated to users' privileges, therefore applying concrete and shared authentication and authorization policies in a way transparent to the user. Two examples of user management interoperability are OpenID and Security Assertion Markup Language (SAML).

3.2.1 Representation of User Models: Shared Format Approach

There is an obvious advantage in utilizing a shared format approach, which imposes the use of a shared syntax and semantics for the representation of user models. This advantage is that a DL system can seamlessly acquire and manage user characteristics discovered by other DL systems. In this way, the DL system may use the existing information for personalization without the user being obliged to input them again.

In order to achieve user model interoperability, the user modelling community has recently focused on ontology based approaches as the basis for the shared format approach. Ontology based approaches have several advantages that originate from the principles of this formalism. The ontological representation of user characteristics allows deducing additional user features based on ontology relations, conditions, and restrictions. The use of an ontology-based user model thus increases the potential for user characteristics to be shared among DL systems. Such an approach is the General User Model Ontology that will be analysed in the following section.

3.2.1.1 General User Model Ontology

Heckmann et al. (Heckmann, Schwartz, Brandherm, Schmitz, & Wilamowitz-Moellendorff, 2005) introduced the General User Model Ontology (GUMO) in order to manage the syntactic and semantic variations in existing user modelling systems. GUMO is based on OWL and is used for the representation of user model characteristics and their interrelationships. The authors collected the user's characteristics that are modelled in user-adaptive systems like the user's heart beat, age, current position, birthplace, or the user's ability to swim. Furthermore, the modelling of user's interests and preferences was analyzed. The construction of GUMO was based on the thought of dividing the descriptions of user model characteristics into three elements: *auxiliary*, *predicate* and *range*. This description is called a *situational statement*. For example, the interest of a user in music could be described in the following way: auxiliary=*hasInteres*t, predicate=*music* and range=*low-medium-high*. The advantage of using GUMO is the semantic uniformity (Heckmann, Schwarts, Brandherm, & Kroner, 2005). The characteristics of GUMO are applied in the user model exchange language called User Modelling Markup Language (UserML) (Heckmann & Kruger, A User Modeling Markup Language (UserML) for Ubiquitous Computing, 2003), which promotes the exchange of user models across systems. UserML was also designed according to the approach of dividing user model characteristics into triples. The advantage of using UserML to model the user model statements is the uniform syntactical relational data structure.

Along with the GUMO and the UserML, a framework is presented that can be used to achieve user model interoperability. There are two kinds of actors involved in this framework: the *u2m.org user model service* and *applications*. The *u2m.org user model service* is an application-independent server for accessing and storing user information and for exchanging this information between different applications. A key feature is that the semantics for all user model characteristics are mapped on to the general user model ontology GUMO (Heckmann D. , 2005). An *application* may add or request information that is stored into the *u2m.org server*.

According to the interoperability framework (cf. Section 2):

- the ***Provider*** is the *u2m.org user model service* while the ***Consumer*** is the *application*;
- the ***Resource*** the two entities are willing to share is a *situational statement*, i.e., the description of a user model characteristic, that should obey to GUMO;
- the ***Task*** is the service the application is planning to support. For example, a typical service can be a personalization mechanism;
- the solution belongs to the **agreement-based approaches** (cf. Section 2.2.1).

Requirements

From the ***Organisational*** point of view, the *Provider* agrees to expose the situational statements in the UserML format. The *Consumer* agrees to acquire situational statements of the *Provider* by interacting with a UserML web service.

From the ***Semantic*** point of view, the *Provider* and the *Consumer* should share a common understanding of the notion of *situational statement*. This is achieved because the framework requires that the *Provider* and the *Consumer* use the General User Model Ontology.

From the ***Technical*** point of view, the *Provider* and the *Consumer* rely on a communication channel based on HTTP or UserML web service.

Results

From the ***Organisational*** point of view, the GUMO approach guarantees that the *Provider* exposes *situational statements* to any Consumer sending proper requests. From the *Consumer* perspective, the GUMO approach guarantees that the Consumer can acquire situational statements from any Provider that uses GUMO. However, this solution incorporates a sort of service level agreement, i.e., a *Provider* should serve a well defined set of incoming requests that comply with the UserML exchange language and the GUMO.

From the ***Semantic*** point of view, the GUMO approach guarantees that the *Provider* and the *Consumer* share a common understanding of the notion of situational statement. This is achieved because the Provider and the Consumer agree to use GUMO and to expose situational statements in the UserML format.

From the ***Technical*** point of view, the *Provider* exposes situational statements through a set of HTTP requests and responses or UserML web services. The *Consumer* can use HTTP requests and responses or UserML web services to gather situational statements from any *Provider*. A basic request looks like:

http://www.u2m.org/UbisWorld/UserModelService.php?

subject=John&auxiliary=hasInterest&predicate=Football

Implementation guidelines

GUMO has to be used in both *Provider* and *Consumer* side. The *Provider* has to support the requests produced in UserML format while the *Consumer* has to provide proper requests in UserML format and consume the responses. Apart from the above requirements the authors didn't provide further implementation guidelines.

Assessment

It is apparent that there are no syntactic or semantic heterogeneity issues to be solved if DL systems adopt GUMO as the shared format approach and expose situational statements in UserML format. All the systems use the shared unified model that is easily exchangeable and interpretable. Nevertheless, the DL systems that exist nowadays are very heterogeneous and dynamic. This makes it impractical, and in some cases even impossible, to use a shared user model. Thus, the General User Model Ontology is suitable for systems that may easily agree to share a common user model format. In such a case, the ***implementation cost*** is insignificant because the *Provider* and the *Consumer* adhere to the shared vocabulary of GUMO and exchange user model characteristics using UserML.

For what is concerned with the ***effectiveness***, being GUMO an agreement based approach it is by definition highly effective for what is captured by the agreement.

3.2.2 User Models and Profiles Conversion

In contrast with the shared format approach, the opposite approach excludes the use of a shared representation for the user model and defines proper algorithms to convert the syntax and semantics of the user model schema characteristics in one system into those used in another system.

Having a common model or a way to move a user profile from one DL to another is not enough. One important issue that should be considered is the issue of *data reconciliation*, which is related to how to reconcile different and in some cases even conflicting user profile characteristics. A user may have specific recorded preferences in a DL but slightly or importantly different ones in another DL. This may be due to several reasons ranging from how the profile was elicited or explicitly created by the user to issues related to user context. A data reconciliation rule helps to define what to do if an actual value in the first DL is different from the corresponding value in the second DL. General approaches include the concatenation of the two values, the replacement of the current value, or the use of a given formula in order a decision to be taken. Examples of such decisions are decisions based on time-stamping (e.g., latest time-stamp wins) or based on a trust value (of the user) for the corresponding DL.

Two approaches that belong to this category are the Generic User model Component (cf. Section 3.2.2.1) and the Framework for User Model Interoperability.

3.2.2.1 Generic User model Component

Van der Sluijs and Houben (Van der Sluijs & Houben, Towards a generic user model component, 2005) introduced the Generic User model Component (GUC) that applies Semantic Web technologies to provide user model server functionalities. There are two kinds of actors involved in the proposed architecture: *GUC* and *UM-based applications*. GUC is a generic component that offers functionalities to store schema models for applications and to exchange user information (user profile) between these models. A *UM-based application* is an entity outside the GUC that uses GUC to store user information. UM-based applications might be applications in the classical sense, but also sensors, agents, and other processes. An application that wishes to use its own schema should (i) "subscribe" to GUC, (ii) upload its schema, which describes the structure of its user model, into the GUC's application schema repository, and (iii) upload or request user information for specific users. If an application schema is stored in GUC, the application can upload instances of that schema for particular users, i.e., for every user that uses an application, an instance of this schema is stored in GUC. Such an instance is called a *user application-view (UAV)*. For every user, GUC keeps a UAV repository that stores a UAV for every application the user uses.

According to the interoperability framework (cf. Section 2):

- the ***Provider*** is GUC while the ***Consumer*** is the *UM-based application*;
- the ***Resource*** the two entities are willing to share is a *user application-view* (UAV) referring to the user information that constitute an instance of the *Consumer*'s schema model and that are stored in the *Provider*'s repository;
- the ***Task*** is the service the application is planning to support. For example, a typical service can be a personalization mechanism;
- the solution belongs to the **mediator-based approaches** (cf. Section 2.2.2) and specifically to the **provider-side mediator approach** because the *Provider* applies schema and instance mapping techniques to support the exchange of user information between applications.

Requirements

From the ***Organisational*** point of view, the *Provider* agrees to store the schema model of each Consumer and exposes *UAVs* in the specific format. The *Consumer* agrees to acquire *user application-views* from the *Provider*.

From the ***Semantic*** point of view, the *Provider* and the *Consumer* should share a common understanding of the notion of *UAV*. For this reason, the Provider stores the schema model of the Consumer. Because the Provider can be used for the exchange of user information between applications, schema and instance mapping techniques should be applied.

From the ***Technical*** point of view, the *Provider* and the *Consumer* rely on a communication channel based on HTTP.

Results

From the ***Organisational*** point of view, the GUC approach guarantees that the *Provider* exposes *UAVs* of the schema models it stores upon proper request. From the *Consumer* perspective, the GUC approach guarantees that the *Consumer* can acquire *UAVs* from the *Provider* that stores its schema model.

From the ***Semantic*** point of view, the GUC approach guarantees that the *Provider* and the *Consumer* share a common understanding of the notion of *user application-view*. This is achieved because the *Provider* stores the schema model of the Consumer. Furthermore, the *Provider* can be used for the exchange of information between user-application views. For this reason, an instance of one *Consumer*'s schema model can be translated into a (partial) instance of another *Consumer*'s schema model. For example, if we want to transfer information from UAV x in UAV y, the information of UAV x is first converted into the structure of UAV y. Second, this converted information has to be integrated into the existing UAV y. This conversion and the subsequent integration constitute an instance mapping. The instance mapping is generated from a schema mapping. A schema mapping from a schema X to a schema Y includes a specification of how all characteristics in schema X are mapped onto the corresponding characteristics in schema Y.

The Provider uses the Shared User Model (S-UM), which includes the most used concepts within the domain, as a means of user model exchange between various Consumers. S-UM can be used as a mediator for the exchange of user information between Consumers by creating a mapping to and from every Consumer and S-UM. Matching and merging techniques can be used to match two input schemas and create a merged schema that is a union of both input schemes. With the matching and merging techniques, a compound ontology of the Consumers' schemas can be constructed. For a user, all the UAVs are merged in the GUC global user model, which is a (partial) instance of the compound ontology.

This structure contains all the information that is known about the user. The global user model is produced by the GUC data manager by applying the mappings to all UAVs in the UAV repository. When a Consumer requests information for a specific user for the first time, the Provider will create a UAV for the user by creating an instance of the corresponding Consumer schema in the application schema repository. The GUC data manager will try to complete this UAV with information stored in the global user model.

From the ***Technical*** point of view, the *Provider* exposes user application-views through HTTP requests and responses. The *Consumer* can use HTTP requests and responses to gather user application-views from the *Provider*.

Implementation guidelines

The authors of GUC have provided some implementation guidelines of instance and schema mappings. A schema mapping is performed in the *Provider* and contains rules that define how all characteristics of a given schema can be mapped to corresponding characteristics in another schema. Schema mappings are delivered by the GUC mapping module. For this, the mapping module requires the source schema, say X, and the target schema, say Y. The mappings are generated based on the similarities between two input schemas and are expressed in the rule language SWRL. As the mapping between schema X and schema Y has to be constructed only once, it can be created by the (human) designer.

information reconciliation is supported by applying the OWL and SWRL techniques. For each *Consumer*, rules can be defined that specify how to reconcile information in the case of a conflict. The information reconciliation rule type helps to define what to do if a value in the transformed UAV already exists in the UAV that it should be integrated in: it is possible that the value is concatenated with the current value, or that the current value is replaced, or that a decision is made based on a given formula.

Irrespectively of the algorithm used for the schema mapping, the result must be examined and possibly be edited by hand before it can be used, because semantic structures may not be interchangeable. Schema characteristics that appear the same may not be interchangeable on instance level. For example, we can consider the related characteristics user-name and password. Even though the semantic meaning of these characteristics might be the same for two Consumers, the concrete values for these characteristics and for a particular user might not be interchangeable for those Consumers.

Assessment

The advantage of using S-UM as a mediator for the exchange of user information between Consumers is that new Consumers can easily be added to the system through only two mappings. The complexity is 2N mappings for N Consumers (Van der Sluijs & Houben, 2006). The disadvantage is that translating in two steps, via S-UM, might result in loss of information. An additional disadvantage is that schema mappings require further human effort and may not always be feasible.

3.2.3 Authentication/Authorisation Protocols for User Management

In the area of user authentication/authorization there are some successful and widely used authentication/authorization protocols. An increasingly frequent problem with standards is that there are too many of them and unfortunately they are designed in such a way that alignment among them is significantly difficult to achieve. Consequently, the need for creating interoperable solutions in this area became imperative. Before analyzing the various proposals, we need to emphasize on the notion of "federated identity".

Federated identity[56], or the "federation" of identity, describes the technologies, standards

[56] http://en.wikipedia.org/wiki/Federated_identity

and use-cases which serve to enable the portability of identity information across otherwise autonomous security domains. The ultimate goal of identity federation is to enable users of one domain to securely access data or systems of another domain seamlessly, and without the need for completely redundant user administration. Federation is enabled through the use of open industry standards and/or openly published specifications, such that multiple parties can achieve interoperability for common use cases. Typical use-cases involve things such as cross-domain web-based single sign-on, cross-domain user account provisioning, cross-domain entitlement management and cross-domain user attribute exchange. Two very important interoperability approaches that support identity federation are OpenID and Security Assertion Markup Language (SAML).

3.2.3.1 OpenID

OpenID[57] is an open, decentralized standard for users' authentication that can be used for access control, allowing users to log on to different services with the same digital identity, where these services trust the authentication body. OpenID replaces the common login process that uses a login-name and a password[58], by allowing a user to log in once and gain access to the resources of multiple software systems.

There are two kinds of actors involved in the framework: *OpenID Providers* and *Relying Parties*. An OpenID Provider is an OpenID Authentication server on which a Relying Party relies for an assertion that a user controls an Identifier. A Relying Party is a Web application that requires proof that the user controls an Identifier. The Relying Party interacts with the User Agent that is user's Web browser which implements HTTP/1.1. The OpenID Provider may also interact with the User Agent.

The OpenID federation mechanism operates in the following manner: The user visits a Relying Party web site which displays an OpenID login form somewhere on its page. Unlike a typical login form with fields for the user name and password, the OpenID login form has only one field—for the Identifier. This form is connected to an implementation of an OpenID client library. A user typically will have previously registered an Identifier with an OpenID Provider. The user types her/his Identifier into the aforementioned OpenID login form.

According to the interoperability framework (cf. Section 2):

- the ***Provider*** is the *OpenID Provider* while the ***Consumer*** is the *Relying Party*;
- the ***Resources*** the two entities are willing to share include (i) an Identifier that is either a "http" or "https" URI (commonly referred to as a "URL") or an XRI and (ii) an assertion in the form of an OpenID protocol message indicating whether the user can be authenticated or not. The Relying Party typically transforms the Identifier into a canonical URL form;
- the ***Task*** is the service the Relying Party is planning to support. Such a service can be any service that requires authentication in order for the user to be able to use it;
- the solution belongs to the **agreement-based approaches** (cf. Section 2.2.1).

Requirements

From the ***Organisational*** point of view, the *Provider* agrees to send positive assertions to the Consumer's authentication requests when it can authenticate that a user controls an Identifier and a user wishes to complete the authentication. Furthermore, when the Provider is unable to identify a user or a user does not or cannot approve the authentication request, the Provider sends a negative assertion to the Consumer. The *Consumer* agrees to expose URI or XRI Identifiers in the normalized

[57] http://openid.net/

[58] http://openid.net/specs/openid-authentication-2_0.html

format that must be absolute HTTP or HTTPS URLs in order to acquire an authentication response from the *Provider*.

From the ***Semantic*** point of view, the *Provider* and the *Consumer* should share a common understanding of the model incorporated by the protocol, i.e., the notion of Identifier as well as OpenID assertions.

From the ***Technical*** point of view, the *Provider* and the *Consumer* rely on a communication channel based on HTTP.

Results

From the ***Organisational*** point of view, the OpenID approach guarantees that the *Provider* submits positive or negative assertions to any Consumer sending proper requests. From the *Consumer* perspective, the OpenID approach guarantees that the Consumer can acquire assertions from any Provider and for any Identifier.

From the ***Semantic*** point of view, the OpenID approach guarantees that the *Provider* and the *Consumer* share a common understanding of the model incorporated by the protocol, i.e., the notion of Identifier as well as OpenID assertions. This is achieved because the Provider and the Consumer use for each identifier the canonical URL form and OpenID is implemented by both the Provider and the Consumer.

From the ***Technical*** point of view, the *Provider* and the *Consumer* that implement OpenID Authentication use only standard HTTP(S) requests and responses.

Implementation guidelines

The OpenID protocol has to be implemented by both the Provider and the Consumer. The Provider has to support the requests envisaged by the protocol and the Consumer has to issue proper requests and consume the responses. A set of libraries[59] have been created to assist the implementation of an OpenID Provider and Consumer. Furthermore, recommendations[60] and tips have been produced for developers who have already implemented OpenID Consumers and/or Providers.

Assessment

Some of the advantages offered by the OpenID standard results from its being an open, decentralized, free framework, which allows Internet users to control their digital life with single identity. The main problems include its vulnerability to phishing and other attacks, the creation of privacy problems, and the lack of trust, which make it unappealing to someone to become an OpenID Consumer.

For what is concerned with the ***effectiveness***, being an agreement based approach it is by definition highly effective for what is captured by the standard.

3.2.3.2 Security Assertion Markup Language (SAML)

The OASIS Security Assertion Markup Language (SAML)[61] standard defines an XML-based framework for describing and exchanging security information between on-line business partners. This security information is specified in the form of portable SAML assertions that applications working across security domain boundaries can trust. The OASIS SAML standard defines precise syntax and rules for requesting, creating, communicating, and using these SAML assertions.

There are two kinds of actors, called system entities that are involved in the framework: *SAML Asserting Party* and *SAML Relying Party*. An Asserting Party is a system entity that makes SAML assertions. It is also sometimes called a SAML authority. A Relying Party is a system entity that uses the received assertions. At the heart of most SAML assertions is a subject (an entity that can be authenticated within the

[59] http://wiki.openid.net/Libraries

[60] http://openidexplained.com/developers

[61] http://docs.oasis-open.org/security/saml/v2.0/

context of a particular security domain) about which something is being asserted. The subject might be a human but might also be some other kind of entity, such as a company or a computer.

According to the interoperability framework (cf. Section 2):

- the ***Provider*** is the *SAML Asserting Party* while the ***Consumer*** is the *SAML Relying Party*;
- the ***Resource*** the two entities are willing to share is a *SAML assertion* that is a piece of data produced by a SAML Asserting Party regarding either an act of authentication performed on a subject, attribute information about the subject, or authorization data applying to the subject with respect to a specified resource. SAML assertions carry statements about a system entity's identity that an Asserting Party claims to be true. The valid structure and contents of an assertion are defined by the SAML assertion XML schema;
- the ***Task*** is the service the Relying Party is planning to support. Such a service can be any service that requires authentication in order for the user to be able to use it;
- the solution belongs to the **agreement-based approaches** (cf. Section 2.2.1).

Requirements

From the ***Organisational*** point of view, the *Provider* agrees to produce assertions using the SAML assertion XML schema. The *Consumer* agrees to acquire SAML assertions from the *Provider*. A trust relationship between the *Provider* and the Consumer should have been established in order for the Consumer to rely on information from the *Provider*.

From the ***Semantic*** point of view, the *Provider* and the *Consumer* should share a common understanding on the notions of *SAML assertion, SAML protocol messages, and SAML bindings.*

From the ***Technical*** point of view, the *Provider* and the *Consumer* rely on a communication channel based on HTTP or SOAP.

Results

From the ***Organisational*** point of view, the SAML approach guarantees that the *Provider* exposes SAML assertions to any Consumer sending proper requests. From the *Consumer* perspective, the SAML approach guarantees that the Consumer can acquire assertions from any Provider. However, this solution subsumes a sort of service level agreement, i.e., a *Provider* should serve a well defined set of incoming requests that comply with the SAML standard. The standard primarily permits transfer of identity, authentication, attribute, and authorization information between *Provider* and *Consumer* that have an established trust relationship.

From the ***Semantic*** point of view, the SAML approach guarantees that the *Provider* and the *Consumer* share a common understanding of the model subsumed by the protocol, i.e., SAML assertions, SAML protocol messages, and SAML bindings. SAML allows for one party to assert security information in the form of statements about a subject. SAML defines three kinds of statements that can be carried within an assertion: authentication statements, attribute statements, and authorization decision statements. SAML protocol messages are used to make the SAML-defined requests and return appropriate responses. The structure and contents of these messages are defined by the SAML-defined protocol XML schema. SAML bindings detail exactly how the various SAML protocol messages can be carried over underlying transport protocols.

From the ***Technical*** point of view, the *Provider* exposes SAML assertions in a SAML protocol response message that must be transmitted using some sort of transport or messaging protocol (HTTP or SOAP). The *Consumer* can issue SAML requests and responses that can be transmitted in well defined HTTP messages to gather the expected information from any

Provider. There is a variety of SAML bindings for various use cases: SOAP (usually over HTTP), PAOS (reverse SOAP), HTTP Redirect, HTTP Post, HTTP Artifact, and SAML URI.

Implementation guidelines

The SAML standard has to be implemented in both *Provider* and *Consumer* side. The *Provider* has to support the requests envisaged by the protocol and the *Consumer* has to issue proper requests and consume the responses. A set of implementation guidelines[62] have been produced that include guidelines for user agent considerations, security considerations, authentication mechanisms, privacy principles, and guidelines for mobile environments.

Assessment

SAML allows security systems and application software to be developed and evolve independently because it offers a set of interoperable standard interfaces that allow for faster, cheaper, and more reliable integration[63]. Many groups benefit from the use of SAML. Producers of security software benefit from having standard schemas and protocols for expressing security information. Application developers benefit from decoupling their software from the underlying security infrastructure. Finally, users benefit because SAML promotes single sign-on (the ability to use a variety of Internet resources without having to log in repeatedly) and personalized user experiences in a privacy-friendly way.

For what is concerned with the ***implementation costs***, Consumers can reduce the cost of maintaining account information by adopting SAML to 'reuse' a single act of authentication (such as logging in with a username and password) multiple times across multiple services.

For what is concerned with the ***effectiveness***, being an agreement based approach it is by definition highly effective for what is captured by the standard.

3.3 Functionality Domain Interoperability Best practices and Solutions

Function Interoperability approaches can be divided in three main classes: *(i)* approaches oriented to resolve interoperability issues at the level of function interface (cf. Section 3.3.1), *(ii)* approaches oriented to resolve interoperability issues at the level of function behaviour (cf. Section 3.3.2) and approaches oriented to *(iii)* resolve interoperability issues at the level of function constraints (cf. Section 3.3.3).

3.3.1 Function Interface Reconciliation Approaches

Several approaches with varying degree of automation can be applied to resolve interface interoperability problems. Nevertheless one may classify them according to their level of automation into: (a) *standard-based* and (b) dynamic/mediation-based approaches. Both kinds of approaches rely upon the use of function descriptions (usually semantically enhanced ones) for the specification of important properties.

As for the standard-based approaches, they propose a relatively static approach consisting in the specification of predefined interfaces for certain types of services. The following ones are discussed: *Function Interface Specification Primitives* (cf. Section 3.3.1.1), *RosettaNet* (cf. Section 3.3.1.2) and *e-Framework* (cf. Section 3.3.1.3).

As for the dynamic/mediation-based approaches, they are essentially based on the (either semi-automated or fully automated) utilization of *Adapters* (Dumas, Benatallah, Hamid, & Nezhad, 2008), which can be provided in either an automated or manual way. All these approaches are mainly based on the use of

[62]http://xml.coverpages.org/SAML-ImplementationGuidelinesV01-8958.pdf

[63] http://saml.xml.org/advantages-saml

appropriate function (or service) specification primitives (cf. Section 3.3.1.1). We need to state here that all these approaches are mainly research outcomes that have not been tested on a product or industrial scale. The following ones are discussed. *Yellin and Storm* (cf. Section 3.3.1.4) propose an approach which facilitates the interoperation of components on an interface and protocol level. Based on the use of appropriate semantics and Finite State Machine model, they provide appropriate mechanisms that are able to (semi)automatically synthesize component adapters. *Benatallah et al.* (cf. Section 3.3.1.5) present a semi-automated approach which exploits manually defined templates for accommodating both interface and behavioral incompatibilities. Differences between services are captured using mismatch patterns which also help in analyzing and resolving them. *Bordeaux et al.* (Bordeaux, Salaün, Berardi, & Mecella, 2004) provide a formal-based approach for evaluating and accommodating the compatibility of services with respect to interface and behavior aspects. They exploit π-calculus to formally represent properties of services conveyed in service description protocols and matchmaking algorithms to evaluate the interoperation of services. *Ponnekanti and Fox* (Ponnekanti & Fox, 2004) have also presented an approach which exploits static and dynamic analysis tools to evaluate the replaceability of services.

3.3.1.1 Function Interface Specification Primitives

Several approaches have been proposed to address the interface specification needs of the Service Oriented Computing domain. These include the following.

WSDL (Booth & Liu, 2007): is an XML-based language used for describing functional properties of Web services. It aims at providing self-describing XML-based definitions that applications, as well as people, can easily understand. WSDL enables one to separate the description of a Web service's abstract functionality from the concrete details of how and where that functionality is offered. This separation facilitates different levels of reusability. Moreover, it supports the distribution of work in the lifecycle of a Web service development and in the production of the WSDL document that describes it.

According to the standard the comprising primitives accommodate the necessary syntactic info to facilitate the invocation of services. Additional properties - specified in other proposal/standards - should be utilized to leverage enhanced operations such as the semantic discovery or the automated mediation of services.

SAWSDL (Farrell & Lausen, 2007): the Semantic Annotations for WSDL and XML Schema (SAWSDL) defines a set of extension attributes for the Web Services Description Language and XML Schema definition language that allows description of additional semantics of WSDL components. The specification defines how semantic annotation is accomplished using references to semantic models, e.g., ontologies. SAWSDL does not specify a language for representing the semantic models, but it provides mechanisms by which concepts from the semantic models, typically defined outside the WSDL document, can be referenced from within WSDL and XML Schema components using annotations.

More specifically SAWSDL focuses on semantically annotating the abstract definition of a service to enable dynamic discovery, composition and invocation of services. The provided extensions annotate parts of a WSDL document such as input and output message structures, interfaces and operations and fit within the WSDL 2.0 (Booth & Liu, 2007), WSDL 1.1 (Christensen, Curbera, Meredith, & Weerawarana, 2001) and XML Schema (Thompson, Beech, Maloney, & Mendelsohn, 2004) extensibility frameworks.

OWL-S (Martin, et al., 2004): The Web Ontology Language for Web Services (OWL-S) is an ontology of services that enables users and software agents, with a high degree of automation, to discover, invoke, compose and monitor services with particular properties that are offered by web resources. OWL-S's approach for semantically describing Web Services is driven by the Upper Ontology according to which each service (instance of Service element) is provided by a resource (instance of Resource element) that (a) presents a ServiceProfile, (b) is described by a ServiceModel and (c) supports a ServiceGrounding.

These features cater for the following aspects:

- The ServiceProfile class answers the question of what the service does, providing all information needed for service-discovery. A service may be presented with more than one profile, and a profile may describe more than one service.
- The ServiceModel describes how a client can use the service and how this service works, i.e., what happens when the service is carried out. In case of services composed of several activities, a ServiceModel can also be used by a service-seeking operation.
- The ServiceGrounding defines the details of how an agent can access the service, by describing communication protocols, message formats, port numbers and other service details that are needed.

WSMO: The Web Service Modeling Ontology (WSMO) is a European initiative which aims to provide a standard for describing semantic web services. It has been based on the work of Fensel and Bussler (Fensel & Bussler, 2002) and it is operated by the SDK Cluster, which is a project cluster of the FP6 projects SEKT[64], DIP[65] and Knowledge Web[66].

WSMO consists of three parts:

- WSMO specifies a formal ontology and language for describing various aspects related to Semantic Web Services;
- WSML is an ongoing work to develop a proper formalization language for semantic web services and for providing a rule-based language for the semantic web;
- WSMX is an ongoing work to create an execution environment for the dynamic discovery, selection, mediation, invocation and inter-operation of the Semantic Web Services. WSMX is going to be a sample implementation for the Web Services Modelling Ontology (WSMO) which describes all the aspects of the Semantic Web Services.

According to its specification, the WSMO ontology is specialized into four distinct types of elements, i.e., Ontology, Service, Goal and Mediator.

- Ontologies introduce the terminology that is used in the other elements,
- Service contains the definition of services,
- Goals describe problems that are addressed by these services and
- Mediators resolve interoperability problems among goals, ontologies or services.

3.3.1.2 RosettaNet

RosettaNet (RosettaNet Community, 2010) is an on-line community and standardization body which has established a set of protocols and standards so as to facilitate B2B transactions among trading partners (requesting and providing partners) adhering to the supply chain business model. The provided standards have been endorsed by more than 500 all-size companies around the world performing a great amount of transactions on a daily basis.

It establishes a set of global and open standards that specify a wide range of collaboration aspects ranging from messages and supported processes to dictionaries and implementation requirements in a technology neutral manner. Appropriate groundings to the predominant message exchange technologies, i.e., Web

[64] http://www.sekt-project.com/

[65] http://dip.semanticweb.org/

[66] http://knowledgeweb.semanticweb.org/

Services, AS/2 and ebMS, for the supported message exchanges have been specified. The range of supported processes (Partner Interface Processes - PIPs) includes seven groups of core business processes that represent the backbone of the trading network. Each group is broken down into segments, i.e., cross-enterprise processes involving more than one type of trading partners.

According to the interoperability framework (cf. Section 2):

- the ***Provider*** is the providing partner offering specific business functionality while the ***Consumer*** is the requesting partner interested in using available business functionality;
- the ***Resource*** the two entities are willing to share is any kind of *supported functionality defined in terms of business processes;*
- the ***Task*** is the service the Provider is willing to offer. Typical services are selling goods or performing inventory checks;
- the solution belongs to the **agreement-based approaches** (cf. Section 2.2.1).

Requirements

From the ***Organisational*** point of view, the *Provider* agrees to expose the provided functionality in the form of web services. The provided service will be available at specific locations. The *Consumer* agrees to access the provided functionality by interacting with the related web service.

From the ***Semantic*** point of view, the *Provider* and the *Consumer* should share a common understanding on the associated entities, business and transaction notions.

From the ***Technical*** point of view, the *Provider* and the *Consumer* rely on standardized web service concepts and technologies, e.g., SOAP, WSDL and HTTP protocols.

Results

From the ***Organisational*** point of view, RosettaNet ensures that the *Provider* will offer the specified functionality in terms that can be understood by the requesting *Consumer*. The *Consumer*, by using the specified mechanisms, e.g., web service standards, will be able to access the provided functionality.

From the ***Semantic*** point of view, this approach guarantees that the collaborating parties will share the same understanding on all related entities and activities, i.e., the notion of products, supported interactions, e.g., selling/buying.

From the ***Technical*** point of view, both the *Provider* and the *Consumer* will use the same technology, e.g., web services and HTTP, for interacting with each other. Data formats and interaction protocols are all predefined and mutually agreed, thus no further mediations are needed.

Implementation guidelines

By relying on well defined and accepted technologies, e.g., web services, and standards, RosettaNet ensures that the implementations required by the interacting parties are straightforward. All parties, e.g., Consumer and Provider, have to implement the necessary functionality in terms specified by the RosettaNet framework. Details on the implementation guidelines and the used mechanisms and approaches are provided by RosettaNet.

Assessment

RosettaNet is an approach which ensures the successful interaction among collaborating parties by relying on mutually agreed and pre-specified mechanisms and concepts. This pre-defined and ‘standardized’ approach has been well tested and accepted by several organizations which have embraced this approach. Therefore, the effectiveness of this approach is already ensured and evaluated.

Regarding the accruing implementation costs, these are relative low as most of the existing organizations have already accepted and embraced the associated implementation technologies. Nonetheless, considerable costs may accrue from transformations and modifications that may be needed in order to ensure the compliance of the back-office technologies and legacy systems with the shared concepts and interaction semantics specified by RosettaNet.

3.3.1.3 e-Framework

Similar to RosettaNet, the e-Framework approach adheres to a pre-defined, standard based manner for ensuring interface, semantics and behavioural compliance of interacting parties. Nonetheless, the application domain of this approach is on Education and Research infrastructures. It also adheres to the Service Oriented Architecture model and aims to provide a knowledge base which will promote interoperability. To ensure this, the provided knowledge base will comprise (*i*) a set of services and their descriptions, (*ii*) sets of service usage models (SUMs), and (*iii*) sets of guides, methodologies and analyses.

According to the interoperability framework (cf. Section 2):

- the ***Provider*** is the providing partner offering specific functionality while the ***Consumer*** is the requesting partner interested in using available functionality;
- the ***Resource*** the two entities are willing to share is any kind of *supported functionality defined in terms of provided services;*
- the ***Task*** is the service the Provider is willing to offer;
- the solution belongs to the **agreement-based approaches** (cf. Section 2.2.1).

Requirements

From the ***Organisational*** point of view, the *Provider* agrees to expose the provided functionality in the form of web services with a pre-specified interface, whilst the *Consumer* agrees to access the provided functionality by interacting with the related web service.

From the ***Semantic*** point of view, the *Provider* and the *Consumer* share a common understanding of all associated entities and transaction notions.

From the ***Technical*** point of view, the *Provider* and the *Consumer* rely on standardized web service concepts and technologies, e.g., SOAP, WSDL and HTTP protocols.

Results

From the ***Organisational*** point of view, it ensures that the *Provider* will offer the specified functionality in terms that can be understood by the requesting *Consumer*. The *Consumer*, by using the specified mechanisms, e.g., web service standards, will be able to access the provided functionality.

From the ***Semantic*** point of view, this approach guarantees that the collaborating parties will share the same understanding of all related entities and activities.

From the ***Technical*** point of view, both the *Provider* and the *Consumer* will use the same technology, e.g., web services, for interacting with each other. Data formats and interaction protocols are all predefined and mutually agreed.

Implementation guidelines

By relying on well defined and accepted technologies, e.g., web services, and standards it ensures that the implementations required by the interacting parties can easily be provided. All parties, e.g., Consumer and Provider, have to implement the necessary functionality in terms specified by the framework. Details on the implementation guidelines and the used mechanisms and approaches are provided by e-Framework.

Assessment

This is an approach which ensures the successful interaction among collaborating parties by relying on mutually agreed and pre-specified mechanisms and concepts. This approach has been tested and accepted by several institutions which have embraced it. Therefore its effectiveness is already ensured.

Regarding the accruing implementation costs these are relative low as most of the existing organizations have already accepted and embraced the associated implementation technologies. Nonetheless, similarly to RosettaNet considerable costs may accrue from transformations and modifications that may be needed in order to ensure the compliance of the back-office technologies and legacy systems with the shared concepts and interaction semantics specified by e-Framework.

3.3.1.4 Yellin and Storm

The approach presented by Yellin and Storm (Yellin & Strom, 1997) facilitates the interoperation of components on an interface and protocol level. The approach includes appropriate mechanisms that are able to automatically synthesize component adapters. The provided adapters enable component composition when they are functionally compatible.

The synthesis process is based on the use of interface mappings between incompatible interface specifications. An interface mapping allows a user to specify the important characteristics required by an adapter that should mediate between components containing these interfaces. The constructed adapters are thus able to accommodate both interface and protocol level incompatibilities.

According to the interoperability framework (cf. Section 2):

- the ***Provider*** is the partner offering a component to be used while the ***Consumer*** is the partner interested in integrating the available components to support the provision of composite applications;
- the ***Resource*** is the component implementing the required functionality*;*
- the ***Task*** is the service the Consumer is offering via the Provider's component;
- the solution belongs to the **automatically constructed mediation approaches** (cf. Section 2.2.2).

Requirements

From the ***Organisational*** point of view, the *Provider* agrees to provide the necessary component description which includes interface and protocol specification primitives, whilst the *Consumer* agrees to access the provided functionality by interacting with the provided component.

From the ***Semantic*** point of view, the *Provider* and the *Consumer* share a common language for the description of component interface and protocol characteristics.

From the ***Technical*** point of view, the *Provider* and the *Consumer* rely on the use of the same component specification language and component provision technology, i.e., CORBA.

Results

From the ***Organisational*** point of view, it ensures that the *Provider* will offer the specified component descriptions in terms that can be understood by the *Consumer*. The *Consumer* based on the provided specifications will be able to integrate the provided components via automatically constructed adapters.

From the ***Semantic*** point of view, this approach guarantees that the collaborating parties will share the same understanding on all related entities and activities. The provided adapters will be able to alleviate the semantic incompatibilities among the collaborating components.

From the ***Technical*** point of view, both the *Provider* and the *Consumer* will use the same technology, e.g., CORBA, for interacting with each other. Data formats and interaction protocols are all specified using the commonly agreed specification languages.

Implementation guidelines

This approach has been evaluated in a research project called Global Desktop project (Huynh, Jutla, Lowry, Strom, & Yellin, 1994). Implementation details on the provided components and the respective interacting components can be retrieved in (Huynh, Jutla, Lowry, Strom, & Yellin, 1994). The provided components are all deployed over CORBA and described using a proprietary language in terms of their protocol and interface characteristics.

Assessment

The provided mechanism is able to generate adapters via the synthesis of simpler parts. The synthesis process is guided by the use of a Finite State Machine model which provides a representation of protocol characteristics. The related **implementation costs** are relatively low as no extra effort is required for the provision of the required specifications.

Regarding its **effectiveness**, the proposed approach cannot ensure the provision of component adapters at all cases. In certain cases the integration of components via the use of automatically constructed adapters is not feasible. Moreover, the provided description mechanisms and synthesis algorithms are unable to capture the semantics of asynchronous interactions among components.

We need to state here that this approach has been tested in the context of the Global Desktop project. No evidence of additional evaluation of the proposed solution has been found.

3.3.1.5 Benatallah et al.

The approach presented by Benatallah at al. (Benatallah, Casati, Grigori, Nezhad, & Toumani, 2005) is a semi-automated approach which exploits manually defined templates for accommodating both interface and behavioral incompatibilities. Differences between services are captured using mismatch patterns which also help in analyzing and resolving them. Mismatch patterns include a template of business logic that can be used to semi-automatically develop adapters to handle the mismatch captured by each pattern. Furthermore, Benatallah et al. provide a number of built-in patterns corresponding to the possible mismatches that have been identified at the interface and protocol levels.

The mechanisms employed for the specification of behavioral and interface descriptions are all based on existing Web service protocols, i.e., BPEL and WSDL. Appropriate adapter templates are already described for many cases. These templates are instantiated and possibly further refined by developers in order to accommodate the incompatibilities among services. More specifically for each adapter the template of required information includes elements such as those presented in Table 1.

Table 1: Adapter information template

Name	Name of the pattern
Mismatch Type	A description of the type of difference captured by the pattern
Template parameters	Information that needs to be provided by the user when instantiating an adapter template to derive the adapter code
Adapter template	Code or pseudo-code that describes the implementation of an adapter that can resolve the difference captured by the pattern
Sample usage	The sample usage section contains information that guides the developer

According to the interoperability framework (cf. Section 2):

- the ***Provider*** is the partner offering specific functionality in terms of a service while the ***Consumer*** is the requesting partner interested in using the Provider's service;
- the ***Resource*** the two entities are willing to share is the implemented functionality offered as a service.
- the ***Task*** is any functionality, the Consumer plans to realise by relying on the Provider's service.
- the solution belongs to the **semi-automated adaptation approaches**.

Requirements

From the ***Organisational*** point of view, the *Provider* agrees to offer the specified service along with the required descriptions in terms that can be understood by the *Consumer*. The provided specifications are described using existing web service related standards such as BPEL and WSDL. The *Consumer* based on the provided specifications will be able to access the provided service via the use of appropriate adapters.

From the ***Semantic*** point of view, both the Consumer and the Provider of a service should have a common understanding of related entities and interaction semantics. Note that this approach does provide for validating the semantic compliance of the Consumer and the Provider parties.

From the ***Technical*** point of view, both the *Provider* and the *Consumer* will use the same technology and standards for the description of interface and behavioural details, i.e., WSDL and BPEL. Additional constraints related to data conformance can be expressed using notation such as XQuery.

Results

From the ***Organisational*** point of view, this approach ensures that the Consumer will be able to interact with the provided service if there is an adapter template that can be used to reconcile the interface and behavioural differences between the provided service and the Consumer request. In case appropriate fine-tuning is required this can be provided on the template code generated by the adaptation template.

From the ***Semantic*** point of view, this approach provides no mechanism which validates and ensures that the collaborating parties will share the same understanding on all related entities and activities. The collaborating parties should have appropriate means which will ensure semantic interoperability.

From the ***Technical*** point of view, both the *Provider* and the *Consumer* will use the same technology i.e., Web services, and widely accepted standards such as WSDL and BPEL.

Implementation guidelines

Both the Consumer and the Provider parties of this approach should adhere to the service oriented computing paradigm. More specifically, the provided services are offered as web services whereas the languages used for the description of interface and behavioural characteristics are WSDL and BPEL respectively. Additional implementation and specification details can be retrieved from (Benatallah, Casati, Grigori, Nezhad, & Toumani, 2005).

Assessment

The proposed approach provides a semi-automated mechanism for the provision of service adapters that are able to accommodate interface and behavioral service discrepancies. The provided mechanism depends on the provision of appropriate adaptation templates which guide the generation of required service adapters.

In terms of implementation costs, the proposed approach doesn’t enforce any compliance to specific interfaces or interaction patterns. Considering that (a) most of the existing functionality is already provided in terms of services, i.e., web services; (b) the provision of interface, e.g., WSDL, and behavioral specifications, e.g., BPEL, can be easily provided; then the implementation costs are relatively low.

As far as the effectiveness of the approach is concerned this primarily depends on the existence of appropriate adaptation templates. Several, adaptation patterns have already been indentified and specified by the proposers, however additional ones related to specific cases should be provided by the interested parties.

3.3.2 Function Behaviour Reconciliation

Function behaviour can be conceived as the set of possible interactions supported by a specific function. Function behaviour expresses the perception of an external observer. It can be described in terms of supported input/output exchanges their (logical or time) ordering and constraints. It therefore becomes apparent that interoperable functions should support interoperable (compatible) behaviours. Incompatible orderings of input/output exchanges or interaction constraints may hinder the integration of functions into more complex ones.

As it has been noted in several research efforts, function interoperability is tightly related to the anticipation of interface, pre/post condition and behaviour concerns. This has been the reason why contemporary approaches catering for the interoperation of functions do not tackle interface, pre/post condition and behaviour issues in isolation. Therefore, both static and dynamic approaches presented in Section 3.3.1 'Function Interface Reconciliation Approaches' resolve behavioural concerns as well. Approaches such as RosettaNet (cf. Section 3.3.1.2) and e-Framework (cf. Section 3.3.1.3), accommodate behavioural concerns through the specification of behavioural patterns (or cases) in the form of PIPs and SUMs respectively. Collaborating parts should adhere to the roles and transactions specified in the predefined behavioural patterns so as to interoperate. In addition to that, there are protocols and standards that have been conceived to capture service behaviour as discussed in Section 3.3.2.1.

For Mediator-based approaches, they rely upon the use of formal behavior specification mechanisms, e.g., Finite State Models and π-calculus, and appropriate algorithms so as to assert and accommodate the behavioral compatibility of distinct functions. Further to the mediation-based approaches presented in Section 3.3.1 'Function Interface Reconciliation Approaches', approaches primarily concerned with the behavior interoperability issues are the following ones:

- **AI-based**: Most of the existing approaches employed for the automated composition of services illustrated in (Jinghai & Xiaomeng, 2004) address behavior interoperability through the use of Finite State Machine Models and appropriate AI planning techniques. State Transition Systems are extracted out of service descriptions (usually semantically enhanced) and, depending on the constraints enforced by each approach, deterministic or non-deterministic planning algorithms are utilized to assert and provide service integrations;
- **Deng et al.**: Deng et al. in (Deng, Wu, Zhou, Li, & Wu, 2006) utilize π-calculus to model the service behavior and the interaction in a formal way. They also propose (i) a method based on the operational semantics of the π-calculus to automate the verification of compatibility between two services and (ii) an algorithm to measure the compatibility degree in a quantitatively manner;
- **Peng et al.**: Along the same lines, Peng et al. (Peng, et al., 2009) utilize a model of service behavior based on Petri-nets with weight. They also propose a formal method to verify and compute service behavior compatibility;
- **Stollberg et al.**: Building upon the WSMO (Fensel & Bussler, 2002), Stollberg et al. (Stollberg, Cimpian, Mocan, & Fensel, 2006) presented a mediation model able to handle and resolve heterogeneity that may occur in the Semantic Web Service domain. More specifically, the presented framework addresses four levels of issues pertaining to the Semantic Web: (*i*) Terminology, (*ii*) Representation Format and Transfer Protocol, (*iii*) Functionality and (*iv*) Business Process.

All these solutions are primarily based on the use of formal representations and models so as to assert the compatibility of services and to construct appropriate adaptors (or mediators) which can reconcile the discrepancies among services.

3.3.2.1 Function Behaviour Specification Primitives

Syntactic and semantic based approaches have been extensively applied in the Service Oriented Computing (SOC) domain for the description of service behavior. Service behavior is normally expressed in terms of performed interactions, i.e., message exchanges and their respective ordering, constraints, etc. perceived through the viewpoint of an external observer. Several protocols and standards have been proposed to accommodate the description of service behavior paving thus the way towards automated service interoperation.

In the following we present some of the most well-known approaches applied in the field of SOC:

- **WS-CDL** (Kavantzas, Burdett, Ritzinger, Fletcher, & Lafon, 2004): The Web Services Choreography (WS-CDL) is a W3C working draft which aims to precisely describe collaborations between any type of party regardless of the supporting platform or programming model used in the implementation of the hosting environment. WS-CDL leverages a "global" viewpoint that facilitates the specification of the common ordering conditions and constraints under which messages are exchanged. The provided specification describes the common and complementary observable behaviour of all the parties involved. Each party can then use the global definition to build and test solutions that conform to it.

  According to (Kavantzas, Burdett, Ritzinger, Fletcher, & Lafon, 2004) the advantages accruing from the introduction of a contract based on a global viewpoint as opposed to anyone endpoint is that "*it separates the overall 'global' process being followed by an individual business or system within a 'domain of control' (an endpoint) from the definition of the sequences in which each business or system exchanges information with others. This means that, as long as the 'observable' sequences do not change, the rules and logic followed within a domain of control (endpoint) can change at will and interoperability is therefore guaranteed*".
- **WSCL** (Banerji, et al., 2002): The Web Services Conversation Language is a W3C Note that facilitates the definition of the abstract interfaces of Web services, i.e., the business level conversations or public processes supported by a Web service. WSCL accommodates the specification of the exchanged XML documents and of the allowed sequencing of these document exchanges. WSCL conversation definitions are XML documents themselves, and therefore can be handled by Web services infrastructures and development tools. WSCL may be used in conjunction with other service description languages like WSDL; for example, to provide protocol binding information for abstract interfaces, or to specify the abstract interfaces supported by a concrete service.

  The WSCL note has been superseded by the WS-CDL language and thus can be only considered as an influencing ancestor which clearly documents the need for such a protocol. WSCL focus was on public processes in which the participants of a Web service engage, thus private application logic or private process were not considered, i.e., the internal implementation and mapping to back-end applications within the various enterprises that are interacting are not taken into account.
- **WS-BPEL** (Alves, et al., 2007): The Web Service - Business Process Execution Language is an OASIS standard providing appropriate notation for the description of abstract and executable business processes. In doing so, it extends the Web Services interaction model and enables it to support business transactions. WS-BPEL defines an interoperable integration model that should facilitate the expansion of automated process integration in both the intra-corporate and the business-to-business spaces.

  Abstract business processes are partially specified processes that are not intended to be executed. Such processes may hide some of the required concrete operational details and serve a descriptive role, with more than one possible use case, including observable behaviour and process templates. WS-BPEL therefore caters for the representation of set of publicly observable behaviours in a standardized fashion. In doing so, it includes information such as when to wait for messages, when to send messages, when to compensate for failed transactions, etc.

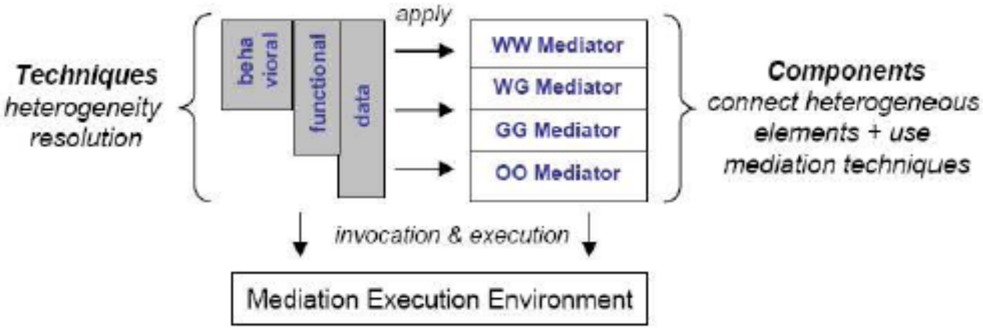

Figure 4. WSMX Mediation Model

The syntactic primitives used for the representation of this information constitute BPEL's Common Base. However, the common base does not have well-defined semantic which is provided by appropriate usage profiles. A usage profile defines the necessary syntactic constraints and the semantic based on Executable WS-BPEL Processes for a particular use case for an Abstract Process. Every Abstract Process must identify the usage profile that defines its meaning via a URI. This is an extensible approach as new profiles can be defined whenever different areas are identified.

- **OWL-S** (Martin, et al., 2004): An OWL-S Service Model conveys the necessary information to facilitate the semantic-based description of service behaviour. Influenced by the protocols applied in Agent-based systems, the OWL-S ServiceModel accommodates appropriate semantic extensions in addition to the exchanged information and the respective ordering. This set of extensions have facilitated the provision of automated approaches towards the integration of services (Sirin, Parsia, Wu, Hendler, & Nau, 2004).

3.3.2.2 Strollbert et al.

The anticipation of incompatibilities is one of the core challenges that the Semantic Web vies to address. The presented approach (Stollberg, Cimpian, Mocan, & Fensel, 2006) provides a model that is able to handle incompatibilities among semantically described services. Based on the WSMO model and WSML descriptions, several types of mediators ranging from syntactic to semantic ones can be provided.

In order to attain a mediator-oriented architecture in accordance to Wiederhold's conception (Wiederhold & Genesereth, 1997), the presented approach distinguishes two dimensions: (1) the *mediation techniques* for resolving different kinds of heterogeneities that can arise within the Semantic Web, (2) *logical components* that connect resources and apply required mediation techniques; these are embedded in a software architecture for dynamic invocation and execution. The applied mediation model is graphically illustrated in Figure 4.

The provided mediators are incorporated into the Web Service Execution Environment (WSMX) which accommodates the execution of service integration. More specifically, to facilitate behavioural issues WSMX provides a Process Mediator (Cimpian & Mocan, 2005). This mediator is based on appropriate service behavioural descriptions. It accommodates several techniques such as message blocking, message splitting or aggregation, acknowledgements generation and so on. Furthermore, being part of WSMX environment, Process Mediator can make use of all the functionalities provided by WSMX regarding message receiving and sending, keeping track of the ongoing conversation, access various Data Mediators, resources and so on. Process mediator is therefore able to anticipate several types of behavioural discrepancies that may come up in the case of semantic web service integration.

According to the interoperability framework (cf. Section 2):

- the ***Provider*** is the partner offering specific functionality in terms of a semantically annotated service while the ***Consumer*** is the requesting partner interested in using the provided service;
- the ***Resource*** is the implemented functionality offered as a semantically described service*;*
- the ***Task*** is the service the Provider is offering;
- the solution belongs to the **semi-automated adaptation approaches**.

Requirements

From the ***Organisational*** point of view, the *Provider* agrees to offer the specified service along with the required semantic-based descriptions in terms that can be understood by the *Consumer*. The provided specifications are described using existing Semantic Web service standards such WSML. The *Consumer* based on the provided specifications will be able to access the provided service via the use of appropriate adapters.

From the ***Semantic*** point of view, both the Consumer and the Provider of a service should use appropriate semantic annotations. The utilized semantic descriptions should be compatible. Their compatibility is ensured via the use of appropriate mediators.

From the ***Technical*** point of view, both the *Provider* and the *Consumer* will use the same technology and standards for the description of services, i.e., WSML.

Results

From the ***Organisational*** point of view, this approach ensures that the Consumer will be able to interact with the provided service if appropriate mediators exist at any of the required levels, e.g., terminology, representation, functionality or business process level.

From the ***Semantic*** point of view, this approach provides a mechanism which validates and ensures that the collaborating parties will share the same understanding of all related entities.

From the ***Technical*** point of view, both the *Provider* and the *Consumer* will use the same technology, i.e., Semantic Web services, and standards such as WSML.

Implementation guidelines

Both the Consumer and the Provider parties of this approach should adhere to the Semantic Web service paradigm. More specifically, the provided services are offered as semantic web services described in WSML. For the application of the provided mediation techniques the WSMX environment should be used as the underlying basis. More details on the required implementation constraints are provided in (Stollberg, Cimpian, Mocan, & Fensel, 2006).

Assessment

The proposed approach caters for the provision of service adapters that are able to accommodate semantic, interface and behavioral service discrepancies. The provided mechanism depends on the use of semantic service descriptions.

In terms of implementation costs the proposed approach doesn’t enforce the compliance to specific interfaces or interaction patterns. The provision of appropriate semantic service descriptions is of relatively low cost. Additional costs may accrue from the use of the WSMX engine as the underlying execution environment.

As far as the effectiveness of the approach is concerned, this primarily depends on the existence of appropriate mediators. In addition the use of semantic based mediation techniques results into considerable performance degradations.

3.3.3 Function Conditions Modelling

Asserting whether two functions may interoperate heavily relies on the assessment of the conditions that must hold prior to and after the invocation of these functions. Pre-conditions and Post-conditions provide a specification of the conditions that hold for a function to be called and the ones that hold after its execution, respectively. The provision of such specifications was a basis upon which formal methods and tools for the validation of software systems have been built.

Several approaches have been utilized up to now to facilitate the specification of such conditions. Assertions along with boolean functions and expressions have been quite keen in programming languages e.g., Eiffel, Java, C/C++, etc. Assertions provide the necessary means to accommodate the specification of pre/post conditions that are evaluated either at design or execution time. Along the same lines, the Service Oriented Computing paradigm utilizes a more declarative approach where such conditions are provided as information elements conveyed in semantic description documents.

Further to these approaches, Unified Modeling Language (UML), which is a widely accepted standard used for the description of software systems, has adopted Object Constraint Language (OCL) as a mechanism used for describing constraints on UML model. Nonetheless as it is documented by OMG, OCL expressions can seamlessly be used to specify application specific constraints on operations/actions that, when executed, do alter the state of the system.

In the following we provide a brief presentation of mechanisms and approaches that have been used for the description of such conditions.

3.3.3.1 Function Pre/Post Condition Specification Primitives

The use of pre/post conditions for the evaluation of software systems has been widely accepted by modern programming languages. Such conditions enable the provision of formal methods and tools that are able to validate systems either at design or execution time. Assertions constitute the most common approach towards the implementation of such mechanisms in several programming languages e.g., Eiffel, Java and C/C++.

Eiffel: Eiffel is an ISO standardized Object-Oriented language that has adopted assertions for the specification of pre and post conditions. Assertions have been accommodated so as to comply with the 'Design By Contract' software engineering approach.

More specifically, the constructs employed by Eiffel for the specification and evaluation of pre/post conditions include:

- Routine precondition denoted with the keyword ‘require’;
- Routine postcondition denoted with the keyword ‘ensure’;
- Class invariant.

Furthermore, the language supports a "Check instruction" (a kind of "assert") and, in the syntax for loops, clauses for a loop invariant and a loop variant.

Java: Assertions is a mechanism that has been introduced in Java version 1.4. An assertion is a statement in the Java programming language that enables one to test his/her assumptions about a program.

Each assertion contains a Boolean expression that one believes it will be true when the assertion executes. If it is not true, the system will throw an error. By verifying that the Boolean expression is indeed true, the assertion confirms the assumptions about the behaviour of a program, increasing the confidence that the program is free of errors.

Though assertions in Java is not a full-blown design-by-contract feature, it can be used as a mechanism to accommodate the specification of pre/post conditions and class invariants.

C/C++: Similar to what is applied in Java, C (and C++) utilizes a macro function called ‘assert’ to evaluate the validity of a logical expression. In the case of an invalid logical expression the program terminates and an assertion violation error is returned.

The evaluation of assertion expressions is controlled via appropriate structures at the compilation time.

As in Java, the ‘assert’ function can be used to specify pre/post conditions and invariants that are evaluated at runtime. Therefore application specific conditions may be set at appropriate places so as to mimic the required and ensured conditions.

In the SOC domain the description of pre/post conditions of services is based on a more declarative approach. Syntactic service descriptions fail to accommodate this need, hence semantic service description approaches are the only ones addressing this topic.

OWL-S and WSMO are two of the most widely known proposals for the semantic description of services that inherently support the description of pre and post conditions. Their support for pre/post conditions has been primarily based on the decision to facilitate the automated utilization and interoperation of such services.

OWL-S: One of the three parts comprising an OWL-S description document is the Service Profile. A service profile provides a description of what a service might offer. An integral part of the provided set of information elements are the preconditions and results (i.e., postconditions) of a service at hand.

More specifically, OWL-S utilizes elements such as ‘*hasPrecondition’* and ‘*hasResult’* to facilitate the description of conditions that should hold for a service to be called and of outcomes that may be achieved. According to the OWL-S specification for a service to be invoked, the precondition specified by the *‘hasPrecondition’* element should be true. Depending on the possible outcomes that a service may return, several instances of the *‘hasResult’* element may be specified.

Each *‘hasResult’* element has a set of associated properties which convey the following information:

- *‘inCondition’*: this property specifies the condition under which this result (and not another) occurs;
- *‘withOutput’* and ‘*hasEffect’*: these properties state what ensues when the condition is true;
- *‘hasResultVar’*: this property declares variables that are bound in the ‘*inCondition’*;

WSMO: An integral property of the Service element is *‘Capability’* which defines what a Web service offers in terms of pre/post conditions, assumptions and effects. According to WSMO a Web service defines one and only one ‘*Capability*’ by specifying the next elements: non-functional properties, imported ontologies, used mediators, shared variables, preconditions, post-conditions, assumptions, and effects.

Preconditions, assumptions, post-conditions and effects are expressed through a set of axioms. A set of shared variables can be declared within the *‘Capability’*. These variables are implicitly all-quantified and their scope is the whole Web service *‘Capability’*. Thus, informally a Web service capability is: for any values taken by the shared variables, the conjunction of the precondition and of the assumption implies the conjunction of the post-condition and of the effect.

3.3.3.2 Function Pre/Post Condition Reconciliation Approaches

Pre/Post condition issues have always been confronted along with interface and behavioral issues. Thus, the approaches presented in

Section 3.3.1 and Section 3.3.2 address this problem as well. A prerequisite in these approaches is the use of formalized representations of pre/post conditions such as the ones used by WSMO and OWL-S.

3.4 Policy Domain Interoperability Best practices and Solutions

Digital libraries (together with digital repositories and data centres) represent the confluence of vision, mandate and the imagined possibility of content and services constructed around the opportunity of use. Underpinning every digital library is a policy framework. It is the policy framework that makes them viable - without a policy framework a digital library is little more than a container for content. Even the mechanisms for structuring the content within a traditional library building as a container (e.g., deciding what will be on what shelves) are based upon policy. Policy governs how a digital library is instantiated and run. The policy domain is therefore a meta-domain which is situated both outside the digital library and any technologies used to deliver it, and within the digital library itself. That is, policy exists as an intellectual construct, that is deployed to frame the construction of the digital library and its external relationships, and then these and other more operational policies are represented in the functional elements of the digital library. Policy permeates the digital library from conceptualisation through to operation and needs to be so represented at these various levels.

Policy interoperability is typically not only point-to-point and bilateral, but wider and richer, implying tier levels across the diverse actors operating at the organisational, semantic and technical levels. Furthermore, some of the most interesting policy interoperability use cases take place either when there are interactions between equals (e.g., a digital library agrees to become interoperable with another on some basis) or according to a hierarchical model of interaction (e.g., like in DRIVER[67], where all participating repositories are required to conform to DRIVER standards).

In this Cookbook, the DL.org Policy Working Group therefore suggests that rather than 'solutions', for policy interoperability it is more appropriate to talk about a 'future' state: not necessarily only best practices but a state of desire that digital libraries stakeholder will try to put into practice. Some elements are not in place today, but would be envisioned as necessary for the interoperability of polices directing digital libraries. Some desired areas for Policy interoperability are e.g., related to access policies (e.g., authentication and authorisation, Service Level Agreements for presenting content) and licensing policies (as documented in the recent 'Public Consultation draft of Europeana Licensing Framework'[68], which provides models for licensing agreements with data providers). In both cases, making policies machine-readable would make them easier to manage. A useful focus would therefore also be on human-machine interaction: for example, licensing policies interoperability might be achieved automatically in the near future.

A cluster of approaches supporting policy interoperability have been identified through a state of the art investigation conducted by the DL.org Policy Working Group, and a survey conducted with the following targeted international organisations and initiatives managing medium/large-scale digital libraries and federated services: ACM Digital Library[69]; Calisphere, California Digital Library (CDL)[70];

[67] http://validator.driver.research-infrastructures.eu/

[68] http://www.europeanaconnect.eu/documents/eConnect_D4.1.1_Europeana%20Licensing%20Framework_v.1.0.pdf

[69] www.portal.acm.org/dl.cfm

[70] www.calisphere.universityofcalifornia.edu/

DANS[71]; DRIVER[72]; E-LIS[73]; Europeana[74]; Ithaka, JSTOR, Portico[75]; LiberLiber[76]; NARA[77]; Nemertes[78]; National Science Digital Library (NSDL)[79]; Padua@Research[80]; UK Data Archive[81]; University of Chicago Digital Library Repository[82]; USGS Digital Library[83]. The results of the survey outline further directions for research in these areas, some of which will be explored more in depth in the final version of the DL.org Cookbook:

- *Approaches for access policy interoperability*: XML; EML - Election Markup Language (OASIS, 2007); METS (Library of Congress); DOI [84]; COUNTER 3 Code of Practice[85]; OpenURL Framework Standard[86]; W3C WAI WCAG - Web Content Accessibility Guidelines[87]; W3C Markup Validation Service[88]; US Federal Government Section 508 Guidelines[89]; DLF ERM Initiative[90];
- *Approaches for preservation policy interoperability*: PRONOM[91]; DROID[92]; JHOVE[93]; UDFR[94]; Global GDFR[95]; Planets Testbed Beta[96]; OAIS[97]; TRAC[98]; DRAMBORA Interactive toolkit[99]; LOCKSS[100]; Portico's Digital Preservation Service[101]; EAD [102]; METS[103]; OAI-PMH[104]; XML[105]; PREMIS[106]; DIDL[107]; DCMI[108]; MARC[109]; ONIX[110];
- Approaches for Network policy interoperability: iRODs[111]; WSDL[112]; XACML[113];

[71] www.easy.dans.knaw.nl

[72] http://www.driver-community.eu/

[73] http://eprints.rclis.org/

[74] www.europeana.eu/

[75] www.ithaka.org

[76] www.liberliber.it/

[77] www.archives.gov/

[78] http://nemertes.lis.upatras.gr

[79] www.nsdl.org

[80] http://paduaresearch.cab.unipd.it

[81] www.data-archive.ac.uk

[82] http://www.lib.uchicago.edu/e/dl/program.php3

[83] http://www.usgs.gov/

[84] www.doi.org/

[85] http://www.projectcounter.org/code_practice.html

[86] http://www.niso.org/kst/reports/standards?step=2&project_key=d5320409c5160be4697dc046613f71b9a773cd9e

[87] http://www.w3.org/TR/WCAG20/

[88] http://validator.w3.org/

[89] http://www.section508.gov/

[90] http://www.clir.org/dlf.html

[91] http://www.nationalarchives.gov.uk/PRONOM/Default.aspx

[92] http://droid.sourceforge.net/

[93] http://hul.harvard.edu/jhove/

[94] http://www.udfr.org/

[95] http://www.gdfr.info/

[96] http://testbed.planets-project.eu/testbed/

[97] http://public.ccsds.org/publications/archive/650x0b1.pdf

[98] http://www.crl.edu/archiving-preservation/digital-archives/metrics-assessing-and-certifying

[99] http://www.repositoryaudit.eu/

[100] http://lockss.stanford.edu/lockss/Home

[101] http://www.portico.org/digital-preservation/

[102] http://www.loc.gov/ead/

[103] http://www.loc.gov/standards/mets/

[104] http://www.openarchives.org/OAI/openarchivesprotocol.html

[105] http://www.w3.org/XML/

[106] http://www.loc.gov/standards/premis/

[107] http://xml.coverpages.org/mpeg21-didl.html

[108] http://dublincore.org/

[109] http://www.loc.gov/marc/

[110] http://www.editeur.org/15/Previous-Releases/

[111] https://www.irods.org/index.php/IRODS:Data_Grids,_Digital_Libraries,_Persistent_Archives,_and_Real-time_Data_Systems

- *Approaches for Intellectual property policy interoperability*: METS[114]; NLM XML DTDs for Journal Publishing, Archiving and Interchange[115]; PREMIS[116]; CC– Creative Commons licences[117]
- Approaches for authentication policy interoperability: XACML[118] ; Shibboleth[119]; Athens[120]
- *Approaches for evaluation and assessment policy interoperability*: DRIVER Guidelines 2.0 Guidelines for content providers; SHAMAN Assessment Framework[121]
- *Approaches for Policy Representation and Enforcement for policy interoperability:* PLEDGE project, AIR Policy Language[122]; iRODS rules[123]; SWRL[124]; Turtle RDF Triples[125]; REWERSE Policy Language[126]; OWL[127]; KAoS[128]; Web Services Policy Framework (WS-Policy)[129]; Web Services Policy 1.5[130]; WSPL[131]; XACML[132]; Rei[133].

In Sections 3.4.1 and 3.4.2, we provide trial descriptions of two sample potential approaches for policy interoperability using the previously described Interoperability Framework (Section 2).

3.4.1 Sample potential approach: EML Overview

The EML - Election Markup Language is a XML-based standard to support end-to-end management of election processes (OASIS, 2007). EML can be used when metadata transformations are needed in distributed environments with multiple digital libraries/repositories/archives.

According to the interoperability framework (cf. Section 2):

- the ***Provider*** is any digital library, repository or archive XY, and the ***Consumer*** is any digital library, repository or archive;
- the ***Resource*** the two entities are willing to share is any kind of original metadata referring to be transferred from one Provider to a Consumer. The same original digital library/repository/archive item might be available in different formats. Once received the transferred item, a new version is created by the Consumer in standard format. All the digital library/repository/archive items must be

[112] http://www.w3.org/TR/wsdl

[113] http://xml.coverpages.org/nlmJournals.html

[114] http://www.loc.gov/standards/mets/

[115] http://xml.coverpages.org/nlmJournals.html

[116] http://www.loc.gov/standards/premis/

[117] http://creativecommons.org/

[118] http://xml.coverpages.org/nlmJournals.html

[119] http://shibboleth.internet2.edu/

[120] http://www.athens.ac.uk/

[121] http://portal.acm.org/citation.cfm?id=1643823.1643899&coll=GUIDE&dl=GUIDE&CFID=63081623&CFTOKEN=61810568

[122] http://dig.csail.mit.edu/TAMI/2007/amord/air-specs.html

[123] https://www.irods.org

[124] http://www.w3.org/Submission/SWRL/

[125] http://www.w3.org/TeamSubmission/turtle/ http://en.wikipedia.org/wiki/Resource_Description_Framework

[126] http://rewerse.net/

[127] http://www.w3.org/TR/owl-features/

[128] http://www.w3.org/2004/08/ws-cc/kaos-20040904

129http://www.ibm.com/developerworks/library/specification/ws-polfram/ http://msdn.microsoft.com/library/en-us/dnglobspec/html/ws-policy.asp

[130] http://www.w3.org/TR/2007/REC-ws-policy-20070904/

[131] http://www-106.ibm.com/developerworks/library/ws-polas/

[132] http://en.wikipedia.org/wiki/XACML http://www.oasis-open.org/committees/xacml

[133] http://rei.umbc.edu/

exchanged using a EML XML-based encoding;

- the ***Task*** is the service that the Service Provider is planning to support. The task poses requirements in terms of the metadata that has to be exposed. However, this is beyond the solution scope, i.e., the solution is open with respect to metadata records that can be exchanged. Typical Service Level Agreements for this kind of scenario can be cross-digital library/repository/archive tasks including : import and export of XML metadata governed by own bibliographic DTD, search, browse, query facilities;
- the solution belongs to the **agreement-based approaches** (cf. Section 2.2.1).

Requirements

From the ***Organisational*** point of view, the *Provider* agrees to transfer the metadata records of its items in original formats. The *Consumer* agrees to acquire metadata records of the *Provider* items and to create new versions in standard formats.

From the ***Semantic*** point of view, the *Provider* and the *Consumer* should share a common understanding on the notions of digital library/repository/archive item, metadata record, metadata schema, XML and EML encoding. In particular, at machine readability level EML is expected to effectively support subsetting of collections, specifications of facets within subsets; creation of transformed versions to facilitate and enrich search; and query capabilities fine-tuned to facilitate business interoperability with primary customers. This can be achieved by complementing the EML approach with agreement-based, mediator-based or blending approaches.

From the ***Technical*** point of view, the *Provider* and the *Consumer* rely on a communication channel based on HTTP and XML.

Results

From the ***Organisational*** point of view, the EML approach ensures that the *Provider* exposes metadata records of its items and other information within a given Service Level Agreement to any client sending proper requests. From the *Consumer* perspective, the EML approach ensures that it can acquire metadata from any Provider implementing it. However, this solution subsumes a sort of Service Level Agreement, *i.e.,* a *Provider* should serve a defined set of incoming requests that comply with the EML approach.

From the ***Semantic*** point of view, the EML approach guarantees that the *Provider* and the *Consumer* share a common understanding of the model subsumed by the protocol, i.e., the notion of item, metadata record, metadata schema, EML encoding. In addition, the approach supports the common sharing of import/export, search, browse and query facilities between *Provider* and *Consumer*.

From the ***Technical*** point of view, XML is the cross-platform glue that passes information to and from the User Interface, mapping seamlessly to the databases and allowing the application to be used in many different digital library/repository/archive types. The *Provider* exposes metadata records and service related information through a defined set of HTTP requests and responses. The *Consumer* can issue defined set of HTTP requests and responses to gather the transferred metadata from any EML *Provider*.

Implementation guidelines

The XML and EML approach needs to be implemented on both *Provider* and *Consumer* side. The *Provider* has to support the requests envisaged by the approach, the *Consumer* has to issue proper requests and consume the responses. A set of implementation guidelines[134] has been produced for EML Version 5.0 (including process and data

[134] http://xml.coverpages.org/eml.html

requirements, Data dictionary, Schema descriptions and XML Schemas) and an implementation guide is available from the Council of Europe[135]. EML Version 5.0 (CS 01) is Candidate for Approval as OASIS Standard.

Assessment

EML has been originally conceived to be a trustworthy approach to e-enabled elections. This makes it an useful candidate for policy interoperability in a distributed collaborative environment, and it is already being used in such a way by some institutions. The case for using Election Markup Language (EML) has been already made by OASIS[136]. A similar rationale can be applied also for digital libraries/repositories/archives were the following benefits are identified[137]:

- Benefits of EML for Providers
    - More choice of products and suppliers;
    - Less dependency on a single supplier;
    - Avoids proprietary lock-in;
    - Stability or reduction in costs;
    - Consistency in adoption of business rules;
    - Supports scalability, transparency and interoperability;
    - Provides basis for accreditation;
- Benefits of EML for Customers
    - Supports trustworthiness of systems;
    - Supports security of transferred data;
    - Provides confidence in the Service Level Agreement implementation;
- Benefits of EML for Suppliers
    - Greater chance of doing business
    - Standardised customer requirements
    - Reduced development costs
    - Accommodates future changes more easily
    - Common core but allows local customisation / extension

3.4.2 Sample potential approach: METS and OAI-PMH

The METS - Metadata Encoding and Transmission Standard (Library of Congress) is a schema for encoding descriptive, administrative, and structural metadata regarding objects within a digital library, expressed using the XML schema. The standard is maintained in the Network Development and MARC Standards Office of the Library of Congress, and is being developed as an initiative of the Digital Library Federation. The Open Archives Initiative Protocol Metadata Harvesting (cf. Section 3.1.1.1) provides an application-independent interoperability framework for metadata sharing. METS and OAI-PMH are used for supporting machine-readable data exchange formats and harvesting of digital objects in distributed environments with multiple digital libraries/repositories/archives. Both implicitly reflect various policies, guidelines, and local specifications particularly in regards to access, metadata standardization, and rights management. Additionally, METS files can also reflect policy for assigning persistent URLs using the ARK specification.

There are two kinds of actors involved in the OAI-PMH framework: Data Providers and Service Providers. A Data Provider manages a metadata repository and uses the OAI-PMH as a means to expose metadata to harvesters. A harvester is a client application operated by a Service Provider to issue OAI-PMH requests to a repository managed by a Data Provider. In the case of an OAI-PMH data provider service, the exposed metadata records can implicitly reflect

[135] http://www.coe.int/t/dgap/democracy/activities/ggis/e-voting/evoting_documentation/Case-for-EML_en.pdf

[136] http://www.oasis-open.org/committees/download.php/26747/The%20Case%20for%20EML%20v2.pdf

[137] Adapted from OASIS Election and Voter Services TC. (2008). *The Case for using Election Markup Language (EML).* White Paper, OASIS.

local metadata specifications (such as for descriptive and rights metadata).

According to the interoperability framework (cf. Section 2):

- the ***Provider*** is the *Data Provider* while the ***Consumer*** is the *Service Provider*;
- the ***Resource*** the two entities are willing to share is any kind of *metadata record* referring to a digital library/repository/archive item and obeying to a metadata schema. The same item might be exposed through multiple metadata records of different formats. All the library/repository/archive items must be exchanged through records compliant with Dublin Core metadata schema (cf. Section 3.1.2.1);
- the ***Task*** is the service the Service Provider is planning to support. The task poses requirements in term of the metadata record that has to be exposed. However this interoperability approach *open* with respect to metadata records that can be exchanged. Typical services are cross-repository tasks including search and browse facilities;
- the solution belongs to the **agreement-based approaches**.

Requirements

From the ***Organisational*** point of view, the *Provider* agrees to expose the metadata records of its items. The *Consumer* agrees to acquire metadata records of the *Provider* items.

From the ***Semantic*** point of view, the *Provider* and the *Consumer* should share a common understanding on the notions of *repository item*, *metadata record* , *metadata schema* and METS XML encoding. In particular, the semantic of the metadata schema should be shared to reach an effective exchange of the metadata records. This can be achieved by complementing the OAI-PMH solution with other agreement-based, mediator-based or blended solutions.

From the ***Technical*** point of view, there are two modalities of incorporating OAI-PMH and METS for interoperability with other digital libraries/repositories. In the first, the OAI-PMH metadata record contains METS content. In the second, the OAI-PMH metadata record contains DC or other descriptive metadata, and the OAI-PMH About part contains related METS content. As for OAI-PMH, the Provider and the Consumer rely on a communication channel based on HTTP and XML.

Results

From the ***Organisational*** point of view, METS supports the Consumer in cross-collection searching on harvested or integrated components from various Provider sources, packaging them together into one XML document for interactions with digital libraries/repositories/archives. OAI-PMH ensures that the Provider exposes metadata records of its items and other information characterising its service (e.g., the metadata schemas supported) to any client sending proper requests. From the Consumer perspective, the OAI-PMH approach guarantees that it can acquire metadata records and other information characterising the service from any Provider implementing it. However, this solution subsumes a Service Level Agreement, *i.e.,* a *Provider* should serve a defined set of incoming requests complying with the OAI-PMH protocol.

From the ***Semantic*** point of view, the adoption of METS and OAI-PMH support the *Provider's* and the *Consumer*'s common understanding of the model subsumed by the protocol, i.e., the notion of item, metadata record and metadata schema. In addition, the approach ensures that *Provider* and *Consumer* share a common way to publish / retrieve diverse types of information (e.g., on the Provider service, metadata formats, sets, records and related identifiers offered by the *Provider*) and that every item is represented through a Metadata Record obeying to the Dublin Core schema (cf. Section 3.1.2.1) and identified via the 'oai_dc' metadata prefix. At schema level, METS supports the encoding of the metadata schema, crosswalks, application profiles and element registries.

From the ***Technical*** point of view, using METS and OAI-PMH the *Provider* exposes metadata records and service related information (e.g., the available metadata formats) through a defined set of HTTP requests and responses. The *Consumer* can issue a defined set of HTTP requests and responses to gather metadata records from any OAI-PMH *Provider*. Metadata records are exposed and gathered through their XML serialisation, compliant with a metadata schema.

Implementation guidelines

METS provides a detailed implementation registry containing descriptions of METS projects planned, in progress, and fully implemented[138]. For the OAI-PMH please refer to Section 3.1.1.1.

For what is concerned to the metadata record associated to repository items, they should either pre-exist in all the schemas that the Provider is willing to expose (one of them must be the DCMS) or be produced via mappings. Moreover, the metadata schemas might pre-exist or be defined for the scope of a specific application domain. In the second case a best practice is that of application profiles (cf. Section 3.1.3).

Assessment

The METS framework allows improving policy interoperability at two levels. At schema level, it supports the encoding of the metadata schema, crosswalks, application profiles and element registries. At repository level, it supports cross-collection searching on harvested or integrated components from various sources, packaging them together into one XML document for interactions with digital libraries/repositories/archives.

OAI-PMH has been conceived to be a lightweight solution to interoperability. Because of this, it is probably one of the most famous interoperability approaches used in the digital library/repository domain.

Using METS and OAI-PMH together facilitates policy interoperability for broad aggregation among distributed digital libraries/repositories, decreasing the costs e.g., in terms of necessary bandwidth and Quality Assurance on migrated or transformed digital content.

3.5 Quality Domain Interoperability Best practices and Solutions

Scientific works dedicated to DLs quality 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 scarcely considered by researchers.

There are specific metrics for estimating content quality, functionality quality, architecture quality, user interface quality, etc. The overall quality of a digital library – which is a 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, the content quality, and the sources functionality affect the quality of the search results.

The granularity of quality can vary a lot, from the establishment and measurement of objective dimensions to strategic guidelines covering heterogeneous aspects of the DL, at different levels. This granularity needs to be taken into account with respect to the organisational, semantic and technical interoperability a digital library or a digital library consortium wants to achieve.

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

[138] http://www.loc.gov/standards/mets/mets-registry.html

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 quality. Quality should 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.

The following paragraphs describe quality interoperability solutions and best practices taking into account respectively different research areas and outcomes from the professional community. They include:

- Data quality interoperability frameworks (cf. Section 3.5.1);
- Web quality interoperability solutions (cf. Section 3.5.2);
- Back end: Ontology-based interoperability models for Web services;
- Front end: Quality standards for Web interfaces, Validators;
- Guidelines, checklists and certificates for DLs which aim to improve standards implementation and application (e.g., OAI-PMH, Dublin Core) across DLs to support interoperability (cf. Section 3.5.3).

The selection has been conducted by the DL.org Quality Working Group.

3.5.1 Data Quality Interoperability Frameworks

In the data quality research field, specific interoperability frameworks have been built for cooperative information systems (CIS). Cooperative information systems are large scale information systems interconnecting diverse systems of autonomous organisations that share common objectives.

Supporting cooperation requires the system to be capable of reflecting both the changes that are decided for its performances (e.g., introducing new technologies) and the continuously ongoing changes of the organizational practices. The problem is how to build information systems which continue to share goals with their organizational environment, human users, and other existing systems as they all evolve. It is the continuous organizational and technological change that makes CIS's a challenge: a CIS is not simply a collection of databases, applications and interfaces, rather it is an architectural framework which maintains consistency among a variety of computer-based systems, user groups, and organizational objectives as they all evolve over time (Scannapieco M. , 2004).

In real world scenarios, usually organisations cannot trust other organisations' data due to the lack of quality certification. As an example, a set of public administrations that, in an e-Government scenario, cooperate in order to provide services to citizens and businesses will replicate many data regarding both citizens and businesses due to data errors and conflicts (Batini & Scannapieco, 2006).

3.5.1.1 DaQuinCIS Framework

The DaQuinCIS Framework, which has been produced within the DaQuinCIS Project[139], covers a large set of issues in the areas of assessment, data correction, object identification, source trustworthiness and data integration (Batini & Scannapieco, 2006) by offering a suite of data quality oriented services.

The DaQuinCIS architecture is based on **peer-to-peer services** for quality improvement and maintenance in Cooperative Information

[139] DaQuinCIS - Methodologies and Tools for Data Quality inside Cooperative Information Systems http://www.dis.uniroma1.it/dq/

Systems. In this architecture heterogeneous and geographically distributed organizations may exchange data and related quality data using a common semi-structured data model based on XML.

The DaQuinCIS architecture (Scannapieco M. , 2004) allows the diffusion of data and related quality. It exploits data replication to improve the overall quality of cooperative data.

The two main components of the architecture are:

- a model for data quality exchange, the D2Q Model (Data and Data quality Model), which is inspired by the data model underlying XML-QL[140], and includes the definitions of constructs to represent data, a common set of data quality properties, constructs to represent them and the association between data and quality data;
- a set of services that realizes data quality functions (Batini & Scannapieco, 2006).

Each organization offers services to other organizations on its own cooperative gateway and also specific services to its internal back-end systems.

Services are all identical and peer, i.e., they are instances of the same software artefacts and act both as servers and clients of the other peers depending of the specific activities to be carried out.

In order to produce data and quality data according to the D2Q model, each organization deploys on its cooperative gateway a Quality Factory service that is responsible for evaluating the quality of its own data. The Data Quality Broker poses, on behalf of a requesting user, a data request over other cooperating organizations, also specifying a set of quality requirements that the desired data have to satisfy; this is referred to as quality brokering function. The Data Quality Broker is, in essence, a peer-to peer data integration system which allows to pose quality-enhanced query over a global schema and to select data satisfying such requirements. The Quality Notification Service is a publish/subscribe engine used as a quality message bus between services and/or organizations (Milano, Scannapieco, & Catarci, 2005).

The DaQuinCIS Framework is conceived for **peer-to-peer interoperability**.

According to the interoperability framework (cf. Section 2):

- the ***Provider*** is the *Organisation XY*, and the ***Consumer*** is the O*rganisation XYZ*, both are **peers;**
- the ***Resource*** the two entities are willing to share are data and quality thresholds;
- the ***Tasks*** DaQuinCIS Framework allows are data assessment, data correction, object identification, source trustworthiness, and data integration (quality driven query processing and instance conflict resolution).

Requirements

From the ***Organisational*** point of view, distributed organisations need to agree to interoperate as **peers**. E.g. *Organisation XY* and *Organisation XYZ* agree to share data and quality thresholds in order to integrate them within a cooperative information system.

From the ***Semantic*** point of view, *Organisation XY* and *Organisation XYZ* should share a common model for data quality exchange, the *D2Q Model* in this case.

From the ***Technical*** point of view, *Organisation XY* and *Organisation XYZ* **interoperate as peers** by relying on a communication channel based on XML.

Results

From the ***Organisational*** point of view, the DaQuinCIS framework ensures that the organisations can share and exchange data and quality thresholds on those data as peers, avoiding duplication of efforts within cooperative information systems.

[140] http://www.w3.org/TR/NOTE-xml-ql/

From the ***Semantic*** point of view, organisations share a common data and data quality model, avoiding misunderstandings and ambiguities on definitions of constructs to represent data, a common set of data quality properties, constructs to represent them and the association between data and quality data.

From the ***Technical*** point of view, a communication channel based on XML allows **data sharing and integration between peers**.

Implementation guidelines

The DaQuinCIS Framework has been tested in the e-Government field. Results and details of the implementation in Italian public administration agencies have been published by the authors (Milano, Scannapieco, & Catarci, 2005). However, the implementation process is at an early stage, despite the need of data quality functionalities in distributed information systems.

Assessment

The DaQuinCIS Framework has been tested with two real data sets owned by Italian public administrations. The obtained results showed that the system is effective in improving the quality of data, with only a limited efficiency overhead (Milano, Scannapieco, & Catarci, 2005).

3.5.1.2 Fusionplex Framework

Fusionplex is a system for integrating multiple heterogeneous and autonomous information sources that uses data fusion[141] to resolve factual inconsistencies among the individual sources (Motro & Anokhin, 2006).

The main principle behind Fusionplex is that all data are not equal. The data environment is not "egalitarian", with each information source having the same qualifications. Rather, it is a diverse environment: some data are more recent, whereas other are more dated; some data come from authoritative sources, whereas other may have dubious pedigree; some data may be inexpensive to acquire, whereas other may be costlier (Motro & Anokhin, 2006). To resolve conflicts, Fusionplex looks at the qualifications of its individual information providers. Thus, it uses metadata to resolve conflicts among data. To accomplish this, the system relies on source *features*, which are metadata on the merits of each information source; for example, the recentness of the data, its accuracy, its availability, or its cost (Motro & Anokhin, 2006). In Fusionplex it is assumed that there are no modelling errors at the local sources (Batini & Scannapieco, 2006), whereas the same instance of the real world can be represented differently in the various local sources due to errors. In order to deal with such instance-level inconsistencies, Fusionplex relies on the *features* metadata described above.

Fusionplex adopts **a client-server architecture**.

According to the interoperability framework (cf. Section 2):

- the ***Provider*** is the *Fusionplex server*, which returns query results to the ***Consumer***, through each *Fusionplex client*;
- the ***Resources*** the two entities are willing to integrate are the *local sources*. Each client passes to the server the name of a database that the client wishes to query and the query itself. The server processes the query and returns its result to the client, which formats it and delivers it to its user;

[141] "Data fusion is generally defined as the use of techniques that combine data from multiple sources and gather that information in order to achieve inferences, which will be more efficient and potentially more accurate than if they were achieved by means of a single source. […] While data integration is used to describe the combining of data, data fusion is integration followed by reduction or replacement. Data integration might be viewed as set combination wherein the larger set is retained, whereas fusion is a set reduction technique with improved confidence" (Wikipedia, <http://en.wikipedia.org/wiki/Data_fusion>).

- the ***Tasks*** the *Fusionplex server* executes are several: (*i*) the query parser and translator parses the client's query, and determines the sources contributions that are relevant to the query; (*ii*) the view retriever retrieves the relevant views from the schema mapping; (*iii*) the fragment factory constructs the query fragments; (*iv*) the inconsistencies detection module assembles a polyinstance of the answer and resolves data conflicts; (*v*) the inconsistencies resolution module resolves data conflicts in each polytuple according to the appropriate resolution policies; (*vi*) the query processor processes the union of all resolved tuples and returns the query results.

Requirements

From the ***Organisational*** point of view, the *Provider* agrees to provide query processing and conflict resolution processes by guaranteeing a certain quality of service. The *Consumer* agrees to share its *local sources.*

From the ***Semantic*** point of view, the *Provider* relies on the *Consumer*'s local source *features*, which are metadata on the merits of each information source; for example, the recentness of the data, its accuracy, its availability, or its cost.

From the ***Technical*** point of view, the *Provider* is implemented in Java and contains the core functionalities described above. At start-up time, the server reads all configuration files, caches all source descriptions and creates temporary tables in a RDBMS. Then it starts listening for incoming connections from the *Consumer*, which connects to the server using simple line-based protocol.

Results

From the ***Organisational*** point of view, the overall architecture of Fusionplex and its tools provide for a flexible *integration service*. The *Consumer* wishing to integrate several information sources logs into the *Provider's server*, provides it with the appropriate definitions, and can begin using its integration services. Fusionplex also provides a client with a graphic user interface (GUI).

From the ***Semantic*** point of view, Fusionplex guarantees that the *Provider* and the *Consumer* share a common understanding of the model subsumed by the architecture, i.e., they rely on local sources contributions for replying to the query.

From the ***Technical*** point of view, each client passes to the server the name of a database that the client wishes to query and the query itself in SQL syntax. The server processes the query and returns its result to the client, which formats it and delivers it to its user.

Implementation guidelines

Fusionplex is still a prototype system.

A Consumer wishing to integrate several information sources logs into the Fusionplex server, provides it with the appropriate definitions, and can begin using its integration.

Assessment

Fusionplex has been tested in the movies information domain and details of the experiment have been provided by the framework's authors (Motro & Anokhin, 2006). Two Internet movie guides were chosen: the Internet Movie Database (http://us.imdb.com/) and the All-Movie Guide (http://allmovie.com/). The system resolved about 15 tuples per second. The authors explained that this performance may not be sufficient for commercial applications, but believe that it affirms the feasibility of their framework.

3.5.2 Web interoperability solutions

3.5.2.1 Back-end solutions: Ontologies for Web Services

Ontology is "*an explicit formal specification of how to represent the objects, concepts, and other entities that exist in some area of interest* *and the relationships that hold among them*"[142]. Several ontologies have been developed for Web services in the field of Quality of Service (QoS) and SLA (Service Level Agreement), which define acceptable levels of service to be met per factors such as availability, accessibility, integrity, performance, etc.

When describing Web services, one of the aspects that need representing is QoS, i.e., the capability of a Web service to meet an acceptable level of service as per factors such as availability and accessibility.

A Service Level Agreement is an agreement between the provider of a service and a customer that defines the set of QoS guarantees and the obligations of the parties. A service provider publishes the set of QoS capabilities that is able to offer in the service registry. A service client (i) specifies the desired QoS requirements for the service and (ii) accesses to the service registry to discover and select the service provider that best meets these requirements based on the advertised capabilities. A negotiation then starts between the client and the provider in order to obtain a SLA that satisfies both parties. During the service execution, the SLA will be the document of reference to monitor and assure that the QoS levels are guaranteed. Through a QoS ontology, the provider and the client have a shared definition of the terms and concepts used in the SLA (Green, 2006; Strasunskas & Tomassen, 2008; Mecella, Scannapieco, Virgillito, Baldoni, Catarci, & Batini, 2003; Uschold & Gruninger, 1996; Fakhfakh, Chaari, Tazi, Drira, & Jmaiel, 2008; Maximilien & Singh, 2004; Zhou & Niemela, 2006; Dobson, Lock, & Sommerville, Quality of service requirements specification using an ontology, 2005; Dobson & Sánchez-Macián, Towards Unified QoS/SLA Ontologies, 2006).

According to the interoperability framework (cf. Section 2):

- the ***Provider*** is the *SLA service provider* while the ***Consumer*** is the *SLA user* that, through an application, defines the set of Quality of Service guarantees and the obligations of the parties;
- the ***Resource*** the two entities are willing to share is the QoS value;
- the ***Task*** is the service the application is planning to support;
- the solution belongs to the **agreement-based approaches**.

Requirements

From the ***Organisational*** point of view, the Provider and Consumer need to agree to perform a matching between Consumer's QoS requirements and Provider's offers in real time.

From the ***Semantic*** point of view, the *Provider* and the *Consumer* should share common definitions of the terms and concepts used in the SLA.

From the ***Technical*** point of view, the *Provider* and the *Consumer* need to rely on a communication channel based on the specific QoS description language, which can be XML on non-XML based.

Results

From the ***Organisational*** point of view, uniformed data definition provides a better view for decision-making and match-making within interoperability scenarios.

From the ***Semantic*** point of view, the QoS ontologies guarantee that the *Provider* and the *Consumer* share common definitions of the terms and concepts used in the SLA, providing a vocabulary for QoS. This is achieved because the Provider and the Consumer agree to use a specific ontology.

From the ***Technical*** point of view, the *Provider* and the *Consumer* negotiate the QoS levels using a communication channel based on the specific QoS description language.

[142] DOI Handbook Glossary, http://www.doi.org/handbook_2000/glossary.html

Implementation Guidelines

Several ontologies for Web services have been defined. The FIPA-QoS, WS-QoS, DAML-QoS and MOQ are briefly discussed.

The ***FIPA-QoS ontology*** (Foundation for Intelligent Physical Agents, 2002) can be used by agents when communicating about the Quality of Service. The ontology provides basic vocabulary for QoS. Additionally, the FIPA-QoS ontology supports two methods to get QoS information: a single query and a subscription. For example, an agent may query current QoS values from another agent using, for example, the *FIPA-QoS interaction protocol* or the agent may subscribe to notifications when something interesting happens in the QoS using the *fipa-subscribe interaction protocol*. These notifications may be dispatched at a predefined interval or when some changes in the QoS occur. The former mechanism (periodic notification) can be used if the agent wants to be informed about the QoS values on a regular basis, for example the value of the throughput every five seconds. The latter mechanism (on occurrence notification) is useful when the agent does not care about QoS values until something relevant to its task happens. For example, an agent that is sending real-time data may need to be informed, when the throughput value drops below the given threshold.

The ***Web service QoS (WS-QoS)*** (Tian, Gramm, Ritter, & Schiller, 2004) framework has been developed to support the dynamic mapping of QoS properties concerning the network performance defined in the Web service layer onto the underlying transport technology at runtime. We call this approach cross layer communication. Further targets of the framework are

- an architecture that allows both service client and service provider to specify requests and offers with QoS properties and QoS classes;
- mechanisms that accelerate the overall Web service lookup and matching process for the client;
- tools that assist application developers with the QoS definition associated with Web services.

The WS-QoS XML schema enhances the current infrastructure of standardized service description, publication, and lookup to allow for the selection of QoS-aware Web services at run-time.

The ***DAML-QoS ontology*** (Zhou, Chia, & Lee, 2004) has been defined to complement the DAML-S ontology with a better QoS metrics model. DAML-S is ontology for describing Web Services. It aims to make Web Services computer interpretable and to enable automated Web service discovery, invocation, composition and monitoring. It defines the notions of a *Service Profile* (what the service does), a *Service Model* (how the service works) and a *Service Grounding* (how to use the service). As a DAML+OIL ontology, DAML-S retains all the benefits of Web content described in DAML+OIL. It enables the definition of a Web services vocabulary in terms of objects and the complex relationships between them, including class, subclass relations, cardinality restrictions, etc. It also includes the XML datatype information. When incorporated DAML-QoS with DAML-S, multiple service levels can be described through attaching multiple QoS profiles to one service profile. One current limitation of DAML-S' QoS model is that it does not provide a detailed set of classes and properties to represent quality of service metrics.

The framework for ***Mid-level Ontologies for Quality (MOQ)*** (Kim, Sengupta, & Evermann, 2007) represents general quality concepts that can be used for Web services, e.g., requirement, measurement, traceability and quality management system. This ontology hasn't been implemented yet, however it aims to be interoperable with existing QoS or measurement ontologies, e.g., SQuaRE (Abramowicz, Hofman, Suryn, & Zyskowski, 2008).

3.5.2.2 Front-end solutions: Web User Interfaces and Validators

The DL.org Quality Working Group is currently investigating the Web User Interfaces quality requirements[143] and the Standard Validators[144] produced by the W3C in order to select the most relevant tools that help DLs to improve the quality of their web interfaces and support interoperability.

3.5.3 Guidelines, checklists, certificates and best practices supporting DL Quality Interoperability

Within the DL field, several guidelines, checklists, and certification methodologies have been produced to solve heterogeneity issues affecting DLs interoperation at organisational, semantic and technical levels. The establishment of common rules, standards application and best practices can have very different scopes (data integration, long-term digital preservation, etc.) and focus(es) (content, functionalities, policies, etc). However, the general aim of those tools is to facilitate cooperation and development within DLs networks and infrastructures.

These are their main common features:

- They represent the result of previous negotiations among different organisation and harmonisations between existing standards, rules and best practices;
- They are conceived and promoted by the DL professional community;
- They have a practical approach;
- They are created to solve specific problems, but they can help to improve the overall quality of the DL;
- They are not “interoperability solutions” but they can help towards their implementation;
- They are intended to improve standard application;
- They broaden the adoption of good practices;
- They often allow flexibility, e.g., DLs can look at their specifications implementing some aspects only;
- They can dramatically improve organisational and/or semantic interoperability among DLs without involving the technical interoperability level. This happens because these tools often arise at a political and managerial level;
- They create a network of users (DLs).

As “quality” is an attribute which can be identified, measured and assessed in any aspect of the DL, in the following tools and descriptions of the quality interoperability aspects are highlighted. Moreover, these tools can be grouped in “document repository”-oriented tools, e.g., *DRIVER guidelines* (cf. Section 3.5.3.1) and *DINI Certification* (cf. Section 3.5.3.2), “research data repository”-oriented tools, e.g., *Data Seal of Approval* (cf. Section 3.5.3.3), and “preservation systems-oriented tools, e.g., *DRAMBORA* (cf. Section 3.5.3.4) and *TRAC* (cf. Section 3.5.3.5).

3.5.3.1 The DRIVER Guidelines 2.0

The DRIVER Guidelines (DRIVER Project, 2008) have been developed in the context of the DRIVER EU project[145]. They constitute a powerful tool to map/translate the metadata used in the repository to the Dublin Core metadata as harvested by DRIVER, but also provide orientation for managers of new repositories to define their local data

[143] In particular, here we refer to the work done by the Web Accessibility Initiative <http://www.w3.org/WAI/>.

[144] Quality Assurance Tools <http://www.w3.org/QA/Tools/>.

[145] DRIVER Digital Repository Infrastructure Vision for European Research http://www.driver-repository.eu

management policies, for managers of existing repositories to take steps towards improved services and for developers of repository platforms to add supportive functionalities in future versions.

The DRIVER Guidelines basically focus on five issues: *collections*, *metadata*, *implementation of OAI-PMH*, *best practices* and *vocabularies and semantics*. Thus the goal is **to reach interoperability on two layers, syntactical** (Use of OAI-PMH & Use of OAI_DC) **and semantic** (Use of Vocabularies).

With respect to ***collections***, the use of "sets" to define collections of open full-text is mandatory for each repository. The use of sets is optional if all resources in the repository (i) are textual, (ii) include not only metadata but also full-text and (iii) are accessible without authorization.

With respect to ***metadata,*** some mandatory and some recommended characteristics have been defined in order to rule out semantic shortcomings arising from heterogeneous interpretations of DC.

With respect to the ***OAI-PMH protocol*** (cf. Section 3.1.1.1), some mandatory and some recommended characteristics have been defined in order to rule out problems arising from the different implementations in the local repository.

The DRIVER Guidelines represent an explicit quality policy for protocol and metadata implementation.

If the mandatory characteristics of the DRIVER Guidelines are met, a repository receives the status of being a validated[146] (Horstmann, Vanderfeesten, Nicolaki, & Manola, 2008) DRIVER provider. If recommended characteristics are met, a repository receives the status of a future-proof DRIVER provider. Validated DRIVER repositories can re-use DRIVER data for the development of local services. They become part of the DRIVER network of content providers.

Requirements

From the ***Organisational*** point of view, the DRIVER Guidelines establish an explicit quality policy for protocol and metadata implementation.

From the ***Semantic*** point of view, the DRIVER Guidelines offer a common vocabulary orientating new and existing digital repositories.

From the ***Technical*** point of view, the DRIVER Guidelines require the *Consumer* is OAI-PMH and OAI_DC compliant.

Assessment

The DRIVER Guidelines 2.0 is a practical tool widely adopted across European document repositories, contributing to their general high quality standard and providing orientation for new and existing repositories.

3.5.3.2 The DINI Certificate

The DINI Certification has been developed in the context of the German Initiative for

[146] The DRIVER Validator can be considered as a practical online tool which has been built to facilitate the compliance assessment. DRIVER offers to local repositories to check the degree of conformance with the Guidelines 2.0 via the web Validator (DRIVER Validator, http://validator.driver.research-infrastructures.eu/). The DRIVER Validator Software has been developed by the National Kapodistrian University of Athens and it is designed for repository managers or 'curators' of repository networks. It runs automated tests for three aspects: (*i*) general compliance with OAI-PMH, (*ii*) compliance with DRIVER-specific recommendations for OAI-PMH implementation and (*iii*) the metadata compliance according to the DRIVER guidelines. Aspect (i) tests the validity of XML according to the OAI-PMH schema in a variety of use patterns to discover flaws in expected behaviour. Aspect (ii) tests several strategies to be within the boundaries of the DRIVER guidelines, such as deleting strategy, batch size or the expiration time of a resumption token. Aspect (iii) looks into the record and tests how the simple Dublin Core fields are used compared to the recommendations in the DRIVER guidelines.

Networked Information (DINI)[147]. DINI certification pursues three main aims:

- to provide a detailed description of the demands on a document and publication server as a service which facilitates the dissemination of scholarly contributions and which involves technology, organisation and processes;
- to pinpoint desirable ways in which this service can be further developed from a technical and organisational point of view;
- to provide users and operators with documentary evidence of a repository's compliance with standards and recommendations.

With the award of a DINI certificate it is possible to attest to the fact that repositories meet well-defined standards of quality.

In order to successfully get the DINI Certificate, the repository must adopt the Open Access repositories principles as a basis for a distributed, machine-based global network for scholarly documents and the OAI-PMH protocol.

Assessment

The DINI Certificate is conceived for the German Open Access repositories. It has been observed that DINI certified repositories have a robust OAI-PMH protocol implementation, homogeneous DC metadata, a low percentage of XML errors, and a high percentage of full-texts. By 2010 DINI Certificate and DRIVER Guidelines v2.0 will be fully compatible.

3.5.3.3 Data Seal of Approval Guidelines

The Data Seal of Approval[148] (DSA) was established by a number of institutions committed to durability in the archiving of research data. By assigning the seal, they not only wish to guarantee the durability of the data concerned, but also to promote the goal of durable archiving in general.

The sixteen DSA quality guidelines are intended to ensure that in the future research data can still be processed in a high-quality and reliable manner, without this entailing new thresholds, regulations or high costs.

Achieving the DSA means that the data concerned have been subjected to the sixteen guidelines of which the assessment procedure consists. The repository will be permitted to display the DSA logo on its homepage and in other locations relevant to its communication in the realm of scientific and scholarly research.

Although the sixteen guidelines regard three stakeholders – the data producer (three guidelines), the data consumer (three guidelines) and the data archive (ten guidelines) – the data archive is seen as the main organisation responsible for the repository. The data archive as an organization that should take care of the overall implementation of the DSA in its own specific field.

In order to acquire the Data Seal of Approval, a Trusted Digital Repository (TDR) is obliged to keep a file directory on the web that is accessible through the homepage of the repository. This so-called assessment directory contains:

- An up-to-date version of the Data Seal of Approval handbook;
- The information leaflet about the Data Seal of Approval Assessment;
- The Data Seal of Approval Assessment.

The DSA Assessment completed by the repository is the starting point for the review procedure, carried out by the DSA Board in order to decide whether or not an organisation is granted the Data Seal of Approval. There is no audit, no certification: just a review on the basis of trust.

[147] German Initiative for Networked Information http://www.dini.de/

[148] The Data Seal of Approval Organisation http://www.datasealofapproval.org/

3.5.3.4 DRAMBORA (Digital Repository Audit Method Based on Risk Assessment)

DRAMBORA (Digital Repository Audit Method Based on Risk Assessment) (Innocenti, Ross, Maceciuvite, Wilson, Ludwig, & Pempe, 2009) is a digital repository audit methodology for self-assessment, encouraging organisations to establish a comprehensive self-awareness of their objectives, activities and assets before identifying, assessing and managing the risks implicit within their organisation.

Within DRAMBORA, preservation systems maintenance is characterised as a risk-management activity. Six stages are implicit within the process. Initial stages require auditors to develop an organisational profile, describing and documenting the repository's mandate, objectives, activities and assets. Latterly, risks are derived from each of these, and assessed in terms of their likelihood and potential impact. Finally, auditors are encouraged to conceive appropriate risk management responses to the identified risk. The process enables effective resource allocation, enabling repository administrators to identify and categorise the areas where shortcomings are most evident or have the greatest potential for disruption[149].

Following the successful completion of the self-audit exercise, organisations can expect to have:

- Established a comprehensive and documented self-awareness of their mission, aims and objectives, and of intrinsic activities and assets;
- Constructed a detailed catalogue of pertinent risks, categorised according to type and inter-risk relationships, and fully described in terms of ownership, probability and potential impact of each risk;
- Created an internal understanding of the successes and shortcomings of the organisation, enabling it to effectively allocate or redirect resources to meet the most pressing issues of concern;
- Prepared the organisation for subsequent external audit whether that audit will be based upon the Trustworthy Repositories Audit & Certification (TRAC) (cf. Section 3.5.3.5), nestor Catalogue of Criteria for Trusted Repositories, or forthcoming Consultative Committee for Space Data Systems (CCSDS) digital repository audit assessment criteria[150].

The DRAMBORA methodology provides auditors with an organisational risk register detailing each risk faced and its status.

3.5.3.5 Trustworthy Repositories Audit & Certification (TRAC)

TRAC (Trustworthy Repositories Audit & Certification: Criteria and Checklist) (Ambacher, et al., 2007) is sponsored by RLG and the US National Archives and Records Administration (NARA), laying the groundwork for international collaboration on digital repository audit and certification between the DCC, RLG (now OCLC-RLG Programs), NARA, NESTOR, and the US Center for Research Libraries.

TRAC offers a set of criteria applicable to a range of digital repositories and archives, from academic institutional preservation repositories to large data archives and from national libraries to third-party digital archiving services, providing tools for the audit, assessment, and potential certification of digital repositories, establishes the documentation requirements required for audit, delineates a process for certification, and establishes appropriate methodologies for determining the soundness and sustainability of digital repositories.

[149] http://www.repositoryaudit.eu/about/

[150] http://www.repositoryaudit.eu/benefits/

3.6 Architecture Domain Interoperability Best practices and Solutions

Architecture Domain interoperability is a multifaceted yet very concrete issue arising whenever two entities, actually two software systems, playing the role of Provider and Consumer are willing to share *Architectural Components* initially owned by the Provider only. Although this problem seems to be well known, in practice this is not the case. It is often mixed and confused with the all the others Interoperability problems discussed in this document.

Moreover, the problem is well beyond the digital library domain. Approaches based on programming practices and methodologies include aspect-oriented programming (Kiczales, et al., 1997), subject-oriented programming (Harrison & Ossher, 1993), multidimensional separation of concerns (Tarr, Ossher, Harrison, & Sutton, 1999) to cite a few.

However, in the digital library domain, interoperability approaches in this domain fall in the following categories: Architectural Component Profile (cf. Section 3.6.1); Standard-based Exploitation of third party Architectural Component (cf. Section 3.6.2); Mediator Services (cf. Section 3.6.3).

3.6.1 Architectural Component Profile

In order to make it possible for a Consumer to exploit an Architectural Component of a third party Provider, the consumer should be provided with a characterisation of such a resource. In a scenario where two systems are willing to share an architectural component, the component profile is a means to reach a common understanding of the component features and gives the provider the characterisation needed to exploit it. This description may assume diverse forms ranging from human-oriented description, e.g., a textual description in natural language, to a machine-understandable one, e.g., WSDL, as in the case of Web Services.

To some extent, the Architectural Component profile is a kind of metadata attached to the Architectural Component. The Consumer system might rely on the explicit knowledge of some of the features characterizing the Architectural Component Profile in order to properly use it.

Two different approaches to Architectural Components Profile Modeling can be distinguished. The first one relates to the definition of a profile that is specific (proprietary) to the system which defines it (a sort of proprietary solution). The second one is aimed to model, by defining a general profile structure, possible interoperability scenarios related to service oriented architectures. This latter approach can be exemplified by the profile defined in the context of the eFramework initiative (cf. Section 3.3.1.3), that seeks to promote the use of the service-oriented approach in the analysis and design of software for use within education and research.

Concrete exemplars of this kind of interoperability solution are: the WS-I Basic Profile (cf. Section 3.6.1.1).

3.6.1.1 WS-I Basic Profile

WS-I Basic Profile (Chumbley, Durand, Pilz, & Rutt, 2010) is a set of non-proprietary Web services specifications, along with clarifications, refinements, interpretations and amplifications of those specifications which promote interoperability among Web services. The WS-I Basic Profile Working Group has currently released Basic Profile 1.2 and Basic Profile 2.0. As far as Service Description is concerned, it relies on WSDL1.1 (Web Services Description Language) whilst for message formatting on SOAP (v1.1 & v1.2 in Basic Profile v1.2 and v2.0 respectively).

3.6.2 Standard-based Exploitation of third party Architectural Component

Every Architectural Component implements one or more functions. If a Consumer entity is able to interact with the Architectural Component hosted by the Provider, it will benefit from such a component by exploiting its functionality. Because of this there are a lot of commonalities with the approaches and solutions discussed in the Functionality Domain Interoperability section (cf. Section 3.3).

Concrete exemplars of this kind of interoperability solution are: SRU (cf. Section 3.6.2.1), i.e., a mechanism for exploiting the search capabilities of another system; OpenSearch (cf. Section 3.6.2.2), i.e., a mechanism for exploiting search facilities of a third party system; SWORD (cf. Section 3.6.2.3), i.e., a mechanism for depositing in third party repositories. Other protocols have been described in previous sections, e.g., OAI-PMH (cf. Section 3.1.1.1).

3.6.2.1 Search/Retrieval via URL (SRU)

SRU (Morgan, 2004; The Library of Congress, 2010) is a standard ANSI-NISO XML-focused search protocol for Internet search queries. The current version is 1.2. It utilizes CQL (Contextual Query Language – previously called Common Query language), a standard syntax for representing queries. The Search/Retrieve operation is the main operation in SRU. It allows the client to submit a Search/Retrieve request for matching records from the server. CQL is a formal language for representing queries to information retrieval systems such as web indexes, bibliographic catalogues and museum collection information. The protocol allows queries and results to be transmitted in different ways, namely, via HTTP Get, via HTTP Post, or via HTTP SOAP (HTTP SOAP is the former SRW).

According to the interoperability framework (cf. Section 2):

- a ***Provider*** is a retrieval system such as a web index, a catalog, etc.; a ***Consumer*** is any client searching for information;
- the ***Resource*** is a third party component realising a search facility;
- the ***Task*** is the functionality known as “search and retrieve”;
- the solution belongs to the agreement-based approaches (cf. Section 2.2.1).

Requirements

From the ***Organisational*** point of view, the Provider exposes its search capabilities in accordance with the SRU requirements. The Consumer agrees to search/retrieve Information Objects in accordance with SRU specifications.

From the ***Semantic*** point of view, Provider and Consumer should have a common knowledge of the semantic associated to the Contextual Query Language.

From the ***Technical*** point of view, the Provider and the Consumer must communicate via HTTP Get, via HTTP Post, or via HTTP SOAP. The incremental benefits of SRU via SOAP are the ease of structured extensions, web service facilities such as proxying and request routing, and the potential for better authentication systems. In the HTTP SOAP – that tries to adhere to the Web Services Interoperability recommendations (cf. Section 3.6.1.1) – clients and servers must support SOAP version 1.1, and may support version 1.2 or higher. This requirement allows as much flexibility in implementation as possible.

Results

From the ***Organisational*** point of view, SRU allows any Provider to expose its search capabilities to any Consumer willing to exploit them.

From the ***Semantic*** point of view, CQL tries to combine simplicity and intuitiveness of expression for simple, every day queries, with the richness of more expressive languages to accommodate complex concepts when necessary.

From the ***Technical*** point of view, SRU also allows simple implementations of both a client and a server. The client is merely an HTML form that generates and submits an SRU-compliant URL to a nominated server.

Implementation guidelines

The standard SRU protocol has to be implemented on both *Provider* and *Consumer* side. The *Provider* has to support the requests envisaged by the protocol, the *Consumer* has to issue proper requests and consume the responses.

A set of implementation guidelines have been produced including guidelines for minimal implementations. Implementation guidelines and a lot of implemented tools are available at the SRU web site.

Assessment

SRU, maintained by the US Library of Congress, is among the most diffused standard XML-focused search protocol for Internet search queries. It is being widely used by a large number of information providers.

In (McCallum, 2006), a comparative study on search protocols and query languages including SRU and Open Search is reported.

In (Denenberg, 2009), Denenberg gave an update on the work ongoing in the OASIS Search Web Services Technical Committee. This work aims to standardize SRU and to reconcile it with the differently originated OpenSearch (cf. 3.6.2.2) as well as with other records-oriented search methodologies. In (Hammond, 2010), Hammond reported on the integration between SRU and OpenSearch.

3.6.2.2 OpenSearch

OpenSearch (opensearch.org, 2010) is a simple means to interface to a search service by declaring a URL template and returning a common syndicated format. Open Search is useful for providing a very low threshold search protocol that primarily supports keyword searching. It accepts the fact that different databases may have differently defined searches and simply uses a keyword search that will be treated as the target sees fit. There is a presumption that the end user may not need to know how the term is being treated by the target.

According to the interoperability framework (cf. Section 2):

- a ***Provider*** is any component realising search facilities; a ***Consumer*** is any entity willing to exploit third party search facilities;
- the ***Resource*** is the search facility realised by a third party component;
- the ***Task*** is the functionality known as “search and retrieve”;
- the solution belongs to the agreement-based approaches (cf. Section 2.2.1).

Requirements

From the ***Organisational*** point of view, the Provider exposes its search capabilities through the OpenSearch protocol. The Consumer searches/retrieves records through a OpenSearch protocol.

From the ***Semantic*** point of view, no requirement exists regarding the query language, as OpenSearch primarily supports keyword searching.

From the ***Technical*** point of view, the Consumer sends a request to the Provider for information about the Provider search capabilities. The Provider sends back the parameter names used locally for search activities. The Consumer then sends a query to the Provider using the “language” and receives retrieved records from the Provider in RSS format.

Results

From the ***Organisational*** point of view, OpenSearch allows any Provider to expose Information Objects to searching, and any Consumer to easily search/retrieve them.

From the ***Semantic*** point of view, Open Search primarily supports keyword based searching while not constraining the query.

From the ***Technical*** point of view, the OpenSearch approach is especially valuable for searching across the many Information Object providers independently from their structure. There is support for search operation control parameters (pagination, encoding, etc.), but no constraints are placed on the query string which is regarded as opaque.

Implementation guidelines

Implementation guidelines and tools are available at the OpenSearch website.

Assessment

OpenSearch is used by most search engines.

3.6.2.3 SWORD

SWORD (Simple Web service Offering Repository Deposit)[151] is a standard mechanism for depositing into repositories and similar systems proposed by the homonymous project under the JISC Repositories and Preservation Programme. Its aim is to address the need for a common Deposit standard with a lightweight solution for populating repositories. SWORD has been conceived as an "application profile"[152] (with extensions) of the Atom Publishing Protocol (APP)[153], which is an application-level protocol for publishing and editing Web resources. In particular, SWORD focuses on two key aspects of the ATOM protocol – the deposit of files, rather than Atom documents, and the extension mechanism for specifying additional deposit parameters. The main result is a profile of the Atom Publishing Protocol which can be used by implementers to create SWORD-compliant deposit clients or SWORD interfaces into repositories, where the client will perform the deposit and the interface will accept it.

According to the interoperability framework (cf. Section 2):

- a ***Provider*** is any entity operating a Repository service while a ***Consumer*** is any entity willing to exploit a third party repository for storing Information Objects;
- the ***Resource*** is the architectural component realising the Repository service;
- the ***Task*** the Consumer is willing to realise is the storage of an Information Object in a third party repository;
- the solution belongs to the **agreement-based approaches** (cf. Section 2.2.1).

Requirements

From the ***Organisational*** point of view, the *Provider* provides its repository with a SWORD-interface and the *Consumer* agrees to deposit metadata and content through this interface.

From the ***Semantic*** point of view, *Provider* and *Consumer* should have a common knowledge of the semantics associated to the SWORD protocol.

From the ***Technical*** point of view, the *Provider* and the *Consumer* communicate according to the SWORD protocol through a communication channel based on HTTP and XML.

Results

From the ***Organisational*** point of view, the SWORD approach guarantees that a *Consumer* can easily deposit in different repositories, while a *Provider* can accept deposit requests from multiple consumers. From the ***Semantic*** point of view, the SWORD approach guarantees that the Consumer and Provider share a common model on the deposit interface.

From the ***Technical*** point of view, depositing results from a two-stage process within APP and SWORD. First, a request from an authenticated user is sent to the implementation for what APP calls the 'service document', this returns details of the collections that user is allowed to deposit to within the repository. At this point, the user may deposit their file into the chosen collection. Various things may prevent success, for example lack of authentication credentials, unacceptable file format or a corrupt MD5 checksum. The repository will send a respond indicating whether or not the deposit is successful.

[151] http://www.swordapp.org/

[152] This notion should not be confused with the notion of Application Profile described in Section 3.1.3.

[153] http://www.ietf.org/rfc/rfc5023.txt

Implementation guidelines

Some guidelines, developed in the context of the PEER project, are reported in (Bijsterbosch, Brétel, Bulatovic, Peters, Vanderfeesten, & Wallace, 2009). PEER is collaboration between publishers, repositories and the research community, which aims to investigate the effects of the large-scale deposit (so called Green Open Access) on user access, author visibility, journal viability and the broader European research environment.

A course on SWORD[154] covering the how it can be used, how it works, and how to get started using it is available at the SWORD initiative web site.

Assessment

Platforms such as DSpace, Eprints, Fedora, IntraLibrary, and Zentity have been provided with SWORD repository implementation allowing clients to deposit publications, data sets, theses, digitized materials, etc.

SWORD has extensively been used in the EU funded project PEER[155], a project started in 2008 with the goal to monitor the effects of large-scale, systematic depositing of authors' final peer-reviewed manuscripts (so called Green Open Access or stage-two research output) on reader access, author visibility, and journal viability, as well as on the broader ecology of European research.

3.6.3 Mediator Services

Standards represent ideally the best interoperability solution when addressing implementation, but they are not always possible or desirable. This because they do not support existing systems that were not built to the standards, and because they may preclude some optimized implementations. Besides that, the standardization process itself, with input from multiple factions with many and varied requirements and intended uses, often takes more time than even the implementation of the standards.

Mediators are components specifically conceived to host the interoperability machinery. These components realise patterns of various genre. Here follows an overview of a selection of such components together with their current contextualization.

Blackboard-based Approaches are based on components that allow asynchronous communication between components in a system. A component willing to interact with another component can write data in the blackboard, these data can be read by the recipient accessing the blackboard. The blackboard is often used to implement a pattern aimed to solve non deterministic problems but due to its nature it can be also used to provide or enhance interoperability. In this sense interaction can consist of an exchange of data between two peers or, in a client/server like model, of an asynchronous invocation of functions of a server by one or more clients.

Connector / Adaptor-based Approaches are based on components that translate one interface for a component into a compatible interface. It is often used to implement Mediators and Brokers components.

Proxy-based Approaches are similar to the *Connector / Adaptor.* The Proxy is a component that provides an interface to another component. The difference is in the fact that it usually represents another "instance" of the target component, it exposes the same interface but allows additional operation over received calls. For example, it can be used when you want to lazy-instantiate an object, or hide the fact that you're calling a remote service, or control access to the object. In this latter case the Proxy is used to limit access to a component, limiting interoperability as well, but usually with the aim of enhancing security.

Mediator-based Approaches are based on components that provide a unified interface to

[154] http://swordapp.org/the-sword-course/

[155] http://www.peerproject.eu

a set of other component interfaces and encapsulate how this set of objects interact. Mediator promotes loose coupling by keeping objects from referring to each other explicitly and enables varying their interaction independently. The implementation of a mediator is based on the use of as many adaptors as the number of components that needs to interact. A mediator can also be used as a "one client to n-servers" interface: for example, a client that wants to execute the same query on both a relational and xml database can ask the mediator, which takes care of all the necessary work.

Broker-based Approaches are based on components that are responsible for coordinating communication, such as forwarding requests, transmitting results and exceptions. It is responsible for coordinating communication in distributed software systems with decoupled components that interact by message exchange. Introducing a broker component allows achieving better decoupling of message producers and message consumers (these roles should be preferred to client/server roles that are rather confusing in a broker-based scenario (Alonso, Casati, Kuno, & Machiraju, 2010)). Usually in a system with several servers, each server registers itself to the broker. Then when the broker receives requests from clients it forwards the request to the correct server and then the answer back to the client. The broker is similar to the mediator in the sense that it represents a specialized version of it.

Registry-based Approaches are based on components used to grant access to other components in a system. Each component in a system registers itself (its interface or other information that allow identification) in the registry. The registry can then be accessed as an interface to the system. It doesn't act like a mediator, taking care of communication between components. Instead it just tells the component asking for information, how to contact the target component.

3.7 Cross-domain Interoperability Best practices and Solutions

3.7.1 Provenance

The notion of *Provenance* is gaining a lot of attention in many domains. It is crucial to making determinations about whether information is trusted, how to integrate diverse information sources, and how to give credit to originators when reusing information. However, there is yet a lack of consensus on what provenance is probably because it is concerned with a very broad range of sources and uses. It has many meanings in different contexts. A working definition of provenance on the Web has been developed by the W3C Provenance Incubator Group[156]: "*Provenance of a resource is a record that describes entities and processes involved in producing and delivering or otherwise influencing that resource. Provenance provides a critical foundation for assessing authenticity, enabling trust, and allowing reproducibility. Provenance assertions are a form of contextual metadata and can themselves become important records with their own provenance.*" In its essence, provenance refers to the sources of information, such as entities and processes, involved in producing or delivering an artefact.

Several initiatives have been promoted to enhance the state of the art in Provenance. Among them, it is worth to cite the above mentioned W3C Provenance Incubator Group[157] and the DCMI Metadata Provenance Task Group[158]. The DCMI Metadata Provenance Group aims to define an application profile (cf.

[156] http://www.w3.org/2005/Incubator/prov/wiki/W3C_Provenance_Incubator_Group_Wiki

[157] http://www.w3.org/2005/Incubator/prov/wiki/W3C_Provenance_Incubator_Group_Wiki

[158] http://dublincore.org/groups/provenance/

Section 3.1.3) that allows for making assertions about description statements or description sets, creating a shared model of the data elements required to describe an aggregation of metadata statements in order to collectively import, access, use and publish facts about the quality, rights, timeliness, data source type, trust situation, etc. of the described statements.

Because of the above fuzziness on the term 'Provenance', it might be very challenging to discuss about provenance interoperability. However, some solutions exist.

3.7.1.1 Open Provenance Model

The *Open Provenance Model (OPM)* (Moreau, et al., 2010) is a model for provenance representation, which is designed to allow provenance information about arbitrary resources to be represented in a technology-independent manner, and to be exchanged between systems by means of a compatibility layer based on a shared provenance model.

The OPM supports a digital representation of provenance for any kind of resources, whether physical objects, or digital data produced by a computer system, or abstract entities or concepts.

According to the interoperability framework:

- a ***Provider*** exposes provenance information about arbitrary resources in conformity with the OPM; such provenance information can be exploited by any ***Consumer*** that is able to comply with the OPM;
- the ***Resource*** the two entities are willing to share is any provenance information about an arbitrary resource according to the OPM;
- the ***Task*** is the functionality that any *Consumer* is planning to support; such a functionality requires the availability of information describing the provenance of a resource;
- the solution belongs to the **agreement-based approaches** (cf. Section 2.2.1).

The Open Provenance Model was originally crafted and released as a result of a workshop (Salt Lake City, August 2007) following the First Provenance Challenge[159], aiming to understand the capabilities of available provenance-related systems and the expressiveness of their provenance representations, and the Second Provenance Challenge[160], aiming to establish interoperability of systems, by exchanging provenance information. A Third Provenance Challenge[161] followed in order to test the original model and define a reviewed version of the model.

Requirements

From the ***Organisational*** point of view, the *Provider* agrees to expose provenance information in conformity with the Open Provenance Model; the *Consumer* agrees to retrieve provenance information in compliance with such provenance model.

From the ***Semantic*** point of view, *Provider* and *Consumer* should have a common knowledge of the semantics associated to the entities involved in the model, i.e., *artifact*s, *processes* and *agents*, and the dependency relationships between them. Moreover, they should agree on the meanings of the additional annotations and properties they would use to enrich the shared model.

From the ***Technical*** point of view, *Provider* and *Consumer* should agree on the representation format to be used to implement the OPM, as well as and the communication protocol. An agreement should be found also on the controlled vocabularies and syntax schemes for additional annotations and properties to be used in the model.

[159] http://twiki.ipaw.info/bin/view/Challenge/FirstProvenanceChallenge

[160] http://twiki.ipaw.info/bin/view/Challenge/SecondProvenanceChallenge

[161] http://twiki.ipaw.info/bin/view/Challenge/ThirdProvenanceChallenge

Results

From the ***Organisational*** point of view, the Open Provenance Model guarantees that *Provider* and *Consumer* represent and exchange provenance information of arbitrary resources in compliance with a shared provenance model.

From the ***Semantic*** point of view, the OPM is an abstract model which defines a provenance directed graph, aiming at expressing how arbitrary objects depended on others and resulted in specific states.

The nodes in the provenance graph are *artifacts*, *processes* and *agents*. An artifact is defined as an immutable piece of state, and may represent the state related to a physical object or a digital representation. Processes represent application activities, that is actions or series of actions performed on artifacts and leading to the production of new artifacts. Agents represent contextual entities controlling processes.

A direct edge in the graph represents a causal dependency between its source (effect) and its destination (cause), so that a path in the provenance graph expresses the causality path that led an object to a given state. Five basic causal relationships are defined: a process used an artifact, an artifact was generated by a process, a process was triggered by a process, an artifact was derived from an artifact, and a process was controlled by an agent.

Each entity in the provenance graph can be annotated with a set of property-value pairs to allow meaningful exchange of provenance information; exemplar annotations may include sub-typing of edges, description of processes, or reference to values of artifacts. Edges can be annotated with time information, as well. Apart from using the predefined annotations, *Provider* and *Consumer* are allowed to define and develop their context-specific properties and values.

From the ***Technical*** point of view, the OPM has a well-defined set of syntactic rules underpinning the generation of valid graphs, as well as a set of inference rules allowing to automatically derive an OPM valid graph from another valid OPM graph, and so to derive indirect provenance information from direct causality relations expressed by edges in a provenance graph. A set of common controlled vocabularies for additional annotations, properties and values is provided, as well. In addition, *Provider* and *Consumer* are allowed to define and implement their own context-specific rules through the definition of an OPM profile, which is a specialization of OPM consisting of customized controlled vocabularies for annotations and properties, profile expansion rules and serialization specific syntax.

The choice of the communication protocol is left to the *Provider* and *Consumer* specific implementation, as well as the model representation format. The OPM currently supports implementations in RDF and XML, but other customized ones may be defined.

Implementation guidelines

An XML schema, namely OPMX[162], and an OWL ontology, namely OPMO[163], are defined in order to implement the OPM specification in XML and RDF. The OPMO is the full ontology extending the OPMV[164], a lightweight vocabulary allowing to express the core concepts of OPM. The OWL ontology and the XML schema were co-evolved, in order for the XML serialization to be convertible into the RDF representation, and viceversa.

A set of Java specifications is available as well, in order to implement the OPM specification in Java code: OPM4J[165] is a Java Library for creating in memory Java representations of OPM graphs and XML serializations. It is part of

[162] http://openprovenance/model/opmx

[163] http://openprovenance/model/opmo

[164] http://purl.org/net/opmv/ns

[165] http://openprovenance.org/java/site/1_1_8/apidocs/org/openprovenance/model/package-summary.html

the OPM Toolbox, a set of command line utilities and Java classes aiming to create Java representations of OPM graphs and to serialize Java representations into XML and RDF, as well as to parse XML and RDF representations and to convert XML representations into RDF and vice-versa.

Assessment

Many systems, such as Taverna, VisTrails and Swift, have been provided with a provenance export capability to OPM. Many other projects are making use of OPM to capture and export provenance information: ProvenanceJS is a JavaScript library that allows for the retrieval and visualization of the provenance information within a Web page and its embedded content offers a provenance export capability to OPM; eBioFlow uses OPM to capture provenance of workflow execution and includes a visualization based on OPM; the Tupelo Semantic Content Repository provides an implementation of OPM.

References to the above mentioned systems and to many other projects implementing OPM can be found on the OPM website[166].

[166] http://openprovenance.org/

4 Interoperability Scenarios

This section is dedicated to discuss common yet complex interoperability scenarios, i.e., scenarios faced while developing large scale Digital Libraries built by interacting with existing systems. These scenarios combine in a coherent way the approaches and solutions discussed in the previous section.

4.1 Digital Library Systems Federation

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 *consumers*, offer functionality over an aggregation of information objects obtained by manipulating objects collected from a set of DLSs, which act here as *providers*. Such providers expose *content resources* the service provider is willing to consume to accomplish its *tasks*.[167] In order to interoperate, the two interlocutors have to face the following challenges: (*i*) "how to exchange the resource", i.e., identifying common data-exchange practices, and (*ii*) "how to harmonize objects", i.e., resolve data impedance mismatch problems arising from differences in data models subsumed by the involved actors. While data exchange across different platforms is typically overcome by adopting XML as lingua-franca and standard data-exchange protocols (such as OAI-PMH, OAI-ORE, ODBC, etc.), the harmonization interoperability challenge has mainly to do with *impedance mismatch issues*, which can be classified as:

- *Data model impedance mismatch*: it is the mismatch between the service provider's data model (i.e., the XML schema capturing structure and semantics of the information objects to be generated) and the data providers' data models (i.e., the XML schema capturing structure and semantics of the information objects to be collected and elaborated).
- *Granularity impedance mismatch*: it is the mismatch between XML encodings of information objects at the service provider's and data providers' sites, which may consider different levels of granularity. For example one DIDLXML file may represent (i.e., package) a set of 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. Figure 5 shows the basic architecture of a data provider federation. With respect to the interoperability framework:

- the ***provider*** is the data provider while the ***consumer*** is the service provider;
- the ***resource*** to be exchanged by the two is an *information object* matching a given *format*, i.e., data model. Examples of resources may be collections of publications, audio and video material,

[167] 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.

compound objects (i.e., sets of interlinked information objects), or "surrogates" of all these, namely metadata descriptions of information objects;

- the ***Task*** is the functionality the service provider is willing to offer to its consumers, be them users or applications.

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 this graph, 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.[168]

As highlighted in Figure 5, in the Digital Library world, the basic interoperability issues occurring between a service provider and a data provider in the need of exchanging information objects are typically overcome by adopting XML as lingua-franca in combination with standard data-exchange protocols – e.g., OAI-PMH (cf. Section 3.1.1.1). XML files have a labelled tree structure and can 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. More specifically, on the data provider side, the 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 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 in order to accomplish a given Task.[169]

The interoperability problem occurs when the exporting and importing component of data providers and service provider do not share the same data model, i.e., speak the same XML schema language. In this case, data and service providers cannot interoperate due to *data model* or *granularity* impedance mismatches and a number of interoperability solutions can be devised. Such solutions imply the construction of special software components on

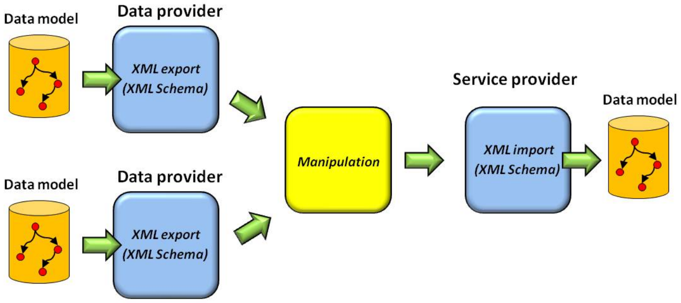

Figure 5. DL Federation: Basic Architecture

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

[169] 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.

the data provider and/or on the service provider side, depending on the specific requirements of the scenario.

In the following we first identify the functional architecture of such solutions, independently of their actual implementation, and then present three specific realizations, relative to common federation scenarios where data providers export metadata records according to the OAI-PMH protocol.

4.1.1 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 solution consists of software components capable of overcoming such differences by applying appropriate XML file transformations (see Figure 6). Implicitly, such transformations convert information objects from one data model onto another.

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 that of the 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 ones).
- *Semantic heterogeneity* occurs at the level of leaves, under two circumstances:
  - ○ 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;
  - ○ the service provider has leaves in the XML schema that do not find a direct correspondent in the data provider XML schema, besides 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

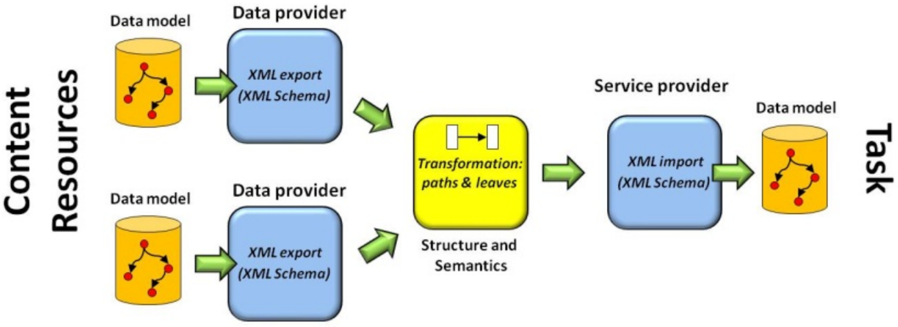

Figure 6. Interoperability issue: Data Model Mismatch

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, from the element names to 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*'s 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 through a given *F* an output leaf.

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 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 with 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" (Repox,[170] D-NET[171]). These consists in named lists of pairs (*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*'s to match unexpected requirements in the future.

4.1.2 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

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

[171] D-NET Software Toolkit, http://www.d-net.research-infrastructures.eu

model mismatch, a granularity impedance mismatch may arise.

The following scenarios typically occur (see Figure 7):

- **(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 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 combination may vary across different application domains, but is often based on shared identifiers and/or external references to those.
- **(1×M: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.

4.1.3 OAI-PMH Repository Federations: common scenarios and solutions

In this section we focus on federations whose data providers are OAI-PMH compatible, i.e., each exporting component of a data provider implements the OAI-PMH Interface, and the service provider, i.e., the consumer, is an application accessing the providers according to the protocol verbs, i.e., its importing component implements an OAI-PMH harvester. In such settings, the resource to be exchanged is a collection of metadata records as conceived by the OAI-PMH protocol data model, while the task is the construction of a uniform "information space" of metadata records

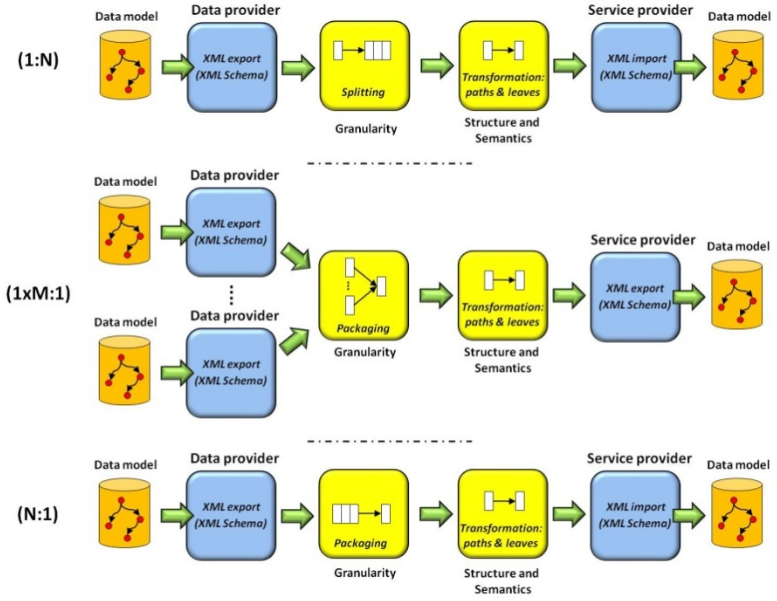

Figure 7. Interoperability issue: Granularity Mismatch

matching the consumer data model. In this sense, we shall assume that such data model embodies the features necessary to implement service provider functionalities.

It is important to note that the term metadata record is often misleading in this context. In fact, OAI-PMH was originally devised to export the XML representation of metadata records, namely structured descriptions of physical or digital objects (many believe that an XML file is by definition metadata, while this is not necessarily the case). However, due to its simplicity, the protocol is increasingly being used as a means to export XML representations of any form of content, not only metadata records. A typical example is that of compound objects, which are portions of a graph of objects, whose XML representation (e.g., XMLDIDL) are sometime exported through OAI-PMH channels (e.g., XMLDIDL representations of compound object at Los Alamos National Library[172]).

In particular, the following section presents three typical scenarios, common in the literature, and for those points at existing solutions. We shall see how, given a DLS federation interoperability problem, the manipulation 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.

4.1.3.1 “Bottom-up” 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)[173].

A special case of bottom-up federations is that 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 OAI-PMH harvester and aggregator dictating the terms of data provider’s inclusion. Although quite rare in practice, due to the difficulties of autonomous and independent organizations to respect external guidelines, this is the case for example for DAREnet-NARCIS[174], 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).

Requirements

From the ***Organisational*** point of view, the consumer has to publish “guidelines for data providers”. Such guidelines, which may have the form of a document or web site instructions, define a SLA that data providers have to respect to be part of the federation. Typically:

- *OAI-PMH compliance* to the verb `getRecords` is mandatory, while the degree of compliancy to other verbs and functionality may vary: incremental harvesting, OAI-set support, etc.
- *Quality-of-service issues*, such as 24/7 availability are another constraint that may vary.
- *Data model compliancy*, that is the metadata format and OAI-Sets to be

[172] LANL, http://library.lanl.gov/

[173] The DRIVER Infrastructure: http://search.driver.research-infrastructures.eu

[174] *DAREnet: Digital Academic Repositories*, http://www.narcis.nl

supported by the data provider: name of the format, list of sets with specification, XML paths and leaf domains of the format.

Data providers willing to be part of the federation must adjust their OAI-PMH publisher services so as to respect the guidelines. In a second stage, once the consumer has “validated” their compliance to the guidelines, data providers are included in the federation and harvested into the aggregation. Validation is a process that may occur at the organizational level, by humans, or at the technical level, through proper tools.

From the ***Semantic*** point of view, the provider and the consumer share a common understanding of the target data model of the consumer, hence on the relative XML schema elements and leaf domains.

From the ***Technical*** point of view, the consumer collects metadata records from the providers relying on the OAI-PMH protocol standard. The consumer is also responsible for providing the tools for data provider administration (registration to participate to the federation and validation). The providers have to realize the transformation, splitting and packaging components required to publish metadata record according to the consumer guidelines.

Results

The required aggregation is achieved and the interoperability issue is solved.

Implementation guidelines

As mentioned above the implementation requires the development of repository administration, harvesting, transformation, splitting, packaging or validation components, depending on the specific scenario at hand. A number of existing tools can be used or be developed from scratch:

- Federation administration: D-NET, Repox
- Harvesting: D-NET, Repox
- Validation: D-NET
- Transformation: D-NET, Repox
- Splitting: D-NET

At the Open Archive initiative web site[175] there is a number of tools dealing with publishing and harvesting using the OAI-PMH protocol. On the consumer side, some of these tools also integrate indexing, searching and UI components.

The D-NET Software Toolkit is a general-purpose tool, which can be customized to manage a set of OAI-PMH data providers, harvest, validate, transform, package, split XML records of arbitrary format.[176] It also offers general purpose components for the construction of service provider functionalities, such as UIs and components dedicated to export the aggregated data through OAI-PMH, ODBC and SRW interfaces.

Repox is a flexible harvester and aggregation software, capable of managing a federation of OAI-PMH repositories to harvest their records, which can be of arbitrary formats, and transform them into records of a given target data model.[177]

Assessment

The solution requires the realization of tools for the construction of a federation of arbitrary numbers of data providers and manipulation components whose complexity depends on the degree of impedance mismatch present between the providers' original data models and the one specified by the consumer guidelines. The evaluation of existing software may ease the implementation process and reduce the realization costs, although this is not always the case: ad-hoc software to be modified, software not documented and maintained, etc.

[175] Open Archives, http://www.openarchives.org/pmh/tools/tools.php

[176] D-NET Software Toolkit, http://www.d-net.research-infrastructures.eu

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

4.1.3.2 "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[178]). 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 its intervention to the relative implementation.

Requirements

From the ***Organisational*** point of view, the providers conform to the OAI-PMH protocol (cf. Section 3.1.1.1) and expose their metadata records, conforming to some local data model, through an HTTP address. In some federations such data model is shared across the providers, in others each data provider preserves its own export metadata format. The consumer is willing to acquire such metadata records from the providers by interacting with the relative OAI-PMH publisher services, hosted at a given base URLs. The providers and the consumer agree on the quality of the data to be transmitted and can interact to explore common solutions to interoperability.

From the ***Semantic*** point of view, the provider and the consumer share a common understanding of the target data model of the consumer, hence on the relative XML schema. The main issue is how to deal with the granularity, semantic and structural impedance issues that may arise when mapping data provider data models onto the consumer data model. For the solution to be implemented, both consumer and providers must elaborate the correct and full mappings between native data models and aggregation data model.

From the ***Technical*** point of view, the consumer collects metadata records from the providers relying on the OAI-PMH protocol standard and has to deal with the impedance issues above to achieve a uniform aggregation. Depending on the specific scenario, the consumer or the providers have to realize the transformation, splitting and packaging components that realize the data model mappings developed together. In "data provider-oriented" solutions, data providers must realize the proper manipulation components in order to export metadata records matching the aggregation data model through OAI-PMH. In "consumer-oriented" solutions, the consumers harvest metadata records conforming to the original data models and then realize the manipulation components required to transform them into records matching the aggregation data model.

Results

The required aggregation is achieved and the interoperability issue is solved.

Implementation guidelines

As mentioned above the implementation requires the development of harvesting, transformation, splitting or packaging components, depending on the specific scenario at hand. A number of existing tools can be used or be developed from scratch:

- Federation administration: D-NET, Repox
- Harvesting: D-NET, Repox
- Transformation: D-NET, Repox
- Splitting: D-NET

At the Open Archive initiative web site[179] there is a number of tools dealing with publishing and harvesting using the OAI-PMH protocol. On the consumer side, some of these tools also

[178] *The European Film Gateway* project: http://www.eureopanfilmgateway.eu

[179] Open Archives, http://www.openarchives.org/pmh/tools/tools.php

integrate indexing, searching and UI components.

The D-NET Software Toolkit is a general-purpose tool, which can be customized to manage a set of OAI-PMH data providers, harvest, validate, transform, package, split XML records of arbitrary format.[180] It also offers general purpose components for the construction of service provider functionalities, such as UIs and components dedicated to export the aggregated data through OAI-PMH, ODBC and SRW interfaces.

Repox is a flexible harvester and aggregation software, capable of managing a federation of OAI-PMH repositories to harvest their records, which can be of arbitrary formats, and transform them into records of a given target data model.[181]

In the case of consumer-oriented solutions, data providers (which often are involved in the definition of the aggregation data model of the consumer) are requested to identify and specify the structural and semantics mappings and the granularity issues to be taken into account on the consumer side to fulfil the harvesting and the aggregation. Often such specifications have the form of documents. In some simpler scenarios, where "flat" metadata format are adopted and no complex semantics transformations are required, data providers may be offered tools through which structural mappings from local data model to aggregation data model (XML schema onto XML schema) can be easily defined by an end user through a graphical interface (Repox).

Assessment

The solution requires the realization of tools for the construction of a federation of arbitrary numbers of data providers and manipulation components whose complexity depends on the degree of impedance mismatch present between the providers and the consumer. The evaluation of existing software may ease the implementation process and reduce the realization costs, although this is not always the case: ad-hoc software to be modified, software not documented and maintained, etc.

Note that data provider-oriented solutions imply high participation cost for the data providers, which have to realize complex data manipulation components in order to participate to the federation. In this sense, consumer-oriented solutions prove to be better scaling for including arbitrary numbers of data providers, which are only required to logistically help the consumer developers at adapting the manipulation components to enable their harvesting and inclusion into the federation. It is not to be underestimated the cost of realizing, in cooperation between data providers, a common schema for the aggregation of the federation; the process of ruling out structural and semantic differences while not loosing information that might be relevant to the aggregation itself is not trivial and often leads to vision and interpretation conflicts.

4.1.3.3 "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,[182] BASE search engine).[183] In such cases, it is the service providers that has to deal with interoperability issues.

[180] D-NET Software Toolkit, http://www.d-net.research-infrastructures.eu

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

[182] *OAIster-OCLC project*, http://www.oclc.org/oaister/

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

Requirements

From the ***Organisational*** point of view, the providers conform to the OAI-PMH protocol (cf. Section 3.1.1.1) and expose their metadata records in Dublin Core format through an HTTP address. The consumer is willing to acquire such metadata records from the providers by interacting with the relative OAI-PMH publisher services, hosted at a given base URLs. The providers are passively accessed and their data harvested and, as such, are not responsible of the quality of the data as required by the consumer.

From the ***Semantic*** point of view, the provider(s) and the consumer(s) only share a common understanding of the notions of Dublin Core data model, hence on the XML schema, and of OAI-PMH sets. However, two main issues must be typically faced by the consumer:

- semantic impedance: the Dublin Core semantics (i.e., domains of the DC fields) adopted across the individual providers typically differ from each other;
- structural impedance: the consumer's data model structure differs from Dublin Core.

For the solution to be implemented, the consumer must therefore know the semantics adopted for Dublin Core at the data providers and, in the case of structural impedance, how the Dublin Core XML schema should map onto the XML schema relative to the aggregation data model.

From the ***Technical*** point of view, the consumer collects metadata records from the providers relying on the OAI-PMH protocol standard and has to deal with the impedance issues above to achieve a uniform aggregation. To this aim, the consumer has to realize the transformation, splitting and packaging components possibly required to construct a uniform aggregation from the Dublin Core records harvested from the providers.

Results

The required aggregation is achieved and the interoperability issue is solved.

Implementation guidelines

As mentioned above the implementation requires the development of harvesting, transformation, splitting or packaging components, depending on the specific scenario at hand. A number of existing tools can be used or be developed from scratch:

- Federation administration: D-NET, Repox;
- Harvesting: D-NET, Repox;
- Transformation: D-NET, Repox;
- Splitting: D-NET.

At the Open Archive initiative web site[184] there is a number of tools dealing with publishing and harvesting using the OAI-PMH protocol. On the consumer side, some of these tools also integrate indexing, searching and UI components.

The D-NET Software Toolkit is a general-purpose tool, which can be customized to manage a set of OAI-PMH data providers, harvest, validate, transform, package, split XML records of arbitrary format.[185] It also offers general purpose components for the construction of service provider functionalities, such as UIs and components dedicated to expose the aggregated data through OAI-PMH, ODBC and SRW interfaces.

Repox is a flexible harvester and aggregation software, capable of managing a federation of OAI-PMH repositories to harvest their records, which can be of arbitrary formats, and transform them into records of a given target data model.[186]

Assessment

The solution requires the realization of tools for the construction of a federation of arbitrary

[184] Open Archives, http://www.openarchives.org/pmh/tools/tools.php

[185] D-NET Software Toolkit, http://www.d-net.research-infrastructures.eu

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

numbers of data providers and manipulation components whose complexity depends on the degree of impedance mismatch present between the providers and the consumer. The evaluation of existing software may ease the implementation process and reduce the realization costs, although this is not always the case: ad-hoc software to be modified, software not documented and maintained, etc.

5 Conclusions

Interoperability issues are among the most challenging problems to be faced when building systems as “collections” of independently developed constituents (systems on their own) that should co-operate with and rely on each other to accomplish larger tasks. Digital Libraries fall in this category as they have to face with interoperability issues very often. This document collects and documents a portfolio of best practices and pattern solutions to common issues faced when developing large-scale interoperable Digital Library systems.

Bibliography

Abramowicz, W., Hofman, R., Suryn, W., & Zyskowski, D. (2008). SQuaRE based Web Services Quality Model. *Proceedings of The International MultiConference of Engineers and Computer Scientists 2008* (pp. 827-835). Newswood Limited.

Alonso, G., Casati, F., Kuno, H., & Machiraju, V. (2010). *Web Services: Concepts, Architectures and Applications.* Springer Berlin Heidelberg.

Alves, A., Arkin, A., Askary, S., Barreto, C., Bloch, B., Curbera, F., et al. (2007). *Web Services Business Process Execution Language Version 2.0*. (OASIS) From http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.html

Ambacher, B., Ashley, K., Berry, J., Brooks, C., Dale, R. L., Flecker, D., et al. (2007). *Trustworthy Repositories Audit & Certification: Criteria and Checklist.* CRL, The Center for Research Libraries.

Appleton, O., Jones, B., Kranzlmüller, D., & Laure, E. (2008). The EGEE-II Project: Evolution Towards a permanent European Grid Initiative. *In L. Grandinetti, editor, High Performance Computing and Grids in Action* , 424-435.

Assante, M., Candela, L., Castelli, D., Frosini, L., Lelii, L., Manghi, P., et al. (2008). An Extensible Virtual Digital Libraries Generator. *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* , 122-134.

Assche, F. (2006). *An Interoperability Framework*. From Learning Interoperability Framework for Europe: http://www.intermedia.uio.no/display/life/An+Interoperability+Framework

Athanasopoulos, G., Candela, L., Castelli, D., Innocenti, P., Ioannidis, Y., Katifori, A., et al. (2010). *The Digital Library Reference Model.* DL.org.

Baker, T., & Keiser, J. (2010). Linked Data for Fighting Global Hunger: Experiences in setting standards for Agricultural Information Management. In D. Wood, *Linking Enterprise Data.*

Banerji, A., Bartolini, C., Beringer, D., Chopella, V., Govindarajan, K., Karp, A., et al. (2002). *Web Services Conversation Language (WSCL) 1.0*. (W3C) From http://www.w3.org/TR/2002/NOTE-wscl10-20020314/

Barwise, J., & Seligman, J. (1997). *Information Flow – The Logic of Distributed Systems.* Cambridge University Press.

Batini, C., & Scannapieco, M. (2006). *Data Quality: Concepts, Methodologies and Techniques.* Springer Berlin Heidelberg.

Benatallah, B., Casati, F., Grigori, D., Nezhad, H. R., & Toumani, F. (2005). Developing Adapters for Web Services Integration. *CAiSE 2005, The 17th Conference on Advanced Information Systems Engineering.* Porto, Portugal.

Berners-Lee, T., Fielding, R., & Masinter, L. *Uniform Resource Identifier (URI): Generic Syntax.* Request for Comments 3986.

Bertino, E., Casarosa, V., Crane, G., Croft, B., Del Bimbo, A., Fellner, D., et al. (2001). *Digital Libraries: Future Directions for a European Research Programme.* San Cassiano: DELOS.

Bijsterbosch, M., Brétel, F., Bulatovic, N., Peters, D., Vanderfeesten, M., & Wallace, J. (2009). *Guidelines for publishers and repository managers on deposit, assisted deposit and self-archiving.* PEER Project.

Bishr, Y. (1998). Overcoming the semantic and other barriers to GIS interoperability. *International Journal for Geographical Information Science , 12* (4), 299–314.

Bizer, C., Cyganiak, R., & Heath, T. (2008). *How to Publish Linked Data on the Web.* From http://sites.wiwiss.fu-berlin.de/suhl/bizer/pub/LinkedDataTutorial/

Bizer, C., Heath, T., & Berners-Lee, T. (2009). Linked Data – The Story So Far. *International Journal on Semantic Web and Information Systems – Special Issue on Linked Data , 5* (3), 1-22.

Booth, D., & Liu, K. C. (2007). *Web Services Description Language (WSDL) Version 2.0 Part 0: Primer*. (W. Recommendation, Producer) From http://www.w3.org/TR/wsdl20-primer

Bordeaux, L., Salaün, G., Berardi, D., & Mecella, M. (2004). When are two Web Services Compatible? *VLDB TES'04.* Toronto, Canada.

Borgman, C. L. (2000). *From Gutenberg to the global information infrastructure: access to information in the networked world.* Cambridge, MA: MIT Press.

Boutsis, S., Candela, L., Di Marcello, P., Fabriani, P., Frosini, L., Kakaletris, G., et al. (2009). *D4Science System High-level Design.* D4Science.

Bruce, T., & Hillmann, D. (2004). The continuum of metadata quality: defining, expressing, exploiting. In D. Hillmann, & E. Westbrooks, *Metadata in Practice* (pp. 238-256).

Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., & Stal, M. (1996). *Pattern-Oriented Software Architecture: A System of Patterns.* John Wiley & Sons .

Bygstad, B., Ghinea, G., & Klaebo, G.-T. (2008). Organizational challenges of the semantic web in digital libraries. *IEEE Innovations in Information Technology IIT 2008*, (pp. 190-194). Al-Ain.

Candela, L., Akal, F., Avancini, H., Castelli, D., Fusco, L., Guidetti, V., et al. (2007). DILIGENT: integrating Digital Library and Grid Technologies for a new Earth Observation Research Infrastructure. *International Journal on Digital Libraries , 7* (1-2), 59-80.

Candela, L., Athanasopoulos, G., Castelli, D., El Raheb, K., Innocenti, P., Ioannidis, Y., et al. (2011). *The Digital Library Reference Model.* D3.2b DL.org Project Deliverable.

Candela, L., Castelli, D., & Pagano, P. (2009). D4Science: an e-Infrastructure for Supporting Virtual Research Environments. *In M. Agosti, F. Esposito, and C. Thanos, editors, Post-proceedings of the 5th Italian Research Conference on Digital Libraries - IRCDL 2009* , 166-169.

Candela, L., Castelli, D., & Pagano, P. (2008). gCube: A Service-Oriented Application Framework on the Grid. *ERCIM News* (72), 48-49.

Candela, L., Castelli, D., & Pagano, P. (2009). On-demand Virtual Research Environments and the Changing Roles of Librarians. *Library Hi Tech , 27* (2), 239-251.

Candela, L., Castelli, D., Ferro, N., Ioannidis, Y., Koutrika, G., Meghini, C., et al. (2008). *The DELOS Digital Library Reference Model - Foundations for Digital Libraries.* DELOS.

Candela, L., Castelli, D., Ferro, N., Ioannidis, Y., Koutrika, G., Meghini, C., et al. (2008). *The DELOS Digital Library Reference Model.* DELOS.

Candela, L., Castelli, D., Ioannidis, Y., Koutrika, G., Pagano, P., Ross, S., et al. (2006). *The Digital Library Manifesto.* DELOS.

Christensen, E., Curbera, F., Meredith, G., & Weerawarana, S. (2001). *Web Services Description Language (WSDL) 1.1*. (W. Note, Producer) From http://www.w3.org/TR/2001/NOTE-wsdl-20010315

Chumbley, R., Durand, J., Pilz, G., & Rutt, T. (2010). *WS-I Basic Profile Version 2.0*. From http://www.ws-i.org/Profiles/BasicProfile-2.0.html

Cimpian, E., & Mocan, A. (2005). WSMX Process Mediation Based on Choreographies. *1st International Workshop on Web Service Choreography and Orchestration for Business Process Management (BPM 2005).* Nancy, France.

Clark, T., & Jones, R. (1999). Organizational Interoperability Maturity Model for C2. *Proceedings of the 1999 Command and Control Research and Technology Symposium.* United States Naval War College, Newport,RI.

Dekkers, M. (2007). *Metadata and modelling for Interoperability.* DCMI.

Denenberg, R. (2009). Search Web Services - The OASIS SWS Technical Committee Work. *D-Lib Magazine , 15* (1/2).

Deng, S., Wu, Z., Zhou, M., Li, Y., & Wu, J. (2006). Modeling Service Compatibility with Pi-calculus for Choreography. *Proceedings of 25th International Conference on Conceptual Modeling.* Tucson, AZ, USA.

Devlin, K. (1991). *Logic and Information.* Cambridge University Press.

DL.org Team. (2010). *DL.org: Digital Library Interoperability, Best Practices and Modelling Foundations*. From www.dlorg.eu

Dobson, G., & Sánchez-Macián, A. (2006). Towards Unified QoS/SLA Ontologies. *IEEE Services Computing Workshops (SCW'06)* (pp. 169-174). IEEE.

Dobson, G., Lock, R., & Sommerville, I. (2005). Quality of service requirements specification using an ontology. *Proceeding of Workshop on Service-Oriented Computing Requirements (SOCCER)* (pp. 1-10). Paris: IEEE.

DRIVER Project. (2008). *DRIVER Guidelines 2.0.* DRIVER Project.

Dumas, M., Benatallah, B., Hamid, R., & Nezhad, M. (2008). Web Service Protocols: Compatibility and Adaptation. *IEEE Data Eng. Bull. , 33* (3), 40-44.

Duval, E., Hodgins, W., Sutton, S., & Weibel, S. L. (2002). Metadata Principles and Practices. *D-Lib Magazine , 8* (4).

European Commission. (2010). *A Digital Agenda for Europe - Communication from the Commission to the European Parliament, the Council, the European Economic and Social Committee and the Committee of the Regions.* Brussels: European Comission.

Fagin, R., Halpern, J. Y., Moses, Y., & Vardi, M. Y. (1995). *Reasoning About Knowledge.* The MIT Press.

Fakhfakh, K., Chaari, T., Tazi, S., Drira, K., & Jmaiel, M. (2008). A comprehensive ontology-based approach for SLA obligations monitoring. *2nd International IEEE Conference on Advanced Engineering Computing and Applications in Sciences (ADVCOMP 2008)*, (pp. 217-222).

Farrell, J., & Lausen, H. (2007). *Semantic Annotations for WSDL and XML Schema*. (W3C Recommendation) From http://www.w3.org/TR/sawsdl/

Fensel, D., & Bussler, C. (2002). The Web Service Modeling Framework WSMF. *Electronic Commerce Research and Applications , 1* (2).

Floros, E., Kakaletris, G., Polydoras, P., & Ioannidis, Y. (2008). Query Processing Over The Grid: the Role Of Workflow Management. In *Grid Computing: Achievements and Prospects* (pp. 1-12). Springer.

Ford, T., Colomb, J., Grahamr, S., & Jacques, D. (2007). A Survey on Interoperability Measurement. *Proceedings of 12th International Command and Control Research and Technology Symposium.* Newport, RI.

Foster, I., Kesselman, C., & Tuecke, S. (2001). The Anatomy of the Grid: Enabling Scalable Virtual Organization. *The International Journal of High Performance Computing Applications , 15* (3), 200-222.

Foster, I., Kesselman, C., Nick, J., & Tuecke, S. (2002). *The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration.*

Foster, I., Zhao, Y., Raicu, I., & Lu, S. (2008). Cloud Computing and Grid Computing 360-Degree Compared. *In Grid Computing Environments Workshop, 2008. GCE ’08* .

Foundation for Intelligent Physical Agents. (2002). *FIPA Quality of Service Ontology Specification.* www.fipa.org.

Fox, E., & Marchionini, G. (1998). Toward a Worldwide Digital Library. *Communications of the ACM , 41* (4), 29-32.

Fox, E., Akscyn, R., Furuta, R., & Leggett, J. (1995). Digital Libraries. *Communications of the ACM , 38* (4), 23-28.

Fraser, M. (2005). Virtual Research Environments: Overview and Activity. *Ariadne , 44*.

Gamma, E., Helm, R., Johnson, R. J., & Vlissides, J. M. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software.* Addison-Wesley Professional.

Geraci, A. (1991). *IEEE Standard Computer Dictionary: Compilation of IEEE Standard Computer Glossaries.* IEEE Press.

Gill, T., & Miller, P. (2002). Re-inventing the Wheel? Standards, Interoperability and Digital Cultural Content. *D-Lib Magazine , 8* (1).

Gill, T., Gilliland, A. J., Whalen, M., & Woodley, M. S. (2008). Glossary. In T. Gill, A. J. Gilliland, M. Whalen, & M. S. Woodley, *Introduction to Metadata.* Getty Publications.

Gonçalves, M., Fox, E., Watson, L., & Kipp, N. (2004). Streams, Structures, Spaces, Scenarios, Societies (5S): A Formal Model for Digital Libraries. *ACM Transactions on Information Systems (TOIS) , 22* (2), 270 –312.

Green, L. (2006). Service level agreements: an ontological approach. *Proceedings of the 8th international conference on Electronic commerce: The new e-commerce: innovations for conquering current barriers, obstacles and limitations to conducting successful business on the internet* (pp. 185-194). ACM.

Gridwise Architecture Council. (2005). *GridWise™ Architecture Council Interoperability Path Forward.* Whitepaper.

Hammond, T. (2010). nature.com OpenSearch: A Case Study in OpenSearch and SRU Integration. *D-Lib Magazine , 16* (7/8).

Harrison, W., & Ossher, H. (1993). Subject-oriented programming: a critique of pure objects. *ACM SIGPLAN Notices , 28* (10).

Heckmann, D. (2005). Distributed user modeling for situated interaction. *35. GI Jahrestagung, Informatik 2005 - Workshop Situierung, Individualisierung und Personalisierung* (pp. 266-270). Conn, Germany: Lecture Notes in Informatics (LNI).

Heckmann, D., & Kruger, A. (2003). A User Modeling Markup Language (UserML) for Ubiquitous Computing. *Proceedings of User Modeling 9th International Conference, UM 2003.* Johnstown, PA, USA.

Heckmann, D., Brandherm, B., Schmitz, M., Schwartz, T., & M., V. W.-M. (2005). GUMO - the general user model ontology. *Proceedings of the 10th International Conference on User Modeling.* Edinburgh, Scotland.

Heckmann, D., Schwarts, T., Brandherm, B., & Kroner, A. (2005). Decentralized user modeling with UserML and GUMO. *Proceedings of 10th International Conference on User Modelling.* Edinburg, Scotland.

Heckmann, D., Schwartz, T., Brandherm, B., Schmitz, M., & Wilamowitz-Moellendorff, M. (2005). GUMO - the general user model ontology. *Proceedings of the 10th International Conference on User Modeling.* Edinburgh, Scotland.

Heery, R., & Patel, M. (2000). Application profiles: mixing and matching metadata schemas. *Ariadne* (25).

Heiler, S. (1995). ACM Computing Survey. *Semantic interoperability , 27*, 271-273.

Hillmann, D. I., & Phipps, J. (2007). Application profiles: exposing and enforcing metadata quality. *Proceedings of the 2007 international conference on Dublin Core and Metadata Applications: application profiles: theory and practice*, (pp. 53-62).

Horstmann, W., Vanderfeesten, M., Nicolaki, E., & Manola, N. (2008). A deep validation process for open document repositories. *ELPUB2008. Open Scholarship: Authority, Community, and Sustainability in the Age of Web 2.0 - Proceedings of the 12th International Conference on Electronic Publishing* (pp. 429-431). ELPUB.

Huynh, T., Jutla, C., Lowry, A., Strom, R., & Yellin, D. (1994). *The global desktop: A graphical composition environment for local and distributed applications.* New York: IBM.

IDABC. (2004). *European Interoperability Framework for pan-European eGovernment Services.* Luxembourg.

Innocenti, P., Ross, S., Maceciuvite, E., Wilson, T., Ludwig, J., & Pempe, W. (2009). Assessing Digital Preservation Frameworks: the approach of the SHAMAN project. *Proceedings of the International Conference on Management of Emergent Digital EcoSystems* (pp. 412-416). ACM.

Innocenti, P., Vullo, G., & Ross, S. (2010). Towards a Digital Library Policy and Quality Interoperability Framework: the DL.org Project. *New Review of Information Networking , 15*, 1-25.

Ioannidis, Y. (2005). Digital libraries at a crossroads. *International Journal of Digital Libraries , 5* (4), 255 – 265.

Ioannidis, Y., Maier, D., Abiteboul, S., Buneman, P., Davidson, S., Fox, E., et al. (2005). Digital library information-technology infrastructures. *International Journal of Digital Libraries , 5* (4), 266-274.

Jinghai, R., & Xiaomeng, S. (2004). A Survey of Automated Web Service Composition Methods. *Cardoso, J., Sheth, P. eds. SWSWPC 2004*, (pp. 43-54).

Kaschner, K., Ready, J. S., Agbayani, E., Rius, J., Kesner-Reyes, K., Eastwood, P. D., et al. (2008). From AquaMaps: Predicted range maps for aquatic species: http://www.aquamaps.org/

Kaschner, K., Watson, R., Trites, A., & Pauly, D. (2006). Mapping world-wide distributions of marine mammal species using a relative environmental suitability (RES) model. *Marine Ecology Progress Series* (316), 285–310.

Kavantzas, N., Burdett, D., Ritzinger, G., Fletcher, T., & Lafon, Y. (2004). *Web Services Choreography Description Language Version 1.0*. (W3C) From http://www.w3.org/TR/ws-cdl-10

Kiczales, G., Lamping, J., Mendhekar, A., Maeda, C., Lopes, C., Loingtier, J.-M., et al. (1997). Aspect-oriented programming . *Proceedings ECOOP'97 — Object-Oriented Programming. 1241*, pp. 220-242. Springer.

Kim, H. M., Sengupta, A., & Evermann, J. (2007). MOQ: Web services ontologies for QoS and general quality evaluations. *International Journal of Metadata, Semantics and Ontologies , 2* (3), 196-200.

Klyne, G., & Carroll, J. (n.d.). *Resource Description Framework (RDF): Concepts and Abstract Syntax.* From http://www.w3.org/TR/rdf-concepts/

Lagoze, C. (2010). *Lost Identity: The Assimilation of Digital Libraries into the Web.* Cornell University.

Lagoze, C., & Van de Sompel, H. (2008). *Open Archives Initiative Object Reuse and Exchange User Guide - Primer*. From http://www.openarchives.org/ore/1.0/primer

Lagoze, C., & Van de Sompel, H. (2001). The open archives initiative: building a low-barrier interoperability framework. *Proceedings of the 1st ACM/IEEE-CS Joint Conference on Digital Libraries, Roanoke, Virginia, United States, JCDL '01* (pp. 54–62). New York, NY: ACM.

Library of Congress. (n.d.). *Metadata Encoding and Transmission Standard (METS) Official Web Site*. From http://www.loc.gov/standards/mets/

Martin, D., Burstein, M., Hobbs, J., Lassila, O., McDermott, D., McIlraith, S., et al. (2004). *OWL-S: Semantic Markup for Web Services*. (W3C) From http://www.w3.org/Submission/OWL-S/

Maximilien, E. M., & Singh, M. P. (2004). Toward autonomic web services trust and selection. *Proceedings of the 2nd international conference on Service oriented computing* (pp. 212 - 221). ACM.

McCallum, S. H. (2006). A Look at New Information Retrieval Protocols: SRU, OpenSearch/A9, CQL, and XQuery. *World Library and Information Congress: 72nd IFLA General Conference and Council.* IFLA.

Mecella, M., Scannapieco, M., Virgillito, A., Baldoni, R., Catarci, T., & Batini, C. (2003). The DaQuinCIS Broker: Querying Data and Their Quality in Cooperative Information Systems. *Journal on Data Semantics , 1*, 208-232.

Milano, D., Scannapieco, M., & Catarci, T. (2005). P2P Data Quality Improvement in the DaQuinCIS System. *Journal of Digital Information Management , 3* (3).

Miller, P. (2000). Interoperability: What it is and why should I want it. *Ariadne , 24*.

Moreau, L., Clifford, B., Freire, J., Futrelle, J., Gil, Y., Groth, P., et al. (2010). The Open Provenance Model core specification (v1.1). *Future Generation Computer Systems* .

Morgan, E. L. (2004). An Introduction to the Search/Retrieve URL Service (SRU). *Ariadne* (40).

Motro, A., & Anokhin, P. (2006). Fusionplex: resolution of data inconsistencies in the integration of heterogeneous information sources. *Information Fusion , 7* (2), 176-196.

Nilsson, M. (2008). *The Singapore Framework for Dublin Core Application Profiles*. From http://dublincore.org/documents/singapore-framework/

NISO. (2004). *Understanding Metadata.* NISO (National Information Standards Organization). NISO.

OASIS Election and Voter Services TC. (2008). *The Case for using Election Markup Language (EML).* White Paper, OASIS.

OASIS. (2007). *OASIS Election Markup Language (EML). Version 5.0 Process and Data Requirements*. (OASIS) From http://docs.oasis-open.org/election/eml/v5.0/os/EML-Process-Data-Requirements-v5.0.htm

OASIS Security Assertion Markup Language. (n.d.). From http://docs.oasis-open.org/security/saml/v2.0/

Open Archives Initiative. (2002). *Open Archives Initiative - Protocol for Metadata Harvesting*. From http://www.openarchives.org/pmh/

Open Archives Initiative. (2008). *Open Archives Initiative Object Reuse and Exchange*. From http://www.openarchives.org/ore/

opensearch.org. (2010). Retrieved 2010 from OpenSearch: http://www.opensearch.org/Home

Paepcke, A., Chang, C. K., Winograd, T., & García-Molina, H. (1998). Interoperability for Digital Libraries Worldwide. *Communications of the ACM , 41*, 33-42.

Pagano, P., Simeoni, F., Simi, M., & Candela, L. (2009). 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.

Park, J., & Ram, S. (2004). Information Systems Interoperability: What Lies Beneath? *ACM Transactions on Information Systems , 22*, 595-632.

Peng, Y., Zheng, Z., Xiang, J., Gao, J., Ai, J., Lu, Z., et al. (2009). A Model of Service Behavior Based on Petri Nets with Weights. *World Congress on Software Engineering*, (pp. 3-6).

Ponnekanti, S. R., & Fox, A. (2004). Interoperability among Independently Evolving Web Services. . *Middleware'04.* Toronto, Canada.

Prud'hommeaux, E., & Seaborne, A. (n.d.). *SPARQL Query Language for RDF.* From http://www.w3.org/TR/rdf-sparql-query/

QUALIPSO project. (2008). Organisational Interoperability. Quality Platforms for Open Source Software.

Ram, S., Park, J., & Lee, D. (1999). Digital Libraries for the Next Millennium: Challenges and Research Directions. *Information Systems Frontiers , 1*, 75-94.

RosettaNet Community. (2010). From RosettaNet Web site: http://www.rosettanet.org

Ross, S. (2003). *Digital library development review.* National Library of New Zealand.

Ross, S. (2007). Digital Preservation, Archival Science and Methodological Foundations for Digital Libraries, . *Keynote Speech at the European Conference on Research and Advanced Technology for Digital Libraries (ECDL) 2007* . Budapest, Hungary.

Ross, S. (2008). Preservation of interoperability and interoperability of preservation. *Third Workshop on Foundations of Digital Libraries* . Aarhus.

Scannapieco, M. (2004). *DaQuinCIS: Exchanging and Improving Data Quality in Cooperative Information Systems.* PhD Thesis in Computer Engineering, Università di Roma "La Sapienza", Dipartimento di Informatica e Sistemistica.

Scannapieco, M., Virgillito, A., Marchetti, M., Mecella, M., & Baldoni, R. (2004). The DaQuinCIS Architecture: a Platform for Exchanging and Improving Data Quality in Cooperative Information Systems. *Information Systems , 29* (7).

Simeoni, F., Candela, L., Kakaletris, G., Sibeko, M., Pagano, P., Papanikos, G., et al. (2007). A Grid-Based Infrastructure for Distributed Retrieval. *In L. Kovacs, 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* , 161-173.

Sirin, E., Parsia, B., Wu, D., Hendler, J., & Nau, D. (2004). HTN planning for web service composition using SHOP2. *Journal of Web Semantics , 1* (4), 377-396.

Stollberg, M., Cimpian, E., Mocan, A., & Fensel, D. (2006). A Semantic Web Mediation Architecture. *Proceedings of 1st Canadian Semantic Web Working Symposium (CSWWS 2006).*

Strasunskas, D., & Tomassen, S. L. (2008). On Significance of Ontology Quality in Ontology-Driven Web Search. *Emerging Technologies and Information Systems for the Knowledge Society* (pp. 469-478). Springer Berlin / Heidelberg.

Tarr, P., Ossher, H., Harrison, W., & Sutton, S. M. (1999). N degrees of separation: multi-dimensional separation of concerns. *Proceedings of the 21st international conference on Software engineering (ICSE '99)* (pp. 107-119). ACM.

The International DOI Foundation. (2005). *The DOI Handbook. Edition 4.2.0, doi:10.1000/186.*

The Library of Congress. (2010). Retrieved 2010 from SRU: Search/Retrieval via URL: http://www.loc.gov/standards/sru/index.html

The Standards Committee of the National Defense Industry Association (NDIA) Robotics Division. (2007). Interoperability Standards Analysis (ISA).

Thompson, H., Beech, D., Maloney, M., & Mendelsohn, N. (2004). *XML Schema Part 1: Structures*. (W. W. Consortium, Producer) From http://www.w3.org/TR/xmlschema-1

Tian, M., Gramm, A., Ritter, H., & Schiller, J. (2004). Efficient Selection and Monitoring of QoS-Aware Web Services with the WS-QoS Framework. *Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence* (pp. 152-158). IEEE Computer Society.

Tolk, A. (2003). Beyond Technical Interoperability – Introducing a Reference Model for Measures of Merit for Coalition Interoperability. *8th International Command and Control Research and Technology Symposium.* CCRP Press.

Tolk, A., & Muguira, J. (2003). The Levels of Conceptual Interoperability Model (LCIM). *IEEE Fall Simulation Interoperability Workshop.* IEEE CS Press.

Tolk, A., Diallo, S., & Turnitsa, C. (2007). Applying the levels of Conceptual Interoperability Model in Support of Integratability, Interoperability, and Composability for System-of-Systems Engineering. *International Journal Systemics, Cybernetics and Informatics , 5* (5).

Uschold, M., & Gruninger, M. (1996). Ontologies: principles, methods and applications. *The Knowledge Engineering Review , 11*, 93-136.

Van der Sluijs, K., & Houben, G. (2006). 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.

Van der Sluijs, K., & Houben, G. (2005). Towards a generic user model component. *Proceedings of the Workshop on Decentralized, Agent Based and Special Approaches to User Modelling, in International Conference on User Modelling*, (pp. 43-52). Edinburgh, Scotland.

Vullo, G., Innocenti, P., & Ross, S. (2010). Towards Policy and Quality Interoperability: Challenges and Approaches for Digital Libraries. *IS&T Archiving 2010, Conference Proceedings.*

Wegner, P. (1996). Interoperability. *ACM Computing Survey , 28*, 285-287.

Wiederhold, G., & Genesereth, M. (1997). The Conceptual Basis for Mediation Services. *IEEE Expert: Intelligent Systems and Their Applications , 12* (5), 38-47.

Yellin, D., & Strom, R. E. (1997). Protocol Specifications and Component Adaptors. *ACM Transactions on Programming Languages and Systems (TOPLAS) , 19* (2), 292–333.

Zeng, M., & Xiao, L. (2001). Mapping metadata elements of different format. *E-Libraries 2001, Proceedings, May 15-17, 2001* (pp. 91-99). New York: Information Today, Inc.

Zhou, C., Chia, L.-T., & Lee, B.-S. (2004). DAML-QoS Ontology for Web Services. *Proceedings of the IEEE International Conference on Web Services.* IEEE Computer Society.

Zhou, J., & Niemela, E. (2006). Toward Semantic QoS Aware Web Services: Issues, Related Studies and Experience. *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence* (pp. 553-557). IEEE Computer Society.

Appendix A. Glossary

Actor Profile: An *Information Object* that models any entity (*Actor*) that interacts with any Digital Library 'system'. An *Actor Profile* may belong to a distinct *Actor* or it may model more than one *Actor*, i.e., a *Group* or a *Community*. (Athanasopoulos, et al., 2010)

Adapter: A component that translates one interface for a component into a compatible interface. It is also known as *Wrapper* or *Connector*.

Broker: A component which is responsible for coordinating communication, such as forwarding requests, as well as for transmitting results and exceptions. It is responsible for coordinating communication in object-oriented distributed software systems with decoupled components that interact by remote service invocations. Introducing a broker component allows to achieve better decoupling of clients and servers. (Buschmann, Meunier, Rohnert, Sommerlad, & Stal, 1996)

Connector: A component that translates one interface for a component into a compatible interface. It is also known as *Adapter* or *Wrapper*.

Controlled Vocabulary: A closed list of named subjects, which can be used for classification. In Library science this is also known as *indexing language*. A controlled vocabulary consists of terms, *i.e.,* particular names for particular concepts.

Data: (*i*) facts and statistics used for reference or analysis. (*ii*) the quantities, characters, or symbols on which operations are performed by a computer. (Oxford Dictionary)

Data Integration**:** an approach aiming at combining data residing in different data sources and providing its users with a unified view of these data (the mediated schema or global schema).

Data Model: an abstract model capturing the distinguishing features of a data set and describing how data are represented.

Digital Library: An organisation, which might be virtual, that comprehensively collects, manages and preserves for the long term rich *Information Objects*, and offers to its *Actors* specialised *Functions* on those *Information Objects*, of measurable quality, expressed by *Quality Parameters*, and according to codified *Policies*. (Athanasopoulos, et al., 2010)

Digital Library System: A software system based on a given (possibly distributed) *Architecture* and providing all the *Functions* required by a particular *Digital Library*. *Actors* interact with a *Digital Library* through the corresponding *Digital Library System*. (Athanasopoulos, et al., 2010)

Digital Library Management System: A generic software system that provides the appropriate software infrastructure both *(i)* to produce and administer a *Digital Library System* incorporating the suite of *Functions* considered fundamental for *Digital Libraries*, and *(ii)* to integrate additional *Software Components* offering more refined, specialised or advanced functionality. (Athanasopoulos, et al., 2010)

GAV: see *Global as View.*

Global as View: a *Data Integration* approach based on the description/characterisation of the mediated schema in terms of a view over the data sources.

Harmonization: *a data manipulation task oriented to make consistent a set of data.*

Interface: *the point of interconnection between two entities.*

Interoperability: *the ability of two or more systems or components to exchange information and to use the information that has been exchanged*. (Geraci, 1991)

LAV: see *Local as View.*

Local as View: a *Data Integration* approach based on the description/characterisation of the data source as a view expression over the mediated schema.

Mediator: a provider of intermediary service linking data sources and application programs (Wiederhold & Genesereth, 1997). An external component hosting the interoperability machinery to mediate between components. Mediation approaches to interoperability are particularly strong in supporting the criteria of autonomy, ease of use and scalability (Paepcke, Chang, Winograd, & García-Molina, 1998).

Ontology: *an explicit formal specification of how to represent the objects, concepts, and other entities that exist in some area of interest and the relationships that hold among them*" (DOI Handbook Glossary, http://www.doi.org/handbook_2000/glossary.html);

Protocol*: a set of guidelines or rules governing governing interactions among parties. In computer science, it is a set of rules governing the exchange or transmission of data between devices;*

Proxy*:* a design pattern making the clients of a component communicate with a representative rather than to the component itself. (Buschmann, Meunier, Rohnert, Sommerlad, & Stal, 1996)

QoS: see *Quality of Service;*

Quality of Service: the capability of a (Web) service to meet a level of service as per factors such as availability and accessibility;

Resource*:* An identifiable entity in the *Digital Library* universe. (Athanasopoulos, et al., 2010)

Service Level Agreement: an agreement between a service provider and a customer that defines the set of Quality of Service guarantees and the obligations of the parties;

Schema mapping*:* the process of transforming the elements of a schema in terms of elements of another schema. It is often used in conjunction with *schema matching*.

Schema matching*:* the process of identifying the similarities between different elements of two diverse schemas. It is often used in conjunction with *schema mapping*.

SLA: see *Service Level Agreement.*

Standard*:* A document established by consensus and approved by a recognized body that provides for common and repeated use, rules, guidelines or characteristics for activities or their results, aimed at the achievement of the optimum degree of order in a given context. (ISO/IEC Guide 2:1996 Standardization and related activities – General vocabulary)

Taxonomy*:* A subject-based classification that arranges the terms in the *controlled vocabulary* into a hierarchy.

Thesauri*:* A *taxonomy* equipped with a richer vocabulary for describing the terms than taxonomies do. It is equipped with other constructs for better describing the world and allowing to arrange subjects in other ways than hierarchies. These constructs include tagging taxonomy terms with properties like "broader term", "related term", "scope note", and "use".

User Profile: *see Actor Profile*

Wrapper: A component that translates one interface for a component into a compatible interface. It is also known as *Adapter* or *Connector*.

Appendix B. Index of Solutions

Appendix C. Experts Notes and Comments

C.1 Implementing a Shared Information Space[187]

One way to achieve Digital Library interoperability is to specify the creation of a standard set of services that are provided by every digital library. Then each digital library can invoke a common access or discovery service on another resource digital library to retrieve data or metadata. This approach assumes that a common purpose exists behind the desire for interoperability. However, the motivation for digital library interoperability may be based on:

- Union catalog construction;
- Federation of existing catalogs;
- Re-purposing of a collection;
- Formation of a new collection;
- Interaction with data processing pipelines;
- Migration of digital material to an archive.

Each of these purposes requires interaction between multiple 'data management systems'. The purposes may be expressed as different sets of required services, implying that service-based interoperability will be difficult to achieve. However, we can look at the basic mechanisms that enable the construction of differentiated services, and examine whether interoperability can still be achieved even when the motivations and services are different.

This note looks at five issues[188]:

(1) Whether interoperability mechanisms need to consider management of the shared information;

(2) Whether the digital library interoperability model should be based on bi-directional interaction between Consumers and Resources;

(3) Whether interoperability should be defined in terms of the properties of the information and data that are being shared between digital libraries;

(4) Whether the interoperability model should be targeted towards all data management applications, not just digital libraries;

(5) Whether a characterization in terms of Organization, Semantic, and Technical views is sufficiently detailed for interoperability between all types of data management systems.

In practice, interoperability between digital libraries should not require modification to any existing digital library. Instead, the interoperability mechanisms should be able to transform between the protocols, semantics, policies, and functions that are already in use. An interoperability mechanism should enable each Provider to be accessed by each Consumer without either the Provider or Consumer requiring modification of their environment, even when they use disparate protocols. For each of the five issues, we look at motivating applications, implications of the proposed extension, and systems that implement the capability.

[187] Editor note: this piece has been authored by Reagan W. Moore. Notes have been added to clarify some of the issues that are raised.

[188] Editor note: these issues subsumes a content centric approach.

On issue 1. In many data management applications, the sharing of information is highly controlled. Policies are associated with the data that are shared. The Consumer is required to assert that the policies will be enforced after they have received the data. This implies that for interoperable digital libraries, policies for use of the shared data must also be exchanged. Simple examples of controlled data include patient data, proprietary data, and classified data. In each example, the acceptable uses of the data require the enforcement of associated policies. The interoperability model needs to encompass simultaneous exchange of policies and data.

Data grids implement this capability through the formation of a shared collection that is controlled by explicit policies. Interoperability is defined in terms of the shared collection that is formed, and the policies that are enforced on the shared collection. In effect, the Resource Digital Library can specify policies that will be enforced even after sharing the data. Policies can be enforced on the shared collection that are derived from the policies that are applied in the resource digital library.

In this perspective, interoperability corresponds to the creation of a new collection that integrates data, information, and policies from each of the contribution resource digital libraries.

On issue 2. A major difficulty with the proposed digital library interoperability model is that the interaction between the Consumer and the Resource needs to be bi-directional.[189]

Consider the issue of scale with future digital libraries holding billions of records and exabytes of data. The Consumer may have many digital objects that they want processed by a service provided by the Resource.

It will not be possible to send all data from the Consumer to the Resource to invoke a specific function. Instead, the function provided by the Resource will need to be sent to the Consumer for application on the Consumer's resources. Future digital libraries will execute their services at the resources where the data are stored.

A Consumer should be able to invoke a task that is exported from the Resource to the Consumer and executed on the Consumer's system. For example, instead of invoking a web service at the Resource, the Resource sends a web applet to the Consumer for execution. Data grids support task distribution for managing distributed data at scale across internationally shared collections. Data grids provide unifying protocols for interacting with multiple independent Resources, and executing tasks at each Resource, or within the shared collection.

This scenario also works in the reverse situation, where the Consumer desired data from the Resource, but requires the Resource to apply processing on the data before transmission. In this case, the Consumer could send a required processing function to the Resource for execution of digital objects stored in the Resource repository.

On issue 3. A third concern is that the digital library interoperability model focuses on interoperability between digital library services. An alternate perspective is to focus on the properties of the shared collection that is being created by the collaboration of two or more digital libraries. The model should define the properties of the shared collection and the set of functions that the shared collection should implement to enforce the desired properties. Data grids do this by installing middleware that maps from the properties of a Resource to the desired properties of the shared collection.[190] The common tasks are implemented through a common invocation mechanism, but executed using the protocol of the specific

[189] Editor note: the framework is conceived to model an interoperability scenario. Neither the notion of Resource nor the quantity of information about it is part of the framework.

[190] Editor note: this is an example of a Mediator-base approach, were the mediator is Provider side.

digital library. This makes it possible for each digital library to implement unique infrastructure, unique protocols, unique semantics, unique policies, and unique procedures. The middleware (data grid) manages the properties of the shared collection independently of each digital library. In effect, the data grid is the interoperability mechanism. The shared collection is the result of all interoperability interactions.

An interoperability model can then be based on the following interaction scheme. Both the Provider and the Consumer interact with a shared collection. The properties for management of the data and information within the shared collection are determined by a Consensus between the Provider and the Consumer. This makes it possible to maintain control over the use of the shared information even after the data and information have left the Provider. Instead of interacting with a Provider, a Consumer interacts with a shared collection. This approach extends agreement-based approaches to enforce policies even after the data have left the Provider. The policies used to manage the shared collection can be different than the policies used by either the Provider or Consumer digital libraries. This approach also extends mediator based approaches, in that the Provider, Consumer, and a third-party Client can all use different protocols. In this approach, the Consumer is differentiated from the Provider in that the Consumer initiates the formation of the shared collection. However the Provider still retains control over what is done with their data through the consensus policies on the shared collection.

On issue 4. A fourth concern is that the interoperability model should function across all types of data management systems: file systems, backup systems, digital libraries, data grids, data processing pipelines, archives, reference collections, etc. These systems typically support a single stage of the scientific data life cycle (organization, sharing, publication, analysis, and preservation).

An implication is that a finite set of operations should enable the instantiation of all desired operations between two digital libraries. In practice, the standard operations correspond to Posix I/O commands, augmented with commands for interacting with databases and structured information. Since programming at the byte level is difficult, aggregations of Posix I/O commands that manipulate well-defined data structures are captured in micro-services that implement basic functions. The micro-services are chained together to implement a desired procedure.

A second implication is that the micro-services need to be invariant across operating systems and data management applications. This requires mapping the augmented Posix I/O protocol standard to the I/O protocol used by each data management application. The same micro-services can then be executed on any operating system for interaction with any data management applications.

A third implication is that the procedures composed through chaining of micro-services need to be explicitly controlled by rules that are applied at each Provider Resource. This is achieved by implementing a distributed Rule Engine. Data can only leave a Resource after compliance with all local policies.

A fourth implication is that the shared collection can be governed by additional rules that are applied uniformly across all Providers. This makes it possible for Providers to retain control of their data, while interoperating with data and information obtained from other Resources through a consensus set of policies and procedures.

A fifth implication is that the micro-services need to exchange structured information that depends upon the underlying operations. It turns out that there is a finite set of data structures that need to be supported across data management applications. Interoperability is based on a set of well-defined:

- Shared data structures. The iRODS system currently supports about 50 well-defined data structures.
- Shared micro-services (basic functions). The iRODS system currently supports about 219 well-defined micro-services.

- Shared state information. The iRODS system currently supports about 209 attributes needed to describe users, resources, data, collection, and rules.
- Shared policies. The iRODS system maps policies to policy enforcement points that are embedded within the data grid framework. Currently about 72 policy enforcement points are instrumented to support each aspect of data management, from user creation, to resource selection, to access control checks, to manipulation of state information.
- Shared federation. Multiple iRODS data grids can control federation through explicit federation policies that define what a user from a Consumer data grid is allowed to do in a Resource data grid. Federation policies can be implemented that enable data flow through chains of data grids, replication of data into a central archive, distribution of data into slave data grids from a master data grid, peer-to-peer exchange of information, and many other interaction models.
- Shared authentication. An iRODS data grid provides single sign-on across all Providers accessed through the data grid. In addition, users within a Consumer data grid can be granted an identity in a Resource data grid through a federation policy. However, authorization is always done by the Resource data grid. The original provider of the data needs to remain in control of how their data are used. The Resource data grid can enforce stronger authorization policies than implemented within the shared collection.
- Shared access methods. The data grid is the intermediary between the access method and the Resource Provider, and does the protocol transformations needed to map between the Resource Provider protocol, the data grid protocol, and client protocols. This makes it possible for any client that accesses the data grid to be able to interact with any type of Resource Provider that is linked by the data grid. A major use of data grid technology is to support access to a legacy data management system through modern access methods, without having to modify the legacy data management system.

On issue 5. A fifth concern is that the characterization of interoperability based on an organizational view, a semantic view, and a technical view is too high level. As discussed above, each of these areas is differentiated into multiple types of interoperability mechanisms:

- Organizational view. This includes the policies enforced on the Provider data grid, the policies enforced on the shared collection, and the policies enforced on federation of shared collections.
- Semantic view. This includes the state information managed by the data grid, the types of queries that can be applied to federated databases, and the transformations that can be imposed on attributes retrieved from the federated databases. The iRODS data grid implements a DataBaseResource (DBR) that encapsulates the SQL or query that may be applied to a remote database. Each request is explicitly contained within a DBR, which is stored as a file within the data grid. All policies and authentication and authorization mechanisms applied on files are also applied on queries to a remote database. The result of the query is encapsulated within a DataBaseObject (DBO) that is also stored as a file within the data grid. All policies enforced on file access can also be enforced on access to the result of a query. An implication is that transformation of semantics between databases now reduces to imposition of policies implemented, for example, as XSLT transformations on XML-encapsulated database tables.
- Technical view. This includes shared data structures, shared micro-services, shared authentication, and shared access methods.

Similarly, the differentiation into Provider, Resource, and Task needs to be expanded to include the shared collection that constitutes re-use of the Provider's data and information:

- Provider can be any data management application, whether digital library, data grid, archive, or data processing pipeline.
- Resource can be a data object, metadata record, policy, procedure, micro-servicer, or aggregations of these entities.[191]
- Task can be a procedure that is controlled by an associated policy. This includes access procedures, transformation procedures for data and metadata, assessment procedures.
- Shared collection contains the data and information that are being re-purposed for use by a Consumer. The shared collection includes not only the data and information, but also the policies and procedures that may be applied, and the assessment criteria that validate appropriate use of the material.[192]

The challenge of characterizing the functionality of a digital library can be simplified by examining this broader space of distributed data management applications. We observe that the technologies that are used to manage the full data life cycle can be subsumed into the following paradigm:

- definition of the driving purpose behind the formation of a collection;
- definition of the set of properties that are required by the purpose;
- definition of the set of policies that enforce the desired properties;
- definition of the set of procedures that are invoked by the policies;
- definition of the set of state information attributes generated or used by the policies;
- definition of the assessment criteria that verify that the properties have been maintained.

From this perspective each stage of the data life cycle corresponds to the set of properties, policies, procedures and state information that fulfill the driving purpose. Consider the following scientific data life cycle:

- Original project data collection formation: the team members decide on the organization policies, the data formats, and the semantics for local use.
- Data grid for sharing the collection with other groups: the team members decide on policies for access, distribution, submission, and derived data products.
- Digital library for formal publication of the collection: the collection policies are extended to ensure compliance with community standards for descriptive metadata, provenance, and organization. Standard services are created for browsing and discovery.
- Data processing pipeline: policies are created for standard data products that are calibrated, projected to the desired geometry and coordinate system, and transformed into physical quantities.
- Preservation environment: policies are created for managing authenticity, integrity, chain of custody, and original arrangement.

In each case the underlying digital libraries continue to provide basic storage and access mechanisms while the data grid enforces the set of policies and procedures for the shared collection. A digital library corresponds to a particular instance of a set of policies and procedures for a specific stage of the data life cycle. To unify the data management applications we can think in terms of virtualization of the data

[191] Editor note: this is exactly the notion of Resource that is part of the presented framework.

[192] Editor note: this notion is needed in content-centric scenarios only. The proposed framework has been conceived to deal with a rich array of Resource including Information Objects, Users, Functions, Policy, Quality Parameters and Architectural Components.

life cycle through the management of the evolution of the policies and procedures as the driving purpose for the collection evolves.

The simplest characterization of a digital library is through the mechanisms that enable virtualization of the data life cycle and automation of digital library functions. This in turn implies the characterization should build upon:

- Policy-based data management which captures policies as computer actionable rules
- Highly extensible data management systems that compose procedures from computer executable functions.

The choice of policies to enforce needs to be decoupled from the choice of implementation of the procedures. Interoperability can then be addressed at multiple levels:

- Specification of the policy that is being enforced
- Invocation of the procedure that the policy controls
- Management of the state information that is generated by the procedure
- Validation of assessment criteria

Data grids implement these capabilities and are being successfully applied to support all stages of the data life cycle. While it is true that each stage uses different policies and procedures, there is a common generic framework in which the policies and procedures can be executed. Data grids provide the multiple levels of virtualization needed to manage formation of shared collections that are governed by a joint consensus.

Interoperability can be considered as the mechanisms that enable the formation of a shared collection across independent organizations. The properties of the shared collection can be managed independently of the original data management environments. Interoperability can be achieved by assembling a data grid that communicates with each digital library. From this perspective interoperability is not the exchange of information between digital libraries but the formation of a new collection that uses resources from multiple digital libraries. This decouples the mechanisms used to support the new shared collection from the mechanisms used to support each of the underlying digital libraries. The implication is that the mechanisms used to support the shared collection must be able to invoke the protocols of each of the original digital libraries.

Consider the following example:

- Data within a Data Conservancy digital library (which uses the OAI-ORE protocol to federate web services) are combined with data within an ArchiveMatica digital library (which chains micro-services composed from Python scripts using Unix pipelines).
- The desired data from each collection are registered into a data grid (such as iRODS). This establishes an identity for the data within the shared collection without moving the data.
- Policies and procedures are defined to enforce the desired properties of the new shared collection.
- Middleware is used to enable the application of the policies and procedures at each digital library.
- Drivers are written for the data grid to map from the middleware protocol to the digital library protocol.
- Similar mappings are created to manage semantics, data format migration, and name space mappings.
- Policies are enforced at each digital library through a distributed rule engine.

A viable approach based on data grids is to consider interoperability from the perspective of the formation of a shared collection. The properties of the shared collection are defined independently of the properties of the original digital libraries. The policies and procedures managing the shared collection are defined independently of the policies and procedures of the original digital libraries. The shared collection requires the virtualization mechanisms that enable it to interact with the policies and procedures of the original digital libraries. Interoperability is then defined as the ability for the data grid to invoke the protocols of the resource digital library, while enforcing the policies of the resource digital library.

An example of a system that enables interoperability is a mounted collection interface. This is the set of information exchange functions that allow a data grid to acquire information from a resource digital library to identify and access individual data objects. There is a set of 22 basic functions that can be mapped to the protocol of each data management application that are sufficient to handle all data manipulation operations.

Interoperability is the set of mechanisms that enable the formation of a shared collection that spans multiple independent digital library systems. Interoperability is characterized by the desired properties of the shared collection that can be created.

A third class of interoperability patterns exists that uses Policy-based approaches on shared collections. Specifically, the iRODS data grid couples agreements with mediation through explicit enforcement of policies and procedures at each Resource.

Other challenges with the interoperability approaches include:

(1) The interoperability mechanisms are decoupled from data management mechanisms. They specify access, but not how management should be enforced on the shared data. The expected management policies need to be made explicit and enforceable through exchange of policies.

(2) There are a wide variety of clients used by data management applications, including: Web services, Web browsers, Workflow systems, Load libraries (Python, Perl, PHP), Java class libraries, Unix shell commands, Grid tools, Digital libraries (Fedora, DSpace), File system (FUSE, WebDAV), Structured data libraries (HDF5, OpenDAP), Dropbox.

Interoperability implies the ability to access a collection not only from another digital library, but also through other types of data clients.

(3) Quality interoperability should be defined within a context based on the properties that are desired for the shared data and information.

Appendix D. Acknowledgements

Many people have contributed to the production of this artefact.

The current version of this artefact benefit from the comments and contributions received by the Digital Library community built and operated in the context of the DL.org Coordination Action. In particular, the following people contributed to the document by providing the DL.org team with comments, visions and insights during their participation in one of the six DL.org operated Working Groups: Kevin Ashley (DCC – Policy WG), Detlev Balzer (EFG project – Content WG), Tiziana Catarci (University of Rome – User WG), Vassilis Christophides (University of Crete – Functionality WG), Genevieve Clavel-Merrin (Swiss National Library – Quality WG), Antonella De Robbio (University of Padua – Policy WG), John Faundeen (USGS – Policy WG), Nicola Ferro (University of Padua – Quality WG), Ed Fox (Virginia Tech – Functionality WG), Stefan Gradmann (Humboldt University – Content WG), Sarah Higgins (DCC – Quality WG), C.H.J.P. Hendriks (Naturalis – Content WG), René van Horik (DANS – Quality WG), Wolfram Horstmann (Bielefeld University Library – Quality WG), George Kakaletris (University of Athens – Functionality WG), Sarantos Kapidakis (Ionian University of Corfu – Quality WG), Akrivi Katifori (University of Athens – User WG), Steve Knight (National Library of New Zealand – Policy WG), Georgia Koutrika (Stanford University – User WG & Functionality WG), Paolo Manghi (ISTI-CNR – Content WG), Natalia Manola (University of Athens – User WG), Carlo Meghini (ISTI-CNR – Functionality WG), Jan Molendijk (Europeana – Architecture WG), Luc Moreau (University of Southampton – Content WG), Andreas Nürnberger (Technical University of Braunschweig – User WG), Pasquale Pagano (ISTI-CNR – Architecture WG), Hans Pfeiffenberger (Alfred Wegener Institute – Policy WG), Axel Poigné (Fraunhofer IAIS – Architecture WG), Andreas Rauber (Vienna University of Technology – Functionality WG), Dirk Roorda (University of Amsterdam – Quality WG), Robert Sanderson (Los Alamos National Laboratory – Architecture WG), Tefko Saracevic (Rutgers University – Quality WG), MacKenzie Smith (MIT – Policy WG), Dagobert Soergel (University of Buffalo – Functionality WG), Manfred Thaller (University of Cologne – User WG).

A special thanks goes to Maria Bruna Baldacci (ISTI-CNR) for her considerable help in improving the readability of most parts of this document.