

# A Functionality Perspective on Digital Library Interoperability

G. Athanasopoulos<sup>1</sup>, E. Fox<sup>2</sup>, Y. Ioannidis<sup>1</sup>, G. Kakaletis<sup>1</sup>, N. Manola<sup>1</sup>, C. Meghini<sup>3</sup>,  
A. Rauber<sup>4</sup>, D. Soergel<sup>5</sup>

<sup>1</sup>Dept. of Informatics and Telecommunications, University of Athens, Greece  
{gathanas, yannis, gkakas, natalia}@di.uoa.gr

<sup>2</sup>Dept. of Computer Science, Virginia Tech, Blacksburg, VA, USA fox@vt.edu

<sup>3</sup>Istituto della Scienza e delle Tecnologie della Informazione, Consiglio Nazionale delle  
Ricerche, Pisa, Italy carlo.meghini@isti.cnr.it

<sup>4</sup>Dept. of Software Technology and Interactive Systems, Vienna University of Technology,  
Vienna, Austria rauber@ifs.tuwien.ac.at

<sup>5</sup>Dept. of Library and Information Studies, University of Buffalo, New York, USA  
dsoergel@buffalo.edu

**Abstract.** Digital Library (DL) interoperability requires addressing a variety of issues associated with functionality. We report on the analysis and solutions identified by the Functionality Working Group of the DL.org project during its deliberations on DL interoperability. Ultimately, we hope that work based on our perspective will lead to improved architectures and software, as well as to greater interoperability, for next-generation DL systems.

## 1 Introduction

A huge volume of information and knowledge is acquired and managed by distinct Digital Libraries (DLs). This leads to problems for academic and public libraries that often work with scores of such DLs and seek to support patrons facing a broad range of systems and services. Similar problems are faced by students, faculty, researchers, scholars, knowledge workers, and the general public. Also of concern is e-science, where labs and centers must use different DLs to address global challenges.

Interoperability among all the DLs needed in each case is a serious concern. Manifesting a broad range of features and capabilities, DL systems employ diverse proprietary solutions and varying applications of a broad range of standards. The problem is further aggravated by the complexity and scale of modern DL systems and problems such as API mismatch, data format mismatch, and missing components.

Interoperability has been the main issue of concern for the DL.org project [4]. Its work is based on the DELOS Digital Library Reference Model [3], in particular, the multi-dimensional representation of the DL domain and the identification of six primary concepts that characterize Digital Libraries: content, users, functionality, policy, quality, and architecture. In this paper, we present results from the discussions

of the DL.org Functionality Working Group [4]. This Working Group is focusing on interoperability with respect to one of these concepts, DL functionality, while still remaining within the broader context of the entire DL space.

## 2 Functions, Interoperability, and Compatibility

There are many definitions of *function*. The Reference Model defines function as: “*an action a DL component or a DL user performs*” [3]. In software engineering, a function is a portion of code or a module that performs a specific task (or action); it is embedded within a larger program but remains relatively independent from the rest of the code. Function interoperability is often concerned with software modules that implement a DL function. The mathematical definition of a function as a mapping from a domain to a range is not important here.

Function interoperability is of particular importance in DLs, as indicated in Table 1. Such interoperability serves three main purposes: i) To provide users of one DL access to the content and functionality of other DLs; ii) To harmonize the user experience provided by different DLs so that the user who has learned to use one DL can use other DLs easily; iii) To save effort in creating new DLs or adding functionality to existing DLs, by reusing existing software components.

**Table 1.** Indicative set of functions where interoperability is especially important.

Behind the scenes	For users
Feature Extraction Classification/Clustering Single Authorization/Single Sign-on Analysis/Statistics operations User Profile Management Harvesting, Aggregating Preservation and Backup	Federated search. Integration of additional external content sources on the fly. Visualization of timelines, maps, videos, etc. Browsing based on same look-and-feel.

To achieve interoperability of functions, one may use a registry that allows for the discovery of software modules that implement sought-after functionality in a given software context for a given user group. Such a registry should show the different ways in which functions can be interoperable.

From a system-based point of view, three important ways to achieve function interoperability are the following:

1. Based on processing (e.g., function Fa may utilize the functionality offered by another function Fb either by directly incorporating the provided functionality within Fa or by calling it as an external service);
2. Based on data/content (e.g., the outcome of function Fa is da which is used by Fb as input via direct exchange / conversion / replacement);
3. Based on cross-function compatibility (e.g., functions Fa and Fb have the same interface).

From a user point of view there are issues related to functionality that concern the compatibility of products. Such compatibility is directional and can be expressed in

two levels. More formally, DL B is product-compatible with DL A with respect to functionality if the following hold:

1. DL B provides all functions that DL A provides (product compatibility with respect to functions provided),
2. DL B uses the same interface with DL A to invoke functions, where interface refers to names of functions, shape and color of buttons, screen layout, command names, and syntax (full product compatibility).

From a framework based point of view, function interoperability requires:

1. an entity-relationship schema,
2. a taxonomy of the ways in which functions can interoperate, and
3. a template for the description of functions and software components.

### 3 Specifications, Solutions, and Ontologies

A function specification, using a template such as the following, facilitates identification of what a function does and how a system or a human may interact with it.

- **Function Behavior:** providing a description of what a function does and of the supported interaction with actors (systems/users)
- **API/Interface Specification:** illustrating the Input and Output data and parameters, data formats/standards, pre-conditions and post-conditions
- **Dependencies/Relationships/Use:** detailing the operating environment in which a function runs; other functions needed; functions that invoke this function; composite functions and workflows
- **Interoperability Concerns:** documenting what is required for interoperability and how does a specific implementation meet these requirements
- **Performance evaluation, assessment, and advice for use**

Specifications of many of the above properties can use existing widely used standards such as IDL, WSDL, SAWSDL, OWL-S, WSMO, or BPEL4WS.

Clearly, there are generic concerns regarding interoperability of functions that cut across all types of software systems and others that are particular to DLs. A function can be implemented as a *service*; thus the Service Oriented Computing (SOC) domain is of particular interest. In that context, there are already proposed static and dynamic solutions. Static solutions can be found in the e-Framework ([www.e-framework.org](http://www.e-framework.org)) or the RosettaNet community ([www.rosettanet.org](http://www.rosettanet.org)). Both initiatives accommodate a standards-based service-oriented approach with well-defined services that facilitate the whole range of the functionality and provided features. On the other hand, dynamic solutions address several of the specified issues and rely on the use of formally defined theories for the automated provision of adapters. Adapters handling incompatibilities (e.g., with interfaces, behavior, and pre/post conditions) have been developed by the SOC research community [1, 2]. These two distinctive types of solutions can serve as the basis for DL function interoperability as well.

Another important aspect when dealing with interoperability is that functions can be related in various ways, including the common relationship of sub-function. As Table 2 shows, the standard DL function “search” has many sub-functions, especially when advanced search is concerned. Thus, it is essential that taxonomic or, even

better, ontological descriptions of DL functions be provided. A thorough treatment of this matter is given elsewhere [5], using the 5S framework; other work involves ontology mappings, alignments, and merging [6].

**Table 2. Sub-functions of search**

Quick Search	Advanced Search
Enter a query and click search	Enter keywords or phrases for selected fields
Enter keywords or phrases for selected field	Select keyword from a list
Limit results	Select Boolean operator (explicit)
Search subscribed titles	Define phrase match (explicit)
Clear	Search within results
	Limit results to (preselection), Sort by (preselection)
	Select display options, Display X results per page
	Display search history

## 4 Conclusions

The Functionality Working Group of the DL.org project has explored issues, approaches, and solutions related to the interoperation of DL functions. Essential are appropriate description mechanisms and registries that will facilitate the publication and discovery of functions. More work is also needed on function taxonomies and ontologies, function composition, and the relationship to interoperability of the other DL concepts, as addressed by other DL.org Working Groups.

**Acknowledgments.** Partially supported by the European Commission under project “DL.org: Digital Library Interoperability, Best Practices and Modelling Foundations”, Contract num: 231551.

## 5 References

1. Benatallah, B., Casati, F., Grigori, D., Nezhad, R., Toumani, F.: Developing Adapters for Web Services Integration. In 17th Conference on Advanced Information Systems Engineering (CAISE), June 2005, Porto, Portugal
2. Bordeaux, L., Salatin, G., Berardi, D., Mecella, M.: When are two Web Services Compatible? In 5th VLDB Workshop on Technologies for E-Services, August 2004, Toronto, Canada
3. Candela, L., Castelli, D., et al.: The DELOS Digital Library Reference Model, Foundations for Digital Libraries, Ver 0.98., DELOS Network of Excellence - Project no. 507618 (2008)
4. DL.org: Digital Library Interoperability, Best Practices and Modelling Foundations. EU funded project, Contract no. 231551, Available at: <http://www.dlorg.eu>
5. Goncalves, M. A., Fox, E. A., and L. T. Watson. Towards a Digital Library Theory: A Formal Digital Library Ontology. *Int. J. Digital Libraries* 8(2): 91-114, April 2008
6. Noy, N.: Semantic Integration: A Survey of Ontology-based Approaches. *ACM SIGMOD Record* 33(4), 2004, pp. 65-70