DL.org

Digital Library Interoperability,
Best Practices and Modelling Foundations

Functionality Working Group

Dagobert Soergel

DL.org
Digital Library Interoperability, Best Practices and Modelling Foundations

Interoperability and Reuse in the Functionality Domain

DL.org
Digital Library Interoperability, Best Practices and Modelling Foundations

Working Group Structure

- **Dagobert Soergel (Scientific Chair**), University at Buffalo
- George Athanasopoulos (WG Leader), University of Athens
- Eleni Toli (Rapporteur), University of Athens
- Vassilis Christophides, FORTH, Crete
- Ed Fox, Virginia Technical Institute
- Yannis Ioannidis, University of Athens
- George Kakaletris, University of Athens
- Natalia Manola, University of Athens
- Carlo Meghini, CNR Pisa
- Andreas Rauber, Technical University Vienna

DL.org
Digital Library Interoperability, Best Practices and Modelling Foundations

Projects and initiatives

Projects represented in the group include:

- CASPAR
- D4Science
- DRIVER
- Europeana

DL.org
Digital Library Interoperability, Best Practices and Modelling Foundations

Outline

- Objectives
  Interoperability use cases
- Function Interoperability Framework
- Conclusion

DL.org
Digital Library Interoperability, Best Practices and Modelling Foundations

Objectives

Interoperability use cases

Ultimate objectives

- **Promote rich functionality** over a wide range of systems with a consistent interface
- **Promote best practices and innovation** by educating DL designers, developers, administrators, and users about the rich array of DL functionality
- **Enable finding and reusing software modules** that implement desired functionality
  - **for developers**: reuse existing modules and design for interoperability
  - **for DL managers**: implement cutting-edge functionality in configuring a DL system
  - **for users**: run a module "on the fly" to accomplish a task.
- **Enable federated search**

Interoperability and reuse scenarios

Scenarios

1 **Find desired functions**, and modules that implement them, and assess their interoperability. Enable **functionality sharing**

2 Enable **content sharing and federated search**

3 Make **switching from one DL to another** easy for the user

Dealing with these scenarios requires

1 Understanding the many ways in which functions interoperate

2 A database with detailed descriptions of functions, revising and extending the DELOS Digital Library Reference Model

Solution: **Function Interoperability Framework**

Interoperability and reuse scenarios
Examples

- The developer of a **Browse** module looks for an **automatic clustering** module to incorporate browsing by cluster

- A DL administrator wants to make available a better image search system

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

Issues in interoperability

- API mismatch
- Mismatch in programming environments
  Needed components missing
- Mismatch in data formats
  (overlap with WG Content)

Interoperability use cases

Exchange of program modules between D4 Science and Driver

- Each system would **describe the functions** it implements (e.g. feature extraction from documents or data transformation using grid resources), considering
  - the semantics of the function (what the program module can do)
  - the technical (and, as relevant, administrative) conditions of use.
- Each system could then search the functions offered by the other and **reuse program modules**.

Interoperability use cases

Single deposit

European project OpenAIRE

- Central portal where users come to deposit their publications.
- The internal deposition service subsequently forwards/deposits them in the corresponding local repositories.
- Requires interoperable functionalities among the various repository platforms.

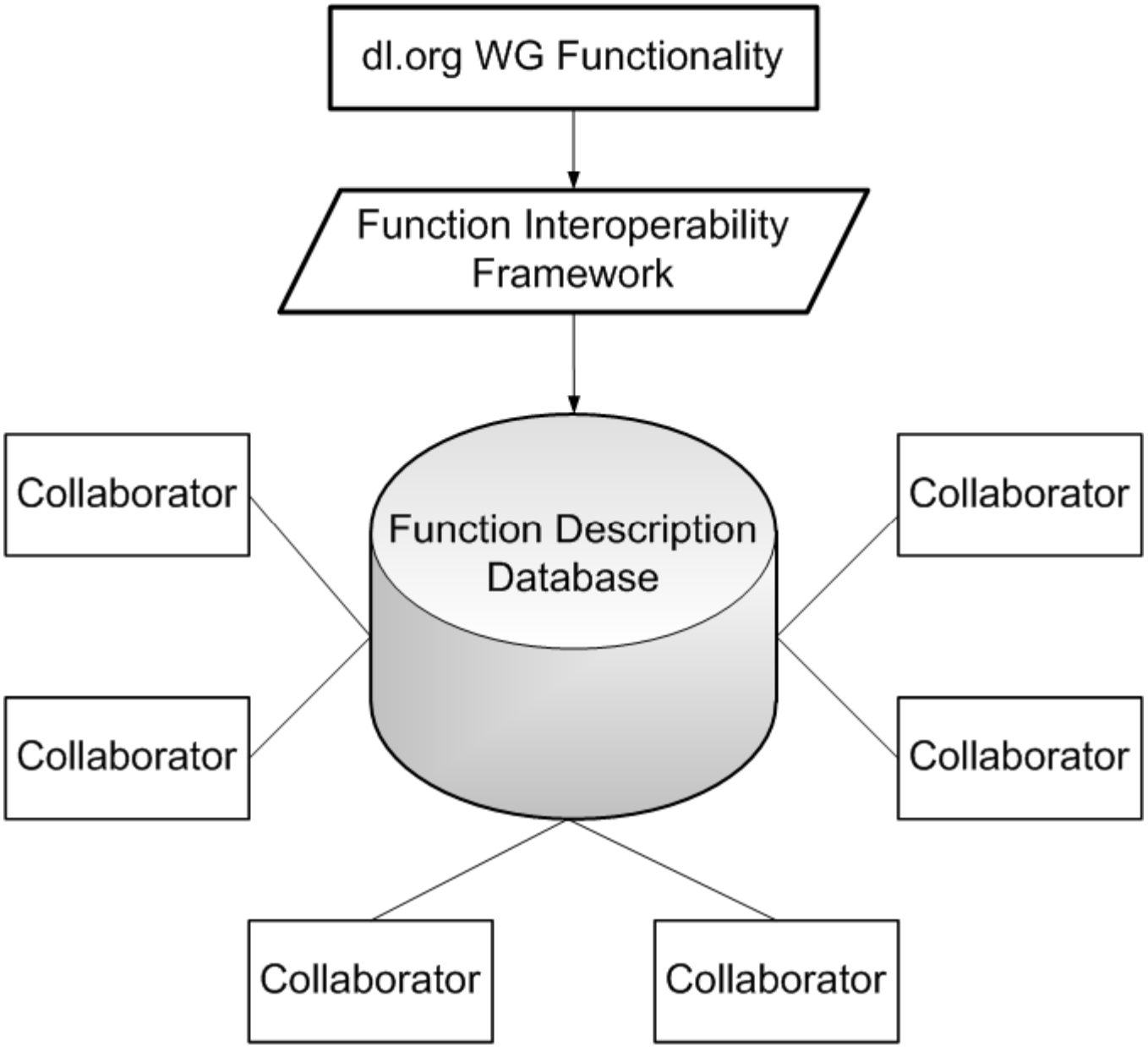
Interoperability use cases

Focusing on behind-the-scenes operations

- **Centralized services** that provide the same service to multiple libraries
  - classification servers
  - format conversion
  - data validation
- The services must be interoperable − based on data − with many DLs
- Can be achieved through
  - standards for data formats
  - standardized authority files.

DL.org
Digital Library Interoperability, Best Practices and Modelling Foundations

The role of the WG Functionality


dl.org WG Functionality
Function Interoperability Framework
Function Description Database
Collaborator
Collaborator
Collaborator
Collaborator
Collaborator
Collaborator

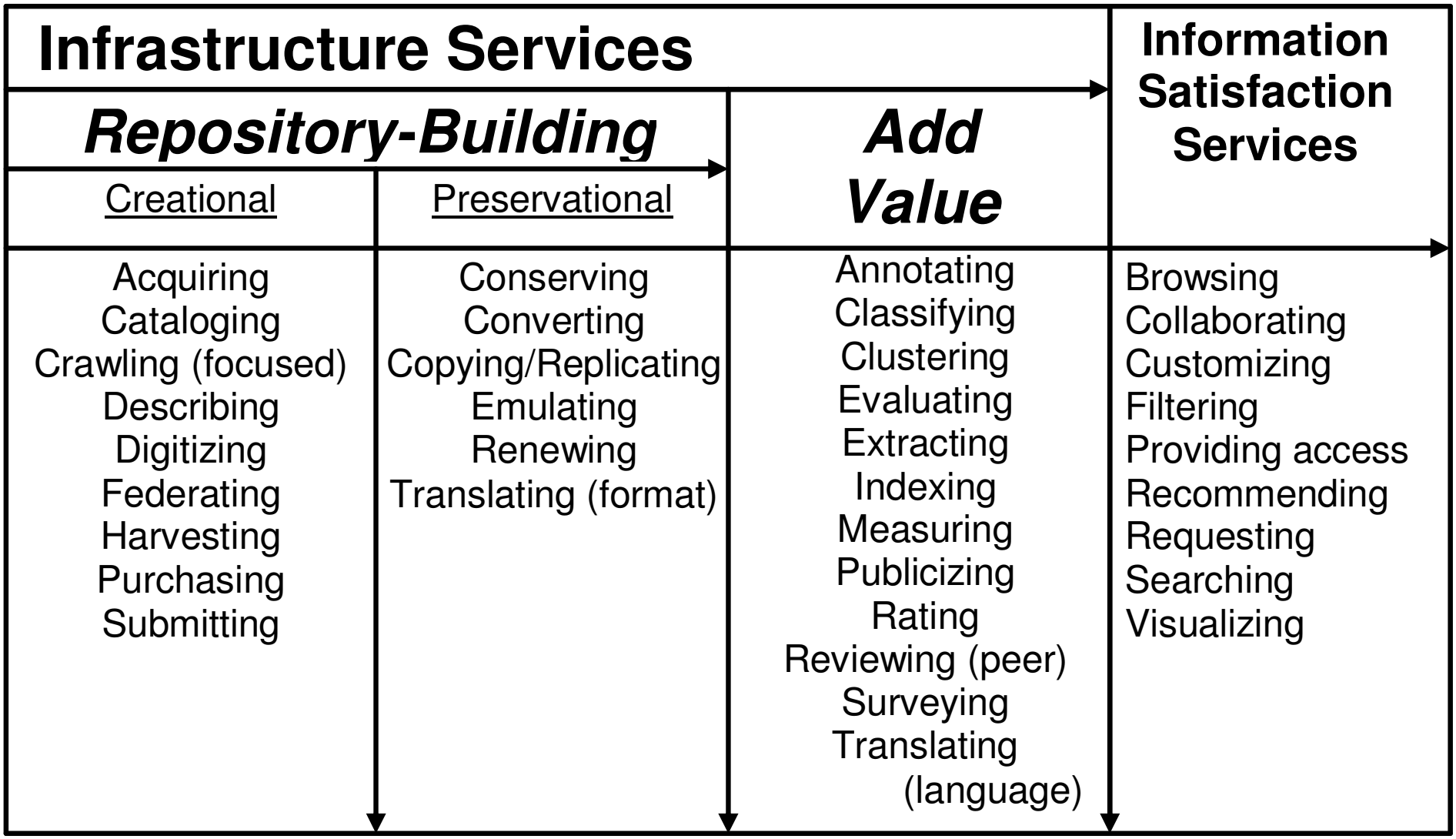
DL.org
Digital Library Interoperability, Best Practices and Modelling Foundations

Function Interoperability Framework

What is a function?

A function in the DLRM is an action a DL component or a DL user performs.

Not restricted to mathematical functions or functions in the programming sense.


Infrastructure Services
Repository-Building
Creational
Preservational
Add Value
Information Satisfaction Services
Acquiring
Cataloging
Crawling (focused)
Describing
Digitizing
Federating
Harvesting
Purchasing
Submitting
Conserving
Converting
Copying/Replicating
Emulating
Renewing
Translating (format)
Annotating
Classifying
Clustering
Evaluating
Extracting
Indexing
Measuring
Publicizing
Rating
Reviewing (peer)
Surveying
Translating (language)
Browsing
Collaborating
Customizing
Filtering
Providing access
Recommending
Requesting
Searching
Visualizing

Functions where Interoperability is specially important

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

Three parts of the Function Interoperability Framework

- An entity-relationship schema
- A taxonomy of ways in which functions can interoperate
- A template for the description of functions and software components

Note: Strong overlap with Architecture WG

E-R schema
for a function description database

Entity types (resource types) (examples)

Function **SoftwareComponent** (a software system, software module, or code snippet) **DesignPattern** (Rike Brecht, Doct. Cons.)	**Data set** **Data format**

Relationship types (examples)

Resource <***hasComponent***> Resource Function <***implementedBy***> SoftwareCo. Function <***represented by***> DesignPattern	Resource <***interoperableWith***> Resource

Terminology

In the following,

function is used broadly to mean either

function or

software component implementing a function

What is function interoperability 1

1 **Interoperability**
(system perspective, focus on software components)

1.1 Composability (f2 can work with f1)

1.2 Replaceability / interchangeability (f2 can replace f1)

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

What is function interoperability 2

A Interoperability of functions based on process

A1 **Interoperability of use (composability)**
Function f1 can use function f2 (conversely, f2 can work in the framework of f1))

A2 Special case: **Interoperability with environment E (composability)**
Function f1 can work in environment E

A3 **Interoperability based on working in same operating environment E (replaceability / interchangeability)**
If Function f1 can operate in environment E
AND f2 can also operate in E,
Then f2 can replace f1

What is function interoperability 2

B Interoperability of functions based on data (content)

B1 Interoperability based on exchanging data (composability)

If Function f1 can operate on the output of f2,

Then f1 and f2 can work together.

f1 and f2 may also exchange data as they run concurrently

B2 Interoperability of functions with data (composability)

If f1 can make use of data set D or of data formatted according to DF

Then Function f1 is interoperable with a data set D or a data format DF

B3 **Interoperability of function based on using same data (replaceability)**

If Function f1 is interoperable with a data set D or data format DF

AND Function f2 is also interoperable with D or DF, respectively,

Then f2 can replace f1

Description / specification /profile of functions

Function Specification: facilitates the identification of what a function does and how one (either a system or a human) may interact with it.

Function description/specification template

- The template shown below applies to
  - the description of a general function, such as *search* or *annotate*;
  - the description of specific software components implementing a function.
  - Not all items apply to the general level, or the description stays very broad, for example with regard to data formats.
- Template focuses on semantics of function specification
- This is a preliminary template. It will be amended as it is applied.

Function specification template

Function Behavior

Description: What is done

Interaction with Actors (Systems/Users)

- Is the function invoked by the user or the system
- What actions does the user take
- What actions does the system take
- Special user groups /roles; user characteristics
- Can the function be applied to different contexts

API/Interface Specification

Dependencies/Relationships/Use

Interoperability Concerns

Function specification template

Function Behavior

API/Interface Specification

Input: Data and parameters, data formats / standards

Output: Data and parameters, data formats / standards

Preconditions

Postconditions

Dependencies/Relationships/Use

Operating environment in which the function runs.

Other functions it needs

Other functions that invoke this function

Other functions invoked. Composite functions

Work flow

Interoperability Concerns

What is required for interoperability (distinguish type of interoperability, for example product compatibility).

How does a specific implementation meet these requirements

Syntax of function specification

Use existing standards

- API/Interface specification
  e.g. IDL, WSDL, SAWSDL, OWL-S, WSMO
- Behavior description
  e.g. OWL-S, WSMO
- Pre and Post conditions
  e.g. WSMO, OWL-S, KIF
- Specification of Composite functions
  e.g. BPEL4WS

Emerging function ontology

Ontology of functions
Function specification vocabulary

Will emerge over time as the database of function descriptions is populated through wide collaboration (crowd-sourcing)

Sub-functions of *search*

Quick Search	Advanced Search
Enter a query and click search	Enter a query and click search
Enter keywords or phrases for selected field	Enter keywords or phrases for selected fields
Limit results to	Select keyword from a list
Search subscribed titels	Select Boolean operator (explicit)
Clear	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

Sub-functions of *annotate*

Select object to be annotated
(need to indicate selection method)

Mark region in the object
(many different methods depending on the object)

Select type of annotation
(highlight, mark with special meaning, text, image, sound)

If text, image, sound

Specify relationship to object to be annotated

Select or create the annotating object
(possibly specifying a region

Annotating within one system

Annotating across systems

DL.org
Digital Library Interoperability,
Best Practices and Modelling Foundations

Conclusion

DL.org
Digital Library Interoperability, Best Practices and Modelling Foundations

Recap

- Objectives
  Interoperability use cases
- WG will produce a
  Function Interoperability Framework
- dl.org should set up an environment in which the DL community can produce a database of function descriptions

DL.org
Digital Library Interoperability, Best Practices and Modelling Foundations

Expected Outcomes

- Interoperability State-of-the-Art survey
- Extensions to the Delos Reference Model
- A Best Practices document
  (DL Technology and Methodology Cookbook)
- One or more papers
- Training course materials

DL.org
Digital Library Interoperability, Best Practices and Modelling Foundations

Take-home message

Unraveling the mysteries of interoperability is harder than you think

Do not clap