



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

G. Athanasopoulos, K. El Raheb¹, E. Fox, G. Kakalettris, N. Manola,
C. Meghini, A. Rauber, D. Soergel

¹Dept. of Informatics and Telecommunications, University of Athens, Greece

k.elraheb@di.uoa.gr

Digital Library Interoperability

function :

“an action a DL component or a DL user performs”

- 1) abstract function (e.g. annotate, browse)
- 2) software component implementing a function

DL Interoperability

Content User

Functionality Architecture

Quality Policy

Prerequisites

- Description
- Publication
- Discovery

A Framework, in appropriate registries, specifying the functions' key characteristics interface, behavior, dependencies, semantics

Functions must comprehend each other !

Goals

- **Functionality sharing:** Find desired functions, and modules implementing them, in a given DL environment
- Enable **content sharing and federated search**
- Make **switching from one DL to another** easy for the user

Scenarios

- The developer of a *browse* module looks for an *automatic clustering* module to incorporate browsing by cluster
- A DL administrator wants to make available a better image search system
- A user found 30 documents in a DL and wants to invoke a *Web service* to create a *multi-document summary*

What's the function interoperability issue?

- **Function Behavior** : description of supported interactions with actors (systems/users), their (logical or temporal) ordering and constraints
- **API Interface Specification** : information about operating a function in a manner of specifying Input and Output parameters
- **Preconditions and PostConditions**: specifications of conditions that should be satisfied prior to and after the execution of a function e.g. assumptions that may hold or quality constraints
- **Dependencies and Relationships** : other functions needed, functions that invoke this function, composite functions and workflows should be also considered.

Types of Function Interoperability

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

Composability (f2 can work with f1)

Replaceability / interchangeability (f2 can replace f1)

(f1 and f2 serve same purpose)

2 **Cross-function (cross-product) compatibility: user perspective**

Similar detailed functionality and user interface

What is needed

a) Standardized shared, rich format for description

b) Standards' compliant interfaces implementation

Function Specification Framework

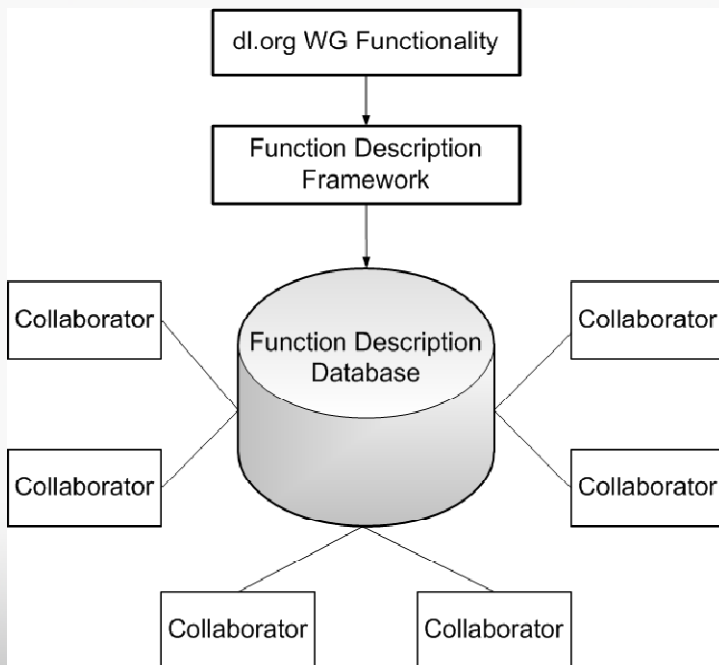
Key Idea

Collaborators can create a database with consistent function descriptions

function description (specification, profile) : what a function does and how a system or a human may interact with it.

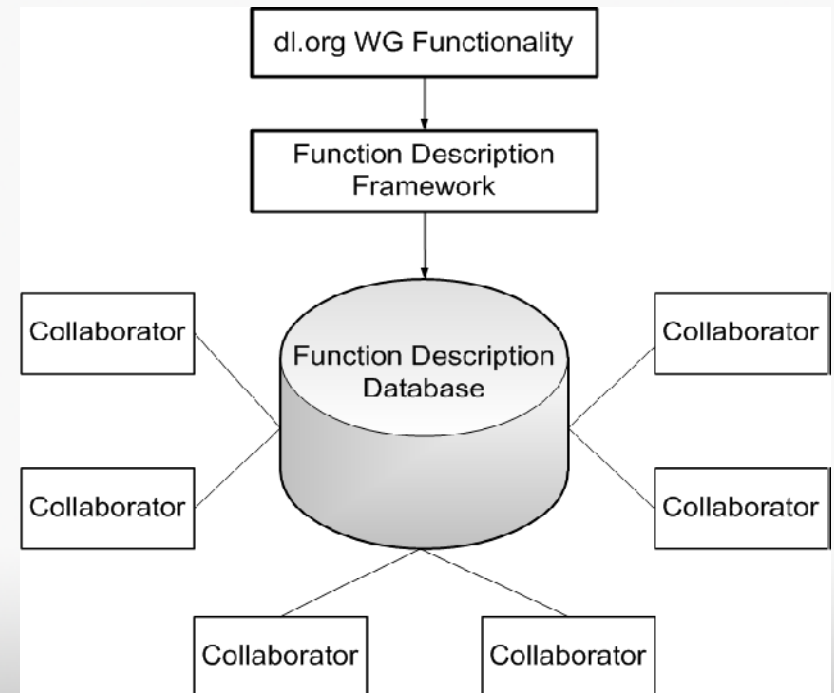
function description framework has three components:

1. An **Entity –Relationship Schema** that defines all the kinds of statements that can be made about a function
2. A **function ontology** provides a **vocabulary** describing the hierarchy of functions
3. A **function description template** that provides a standard way to organize all the statements about a function



Function Description Framework Components

1. **Entity –Relationship Schema**
2. *Function ontology*
3. *Function Description Template*



Entity Relationship Schema Examples

Entity Types

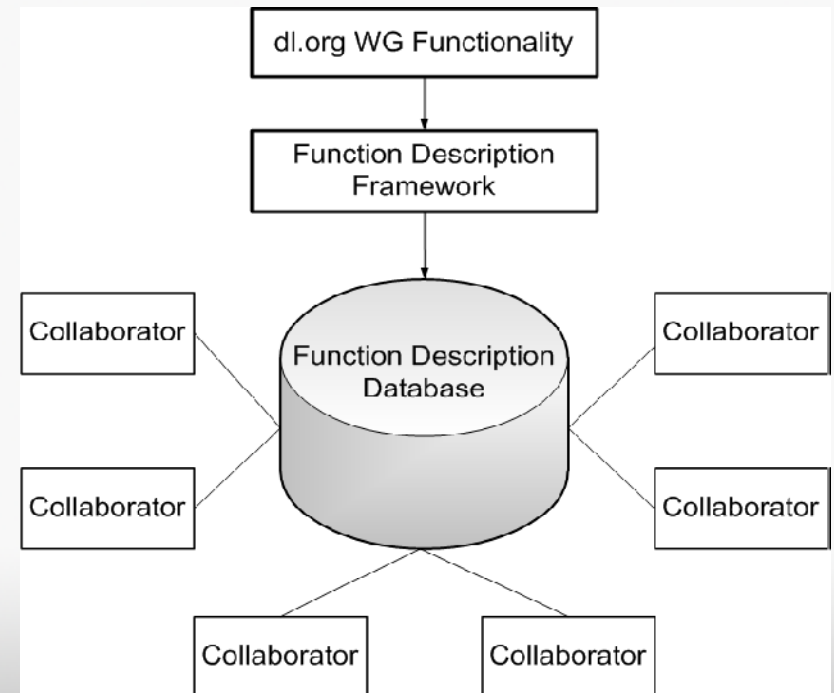
- **Resource** (Function, DataSet, orDataFormat)
- **Function**
- **SoftwareComponent** (a software system or module, or a code snippet)
- **DesignPattern**
- **InteropType**
- **DataSet**
- **DataFormat**

Relationship types

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

Function Description Framework Components

1. Entity –Relationship Schema
2. **Function ontology**
3. *Function Description Template*





Function Ontology Example: Search

Search hierarchy:

discover

 navigate

 browse

search

 quick search

 advanced search

Quick Search

- Enter a query and click search
- Enter keywords/phrases for selected fields
- Limit results to
- Search subscribed titles
- Clear

Advanced Search

- Enter a query and click search
- Enter keywords or phrases for selected fields
- Select keyword from a list
- Select Boolean operator (explicit)
- 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

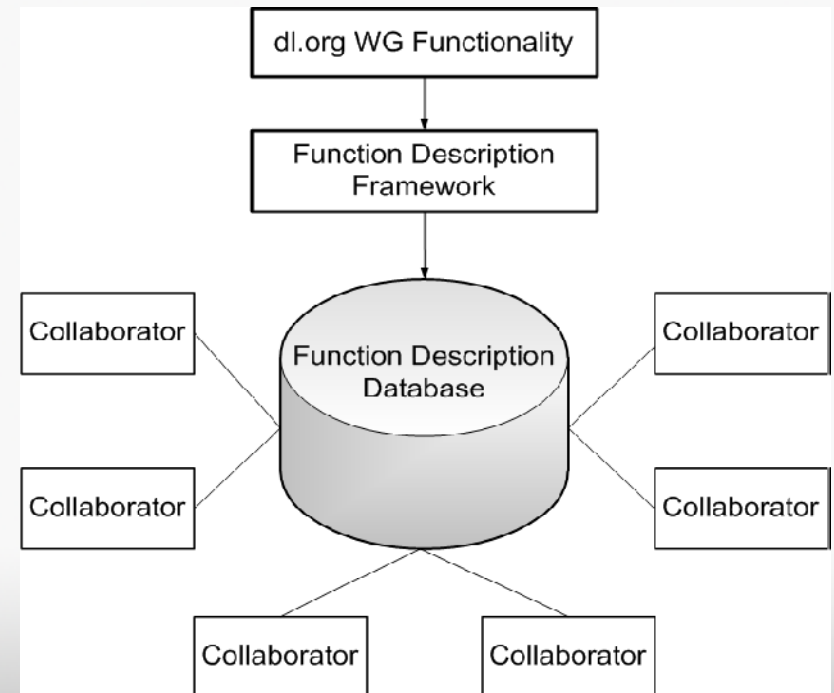
Function Ontology Example: Annotate

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

Function Description Framework Components

1. *Entity –Relationship Schema*
2. *Function ontology*
3. *Function Description Template*



Function Description Template

- **Function Behavior:** What does the function do. This requires the vocabulary of the function ontology
 - **Interaction with Actors (Systems/Users)**
 - Is the function invoked by the user or the system
 - What actions does the user take
 - What actions does the system take
 - Special user groups /roles; user characteristics
 - Can the function be applied to different contexts
 - **API/Interface Specification**
 - Input: Data and parameters, data formats / standards
 - Output: Data and parameters, data formats / standards
- Preconditions and Postconditions:**
- What should hold prior to the execution of the function
 - What should hold once function execution is completed

Function Description Template

- **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
- How does a specific implementation meet these requirements

- **Assessment. Performance. Advice for use**

- For specific types of functions, such as search, the template should include quality parameters -software evaluation criteria

- **Usage conditions**

- Rights, type of license, costs, etc.

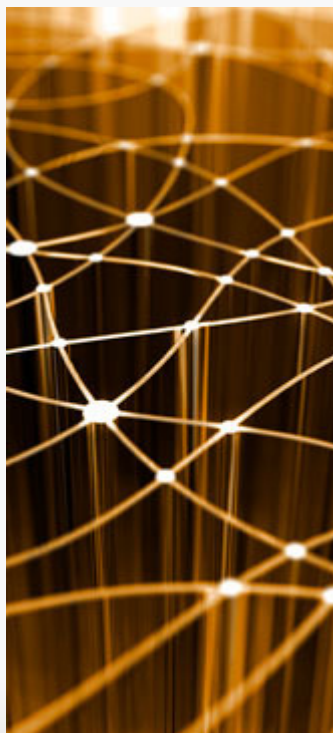
Implementation Scenarios

- Function Specification:
 - Use of **Service-Oriented Computing (SOC)** standards e.g. **WSDL, SAWSDL, OWL-S, WSMO, BPEL4WS**
- Function Publication & Discovery:
 - Based on existing **Service Registries**
 - Syntax based**, e.g., ebXML, UDDI
 - Semantics based**, e.g., **SpiDeR**, METEOR-S , DIRE, PYRAMID-S, Ockham, e-Framework

Looking for: **Scalability, fault tolerance**, rich **syntactic/semantic** format standards, description of additional properties (**behavioral analysis, evaluation** assessment, **interoperability**)

Closing Remarks

- **Benefits resulting from the Framework**
 - Better assertion and **extensibility** steering the of existing DLs for **DL managers**
 - Easier identification and **reuse** of appropriate implementations by **developers** and **system integrators**
 - **On-demand integration** of additional functionality, i.e. to satisfy on-the fly the needs of end-users
 - **Software integration** (composition) algorithms and mechanisms, e.g. AI planning techniques
- **Open Challenges**
 - **Publishing** the specifications in a manner that can be **discovered** and understood among diverse systems is still a big **challenge (under investigation)**
- **Future Work**
 - **Merging** and **aligning** the framework with other DL domains (content, user, quality, policy, architecture)



Thank you

