DL.org

Digital Library Interoperability, Best Practices and Modelling Foundations

The Digital Library Reference Model: Functionality Domain

Carlo Meghini
CNR-ISTI
carlo.meghini@isti.cnr.it

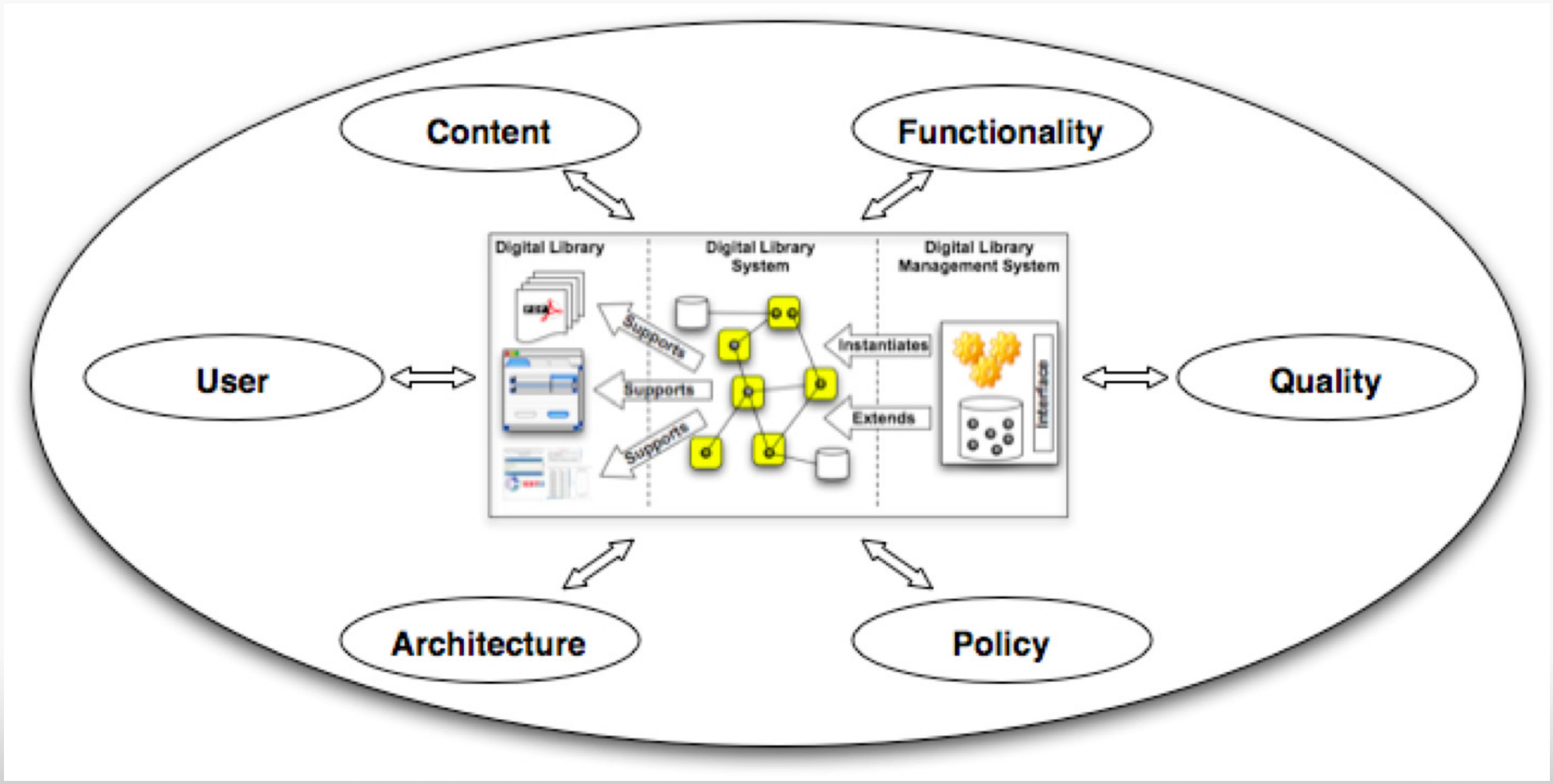
DL.org Autumn School, Athens, 3-8 October 2010

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

Outline

- The context
- The functionality domain
- A scenario

Reference Model



Role of the Reference Model

The *Functionality* concept encapsulates the services that a Digital Library offers to its different users.

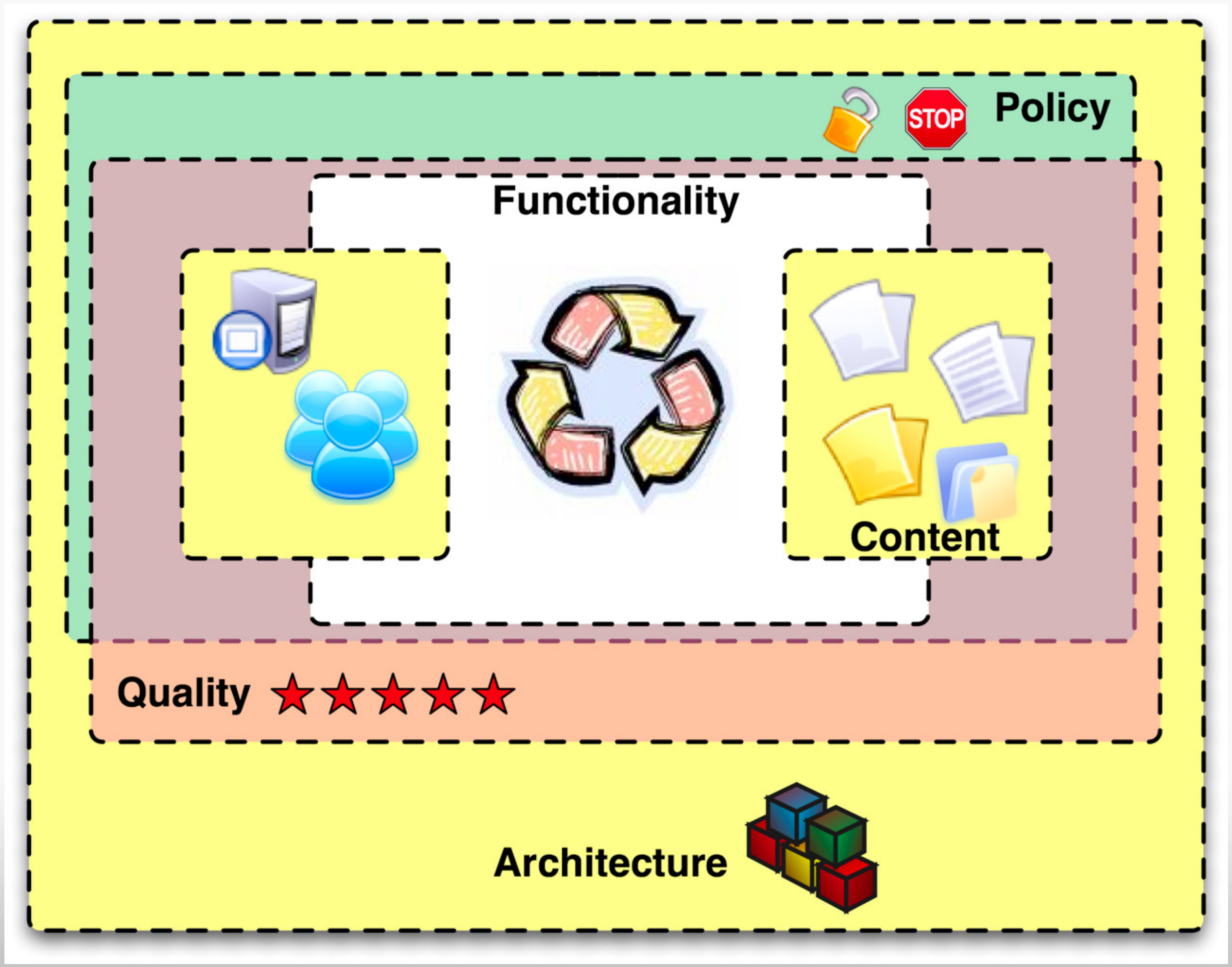
There is no limit to the set of functionalities that a DL can offer to their users:

- technologies evolve
- business models evolve
- expectations evolve

What is the role of a Reference Model?

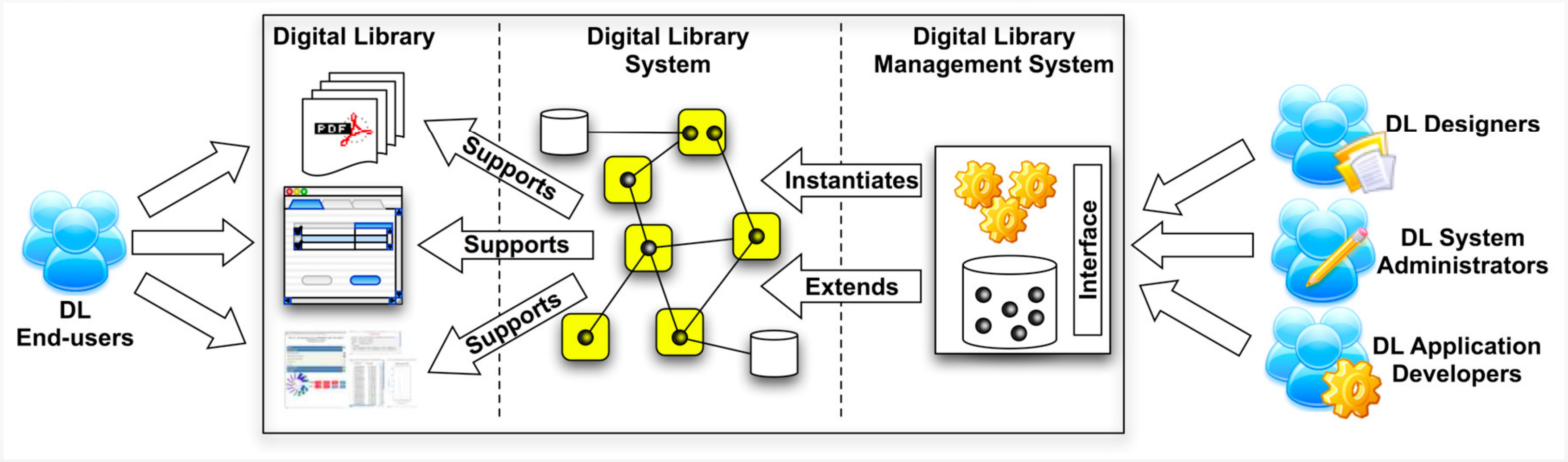
To lay down the building blocks, as of today

Functionality is the core



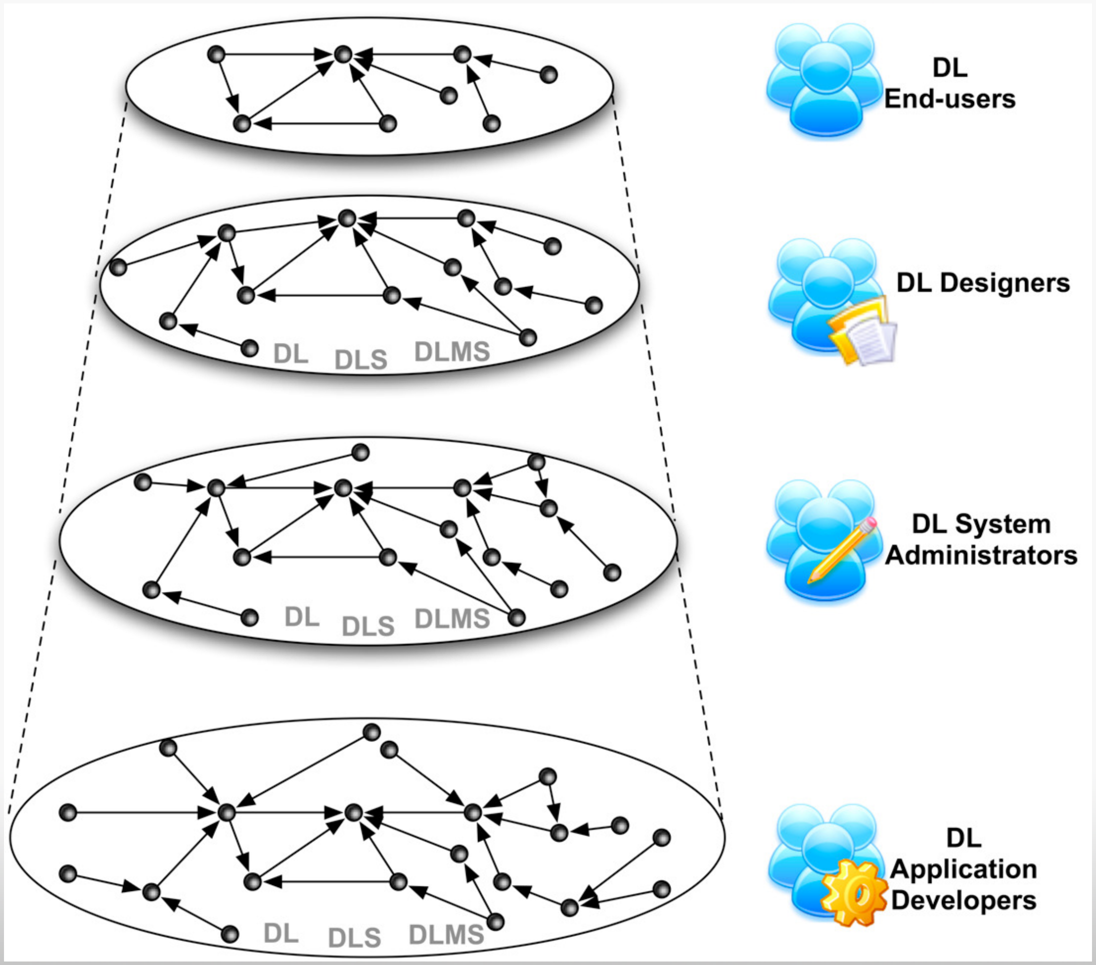
Functionality for ?

- Actors who use the DL functionality:



Digital Library Interoperability, Best Practices and Modelling Foundations

Relationships among users



The Functionality Domain

Captures all processing that can occur on *Resources* and activities that can be observed by *Actors* in a Digital Library

Function

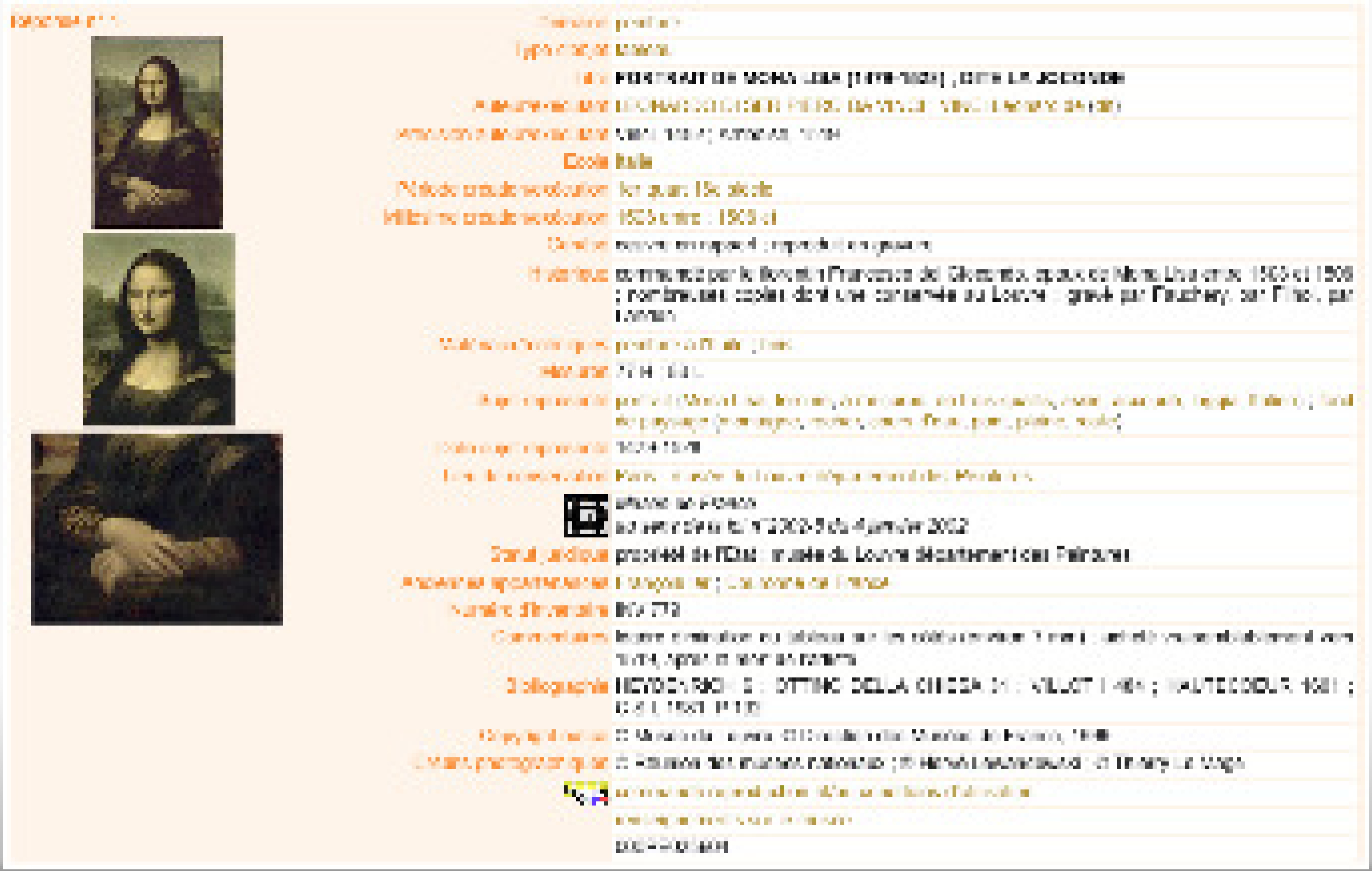
Wikipedia (Oct. 4, 2010, 11:12:56 CET)

- *In the abstract set-theoretic approach, a function is a relation between the domain and the codomain that associates each element in the domain with exactly one element in the codomain.*
- *An example of a function with domain {1,2,3} and codomain {2,3,4} associates 1 with 2, 2 with 3, and 3 with 4*.

Function specification

- A table of values is a common way to specify a function in statistics, physics, chemistry, and other sciences.
- And also in information systems
  - and therefore in DLs

A special kind of function



[illegible]	peinture
Type d'objet	tableau
Titre	PORTRAIT DE MONA LISA [illegible] , DITE LA JOCONDE
Auteur/exécutant	LEONARDO DI SER PIERO DA VINCI [illegible]
[illegible]	[illegible]
Ecole	Italie
Période création/exécution	1er quart 16e siècle
Millésime création/exécution	1503 entre ; 1506 et
[illegible]	[illegible]
Historique	commandé par le florentin Francesco del Giocondo, époux de Mona Lisa entre 1503 et 1506 ; nombreuses copies dont une conservée au Louvre ; gravé par Fauchery, par Filhol, par Lorichon
Matériaux/techniques	peinture à l'huile [illegible]
Mesures	77 H ; 53 L
[illegible]	[illegible]
[illegible]	[illegible]
Lieu de conservation	Paris ; musée du Louvre département des Peintures
	[illegible] 2002
Statut juridique	propriété de l'Etat ; musée du Louvre département des Peintures
[illegible]	[illegible]
Numéro d'inventaire	INV 779
[illegible]	[illegible]
Bibliographie	HEYDENREICH [illegible] ; OTTINO DELLA CHIESA [illegible] ; VILLOT [illegible] ; HAUTECOEUR [illegible] ; [illegible]
Copyright notice	[illegible]
Crédits photographiques	© Réunion des musées nationaux ; © Hervé Lewandowski ; © Thierry Le Mage
	[illegible]
	[illegible]
	[illegible]

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

In Computer Science

- A function is a (logical) machine that performs a specific task.
  - In a DL, there are a number of such machines, ready to be used by the (authorized) user
- In order to use a function, the function must be applied
  - i.e. the machine must be started
    - push a button on a GUI
    - enter a URL is a browser
    - type some text in a terminal window

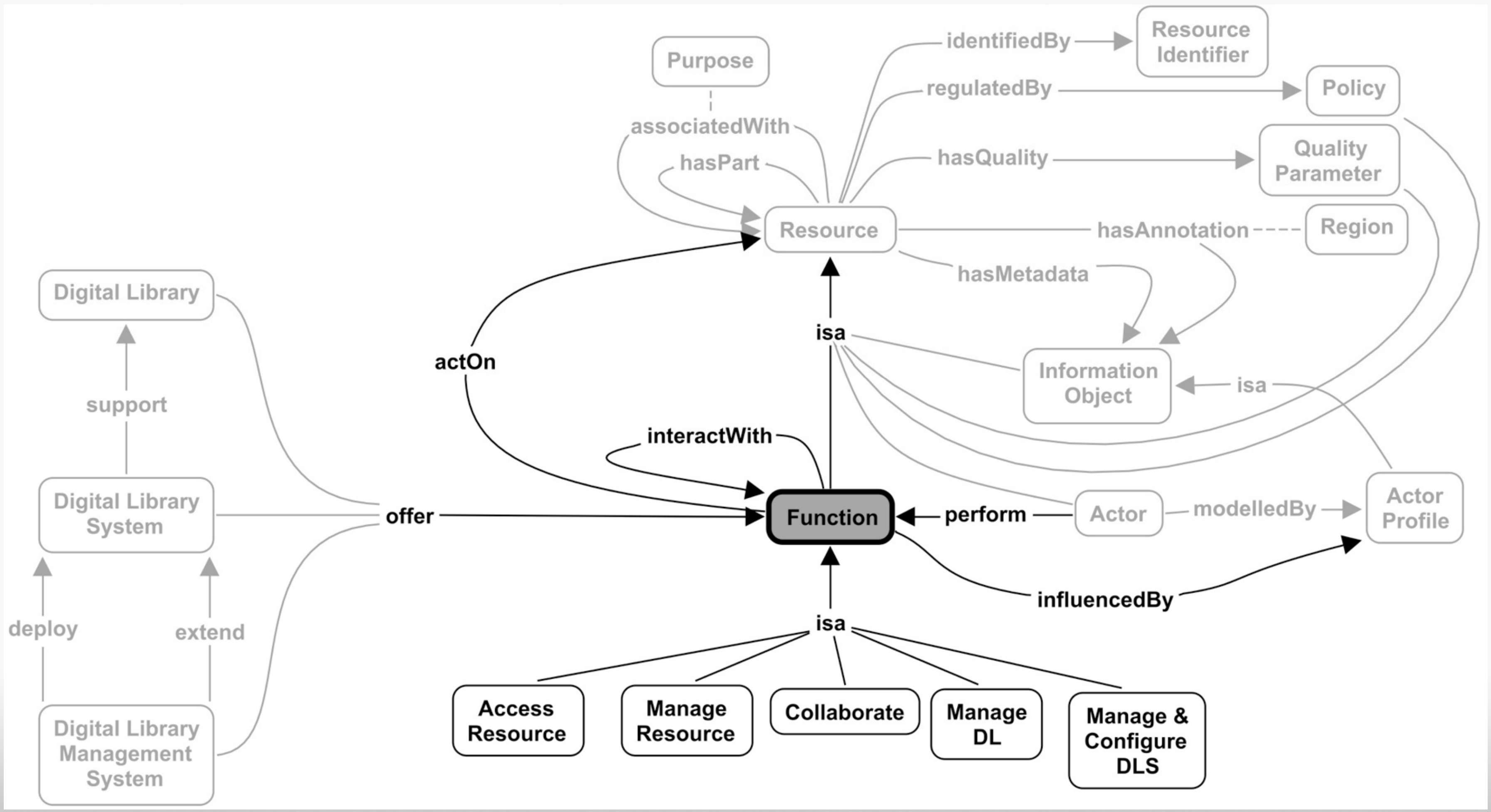
DL.org
Digital Library Interoperability, Best Practices and Modelling Foundations

In Computer Science

- When applied, a function becomes a process:
  - has input parameters
  - has output parameters
  - may change the DL
- Querying
- Inserting an object
- but there is a lot more …

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

The Map



A Function is-a Resource

- it has a unique identifier (Resource Identifier)
  - is it an information resource or a non-information resource?
- it has structure:
  - can be atomic
    - composed of no other functions
  - the composition of simpler functions, which results in an arbitrarily structured workflow:
    - <hasPart> A function has functions as its parts
    - <associatedWith> A function is associated with a function for a Purpose

A Function is-a Resource

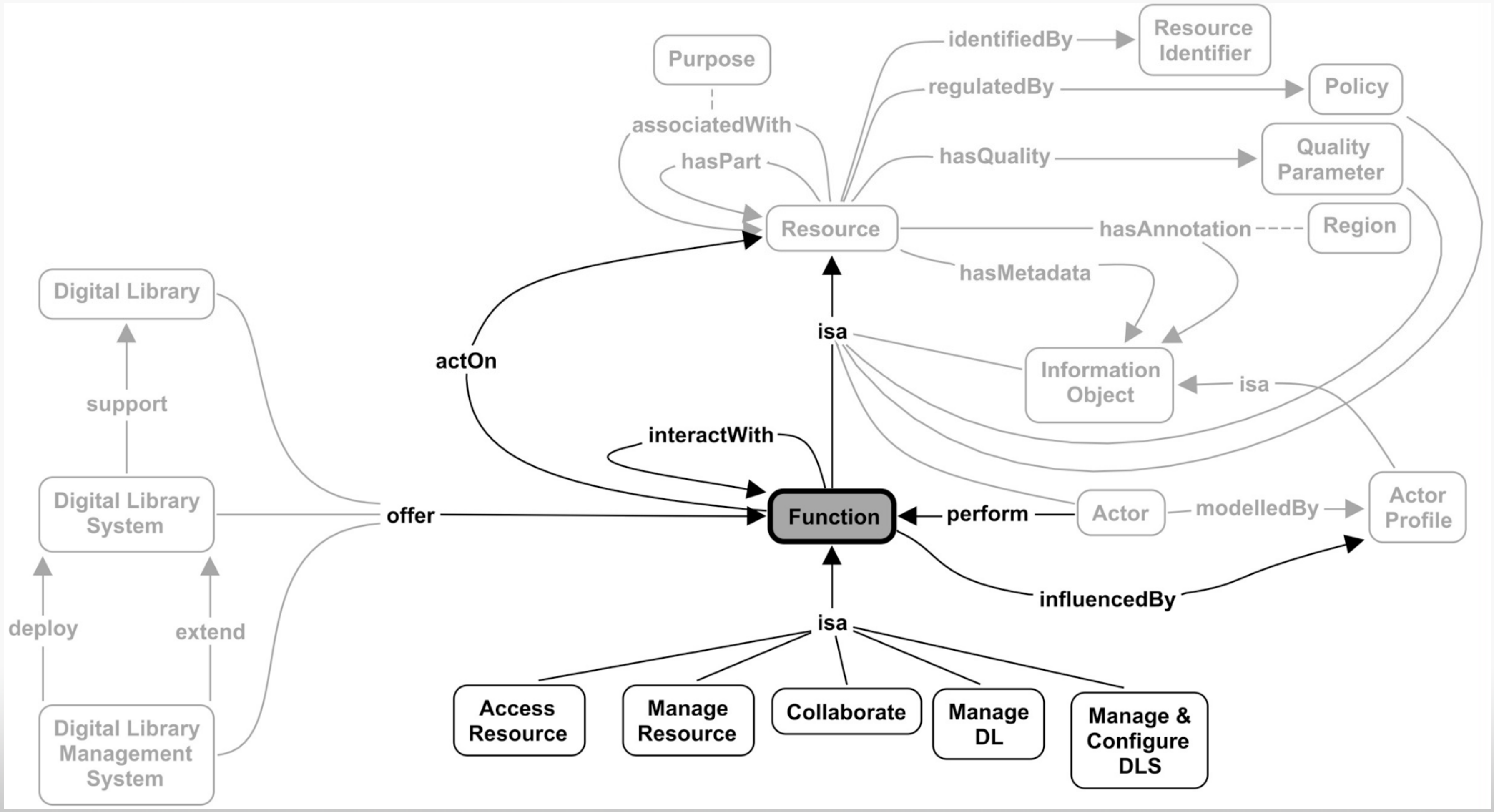
- it is characterised by various Quality Parameters covering various quality aspects (<hasQuality>)
  - synchronous vs. asynchronous
  - efficient
  - robust
  - state-full vs state-less
  - CPU-bound vs I/O-bound
- its lifetime and behaviour are regulated by Policies (<regulatedBy>)
  - which Actors are allowed to perform the Function in a certain context
  - which billing schema applies to the Function
- it can be enriched with Metadata (<hasMetadata>)
  - there are many languages for describing functions, from technically-oriented ones (such as WSDL) to more semantically-oriented ones (DAML-S)
  - Functions are searchable, like any other resource

A Function is-a Resource

- it can be enriched with Annotation (<hasAnnotation>)
  - pre-formalization
  - helping the interpretation
  - social activity

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

The Map



Digital Library Interoperability, Best Practices and Modelling Foundations

A Function is a Function

- A Function acts on Resources (<actOn>)
  - Resources means not only Information Objects but also Functions, Actor profiles, Policies, etc.
- A Function interacts with other Functions (<interactWith>)
  - Orders functions within a workflow

How is a Function born?

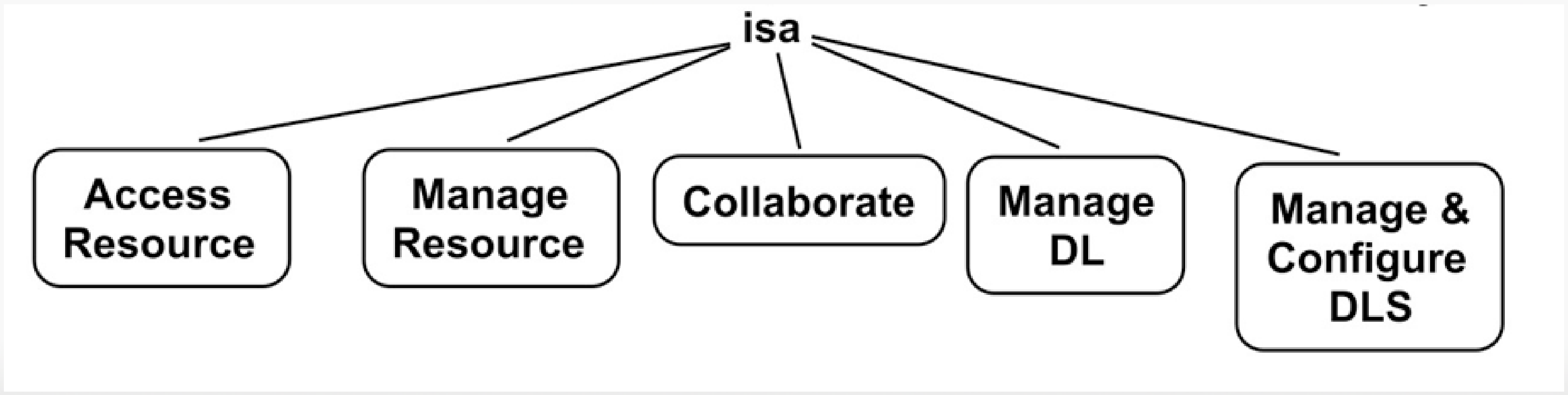
The result of a complex process/chain.

A simplified waterfall picture (largely inspired by the process in Europeana):

- Business sees a new opportunity and defines a need
- Potential users elaborate the need in form of a (system of) requirement(s)
- Conceptual modellers turn requirements into a functional specification
- Developers turn the functional specification into a technical specification
- Business performs cost/benefit analysis on the technical specification and (sometimes) signs it off for implementation
- Developers turn the technical specification into software
- Quality controllers test the software to check whether it meets quality parameters
- Users tests the software to check whether it meets the initial requirements
- System administrators deploy the software into the architecture, and subsequently make sure the software operates correctly as the context around the DL evolves.

What Functions in a DL?

- Each Digital Library may have its own set of *Functions* depending on its underlying business models.
- *Function* is specialised into five other concepts that still represent quite general classes of activities.



Access

Access Functions help in identifying and obtaining *Resources.*

Access Resource encompasses all machines related to

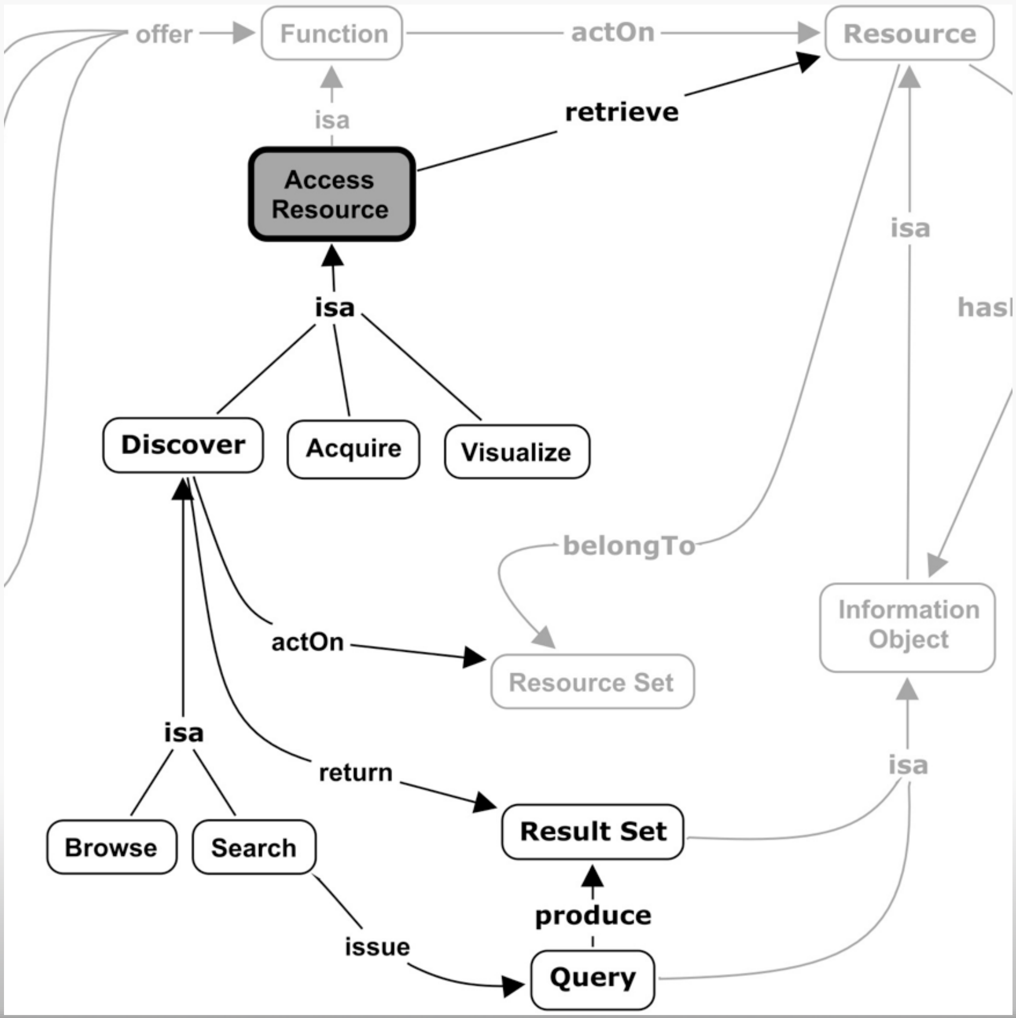
- requesting
- locating
- retrieving
- transforming
- representing in a ‘material form’

a *Resource*

Access Functions do not modify the DL

Access functions

- C32 Access Resource
  - C33 Discover
    - C34 Browse
    - C35 Search
  - C36 Acquire
  - C37 Visualise

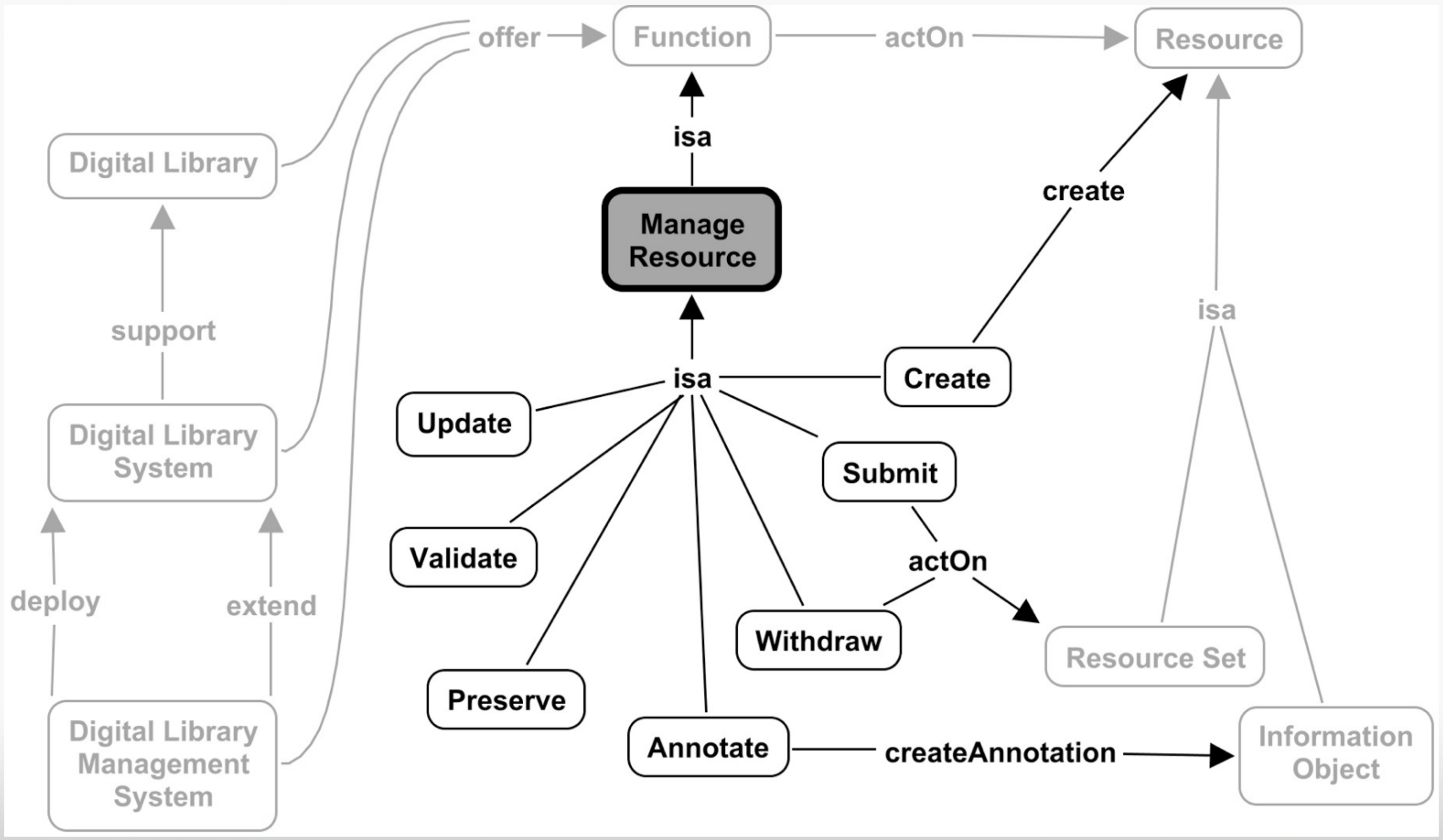


Manage Resource

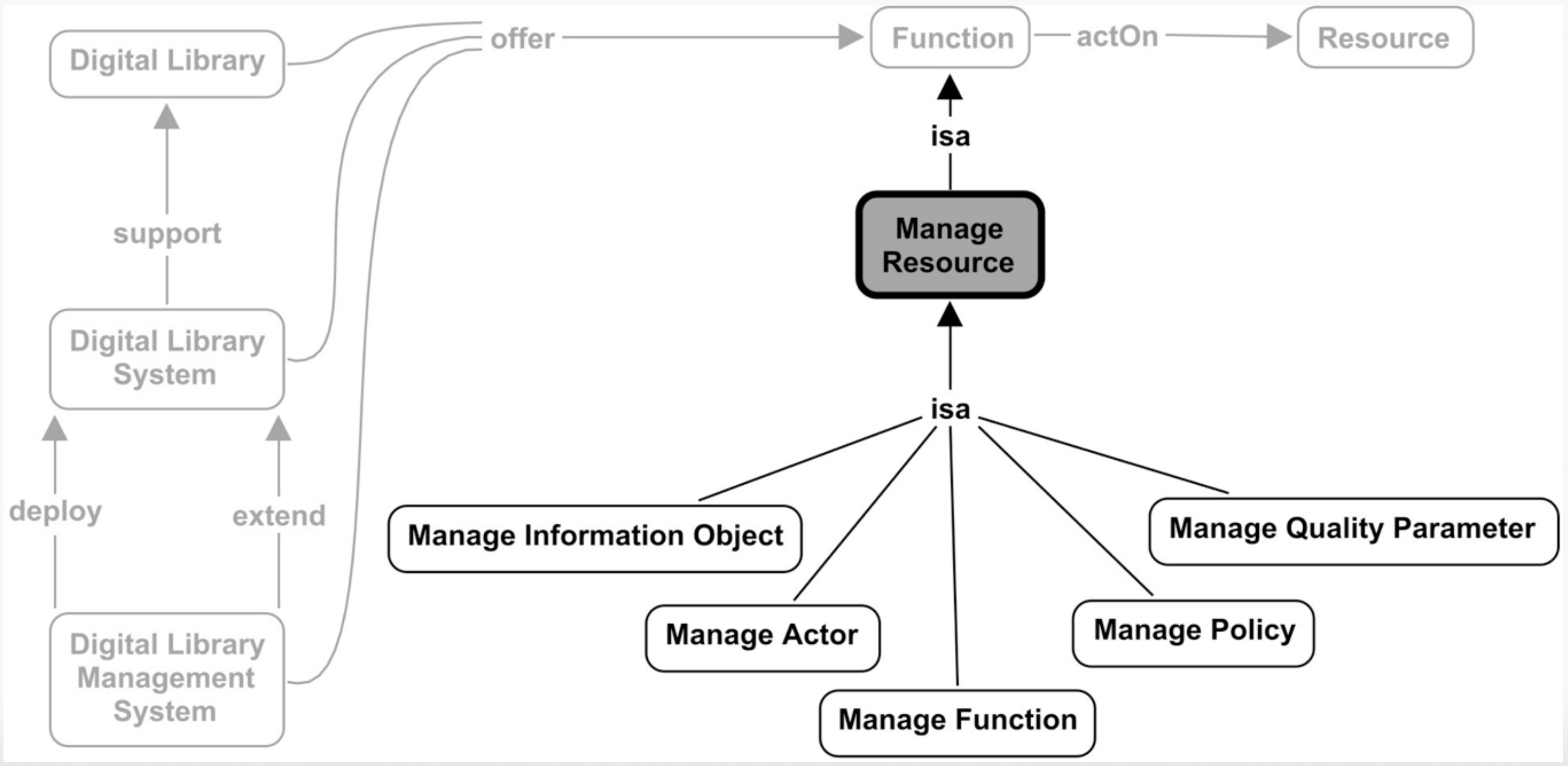
- includes all machines for
  - creating new Resources
  - inserting them into the DL
  - deleting old Resources
  - updating existing Resources,
  - converting or transformating existing Resources.

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

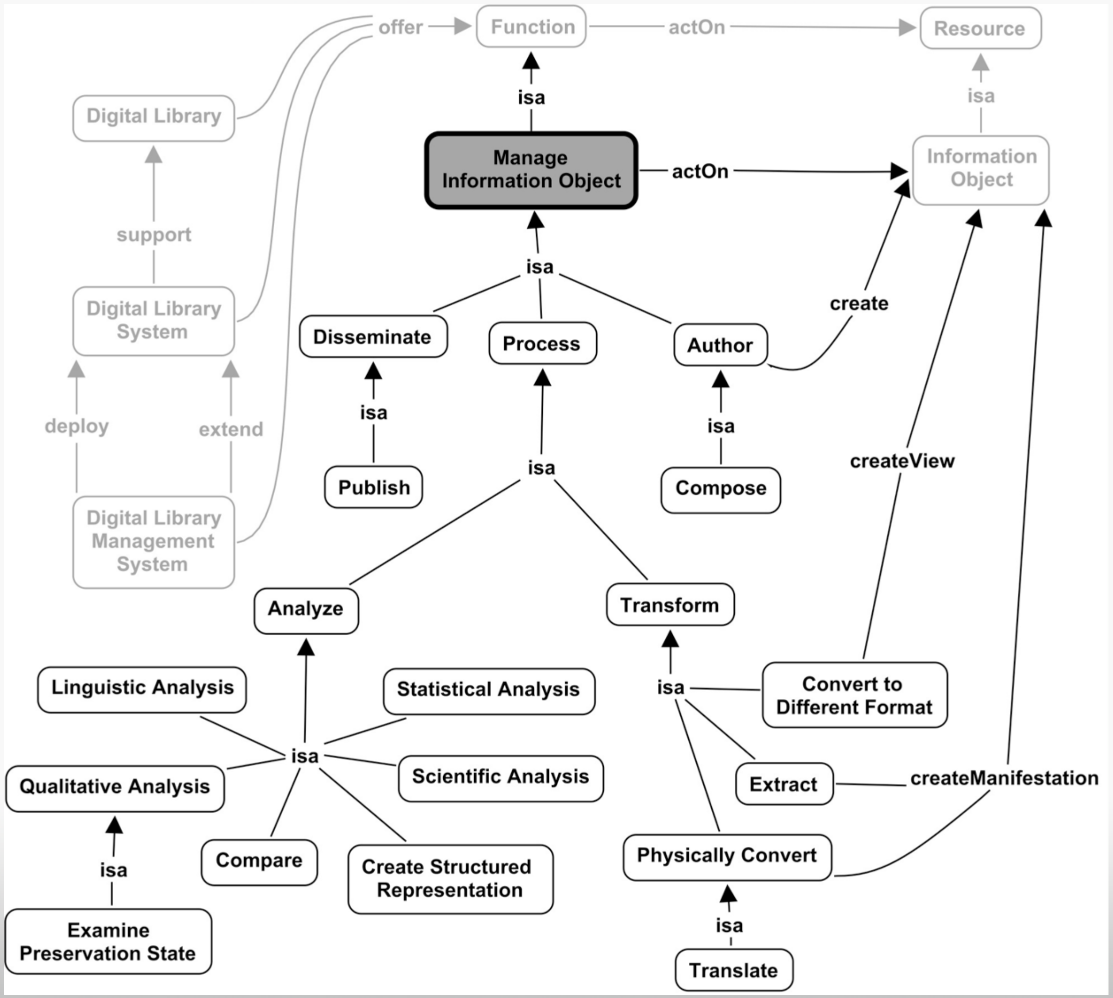
Manage Resource for all Resources



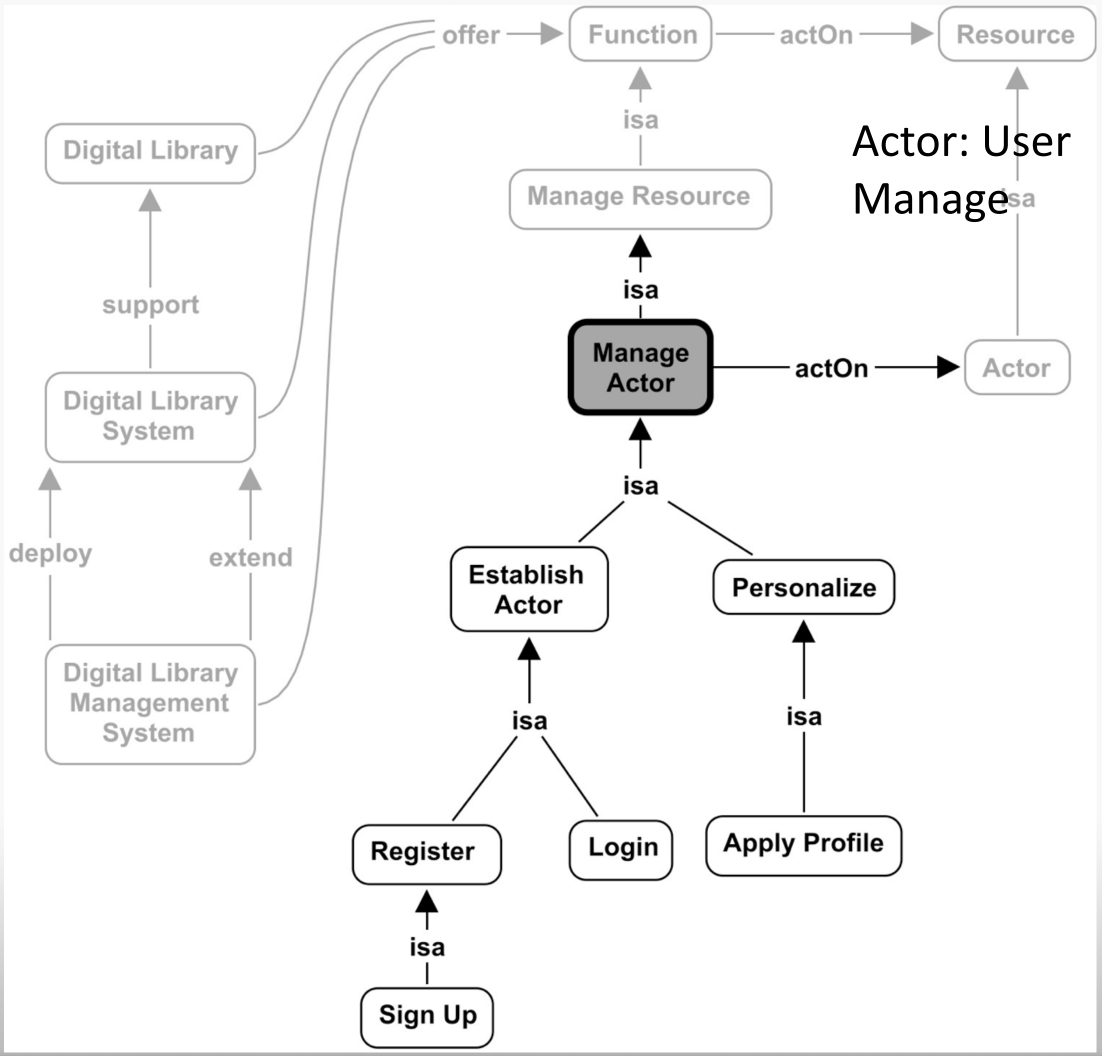
Manage Resource per Resource Type



Manage Information Object



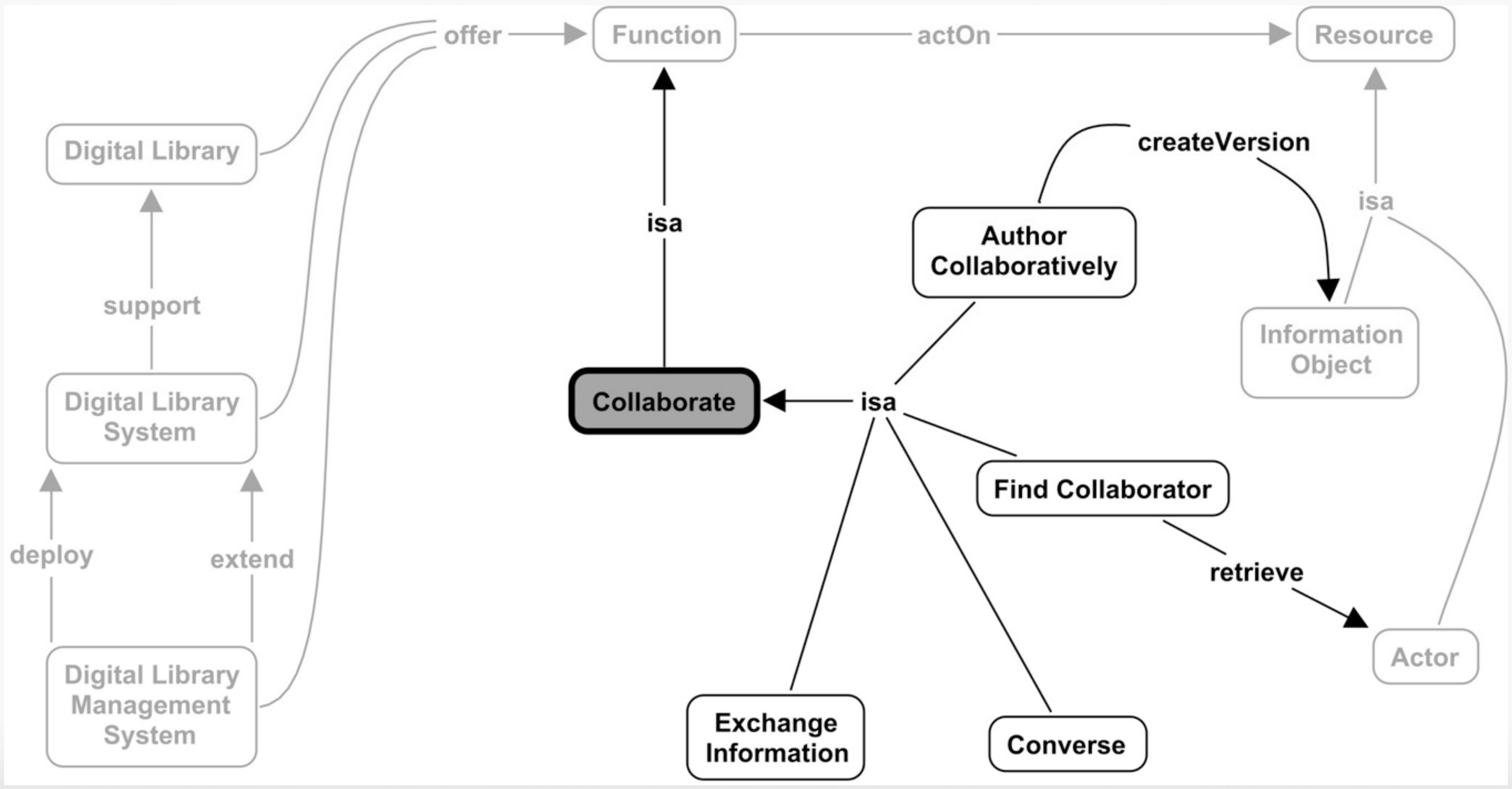
Manage Actor



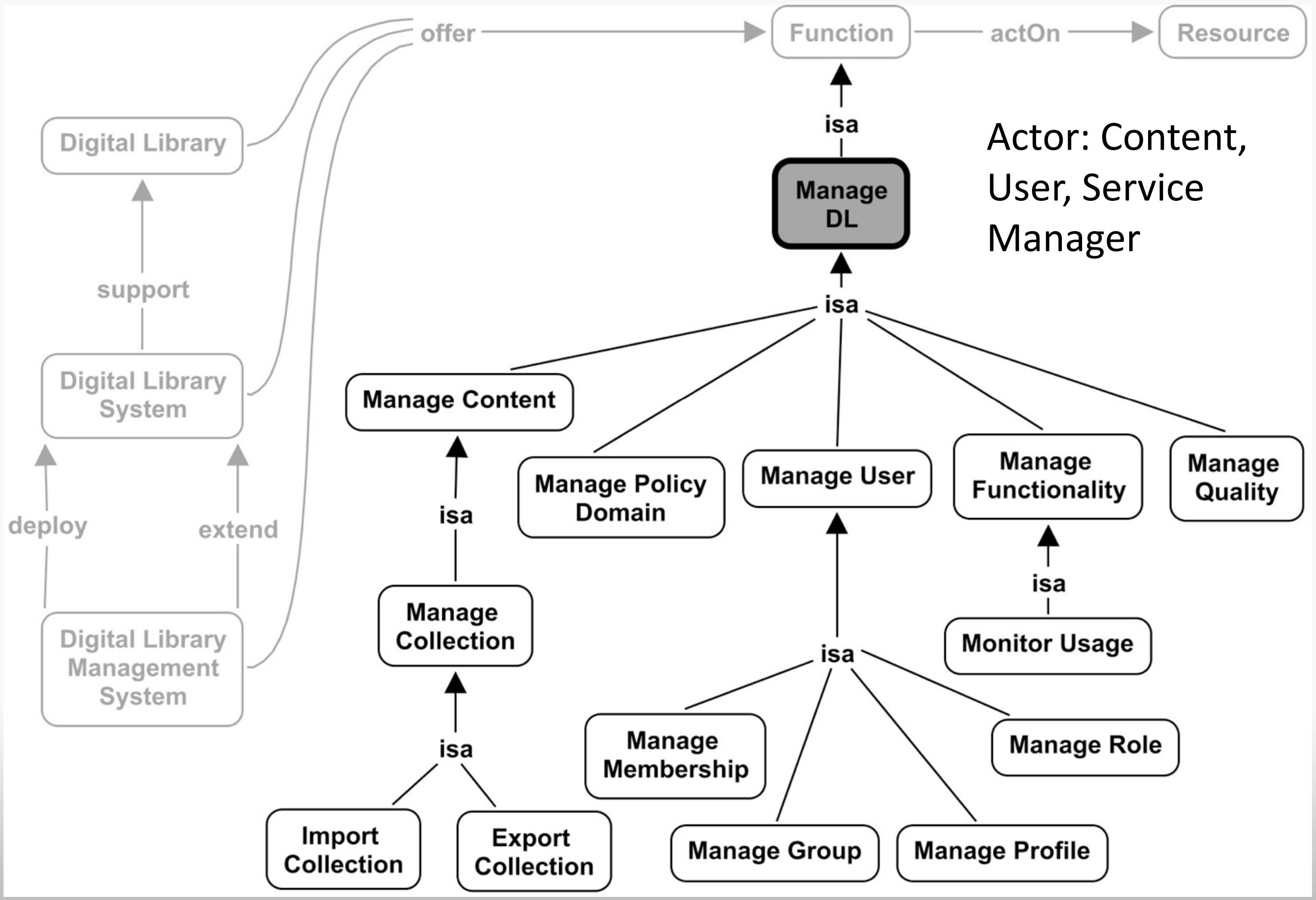
Actor: User
Manage

Digital Library Interoperability, Best Practices and Modelling Foundations

Collaborate

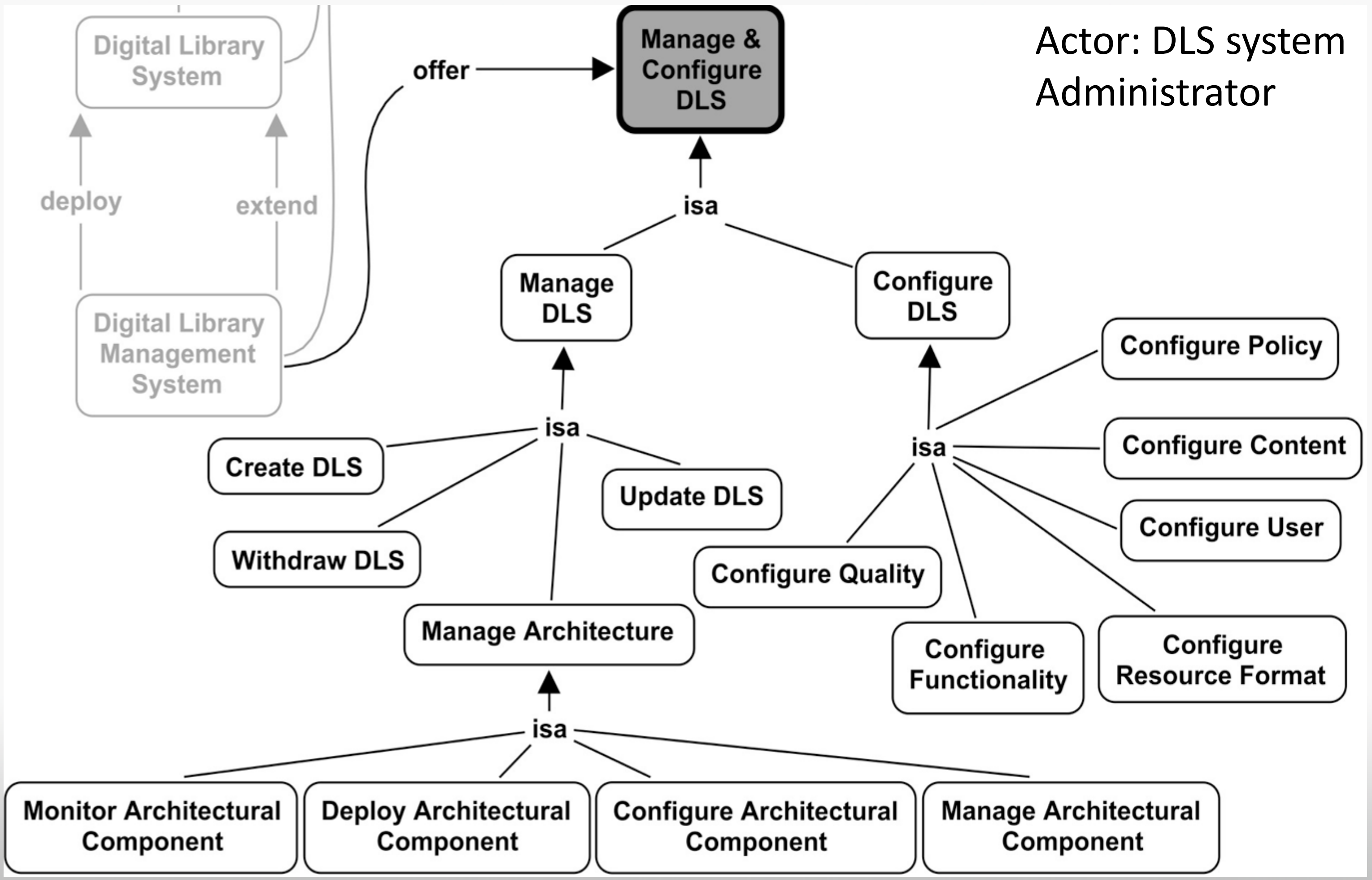


Manage DL



Actor: Content, User, Service Manager

Manage & Configure DLS



Actor: DLS system Administrator

Scenario: InfoSer

The agent of the scenario is a DL designer, D.

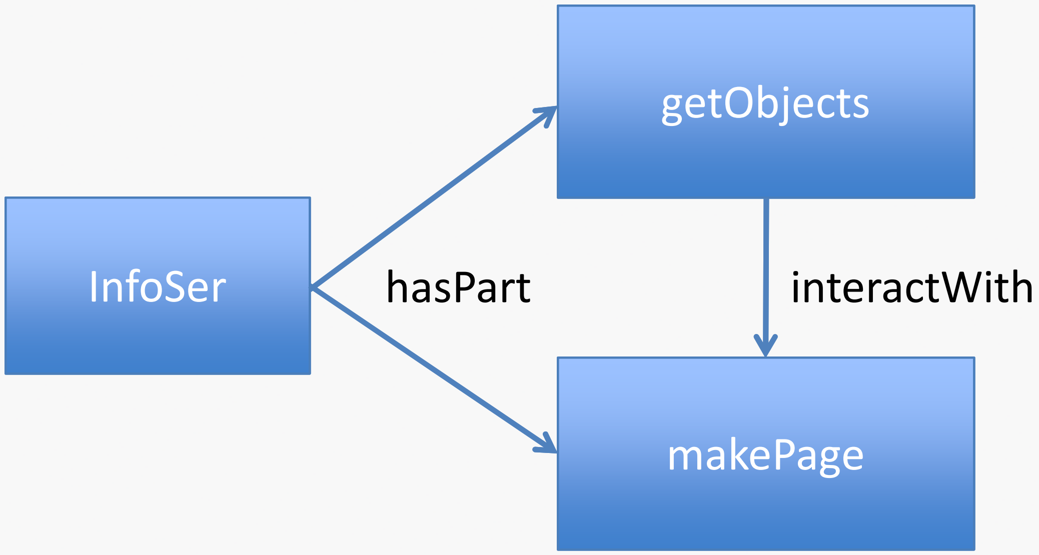
D has been given by the DL Director the task of designing an information service that offers to the user, for a given topic:

- a list of resources related to the topic in the DL
- a list of the items that can be downloaded from the DL (either freely or by paying some fee)

InfoSer

- How to implement InfoServ?
- InfoServ will be implemented as a complex function, built on top of 2 simpler functions:
  - a function for collecting the DL objects about the given topic: getObjects(topic)
    - and identifying the DL objects that are downloadable
  - a function for constructing the result and serve it to the user: makePage(result)

Functions



getObjects(topic)

- Discover objects by querying the associated (<hasMetadata>) InformationObject on the topic properties:
  - depends on the ResourceFormat associated (<hasFormat>) with the InformationObject
    - DC: dc:subject, dc:coverage
    - CIDOC CRM: P129 is about (is subject of)
- The query must specify the relevant properties:
  - properties to be displayed to the user
  - properties for understanding whether the object is downloadable
- Return the ResultSet

makePage

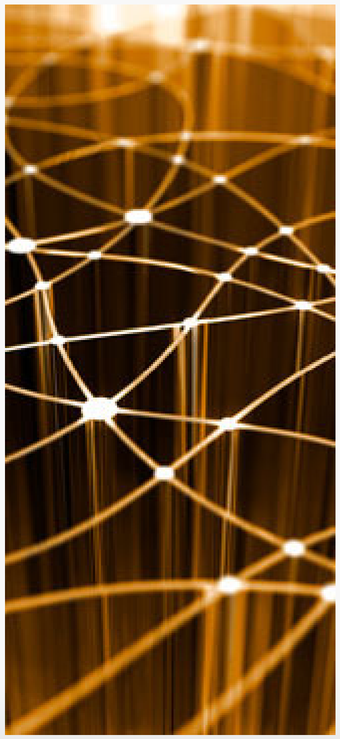
- Create a new resource in the DL
  - we want to persist the information for further, multiple re-use
  - the resource has a URI and has 3 manifestations:
    - the result as an HTML document
    - the result as a PDF document
    - the result as an RDF graph
- Display the resource to the user
- The user will Visualize the Resource

Extended scenario

- The exercise for Friday is based on a variation of the same scenario:
- The youngest child (8) of the DL director (45) has used InfoSer for her home work and found it … improvable.
- She then suggested her father to improve the InfoSer service by using existing resources on the Web:
  - Europeana for finding relevant resources
  - Amazon for offering purchasable items
- Additionally:
  - Wikipedia for giving an account of the topic
- The next day, D (i.e. you) gets a new task.

DL.org

Digital Library Interoperability, Best Practices and Modelling Foundations



Thank you

wiki.dlorg.eu/index.php/Reference_Model