Modelling the Web

Nicolas Spyratos[1] Carlo Meghini[2]

[1]UP Sud - LRI, Paris
[2]CNR - ISTI, Pisa

Outline

Outline

Outline

Outline

Outline

Outline

Motivations

Motivations:

- To facilitate operations in Digital Libraries (DLs), especially the discovery and re-use of objects.
- To create a yardstick, against which to “measure” DLs.
- To highlight the mathematical structure underlying a DL.

In a way that is:

- *As simple as possible, but not simpler.*
- Compliant with the Web (the largest DL ever).

Goal

We need a level of abstraction over the overwhelming amount of details involved in the management of a DL, *i.e.*, a *data model.*

Operations provided by the model:

- *describe* an object of interest according to the vocabulary of the community;
- *discover* objects of interest based on content and/or description;
- *view* the content of a discovered object;
- *identify* an object of interest, in the sense of assigning to it an identity;
- *re-use* objects in a different context.

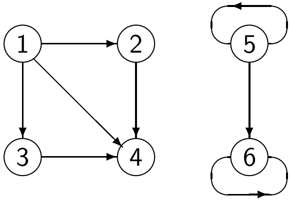
We want to define these operations and give algorithms for their implementation.

Mathematical preliminaries

We use one modelling tool: set-valued functions, which sometimes we view as graphs or binary relations.

A={1,2,3,4,5,6,7,8}

a	$f(a)$
1	{2,3,4}
2	{4}
3	{4}
4	{}
5	{5,6}
6	{6}
7	und.
8	und.



A : any non-empty set

$\mathcal{P}(A)$: the powerset of A

A *set-valued function* f on A is a partial function assigning to each element a in its domain of definition, a possibly empty subset of A :

$$f: \ A \longrightarrow \mathcal{P}(A)$$

$f(a)$: the *image* of a under f

$def(f)$: the domain of definition of f

$range(f) = \bigcup \{f(a) \mid a \in def(f)\}$

f partitions A into two subsets:

- the *active* objects, $\mathrm{act}(f)$, the objects that appear in f (either in the domain or the range of f):

$$\mathrm{act}(f) = \mathit{def}(f) \cup \mathit{range}(f)$$

- the *inactive* objects, $\mathrm{inact}(f)$, the objects that do not appear in f

$$\mathrm{inact}(f) = A \setminus \mathrm{act}(f)$$

a	$f(a)$
1	{2,3,4}
2	{4}
3	{4}
4	{}
5	{5,6}
6	{6}

$A = \{1, 2, 3, 4, 5, 6, 7, 8\}$
$\mathit{def}(f) = \{1, 2, 3, 4, 5, 6\}$
$\mathit{range}(f) = \{2, 3, 4, 5, 6\}$
$\mathrm{act}(f) = \{1, 2, 3, 4, 5, 6\}$
$\mathrm{inact}(f) = \{7, 8\}$

An active object a is:

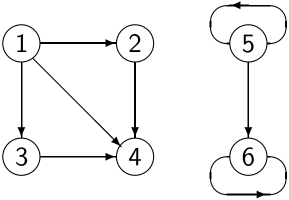
- *initial* if it is not in the image of any other object:

$$a \in def(f) \text{ and } [\ (\forall x \in def(f))\ a \in f(x) \rightarrow x = a\ ]$$

- *terminal* if either it is not an identifier, or it is an identifier and belongs to its own image:

$$a \in range(f) \text{ and } [\ a \in def(f) \rightarrow a \in f(a)\ ]$$

- *intermediate* if it is neither initial nor final.



initial: $\{1, 5\}$

terminal: $\{4, 6\}$

intermediate: $\{2, 3\}$

Digital Objects

A DL includes a set of digital objects.

A DL is very different from a traditional information system, which contains *representations.*

Intuitively, we think of a digital object as a piece of information in digital form such as a PDF document, a JPEG image, a URI and so on.

As such, a digital object can be processed by a computer, for instance it can be stored in memory and displayed on a screen.

$\mathcal{O}$: a collection of digital objects.

We assume $\mathcal{O}$ to be non-empty and countable.

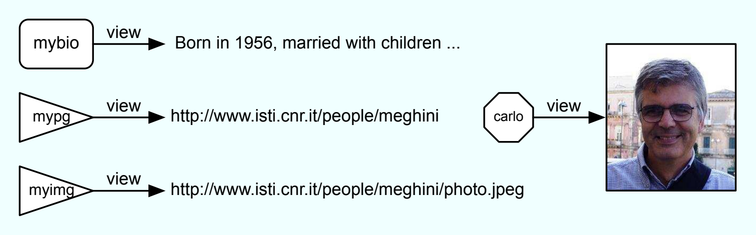
Objects in $\mathcal{O}$ have a *view,* a *content* and a *description.*

View

We assume that each digital object can be *viewed* using an appropriate mechanism.

view(o) : the view of o

view is a total function having the set O as domain. The range of view is outside the scope of our model.



Content

We define *content* over O to be a set-valued function cont on O :

$$\text{cont} : \mathrm{O} \rightarrow \mathcal{P}(\mathrm{O})$$

such that for each object $o \in \mathit{def}(\text{cont})$, $\text{cont}(o)$ is a *finite, possibly empty* set of objects.

$\text{cont}(o)$: the *content* of o

$\mathit{def}(\text{cont})$: the *identifiers*

document: a rendering of some content on a specific device

- we do not exclude the case in which $o \in \text{cont}(o)$
- content is dynamic (in time and space).

Special objects

Given a content function:

- the inactive objects are those not used currently, but available. They may enter the content function either as identifiers or as elements of content at any later point in time.

- the initial objects: identifiers of *collections*.

    - A special category: objects with empty content

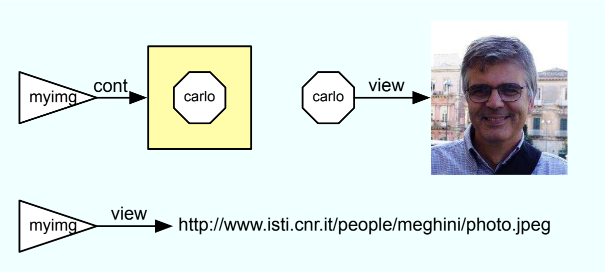
- the terminal objects: "pure" content objects, contributing to the content by their view.

An image identified by a URI

myimg: a digital object (a URI)
view(myimg)=`http://www.isti.cnr.it/people/meghini/photo.jpeg`

carlo: a digital object (an image)
view(carlo)=*a photograph*

cont(myimg)={carlo}



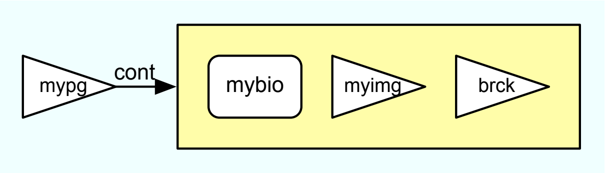
An Web page

mypg: a digital object (a URI)
view(mypg)=`http://www.isti.cnr.it/people/meghini/index.html`

mybio: a digital object (a text)
view(mybio)="Born 1956, married with children, ..."

brck: a digital object (a URI)
view(brck)=`http://www.bricksfactory.org`

cont(mypg)={mybio,myimg,brck}



Versions

The user is working on a text, of which he wants to maintain versions:

- folder o
  - file o_1
    - text t_1 (view(t_1) : the initial text)
  - file o_2
    - text t_2 (view(t_2) : the modified text)

We view o as the identifier of our text and o_1 and o_2 as two versions of it.

Which version represents o at any point in time? any of the two, depending on context.

The versions of o are alternatives for o, not necessarily its evolution in time.

The *versions* over O :

$$\text{vers} : \text{O} \rightarrow \mathcal{P}(\text{O})$$

such that for each object $o \in \mathit{def}(\text{vers})$, $\text{vers}(o)$ is a finite, possibly empty set of objects not containing o.

$\text{vers}(o)$: the *versions* of o.

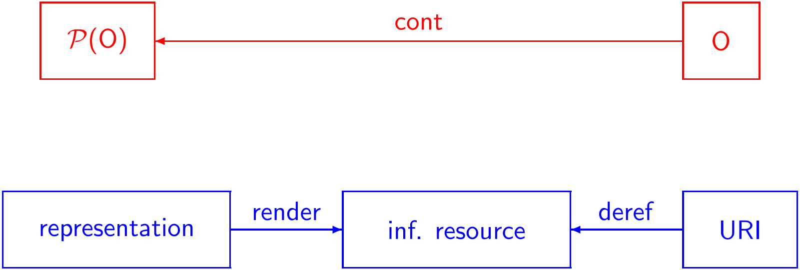
Relationship with the Web architecture

The web architecture is based on three fundamental notions: *resource, representation* and *identifier.*

- A resource "can be anything that has identity".
  - An *information resource* is a resource all of whose "essential characteristics can be conveyed in a message".
- A representation is "data that encodes information about resource state".
- An identifier is "an object that can act as a reference to something that has identity". The Web uses a single global identification system: the Uniform Resource Identifiers (URI).

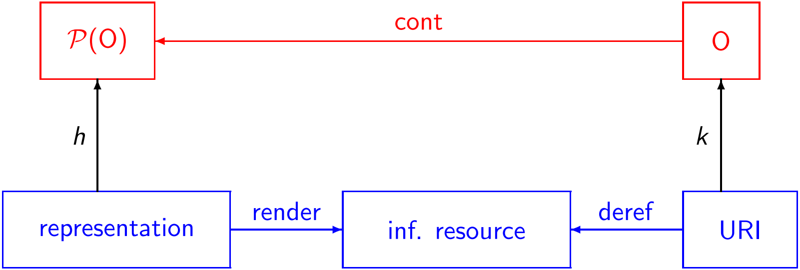
A resource is obtained by *de-referencing* its URI, which for HTTP URIs implies *rendering* one of its representations.

$$\boxed{\mathcal{P}(O)} \xleftarrow{\text{cont}} \boxed{O}$$

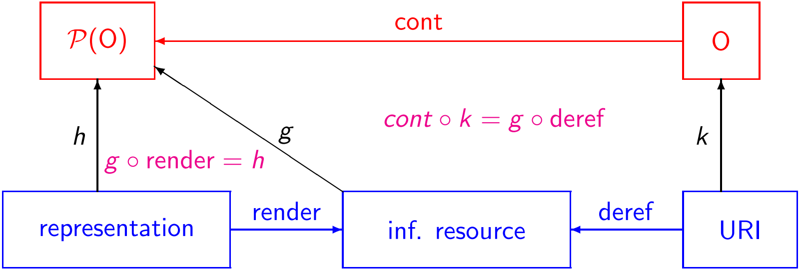

$\mathcal{P}(\mathrm{O})$
cont
O
representation
render
inf. resource
deref
URI

$$
\begin{tikzcd}
\mathcal{P}(O) & & O \arrow[ll, "\text{cont}"'] \\
\text{representation} \arrow[u, "h"] \arrow[r, "\text{render}"] & \text{inf. resource} & \text{URI} \arrow[l, "\text{deref}"']
\end{tikzcd}
$$

- h associates each representation to the set of objects it contains



- h associates each representation to the set of objects it contains
- k associates each URI to an identifier, 1:1



- h associates each representation to the set of objects it contains
- k associates each URI to an identifier, 1:1

Given h and k, there is a unique g which satisfies the constraints.

Descriptions support the interpretation, the discovery, and the management of content.

Descriptions are statements about the DL objects and related entities.

A description: a set of (subject label object) triples.

Notice: any object in O can be used in a triple.

The descriptions in a DL are a finite set of triples $\mathsf{T} \subseteq \mathsf{O} \times \mathsf{O} \times \mathsf{O}$

A *description forming* function over O :

$$\mathsf{dform} : \mathsf{O} \rightarrow \mathcal{P}(\mathsf{T})$$

such that for each object $o \in \mathit{def}(\mathsf{dform})$, $\mathsf{dform}(o)$, is a finite, non-empty set of triples.

def(dform) : the *description identifiers.*

Intermediate objects allow to make statements about descriptions, *i.e.*, metadata about metadata.

In RDF, triple reification is defined to obtain the same affect.

$(o, \mathrm{dform}(o))$: a named graph.

Next, we link objects and their descriptions.

description over O :

$$\mathrm{desc} : \mathrm{O} \longrightarrow \mathcal{P}(\mathrm{O})$$

such that for each object $o \in \mathit{def}(\mathrm{desc})$, $\mathrm{desc}(o)$, is a finite, possibly empty set of description identifiers, *i.e.*, we require

$$\mathit{range}(\mathrm{desc}) \subseteq \mathit{def}(\mathrm{dform}).$$

$\mathrm{desc}(o)$: the *descriptions* of o.

Conclusions and future work

We have the initial elements of a DL model, compliant with the web architecture (as well as with OAI-ORE).

Next steps:

- To move towards RDF Schema?
- query language
- data manipulation language
- implementation

Thank you!

Any question?