Modelling the Web

Nicolas Spyratos\textsuperscript{1} \hspace{1cm} Carlo Meghini\textsuperscript{2}

\textsuperscript{1}UP Sud - LRI, Paris
\textsuperscript{2}CNR - ISTI, Pisa
Outline

1. Motivation and goal
2. Mathematical preliminaries
3. Digital Objects
   - View
   - Content
   - Versions
4. Relationship with the Web architecture
5. Descriptions
6. Conclusions
Outline

1. Motivation and goal
2. Mathematical preliminaries
3. Digital Objects
   - View
   - Content
   - Versions
4. Relationship with the Web architecture
5. Descriptions
6. Conclusions
Outline

1 Motivation and goal
2 Mathematical preliminaries
3 Digital Objects
   - View
   - Content
   - Versions
4 Relationship with the Web architecture
5 Descriptions
6 Conclusions
Outline

1. Motivation and goal

2. Mathematical preliminaries

3. Digital Objects
   - View
   - Content
   - Versions

4. Relationship with the Web architecture

5. Descriptions

6. Conclusions
Outline

1. Motivation and goal
2. Mathematical preliminaries
3. Digital Objects
   - View
   - Content
   - Versions
4. Relationship with the Web architecture
5. Descriptions
6. Conclusions
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).
Modelling the Web

Motivation and goal

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

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

<table>
<thead>
<tr>
<th>( a )</th>
<th>( f(a) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>{2,3,4}</td>
</tr>
<tr>
<td>2</td>
<td>{4}</td>
</tr>
<tr>
<td>3</td>
<td>{4}</td>
</tr>
<tr>
<td>4</td>
<td>{}</td>
</tr>
<tr>
<td>5</td>
<td>{5,6}</td>
</tr>
<tr>
<td>6</td>
<td>{6}</td>
</tr>
<tr>
<td>7</td>
<td>und.</td>
</tr>
<tr>
<td>8</td>
<td>und.</td>
</tr>
</tbody>
</table>
$A$ : 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 $A$:

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

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

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

range$(f) = \bigcup \{ f(a) \mid a \in \text{def}(f) \}$
\( f \) partitions \( A \) into two subsets:

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

\[
\text{act}(f) = \text{def}(f) \cup \text{range}(f)
\]

- the \textit{inactive} objects, \( \text{inact}(f) \), the objects that do not appear in \( f \)

\[
\text{inact}(f) = A \setminus \text{act}(f)
\]

\begin{tabular}{|c|c|}
  \hline
  \( a \) & \( f(a) \) \\
  \hline
  1 & \{2,3,4\} \\
  2 & \{4\} \\
  3 & \{4\} \\
  4 & \{} \\
  5 & \{5,6\} \\
  6 & \{6\} \\
  \hline
\end{tabular}

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

\( \text{def}(f) = \{1, 2, 3, 4, 5, 6\} \)

\( \text{range}(f) = \{2, 3, 4, 5, 6\} \)

\( \text{act}(f) = \{1, 2, 3, 4, 5, 6\} \)

\( \text{inact}(f) = \{7, 8\} \)
An active object $a$ is:

- **initial** if it is not in the image of any other object:
  \[ a \in \text{def}(f) \text{ and } (\forall x \in \text{def}(f)) \ a \in f(x) \rightarrow x = a \]

- **terminal** if either it is not an identifier, or it is an identifier and belongs to its own image:
  \[ a \in \text{range}(f) \text{ and } a \in \text{def}(f) \rightarrow a \in f(a) \]

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

![Diagram of active objects]

- 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.

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

We assume $O$ to be non-empty and countable.

Objects in $O$ have a view, a content and a description.
We assume that each digital object can be \textit{viewed} using an appropriate mechanism.

\texttt{view}(o) : the view of \texttt{o}

\texttt{view} is a total function having the set \texttt{O} as domain. The range of \texttt{view} is outside the scope of our model.
We define \textit{content} over \( O \) to be a set-valued function \( \text{cont} \) on \( O \) :

\[
\text{cont} : O \rightarrow \mathcal{P}(O)
\]

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

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

\( \text{def}(\text{cont}) \) : the \textit{identifiers}

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

- folder $o$
  - file $o_1$
    - text $t_1$ ($\text{view}(t_1)$: the initial text)
  - file $o_2$
    - text $t_2$ ($\text{view}(t_2)$: 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 $O$:

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

such that for each object $o \in \text{def}(\text{vers})$, $\text{vers}(o)$ 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.
$P(O)$ cont $O$
Modelling the Web

Relationship with the Web architecture

\mathcal{P}(O) \xrightarrow{\text{cont}} O

representation \xrightarrow{\text{render}} \text{inf. resource} \xleftarrow{\text{deref}} \text{URI}
$h$ associates each representation to the set of objects it contains
\( \mathcal{P}(O) \) associates each representation to the set of objects it contains.

\( k \) associates each URI to an identifier, 1:1.
$\mathcal{P}(O)$

- $h$ 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.

$g \circ \text{render} = h$

$cont \circ k = g \circ \text{deref}$
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 $T \subseteq O \times O \times O$

A description forming function over $O$:

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

such that for each object $o \in \text{def}(dform)$, $dform(o)$, 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, dform(o)) : a named graph.

Next, we link objects and their descriptions.
description over $O$:

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

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

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

desc$(o)$: the \textit{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?