

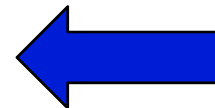


Calcolatori, Internet e il Web



Refresher on Computer Fundamentals and Networking

- History of computers
- Architecture of a computer
- Data representation within a computer
- Computer networks and the Internet
- The World Wide web
- Introduction to XML





Evolution of computer market (1940-2000)



- Military applications in early 40s
- Scientific/research applications in late 40s
- Commercial applications appear in early 50s
- Monopoly of IBM starts with 650, 701, 702
- Monopoly of IBM continues with 7070, 7090 and the 360 series, starting the “mainframe era” (in the 60s and 70s)
- Arrival of the “minicomputers” in the 70s
- Arrival of the PC in the 80s
- Arrival of the Internet in the 90s
- Arrival of the Web in the 90s



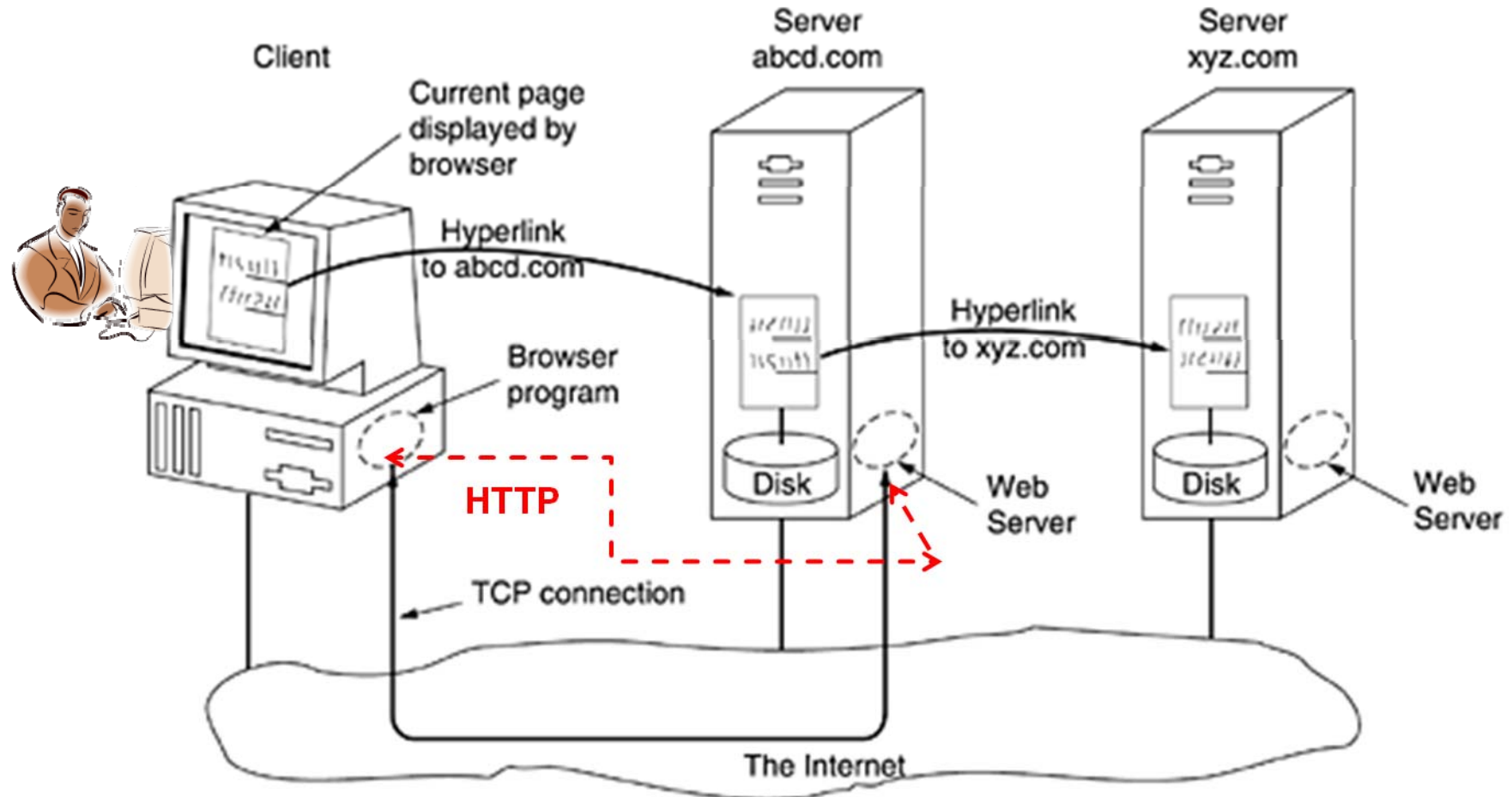
The World Wide Web



- Combination of computer technology and communication technology
- It all started with the “hyperlink”
- Then came the “browser” (Mosaic)
- Then came the first wave
- Then came the “dot come, dot gone”
- Then came the second wave
- Finally came the “information explosion”
 - An estimate of 500 to 1000 million hosts
 - An estimate of 30 to 50 billion pages on line
- And now we are in Web 2.0 (with Web 3.0 already happening)



The Web architecture





The editors



- Text processing applications started already in the early days of the computers (sixties)
- A “text processor” (or editor) has two main functions:
 - processing the text (delete, replace, insert, etc.)
 - specifying the format (bold, center, new line, etc.)
- The first editors were using a “mark up” language (i.e. commands intermixed with the text) to provide formatting instructions (only limited interactivity available through typewriter-like terminals)
- The “second generation” editors were using the WYSIWYG paradigm: What You See Is What You Get (much better interactivity available with display and mouse)



The hyperlink



- The idea of the “hyperlink” was (experimentally) proposed in the sixties, as a feature of a “smart editor”
 - selecting a portion of the text, it was possible to open a second document, in addition to the one being edited (very awkward to use on a typewriter-like terminal)
- With the arrival of display screens and the mouse (eighties) the hyperlink came back in “3D documents”
 - clicking on a portion of the text it was possible to open a second document, which was maintained as a second (virtual) screen behind the first one
- With the arrival of the (fast) internet, it became the “web hyperlink”
 - clicking on a portion of the text it was possible to open a second document, coming from a different computer



The browser



- With the arrival of the (web) hyperlink, the problem was then how to properly display a (web) page that had been generated on a different computer, possibly with a different (wysiwyg) editor
- The solution was the definition of HTML (Hyper Text Markup Language), i.e. a standard mark up language, and the implementation of smart editors (the browser) capable of correctly displaying pages formatted with HTML, regardless of where they were coming from



What is XML?



- The acronym means **eXtensible Markup Language**
- It is used to describe “data” in a way which is simple, structured and (usually) readable also by humans
- Developed at the end of the nineties by W3C (World Wide Web Consortium) as a simplification of the language SGML (Standard Generalized Markup Language), from which also HTML was derived
- Initial objective was to define a standard language for exchanging information on the Web



Simple example



- Define the structure of a (simple) telephone book:
 - *Name; Surname; Telephon.*
- Example of a text file (phonebook.txt):
Mario; Rossi; 031221222
Francesco; Neri; 123876453
Michele; Bianchi; 022121222
- Possible alternative
Mario;Rossi;031221222■Francesco;Neri;123876453■
Michele;Bianchi;022121222■
- Use of “delimiters” to separate the elements of the structure (e.g. “;” and “■”)



Simple example in XML



```
<phonebook>
  <entry>
    <firstname>Mario</firstname>
    <lastname>Rossi</lastname>
    <phone>031221222</phone>
  </entry>
  <entry>
    <firstname>Francesco</firstname>
    <lastname>Neri</lastname>
    <phone>123876453</phone>
  </entry>
  <entry>
    <firstname>Michele</firstname>
    <lastname>Bianchi</lastname>
    <phone>022121222</phone>
  </entry>
</ phonebook >
```

- The delimiters have a name (elements or tags)
- A hierarchical structure is becoming apparent, by means of “opening tags” and “closing tags” nested one within the other



XML is not HTML



- The main purpose of HTML is to describe how to render (visualize) the data
- HTML tags are pre-defined (by W3C)
- The main purpose of XML is to describe how the data is structured and what is the contents of the components of the structure
- XML tags can be freely defined by the user, but....
 - they do not give any “formal” indication about the meaning (the semantics) of the data delimited by the tags
 - they do not give any indication on how the data can be represented, processed, utilized
- XML is only a way to create “languages” with a standard syntax to describe the structure of data



Structure of a XML document



```
<?xml version="1.0" encoding="UTF-8"?>
```

Prolog

```
<personnel>  
  <employee>  
    <First>Fred</First>  
    <Last>Landis</Last>  
    <Title>Project Manager</Title>  
    <Phone>123-456-7890</Phone>  
    <Email>f.landis@nanonull.com</Email>  
  </employee>  
  <!--This is a comment-->  
</personnel>
```

Root



The XML elements



- There must be only one root element
- Tag elements must be closed
- Tag names can be any string of letters and numbers, not containing spaces (blank characters) and not starting with a number or special character
- XML distinguishes small letters and capital letters, e.g. `<title>` is different from `<Title>`
- Elements can contain “text” or other elements (can also be void)



Special characters



Some special characters cannot appear in the value (the “text”) of an element

name	meaning
&amp;	&
&lt;	<
&gt;	>
&apos;	'
&quot;	”



Attributes of an element



- Attributes can be defined in the opening tag, to provide additional information about the element, e.g.
`<actor role="primary">Brad Pitt</actor>`
- Whether the “additional information” is provided as an attribute or in a nested element is decided by the designer of the XML definitions

```
<person>  
  <name>Mario</name>  
  <surname>Rossi</surname>  
  <sex>male</sex>  
</person>
```

```
<person sex="male">  
  <name>Mario</name>  
  <surname>Rossi</surname>  
</person>
```

```
<person surname="Rossi">  
  <name>Mario</name>  
  <sex>male</sex>  
</person>
```



Conflicts in tag names



In XML, element names are defined by the developer. This often results in a conflict when trying to mix XML documents from different XML applications.

```
<table>  
  <tr>  
    <td>Room1</td>  
    <td>Room2</td>  
  </tr>  
</table>
```

HTML table

```
<table>  
  <name>Coffee Table</name>  
  <width>80</width>  
  <length>120</length>  
</table>
```

piece of furniture



Use of prefixes



```
<h:table>
  <h:tr>
    <h:td>Room1</h:td>
    <h:td>Room2</h:td>
  </h:tr>
</h:table>
```

This comes from the
“namespace” **h**

```
<f:table>
  <f:name>Coffee Table</f:name>
  <f:width>80</f:width>
  <f:length>120</f:length>
</f:table>
```

This comes from the
“namespace” **f**



Declaration of namespaces



```
<root  
xmlns:h="http://www.w3.org/TR/html4/"  
xmlns:f="http://www.w3schools.com/furniture">
```

Declaration of
“namespace” h

```
<h:table>  
  <h:tr>  
    <h:td>Room1</h:td>  
    <h:td>Room2</h:td>  
  </h:tr>  
</h:table>
```

Declaration of
“namespace” f

```
<f:table>  
  <f:name>Coffee Table</f:name>  
  <f:width>80</f:width>  
  <f:length>120</f:length>  
</f:table>  
  
</root>
```

Namespaces are
“resources” and are
identified by their URI
and (usually) are
specified as attributes
of the root



Definition of document types



- Need to define exactly the “abstract” structure of a given XML document, regardless of its actual content
 - Element (tag) names
 - Content of each element (simple or complex)
- DTD: Data Type Definition (deprecated)
 - Uses a syntax different from XML
- XML Schema
 - Uses the same syntax as any other XML document



An XML document



```
<?xml version="1.0" encoding="ISO-8859-1"?>
<shiporder orderid="889923">
  <orderperson>John Smith</orderperson>
  <shipto>
    <name>Ola Nordmann</name>
    <address>Langgt 23</address>
    <city>4000 Stavanger</city>
    <country>Norway</country>
  </shipto>
  <item>
    <title>Empire Burlesque</title>
    <note>Special Edition</note>
    <quantity>1</quantity>
    <price>10.90</price>
  </item>
  <item>
    <title>Hide your heart</title>
    <quantity>1</quantity>
    <price>9.90</price>
  </item>
</shiporder>
```



An XML schema (shiporderschema.xsd)



```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="shiporder">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="orderperson" type="xs:string"/>
      <xs:element name="shipto">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="name" type="xs:string"/>
            <xs:element name="address" type="xs:string"/>
            <xs:element name="city" type="xs:string"/>
            <xs:element name="country" type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="item" ... > ..... </element>
    </xs:sequence>
    .....
  </xs:complexType>
</xs:element>
</xs:schema>
```



An XML schema



```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="shiporder">
  <xs:complexType>
    <xs:sequence>
      .....
      <xs:element name="item" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="title" type="xs:string"/>
            <xs:element name="note" type="xs:string" minOccurs="0"/>
            <xs:element name="quantity" type="xs:positiveInteger"/>
            <xs:element name="price" type="xs:decimal"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="orderid" type="xs:string" use="required"/>
  </xs:complexType>
</xs:element>
</xs:schema>
```



The XML schema



- defines elements that can appear in a document
- defines attributes that can appear in a document
- defines which elements are child elements
- defines the order of child elements
- defines the number of child elements
- defines whether an element is empty or can include text
- defines data types for elements and attributes
- defines default and fixed values for elements and attributes



Namespaces



- A schema is a collection (vocabulary) of type definitions and element declarations
- Each different schema can be assigned a unique name (i.e. a URI) which indicates the “namespace” defined in the schema
- The namespace(s) are useful to “validate” an “instance document”
- The namespaces used in an instance document are (usually) declared as attributes of the root



Target name space



```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.example.com/shiporderschema">
<xs:element name="shiporder">
  <xs:complexType>
    <xs:sequence>
      .....
    </xs:sequence>
    <xs:attribute name="orderid" type="xs:string" use="required"/>
  </xs:complexType>
</xs:element>
</xs:schema>
```



Referencing a schema



```
<?xml version="1.0" encoding="ISO-8859-1"?>
<shiporder orderid="889923"
  xmlns="http://www.example.com/shiporderschema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.example.com/shiporderschema
    http://www.example.com/shiporderschema.xsd">

  <orderperson>John Smith</orderperson>
  <shipto>
    .....
  </shipto>
  <item>
    .....
  </item>
  <item>
    .....
  </item>
</shiporder>
```



XML and the Web



XML is the de-facto standard for the exchange of information and metadata descriptions on the Web

- TEI (Text Encoding Initiative)
 - Specifications of encoding methods for machine-readable texts, chiefly in the humanities, social sciences and linguistics.
- EAD (Encoded Archival Description)
 - XML markup designed for encoding archival information
- PREMIS (Preservation Metadata)
 - A data dictionary and supporting XML schemas for core preservation metadata needed to support the long-term preservation of digital materials.



XML and the Web (Digital Libraries)



- MARCXML
 - MARC 21 data in an XML structure
- MODS (Metadata Object Description Standard)
 - XML markup for **selected metadata** from existing MARC 21 records as well as original resource description
- MADS (Metadata Authority Description Standard)
 - XML markup for **selected authority data** from MARC21 records as well as original authority data
- METS (Metadata Encoding & Transmission Standard)
 - Structure for encoding descriptive, administrative, and structural metadata
- MIX (NISO Metadata for Images in XML)
 - XML schema for encoding technical data elements required to manage digital image collections



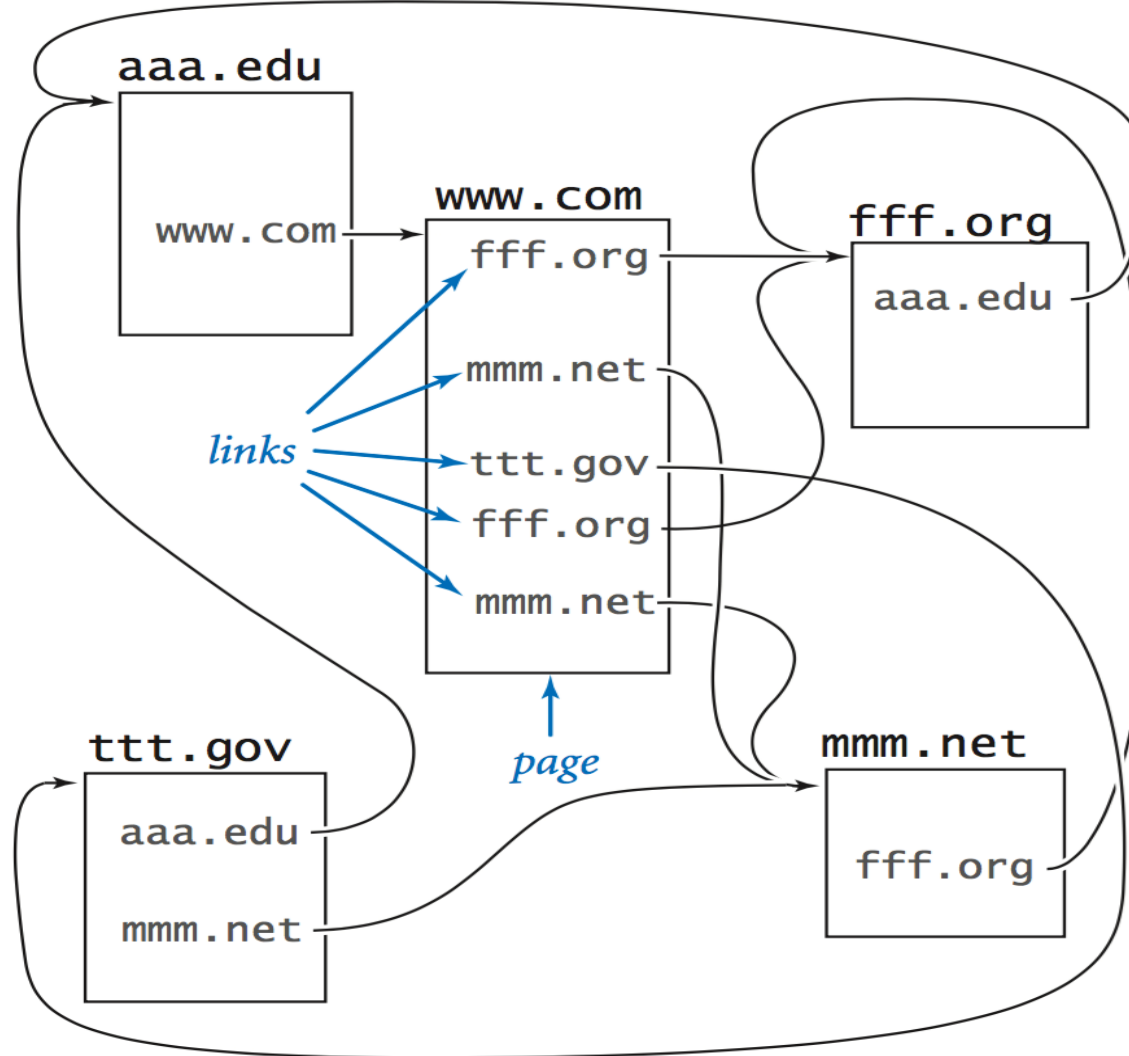
Evolution of information on the Web



- The World Wide Web is called “web” as the information can be seen as a graph of web pages interconnected by hyperlinks
- The hyperlink is “anonymous”, i.e. it does not say what is at the other end. The human, reading the words associated with the hyperlink, can figure out what could be at the other end of the hyperlink (that is very difficult for a program)
- In the Web of Linked (Open) Data the hyperlinks have a “type”, which (in a given vocabulary) provide an indication (to a human or to a program) on what should be expected at the other end of the link



The hyperlinks

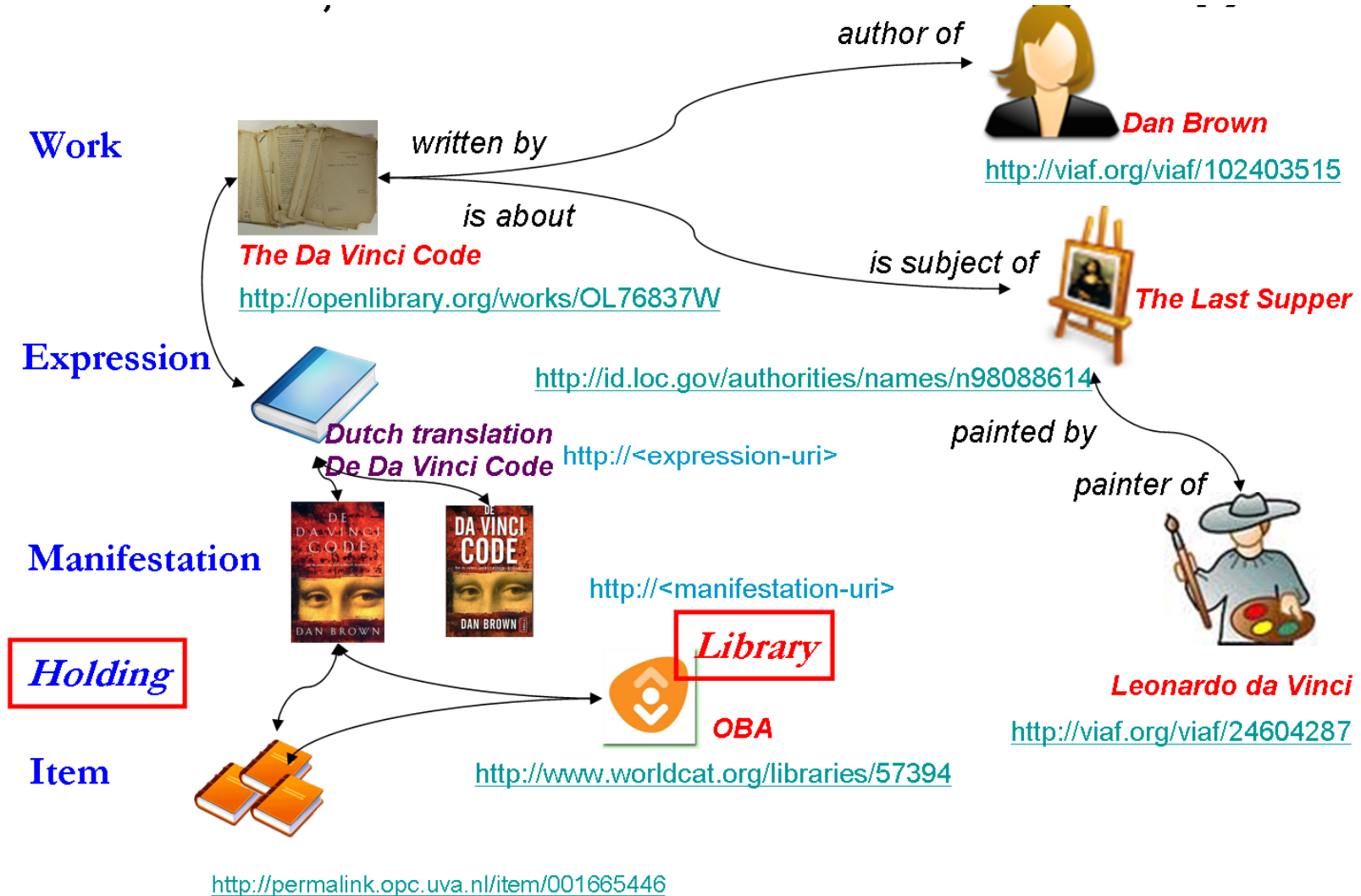




The World Wide Web

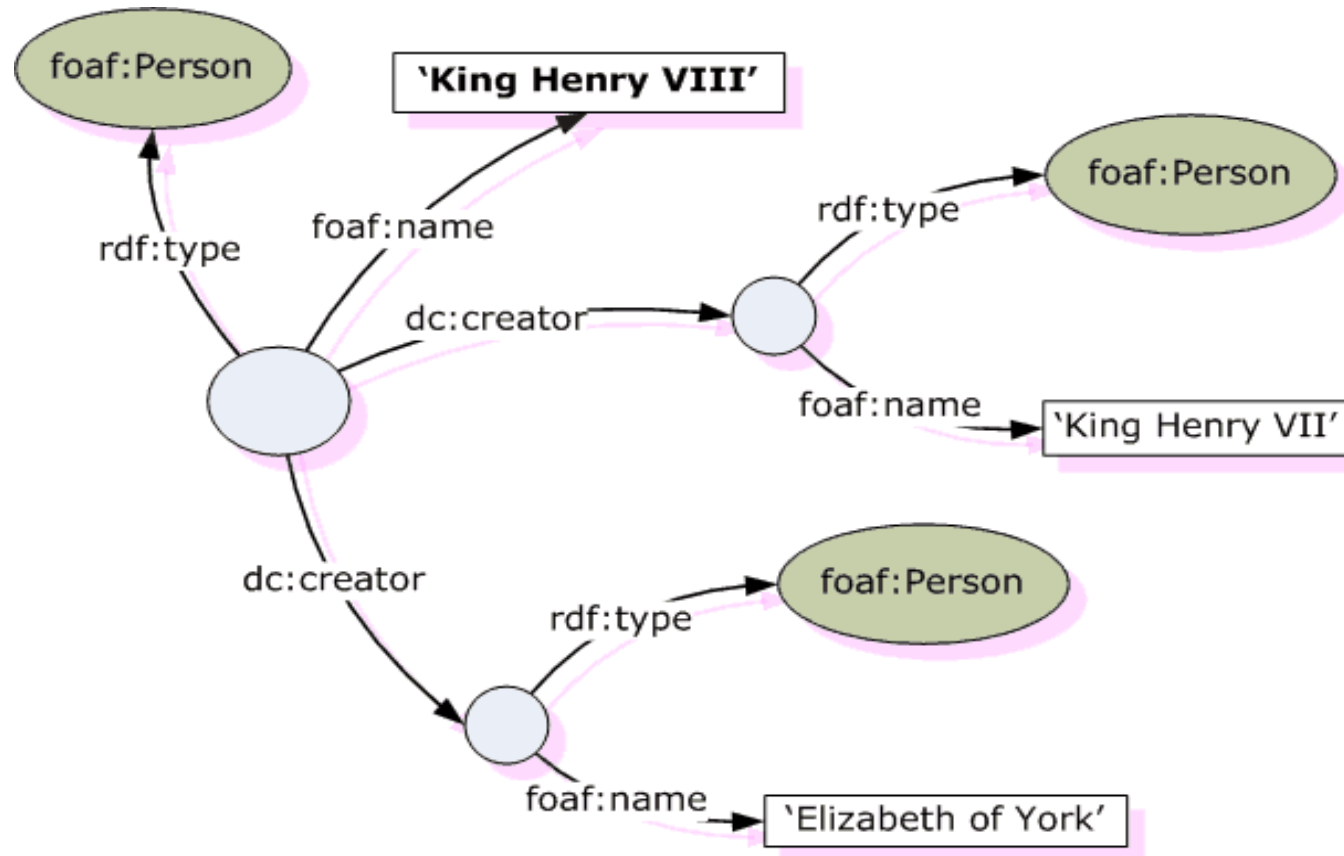


Web of Linked Data





Vocabularies of linked data



```
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns# >  
prefix dc: <http://purl.org/dc/elements/1.1/>  
prefix foaf: <http://xmlns.com/foaf/0.1/>
```