§Location (URL)

In the initial analysis, this element was intended to include addresses that can be used for accessing the dataset (for download, for dynamic query, for requesting delivery), i.e. a URL for a networked resource, but possibly a physical address for an analogue or offline resource.

I think it is inadvisable to 'unpack' or describe URLs with further metadata properties. I suggest that the value of the URL (HTTP URI) is that it is formally defined and atomic. There is nothing more than can be reliably asserted about a URL.

Paul Walk - happy to argue about this :-)

A typical URL would have the form http://www.example.com/index.html, which indicates a protocol (http), a hostname (www.example.com), and a filename (index.html)

From Wikipedia: A Uniform Resource Locator (URL), commonly informally termed a web address (a term which is not defined identically) is a reference to a web resource that specifies its location on a computer network and a mechanism for retrieving it. A URL is a specific type of Uniform Resource Identifier (URI), although many people use the two terms interchangeably. A URL implies the means to access an indicated resource and is denoted by a protocol or an access mechanism, which is not true of every URI. Thus http://www.example.com is an URL, while www.example.com is not. URLs occur most commonly to reference web pages (http), but are also used for file transfer (ftp), email (mailto), database access (JDBC), and many other applications…

My take: A URL commonly uses the HTTP protocol to reference a location. It is partly made up of a "domain name" where a domain service is used (DNS - Domain Name Service) that translates back to the IP address of the host/server. It is useful to use a "domain name" to enable dynamic assignment of IP address to the host/server. Sometimes a "subdomain is used" to further subdivide the "domain name" to different hosts/servers.

For example, Google has domain name: "http://www.google.com".

It has a number of subdomains including:

- "http://drive.google.com"
- "http://pages.google.com"

Port numbers are sometimes provided (if default port 80 is not used) as part of the "URL" to instruct the client to communicate with the server on a specific port.

Unpacking a URL can be possibly done into the following parts:

- Protocol (http, ftp, mailto, jdbc, etc.)
- Subdomain (www or other)
- Domain name (name.com, name.de, etc.)
- Port number (80 or other)
- Directory (path to the page, if none is provided, server uses root web directory)

- Page (if no page is provided, server uses default page)

It can be argued that URL does not need further unpacking which can be the case for most users but to understand the "sum of the parts" and sometimes to describe the parts in a data representation such as XML, you may need to break it down further.

Data Representation Example:

```
<URL>
        <protocol>https</protocol>
        <domain>
                <subdomain>drive</subdomain>
                <domain_name>google.com</domain_name>
        </domain>
        <port_number>80</port_number>
        <directory>intl/en/drive/</directory>
        <page>index.html</page>
</URL>
```

OR simply:

```
<URL>https://www.google.com/intl/en/drive/</URL>
```

In which case it is up to software to parse it and understand it further, rather than putting the separate parts together to form the "URL".

Sometimes it helps to define the various parts because it is sometimes not provided in the URL itself. In this example, index.html is not provided in the URL.

Sharief Youssef, NIST

Ted Habermann -
Breaking URLs down into atomic pieces makes them harder to use and confusing for many metadata providers. It may make sense to focus here on other information that can make the URL more useful to users. Three things come to mind: titles, descriptions, and functions. The ISO TC211 model for online resources includes all three.

Mappings

Please give below:

- The name of your data archive, repository, catalogue, etc.
- For your native or internal metadata scheme (or one you normally use), please list the elements that correspond to this high-level element. For each one:
    - Give the name of the element
    - Give any constraints on the value (controlled vocabulary, syntax/encoding)

- Highlight any divergence in the semantics of the element from the discussion above.

Keith Jeffery comments 20200601

The key purpose of the Location element is to provide a route to reach the digital object (metadata, data, software source code, workflow description…) of interest that is described / characterised by other metadata elements.

The most popular method of defining location is with a URL https://www.w3.org/Addressing/URL/url-spec.html . For the purposes of management of metadata (and hence management of access to the described digital object) the sub-elements of a URL are not helpful since many of the sub-elements of a URL would be documented in other metadata elements (e.g. Organisation).

The element location is not to be confused with spatial or temporal location which are handled by other elements (spatial coordinates, temporal coordinates).