Perspectives on using Schema.org for publishing and harvesting metadata at Europeana

Valentine Charles, Richard Wallis, Antoine Isaac, Nuno Freire and Hugo Manguinhas | SWIB 2017
European Cultural Heritage on the Web

The main goal of Europeana is to provide access to cultural heritage and encourage people to engage with culture.

- And the main access point is the Web!
- It is crucial for Europeana to be recognised as a trusted and authoritative repository of cultural heritage by the search engines.
Perspectives on using Schema.org for publishing and harvesting metadata at Europeana
Data in Europeana

Publication of data on the Web supported by the Europeana Data Model (EDM)

- It enables the representation of:
  - structured and open data (CC0 license)
  - rich in links between objects and their digital representations
  - links to controlled vocabularies and datasets (e.g. Geonames, DBpedia, Wikidata)
Schema.org

- **Schema.org** is developed as a vocabulary, following the Semantic Web principles.
- It is a collaborative and community-based activity, and its main platform of collaboration is the W3C Schema.org Community Group.
- Its main application is in web pages, where data can be referenced or embedded in many different encodings (e.g., RDFa, Microdata and JSON-LD).
- (Digital) Cultural heritage objects can be represented in Schema.org.
- Schema.org can also be extended:
  - The **Bibliographic Extension** provides additional properties and types to describe bibliographic resources.
  - The **Architypes extension** currently works on identifying relevant types and properties to describe archives and their contents.
Mapping EDM to Schema.org
Objective: a Schema.org representation of Europeana EDM, being as rich as possible and tailored to Europeana’s realities and user needs

- schema:CreativeWork and several of its refining subclasses such as schema:VisualArtwork, schema:Book, schema:Painting, schema:Sculpture, and schema:Product can be matched to edm:ProvidedCHO
  - subclasses may be used with more specific properties than the ones available for schema:CreativeWork such as schema:artMedium for schema:VisualArtwork.

- schema:MediaObject and its subclasses schema:ImageObject, schema:VideoObject, schema:AudioObject can be matched to edm:WebResource

- the schema:Person, schema:Place and schema:Organization classes match the semantics of EDM contextual classes edm:Agent, edm:Place and foaf:Organization.
Examples of mapping issues

- Mapping edm:ProvidedCHO to subtypes of schema:CreativeWork (e.g. schema:Book, schema:Painting, schema:Sculpture, schema:ImageObject) will require a mapping with dc:type.

- Mapping edm:Webresource to more specific subtypes of schema:MediaObject (e.g. schema:ImageObject, schema:AudioObject, schema:VideoObject) will require a mapping between MimeTypes, file extensions, etc. to ascertain the correct type.

- artMedium/artform/artworkSurface
  - These are properties of schema:VisualArtwork indicating the physical type of artwork such as sculpture, painting, drawing, etc.
Making the most of your strings

different strategies to expand strings into entities

A minimal requirement is to expand strings into an entity description

```
1    "creator": "J .M. Smith"

Should be expanded to:
```
```
1    "creator": {
2        "@type": "Person"
3        "name": "J. M. Smith"
4    }
```
Making the most of your strings

different strategies to expand strings into entities

1. Implicit Blank Nodes (nested output)

```json
{
    "@id": "http://data.europeana.eu/item/2021618/internetserver_Details",
    "@type": ["CreativeWork", "VisualArtwork"],
    "name": "Mona Lisa | Leonardo da Vinci",
    "creator": {
        "@type": "Person",
        "name": "Luigi Calamatta"
    }
}
```

2. Explicit Blank nodes

```json
{
    "@id": "http://data.europeana.eu/item/2021618/internetserver_Details",
    "@type": ["CreativeWork", "VisualArtwork"],
    "name": "Mona Lisa | Leonardo da Vinci",
    "creator": "_:p0",
}
{
    "@id": "_:p0",
    "@type": "Person",
    "name": "Luigi Calamatta"
}
```

3. Entity Reference

```json
{
    "@id": "http://data.europeana.eu/item/2021618/internetserver_Details",
    "@type": ["CreativeWork", "VisualArtwork"],
    "name": "Mona Lisa | Leonardo da Vinci",
    "creator": "http://data.europeana.eu/item/p12345",
}
```
URI design

- Make sure to distinguish URIs of the resources from the URI of the Web Page

```json
"@graph": [
  {
    "@id": "http://www.europeana.eu/portal/record/2021618/internetserver_Details",
    "@type": "WebPage",
    "mainEntity": "http://data.europeana.eu/item/2021618/internetserver_Details"
  },
  {
    "@id": "http://data.europeana.eu/item/2021618/internetserver_Details_kunst_2",
    "@type": ["CreativeWork", "VisualArtwork"],
    "name": "Mona Lisa | Leonardo da Vinci",
    "creator": ["http://data.europeana.eu/item/p12345", "Luigi Calamatta"]
  }]
```
Practicalities for publishing
Schema.org at
Europeana.eu
Schema.org data embedded within html pages

**Objective:** to enable external organizations in general, and Search Engines in particular, to consume the data into their Knowledge Graphs of resources on the web.

- Embedding Schema.org data shouldn’t impact the primary purpose of the html pages in supporting human interaction, we therefore recommend to separate the interface concerns from the underlying data structures.
- The user interface design requirements of Europeana websites may change independently of the underlying data structures.
- The Schema.org vocabulary will evolve as well as the modeling and quality of data stored by Europeana.
- A standard approach is to ‘bolt-on’ the structured data to the page construction.
- It consists in inserting a section in the page source code, containing the structured data, that does not impact on its visual output.
JSON-LD output

- JSON-LD format inserted into a html script tag

```html
<script type="application/ld+json">
{
    "@context": "http://schema.org",
    "@graph": [
    {
        "@id": "http://data.europeana.eu/item/2021618/internetserver_Details_kur",
        "@type": ["CreativeWork", "VisualArtwork"],
        "name": "Mona Lisa I Leonardo da Vinci",
    }
    ]
}</script>
```
Generation of Schema.org data

On-the-fly

- the source data being read as EDM from storage and then being passed through a mapping/conversion process.
  
  + no extra data is stored to support Schema.org; also changes to mapping rules are instantly available.
  
  - system loading, and difficulty in supporting complex dependencies in data mapping.

Batch creation

- An alternative is that the resource data is batch processed
  
  + not needing processing to extract data for display.
  
  - difficult to cope with mapping changes and re-indexing of databases.

Combined approach (on-the-fly & batch creation)

- Standard web caching techniques to limit loading requirements
Objective: get search engines to crawl and consume data from the pages describing Europeana resources.

- Sitemaps inform search engines about which of the website URLs are available for crawling and some additional information that will enable the website to be crawled more effectively.

- Sitemaps need to be provided and well maintained for all pages that contain Schema.org data on the Europeana websites.
  - Sitemaps are regularly updated to indicate new and updated pages.
  - This needs to take into account pages that visually may not have changed, but have data output that has changed.
  - Not indicating changed pages, or wrongly indicating that pages have changed, can result in a site not being fully crawled and data not being consumed.
Europeana as a harvester of Schema.org
Harvesting data using Schema.org sitemap

- Schema.org sitemap can also be used as a point of reference for harvesting data
  - the mechanism to aggregate Schema.org data can start the same way as for crawling ordinary web pages.
  - Then the process is comparable to the one for ordinary web pages, which is based on following the hyperlinks within the HTML.

In the particular case of digital library websites, sitemaps help dealing with some typical discovery problems faced by CH institutions:
- They enable web crawlers to reach areas of the website that are not available through the browsable interface.
- There are chances that the web crawlers will overlook some of the new or recently updated content.
Europeana harvesting Schema.org

- A new mapping from Schema.org to EDM is required
Conclusion

● It is possible to represent Europeana data resources using the Schema.org vocabulary.

● We will implement the mapping in our API output (planned for end of next year).

● We will work on further recommendations and/or specifications to enable the provision of Schema.org metadata interoperable with EDM.

⇒ More details in the Code4Lib paper