Aggregating media fragments into collaborative mashups: standards and a prototype

Philippe Duchesne
phd@highlatitud.es

Abstract. This paper presents the ongoing CrossLinks project aimed at building a data mashup standard and platform. The presented work is threefold. It first reviews existing syntaxes for URI fragments, builds upon them to formalize a uniform way to reference fragments of any media type, thereby allowing hyperlinking of sub-elements. These data fragments are then stitched together by devising a model and exchangeable format for aggregated data views. Lastly, a platform is developed to ease authoring, storage, exchange and most importantly visualization of such data mosaics.

Keywords. Linked Data, URI Fragments, Media types, Data mashups, Collaboration, Visualization

1. Introduction

Now that large amounts of open data become available, along with efficient visualization tools for their respective media types, one of the next challenges is to make sense of these data in the scope of particular domains and use cases. A fair part of making that sense lies in the ability to link parts of data elements and tie them together in mosaics that carry more information than the sum of these elements.

When curating data to offer a unified and structured view of heterogeneous subsets that fits their usecase, users face problems such as:

- they do not own the various data sources they are referring to, nor can they edit them
- they need to collect and structure only parts of resources identified by URIs
- they need to store the result of their curation, exchange it and possibly collaboratively edit and annotate it
- they need to view that result in a transparent interface that overlays resources with their respective annotations and related fragments

This paper is about the CrossLinks project, aimed at building an application for authoring, exchanging and consuming such aggregated data views. It presents both the prototype itself, and the abstract and formal approach that underpins it.

2. Referencing fragments

The project first has to address the notion of URI fragments, and the lack of support thereof by current mainstream tools and standards. Current mass-market usages of URI fragments mostly include HTML anchored elements (the only one consistenly supported by browsers), but also the time dimension in YouTube online player, or the line numbering dimension in some PDF viewers. Named fragments are also

noticeably used to identify named elements of RDF graphs [6]. The URI specification [1] itself states that fragment syntax depends on resolved URI media type, and therefore usually on vendor-specific viewers for these media types.

Several recent specifications propose ways to formalize and extend usage of URI fragments. Most noticeably, the Media Fragments URI recommandation [2] focuses on *temporal* and *spatial* dimensions (using respectiveley the `t` and `xywh` fragment elements), but also covers the *track* and *id* dimensions, in a work targeting video and audio media types, although their use can obviously be extended to any media type offering said dimensions.

Likewise, an IETF Internet draft by Hausenblas and Wilde [3] focuses on URI fragments for the CSV mime type, defining fragment elements such as `row`, `col` or `cell`. Their work is very important in the scope of this project, as the ability to integrate and portray fragments of tabular data is a key feature in the scope of open data curation.

Building on these efforts, the CrossLinks project suggests extra dimensions and corresponding fragment syntaxes that should be considered to properly cover the range of media types that are relevant. Namely, an `xpath` fragment element based on XPath expressions [8] is proposed for tree-structured data types such as XML, HTML or JSON. Another fragment element (`bbox`) defined after the OGC WMS [7] specification, is dedicated to geospatial datatypes, as these constitute a fair part of data curation usecases, and deserve a dedicated fragment expression distinct from the pixel-wise `xywh`, both for conceptual and practical reasons.

Table 1 below summarizes these fragment expressions chosen in the project. In this table, the `page` fragment is retained, although it overlaps the `line` fragment. The `ElementId` fragment is retained too, because of its wide acceptance (HTML, RDF). It is however not coherent (it has no explicit fragment identifier - it acts as the default fragment dimension).

Fragment syntax	Representative media types	Description
`#t=`*`timerange`*	`video/*, audio/*`	used for temporal fragments, as defined in MediaFragments
`#xywh=...`	`image/*, video/*`	used for fragments in pixel space, as defined in MediaFragments
`#xpath=...`	`text/html, application/xml, application/json`	used for any tree-structured media type
`#bbox=`*`minx`*`,`*`miny`*`,`*`maxx`*`,`*`maxy`	`application/vnd.google-earth.kml+xml, application/gml+xml`	used for fragments in the geospatial space
`#col=`*`range`*`&row=`*`range`	`text/csv`	used for tabular data fragments
`#line=`*`range`	`text/plain`	as defined in RFC5147
`#page=...`	`application/pdf`	used for paged media, such as pdf or slides
`#ElementId`	`application/rdf+xml`	identifies an element by ID in the namespace of the RDF resource

Table 1. *Proposed fragment expressions, and their respective media types (not exhaustive)*

3. Building data mosaics

Assuming URI fragments as defined above are supported, the next step is to be able to gather relationships between data fragments to form a "linked fragments bundle" that conveys information about a specific topic or context of work, thereby constituting a contextual data view. Current standards and tools allow to express links between resources (e.g. RDF), but there lacks a comprehensive format to easily structure and exchange such data views.

The CrossLinks project defines a simple abstract model that fits those needs, based on a folder structure containing resource references, links between those resources and possibly annotations from collaborating users. The term 'resource' here refers to the result of the resolution of a URI that potentially contains a *fragment* expression, i.e. whole or part of an online dataset. Using JSON as an arbitrary encoding for the sake of an example, this can yield:

```
{"name": "Sample Link Set",
 "content" : [
   {"name": "Discovering JavaScript Object Notation with Douglas Crockford",
    "src": "http://youtu.be/kc8BAR7SHJI",
    "links": [
       {"src":"#t=40,42",
        "target": "http://example.org/foo.html#xpath=...",
        "description": "Discovering Javascript Object Notation"},
       {"src":"#t=02:23,02:25",
        "target": "http://example.org/bar.csv#col=2-3&row=0-100",
        "description": "Getting JSON accepted"},
       {"src":"#t=03:25,03:28",
        "description": "Here he explains why XML is wrong"},
       {"src":"#t=04:00,04:20&xywh=30,30,100,50",
        "description": "Highlight a 100x50 rectangle in the video area"}
    ]},
   {"name": "Some KML document",
    "src": "http://example.com/countries.kml",
    "links": [
       {"src":"#bbox=2.5,49.5,6.4,51.5",
        "target": "http://example.org/belgium.html",
        "description": "This is belgium"},
       {"src":"#bbox=2.5,49.5,6.4,51.5&t=20,30",
        "description": "This annotation appears over Belgium only between [20,30]"}
    ]},
 ]}
```

This example highlights some of the possible usages of fragments and links across media types, e.g. how temporal and pixelspace segments of an online video can be annotated with fragments of other online resources such as an HTML page paragraph, some columns of a CSV table, or merely with some inline text from the user.

The use of JSON and the structure of the objects here is only for the sake of the example, and does not constitute a formal encoding suggestion. Besides, several useful attributes are omitted throughout this example for the sake of conciseness, such as title, mime types, tags, or semantic qualifier for the links. A detailed description of the abstract model and encoding is part of the CrossLinks project.

4. Implementation

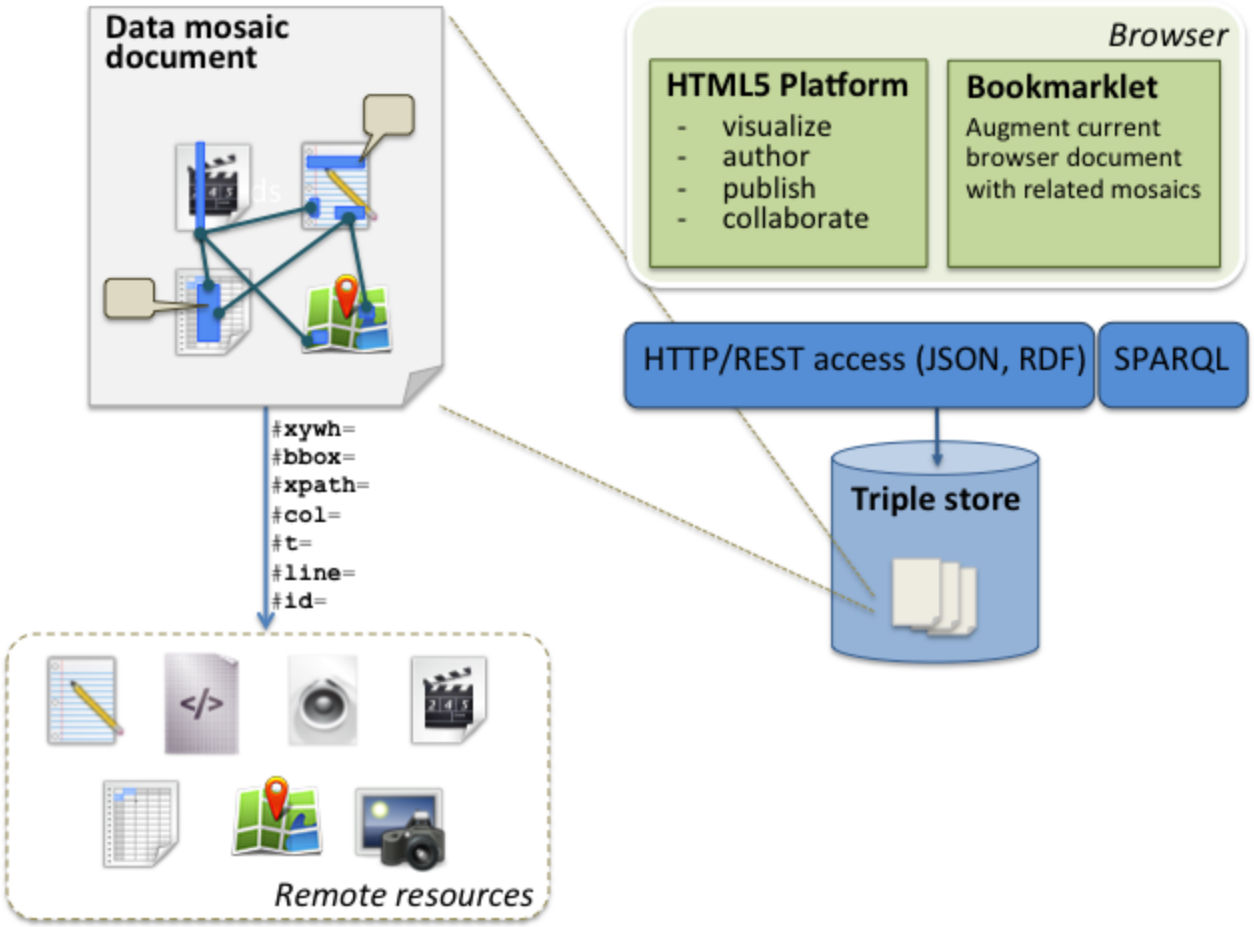

Fig 1. *Client-Server overview*

The server features a triple store to hold data mosaics, using an RDF encoding of the abstract model defined above. That data store is exposed through an HTTP/REST interface to manipulate data mosaics via JSON representations, mostly for a convenient access from a browser-based platform. The server also exposes a SPARQL endpoint for finer-grained querying.

The client platform is an HTML5-based application that integrates several state-of-the-art libraries for the rendering of all the media types supported, namely at this stage video, audio, html, tabular, geo, including some custom viewers for content publishing platforms such as YouTube or Slideshare. Beside the main client that offers extended authoring capabilities, a major feature of this HTML platform is the existence of bookmarklets that allow users to overlay a page being browsed with content from the server, offering an augmented view of the browsed document. This ability to dynamically overlay the page being browsed with annotations and media fragments is crucial to offer a seamless experience and avoid breaking the user's train of browsing thought (or better capture it when in authoring mode).

5. Conclusion & Perspectives

If properly standardized and used, URIs fragments can be leveraged to produce finer-grained linked data, bringing semantics to any subparts of an online resource.

Along with URI fragments, data mosaics constitute aggregated views that are very useful to convey domain-specific knowledge and perspective, enabling users to easily stitch together and publish added value data mashups. Because such data mosaics exist and can evolve independently of the related resources, they are resources on their own, like multi-faceted patchworks of remote data that can be collaboratively authored, annotated and exchanged.

A user interface that offers a seamless overlay of those data mosaics onto the browsing habits of users proves to be a major feature to engage users in the data curation process, and offers a new browsing experience.

In terms of uses cases, and using the W3C Working Draft "*Linked Data Platform Use Cases and Requirements*" [9] as a reference nomenclature, this platform prototype presented here targets the "Manage containers", "Update existing resource" (through enrichment), "Aggregate resources" and "Manage media resources" use cases.

Regarding implementation, several features of this prototype (like standardized fragment support, enhanced media types rendering, cross-domain data mashups) should be considered as out-of-the-box capabilities of mainstream browsers, as they are generic and are potential building blocks for browser-based linked data solutions.

6. References

1. Berners-Lee T., Fielding R., and L. Masinter (2005) *Uniform Resource Identifier (URI): Generic Syntax*, IETF, RFC 3986, STD 66. http://tools.ietf.org/html/rfc3986
2. Mannens E, Troncy R, Pfeiffer S, Van Deursen D (eds) (2012) *Media fragments URI 1.0.* W3C Proposed Recommandation. http://www.w3.org/TR/media-frags/
3. Hausenblas M., Wilde E. (eds) (2013) *URI Fragment Identifiers for the text/csv Media Type.* IETF Internet Draft. http://tools.ietf.org/html/draft-hausenblas-csv-fragment-02
4. Duerst M., Wilde E. (2008) *URI Fragment Identifiers for the text/plain Media Type*, IETF, RFC 5147. http://tools.ietf.org/html/rfc5147
5. Tom Heath and Christian Bizer (2011) *Linked Data: Evolving the Web into a Global Data Space* (1st edition). Synthesis Lectures on the Semantic Web: Theory and Technology, 1:1, 1-136. Morgan & Claypool. http://linkeddatabook.com/editions/1.0/
6. Klyne G., Carroll J. J. (2004) *Resource Description Framework (RDF): Concepts and Abstract Syntax*. World Wide Web Consortium. http://www.w3.org/TR/rdf-concepts/
7. de la Beaujardiere J. (ed) (2006) *OpenGIS Web Map Service (WMS) Implementation Specification*, OGC 06-042. http://www.opengeospatial.org/standards/wms

8. World Wide Web Consortium (2010) *XML Path Language (XPath) 2.0*, W3C Recommandation. http://www.w3.org/TR/xpath20
9. Battle S., Speicher S (2013) *Linked Data Platform Use Cases and Requirements*, W3C Working Draft. http://www.w3.org/TR/ldp-ucr/