SPARQL-MM - Extending SPARQL to Media Fragments

Thomas Kurz[1], Sebastian Schaffert[1], Kai Schlegel[2], Florian Stegmaier[2], and Harald Kosch[2]

[1] Salzburg Research, Salzburg, Austria,
thomas.kurz@salzburgresearch.at,
www.salzburgresearch.at
[2] University of Passau, Passau, Germany,
www.dimis.fmi.uni-passau.de

Abstract. Interconnecting machine readable data with multimedia assets and fragments has recently become a common practice. But specific retrieval techniques for the so called Semantic Multimedia data are still lacking. On our poster we present SPARQL-MM, a function set that extends SPARQL to Media Fragment facilities by introducing spatio-temporal filter and aggregation functions.

Keywords: SPARQL, Semantic Web, Media Fragments

1 Introduction

In the recent past Multimedia and the Semantic Web have moved closer together, reflected in many efforts like W3C recommendations for Media Annotations [2] and Fragment URIs [7], which are also taken up by industry like described e.g. in [4] and [5]. These efforts mainly focus on a standardized representation of multimedia metadata and the connection to knowledge using Semantic Web technologies, but currently do not consider multimedia specific query functionalities. On our poster we present SPARQL-MM, an extension for SPARQL that introduces spatio-temporal filter and aggregation functions to the de-facto standard query language in the Semantic Web. In the following we give an example, how media is currently represented in the Semantic Web and which functions are missing for current SPARQL.

2 The hidden knowledge of fragment annotations

Figure 1 outlines an example of an annotated video showing the winners ceremony of an extreme sports event. We use Media Fragment URIs to link annotations to specific spatio-temporal parts of the video. The specification provides a media-format independent, standard means of addressing media fragments on the Web using Uniform Resource Identifiers. It supports particular name-value pairs, e.g. `('t='start','end)` for temporal and `('xywh=',x','y','width`

','height) for regional fragments. In our example *Connor Macfarlane* appears from second 194 to 198 on the left side, while *Lewis Jones* is marked from second 193 to 198 on the right side. If a user wants to retrieve the (spatio-temporal) snippet, that covers exactly these both person, she may issue a query like: "Give me the spatio-temporal snippet that shows *Lewis Jones* right beside *Connor Macfarlane'*". Currently SPARQL does not support such queries, because some necessary information is not explicitly expressed in RDF, but hidden within the Media Fragment URIs. Neither does SPARQL support filter functions like *rightBeside* or *temporalOverlap*, nor aggregation functions like *boundingBox*.

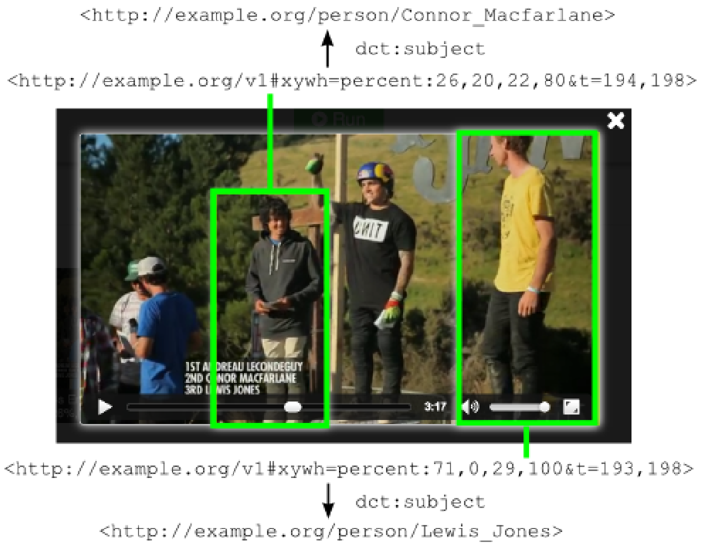
<http://example.org/person/Connor_Macfarlane>
dct:subject
<http://example.org/v1#xywh=percent:26,20,22,80&t=194,198>
1ST ANDREAU LECONDEGUY
2ND CONNOR MACFARLANE
3RD LEWIS JONES
3:17
<http://example.org/v1#xywh=percent:71,0,29,100&t=193,198>
dct:subject
<http://example.org/person/Lewis_Jones>

Fig. 1. A sample for an annotated video

3 SPARQL Multimedia Functions

Table 1 lists all functions that we took into account for our extension. We follow well known standards like DE-9IM [3] for topological and [1] for temporal relations. The parameters r1, r2 are of type media fragment resource, the boolean flag decides, if equal ranges are included or not (whereby false is default). A detailed description of all functions in human and machine readable format (following the sparql-service-description extension for describing SPARQL extensions and function libraries[3]) can be found on the source repository of our reference implementation[4]. Each function is identified by a unique URI but all together share

[3] http://www.ldodds.com/schemas/sparql-extension-description/

[4] http://github.com/tkurz/sparql-mm/

the same base URI `mm:<http://linkedmultimedia.org/sparql-mm/functions#>`. Currently the implementation uses the OpenRDF Sesame[5] API and its extension interfaces, which makes it backend agnostic but requires expensive and inefficient in-memory calculations. We plan to improve this by a backend specific implementation for KiWi Triplestore[6] in combination with SQL/MM [6].

	Relation Function	Aggregation Function

Spatial	`mm:spatialCovers(r1,r2[,flag])`
	`mm:spatialDisjoint(r1,r2)`
	`mm:spatialEqual(r1,r2)`
	`mm:spatialIntersects(r1,r2[,flag])`
	`mm:spatialTouches(r1,r2)`
	`mm:bottom(r1)`
	`mm:isAbove(r1,r2)`
	`mm:isBelow(r1,r2)`
	`mm:left(r1)`
	`mm:leftBeside(r1,r2)`
	`mm:right(r1)`
	`mm:rightBeside(r1,r2)`
	`mm:top(r1)`
		`mm:spatialBoundingBox(r1,r2)`
		`mm:spatialIntersection(r1,r2)`

Temporal	`mm:after(r1,r2)`
	`mm:before(r1,r2)`
	`mm:temporalContains(r1,r2[,flag])`
	`mm:temporalEqual(r1,r2)`
	`mm:finishes(r1,r2[,flag])`
	`mm:temporalMeets(r1,r2)`
	`mm:temporalOverlaps(r1,r2[,flag])`
	`mm:starts(r1,r2)`
		`mm:temporalIntersection(r1 r2)`
		`mm:temporalBoundingBox(r1,r2)`
		`mm:temporalIntermediate(r1,r2)`

Combined		`mm:boundingBox(r1,r2)`
		`mm:intersection(r1,r2)`

Table 1. SPARQL-MM Functions

Using SPARQL-MM functions we can now formulate the users need from section 2 ("Give me the spatio-temporal snippet that shows *Lewis Jones* right beside *Connor Macfarlane*"), as a SPARQL query like in Listing 1.1. We use *mm:temporalOverlaps* to get fragments that appear in the identical temporal sequence. *mm:rightBeside* handles the spatial relation and *mm:boundingBox* merges every two fragments that match the filters. You can test the query by selecting the example from the samples menu in the demo page[7].

[5] http://www.openrdf.org/

[6] http://marmotta.apache.org/kiwi/triplestore.html

[7] http://demos.mico-project.eu/sparql-mm/sparql-mm/demo/index.html

4 Conclusion and Further Work

In this paper we presented SPARQL-MM, a function set that adds spatio-temporal filters and aggregators to SPARQL. In further steps we will extend the function set to fragment feature extractors (e.g. *getDuration*) that can for example be used for sorting. We also try to include more complex spatial structures like polygons, e.g. by extending the MediaFragments URIs with a `rel` attribute that points to SVG shapes.

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX mm: <http://linkedmultimedia.org/sparql-mm/functions#>
PREFIX ma: <http://www.w3.org/ns/ma-ont#>
PREFIX dct: <http://purl.org/dc/terms/>

SELECT (mm:boundingBox(?l1,?l2) AS ?two_guys) WHERE {
    ?f1 a ma:MediaFragment; ma:locator ?l1; dct:subject ?p1.
    ?p1 foaf:name "Lewis Jones".
    ?f2 a ma:MediaFragment; ma:locator ?l2; dct:subject ?p2.
    ?p2 foaf:name "Connor Macfarlane".
    FILTER mm:rightBeside(?l1,?l2)
    FILTER mm:temporalOverlaps(?l1,?l2)
}
```

Listing 1.1. A SPARQL-MM query

SPARQL-MM is implemented as Open Source project hosted on GitHub.

5 Acknowledgments

SPARQL-MM is developed within MICO, a research project partially funded by the European Commission 7th FP (grant agreement no: 610480).

References

1. J.F. Allen. Maintaining Knowledge About Temporal Intervals. *Communications of the ACM*, 26(11):832–843, 1983.
2. Werner Bailer et al. Ontology for media resources 1.0. W3C recommendation, W3C, February 2012. http://www.w3.org/TR/2012/REC-mediaont-10-20120209/.
3. Eliseo Clementini et al. A small set of formal topological relationships suitable for end-user interaction. In *SSD*, volume 692 of *Lecture Notes in Computer Science*, pages 277–295. Springer, 1993.
4. Thomas Kurz et al. Semantic enhancement for media asset management systems. *Multimedia Tools and Applications*, pages 1–27, 2012. 10.1007/s11042-012-1197-7.
5. Lyndon J. B. Nixon et al. Connectme: Semantic tools for enriching online video with web content. In *I-SEMANTICS (Posters & Demos)*, volume 932 of *CEUR Workshop Proceedings*, pages 55–62, 2012.
6. Knut Stolze. SQL/MM Spatial - The Standard to Manage Spatial Data in a Relational Database System. In *BTW*, pages 247–264, 2003.
7. Raphaël Troncy et al. Media fragments URI 1.0 (basic). W3C recommendation, W3C, September 2012. http://www.w3.org/TR/2012/REC-media-frags-20120925/.