

SPARQL-MM - Extending SPARQL to Media Fragments

Thomas Kurz¹(✉), Sebastian Schaffert¹, Kai Schlegel², Florian Stegmaier²,
and Harald Kosch²

¹ Salzburg Research, Salzburg, Austria
thomas.kurz@salzburgresearch.at
<http://www.salzburgresearch.at>

² University of Passau, Passau, Germany
<http://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, 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*.



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 libraries¹) can be found on the source repository of our reference implementation². Each function is identified by a unique

¹ <http://www.ldodds.com/schemas/sparql-extension-description/>

² <http://github.com/tkurz/sparql-mm/>

URI but all together share the same base URI `mm:<http://linkedmultimedia.org/sparql-mm/functions#>`. Currently the implementation uses the OpenRDF Sesame³ 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 Triplestore⁴ in combination with SQL/MM [6].

Table 1. SPARQL-MM functions

	Relation function	Aggregation function
Spatial	<code>mm:spatialCovers(r1,r2[,flag])</code>	<code>mm:spatialBoundingBox(r1,r2)</code>
	<code>mm:spatialDisjoint(r1,r2)</code>	<code>mm:spatialIntersection(r1,r2)</code>
	<code>mm:spatialEqual(r1,r2)</code>	
	<code>mm:spatialIntersects(r1,r2[,flag])</code>	
	<code>mm:spatialTouches(r1,r2)</code>	
	<code>mm:bottom(r1)</code>	
	<code>mm:isAbove(r1,r2)</code>	
	<code>mm:isBelow(r1,r2)</code>	
	<code>mm:left(r1)</code>	
	<code>mm:leftBeside(r1,r2)</code>	
	<code>mm:right(r1)</code>	
	<code>mm:rightBeside(r1,r2)</code>	
	<code>mm:top(r1)</code>	
Temporal	<code>mm:after(r1,r2)</code>	<code>mm:temporalIntersection(r1 r2)</code>
	<code>mm:before(r1,r2)</code>	<code>mm:temporalBoundingBox(r1,r2)</code>
	<code>mm:temporalContains(r1,r2[,flag])</code>	<code>mm:temporalIntermediate(r1,r2)</code>
	<code>mm:temporalEqual(r1,r2)</code>	
	<code>mm:finishes(r1,r2[,flag])</code>	
	<code>mm:temporalMeets(r1,r2)</code>	
	<code>mm:temporalOverlaps(r1,r2[,flag])</code>	
	<code>mm:starts(r1,r2)</code>	
Combined		<code>mm:boundingBox(r1,r2)</code>
		<code>mm:intersection(r1,r2)</code>

Using SPARQL-MM functions we can now formulate the users need from Sect. 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`

³ <http://www.openrdf.org/>

⁴ <http://marmotta.apache.org/kiwi/triplestore.html>

merges every two fragments that match the filters. You can test the query by selecting the example from the samples menu in the demo page⁵.

```

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(?i1 ,?i2) AS ?two_guys) WHERE {
  ?f1 a ma:MediaFragment; ma:locator ?i1; dct:subject ?p1.
  ?p1 foaf:name "Lewis Jones".
  ?f2 a ma:MediaFragment; ma:locator ?i2; dct:subject ?p2.
  ?p2 foaf:name "Connor Macfarlane".
  FILTER mm:rightBeside(?i1 ,?i2)
  FILTER mm:temporalOverlaps(?i1 ,?i2)
}

```

Listing 1.1. A SPARQL-MM query

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.

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

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

References

1. Allen, J.F.: Maintaining knowledge about temporal intervals. *Commun. ACM* **26**(11), 832–843 (1983)
2. Bailer, W., et al.: Ontology for media resources 1.0. W3C recommendation, W3C, February 2012. <http://www.w3.org/TR/2012/REC-mediaont-10-20120209/>
3. Clementini, E., di Felice, P., van Oosterom, P.: A small set of formal topological relationships suitable for end-user interaction. In: Abel, D.J., Ooi, B.-C. (eds.) *SSD 1993*. LNCS, vol. 692, pp. 277–295. Springer, Heidelberg (1993)
4. Kurz, T., et al.: Semantic enhancement for media asset management systems. *Multimedia Tools Appl.* 1–27 (2012). doi:10.1007/s11042-012-1197-7
5. Lyndon J.B.N., et al.: Connectme: semantic tools for enriching online video with web content. In: *I-SEMANTICS (Posters & Demos)*. vol. 932, CEUR Workshop Proceedings, pp. 55–62 (2012)

⁵ <http://demos.mico-project.eu/sparql-mm/sparql-mm/demo/index.html>

6. Stolze, K.: SQL/MM spatial - the standard to manage spatial data in a relational database system. In: BTW, pp. 247–264 (2003)
7. Troncy, R., et al.: Media fragments URI 1.0 (basic). W3C recommendation, W3C, September 2012. <http://www.w3.org/TR/2012/REC-media-frags-20120925/>