

University of Passau
Department of Informatics and Mathematics
Chair of Distributed Information Systems

Doctoral Thesis

Adapting Semantic Web Information Retrieval to Multimedia

Dipl. Inf. Thomas Kurz

October, 2019

Advisor: Prof. Dr. Harald Kosch
Second advisor: Prof. Dr. Sören Auer



"Computer Science is no more about computers
than astronomy is about telescopes."

Edsger W. Dijkstra (1930-2002)

Für Katharina, Paul und Helene.

Abstract

The amount of audio, video and image data on the Web is immensely growing, which leads to data management problems based on the hidden character of Multimedia. Therefore the interlinking of semantic concepts and media data with the aim to bridge the gap between the Internet of documents and the Web of Data has become a common practice. However, the value of connecting media to its semantic meta data is limited due to lacking access methods and the absence of an adapted query language specialized for media assets and fragments. This thesis aims to extend the standard query language for the Semantic Web (SPARQL) with media specific concepts and functions. The main contributions of the work are an exhaustive survey on Multimedia query languages of the last 3 decades, the SPARQL extension specification itself and an approach for the efficient evaluation of the new query concepts. Additionally I elaborate and evaluate a meta data based media fragment similarity approach, which provides a basis for further language extensions.

Kurzzusammenfassung

Das Wachstum multimedialer Daten wie Audio, Video und Bilder war in den letzten Jahren immens. Das Besondere an dieser Art der Daten ist die versteckte Semantik, die sich nur schwer mit herkömmlichen Information Retrieval Funktionen verbinden lässt und dadurch zu Problemen im Management der Multimedia Daten führt. Konzepte des Semantic Web eignen sich allerdings sehr gut, diese Lücke zu schließen, was sich in vielen Szenarien bereits positiv etabliert hat. Nichtsdestotrotz fehlen mit geeigneten Zugriffsmethoden und einer adaptierten Anfragesprache wichtige Teile, um dieses Konzept der verlinkten Multimedia Daten abzurunden und voll in einem End-to-End Prozess zu verwenden. In dieser Arbeit stelle ich eine Erweiterung der Standard-Anfragesprache im Semantic Web (SPARQL) um multimedia-spezifische Funktionen vor. Der wissenschaftliche Beitrag lässt sich dabei in drei Teile gliedern: Ein umfassendes Survey zu Multimedia Anfragesprachen der letzten 30 Jahre, die Erweiterung von SPARQL inklusive einer geeigneten Methodik zur Anfrageoptimierung, sowie ein Ansatz zur fragment-basierten Ähnlichkeitsberechnung von Bildern mit zugehöriger Evaluierung.

Acknowledgement

This work is the result of a long journey. It would never had been possible without the substantial mental and professional support of many people. This is the place to say thanks - without you I'd never have made it.

I want to thank my supervisor Harald Kosch for his patience, his unbreakable faith in me and the friendly pushes in the right direction.

I am grateful for so many inspiring colleagues I was allowed to work with in the last decade at Salzburg Research, Redlink, the Mico project and the Chair of Distributed and Multimedia Information Systems at the University of Passau. They had a major influence on my work (in alphabetical order): Patrick Aichroth, Emanuel Berndl, Mario Döllner, Jakob Frank, Micheal Granitzer, Andreas Gruber, Werner Haring, Kai Schlegel, Christian Weigel, and Rupert Westenthaler.

In particular I want to thank Sebastian Schaffert who guided me in scientific work and taught me a so many technological skills, Sigi Reich for his continuous support during my work at Salzburg Research, and Florian Stegmaier for pushing me to actually start a doctoral thesis and the good time we had during basic and higher studies.

I want to thank my parents Albert and Rosemarie who enabled so many things in my life and provided me the opportunity to become happy in the areas of my interest. I thank my sister Christine, my grandparents, my aunts and uncles, my family and my family by marriage, and last not least my friends - you all give me an environment where I can always be who I am.

I dedicate this thesis to the most important thing in my life - my little family - Katharina, Paul and Helene. You are everything!

Contents

List of Tables	xiii
List of Figures	xvi
I Preface	1
1 Introduction	3
1.1 Motivation	3
1.2 Contribution	5
1.3 Overview	6
II Related Work	7
2 Linked Media	9
2.1 The Semantic Web	9
2.2 Semantic Web Query Languages	15
2.3 Extension-Mechanism of SPARQL	18
2.4 The Linked Data Movement	20
2.5 Media in the Web of Data	21
2.6 Conclusion	28
3 Multimedia Query Languages	29
3.1 Survey of Multimedia Query Languages	30
3.1.1 Historical Overview	30
3.1.2 Early works in the 80s	31
3.1.3 Extended works between 1990 and 2000	31
3.1.4 Works from 2000 until now	32
3.1.5 Detailed View on representatives	34
3.2 Requirements of Multimedia Query Languages	44
3.2.1 Preliminaries	45
3.2.2 General requirements of query languages	45
3.2.3 Specific requirements of Multimedia query languages	46
3.3 Conclusion	49
III Semantic Multimedia	51
4 Application Scenarios	53
4.1 Requirements gathering	53

4.2	Image Retrieval	54
4.3	Image/Video Retrieval	56
4.4	Conclusion	57
5	Basic Model for a Semantic Web Multimedia QL	59
5.1	Modeling Multimedia	60
5.2	SPARQL Algebra	65
5.2.1	SPARQL Abstract Query Syntax	65
5.2.2	SPARQL query string translation	67
5.3	Conclusion	69
6	Class and Property Model for Extensions	71
6.1	Design Principles	71
6.2	Class Model	71
6.3	Instant Model	75
6.4	Alignment to Existing Models	76
6.5	Excursion: Extending Media Fragments URIs	78
6.5.1	Media Fragment URI Extensions	78
6.5.2	Related approaches for Media Fragment URI extensions	81
6.5.3	Mapping Media Fragments URI Extensions to the Model	82
6.6	Conclusion	83
IV	Multimedia Extension for SPARQL	85
7	Sparql-MM Functions	87
7.1	Extension Functions	87
7.2	Spatial Relations, Aggregations and Properties	88
7.2.1	Topological Relations	88
7.2.2	Directional Relations	90
7.3	Temporal Relations, Aggregations and Properties	91
7.4	Spatio-Temporal Property and Function Specification	92
7.5	Conclusion	104
8	Optimization	107
8.1	SPARQL Filter Optimization	107
8.1.1	Spatio-temporal Indexes	107
8.1.2	SPARQL optimization approaches	110
8.1.3	Optimizing SPARQL	114
8.2	Considering Filters for SPARQL query optimization	116
8.2.1	Experimental proof of selectivity assumption	116
8.2.2	Filters and Edge costs	118
8.2.3	Query plan search	120
8.3	Conclusion	123

9 Evaluation	125
9.1 Example	125
9.1.1 Example: Translate SPARQL to ESG	125
9.1.2 Example: Calculate costs for nodes and vertices	126
9.1.3 Example: Find most cost efficient plan	127
9.2 Evaluation Environment	128
9.3 Results	133
9.4 Conclusion	144
V Semantic Multimedia Relations	145
10 Semantic Distance of Media Fragments	147
10.1 Semantic Distance	147
10.2 Spatio-temporal Fragment Distance	151
10.3 Semantic fragment similarity	154
10.4 Conclusion	156
11 Evaluation	157
11.1 Evaluation Environment	157
11.2 Results	158
11.3 Conclusion	161
VI Summary	163
12 Résumé	165
12.1 Conclusion	165
12.2 Further Work	167
A Directory for Publications	169
A.1 Books and Articles	169
A.2 Proceeding Papers	173
B Supplementary Material	175
B.1 Prefixes	175
B.2 Linked Media Fragment Ontology (LMO)	176
B.3 Categories in Optimization Evaluation	181
B.4 Optimized query plans	183
Bibliography	187

List of Tables

3.1	Query-by-example with WS-QBE: 1	41
3.2	Requirements for Multimedia Query Languages	50
5.1	Truth table for rightBeside function on rectangular image fragments	62
6.1	Mapping: Media Fragments URI	77
6.2	Mapping: Media Fragments URI Extension	82
7.1	SPARQL-MM Function Types	87
8.1	Results of the function selectivity experiment	118
9.1	Evaluation Results: Query 1	134
9.2	Evaluation Results: Query 2	135
9.3	Evaluation Results: Query 3	136
9.4	Evaluation Results: Query 4	137
9.5	Evaluation Results: Query 5	138
9.6	Evaluation Results: Query 6	139
9.7	Evaluation Results: Query 7	140
9.8	Evaluation Results: Query 8	141
9.9	Evaluation Results: Query 9	142
9.10	Evaluation Results: Query 10	143
B.1	Prefix-Table	175
B.2	Categories in Optimization Evaluation	181
B.3	Evaluation: Queryplan 1	183
B.4	Evaluation: Queryplan 2	183
B.5	Evaluation: Queryplan 3	183
B.6	Evaluation: Queryplan 4	184
B.7	Evaluation: Queryplan 5	184
B.8	Evaluation: Queryplan 6	184
B.9	Evaluation: Queryplan 7	185
B.10	Evaluation: Queryplan 8	185
B.11	Evaluation: Queryplan 9	185
B.12	Evaluation: Queryplan 10	186

List of Figures

2.1	Semantic Web Layer cake	9
2.2	SKOS example in SKOSJS	13
2.3	Descriptive example of a Media Fragment	25
2.4	Baseline Web Annotation Model	26
3.1	Multimedia Query Languages in the years 1980 to 2000	31
3.2	Multimedia Query Languages in the years 2001 to 2015	33
3.3	SQL/MM geometric type hierarchy	36
3.4	MQuery: visual query example	41
3.5	Query-by-example with WS-QBE: 2	41
3.6	MPQF Input Query Format	42
3.7	MPQF Output Query Format	43
4.1	Diagram of requirement gathering	54
5.1	Example for rectangular image fragments	62
5.2	Example for an animated video fragment	65
6.1	SPARQL-MM basic classes and relations	72
6.2	Sample object Circle	75
6.3	Sample object Interval	76
6.4	Sample animation	77
6.5	Fragment Extension: Shape	79
6.6	Fragment Extension: Transformation	80
6.7	Fragment Extension: Animated Transformation	81
7.1	Clementini-Matrix	88
7.2	Example for DE9im	88
7.3	Models of directional relations	90
7.4	Allen’s 13 basic temporal relations	91
8.1	R-tree example	108
8.2	Sample image: Alices Birthday 2012	110
8.3	Non-optimal evaluation tree	112
8.4	Hypothetical progression of join-tables sizes	113
9.1	Example ESG	126
9.2	Example result image for Query 5	130
9.3	Join Evaluation Graph: Query 1	134
9.4	Join Evaluation Graph: Query 2	135
9.5	Join Evaluation Graph: Query 3	136

9.6	Join Evaluation Graph: Query 4	137
9.7	Join Evaluation Graph: Query 5	138
9.8	Join Evaluation Graph: Query 6	139
9.9	Join Evaluation Graph: Query 7	140
9.10	Join Evaluation Graph: Query 8	141
9.11	Join Evaluation Graph: Query 9	142
9.12	Join Evaluation Graph: Query 10	143
10.1	Simplified ontology example graph	148
10.2	Implicite semantic term relationships in Word2Vec models	150
10.3	Media Similarity Evaluation: Example of an annotated image	152
10.4	Linked Media Fragment Distance: Idea	152
11.1	Test UI for Similarity Metrics Evaluation	159
11.2	Similarity Metrics Evaluation: Options selected in AVG	159
11.3	Similarity Metrics Evaluation: Sum of values per image	160
11.4	Similarity Metrics Evaluation: Weighted sum of values per image	160

Part I

Preface

Introduction

"A Little Semantics Goes a Long Way"¹

James Hendler

1.1 Motivation

During the last decade the amount of image and video content in the Web has increased rapidly. The main reasons for this trend are the rise of Web 2.0 with the associated tremendous growth of user generated content and the common accessibility to media production hardware like smart phones with integrated high-resolution cameras. In combination with video cutting freeware as well as free video streaming platforms, the production and distribution of Multimedia content is much cheaper and easier than it was a few years earlier. The lowering of barriers has enabled many people to engage in both, web-video/image production and consumption. This trend also happens in companies and institutions offering more and more commercially produced Multimedia content on the Web. Often they make use of special channels or platforms to strengthen their brand, offer company produced content to potential customers (e.g., using newsletter functions) and exclude undesirable information from those channels. There are two main issues that channel operators are concerned about: a) how to offer the right content to the right people and b) how to provide background information to consumers and thus increase the average duration of site visits while avoiding the propagation of incomplete or wrong information through any third parties.

In parallel to the Web 2.0 efforts, there is a research-led trend with the vision of a future Web in which information will no longer be confined to human understandable texts and media, but also be presented in machine-readable and machine-interpretable formats. This new Web will allow machines to understand the semantics of data and their relations, and so be able to (re)use and present it in a smarter way. Sir Tim Berners-Lee, the founder of the World Wide Web, called this vision the Semantic Web or the Web of Data [BLHL01]. In the last few years this vision has become more real because an increasing number of content providers has been publishing their data according to Semantic Web standards in order to open their data for further use. The current state of the so called Linked Open Data cloud is visualized in the Linking Open Data cloud diagram ². By September

¹Taken from <http://www.cs.rpi.edu/~hendler/LittleSemanticsWeb.html>

²LOD cloud: <http://richard.cyganiak.de/2007/10/lod/>

2011, the cloud contained 295 data sets, which are interlinked by around 504 million links [W3C07]. The crawl from 2014 discovered 1014 data sets³, and a current report mentions 1231 sets⁴, and the cloud is still growing. Even if the description of digital media resources with metadata properties has a long history in research and industry [D⁺11] Multimedia assets played a subsidiary role at the first steps of the Web of Data. In order to improve this situation, the W3C initiated the Video in the Web activity⁵. The associated Working Groups recommended a media-format independent standard for addressing media fragments on the Web using Uniform Resource Identifiers. This format supports particular name-value pairs, like (`t='start', 'end'`) for temporal and (`xywh='x', 'y', 'width', 'height'`) for regional fragments. A further group developed a common description practice for many different media objects and formats on the Web by providing an ontology [C⁺12] and API [BPL⁺14]. More complex ontologies that fulfill many higher-level requirements for media annotation like COMM⁶ (more or less a re-engineering of MPEG-7 using DOLCE), M3O⁷ or RICO are not widely accepted precisely because of their complexity [KGD⁺14], which is a big hurdle for Web users and developers. A model, which is not restricted to media annotation but about annotations on the Web in general, has been introduced as Open Annotation Data Model (OADM) [SCdS13] and turned into a W3C standard [CYS17] in 2017. It allows the creation of annotations that are easily shareable between platforms, while trying to satisfy complex requirements and being as easy as possible at the same time. Both, the Ontology of Media Resources and the Web Annotation Model, support Media Fragment UIRs for fragment identification.

Even if there are many approaches to publish interlinked media data, a well-suited solution for Multimedia retrieval in the Semantic Web is lacking. The de-facto standard query language for RDF (SPARQL) [HS13] allows expressing discrete queries across diverse data sources, where the data is represented as RDF. It includes features like basic conjunctive patterns, value filters, optional patterns, and pattern disjunction. SPARQL is extendable in many ways and thus allows to add functionality that goes beyond the specification of either the SPARQL query language or the SPARQL protocol.

³State of the LOD cloud 2014: <https://tinyurl.com/y2wj2j3o>

⁴State of the LOD cloud June 2018: <https://lod-cloud.net/>

⁵Video in the Web activity: <http://www.w3.org/2008/WebVideo/>

⁶COMM: <http://comm.semanticweb.org/>

⁷M3O: <http://m3o.semantic-multimedia.org/ontology/2009/09/16/>.

The main aims of my thesis are:

1. the definition of features for a Multimedia query language based on the analysis of historical query languages, its requirements and real world use cases, and
2. the identification of missing parts in the State of the Art of Multimedia and the Semantic Web, as well as
3. the proposal and evaluation of possible solutions to turn media items into full citizens of the Web of Data.

When I started this thesis there has been no extension that brings Multimedia specific features like spatio-temporal aspects or media similarity into SPARQL. Since then, there have been a few approaches to overcome this problem. In [SPM⁺16] the authors define an OWL based ontology for describing spatio-temporal relations and SPARQL based access to media. The authors of [FBH17] describe an ontology driven strategy to overcome the lacking, and a simple approach that enables temporal media fragment queries is described in [NW18]. The mentioned approaches substantially differ from my work in the basic approach (as they do not use SPARQL function extensions), feature completeness (as the works considers just smaller subsets of media related queries), and well defined query plan optimization (as the authors do not discuss this in detail).

1.2 Contribution

In this thesis techniques and methods are elaborated that integrates the two topics of Semantic Web and Multimedia information retrieval. The elaborations are tested within several real world scenarios in order to evaluate the theoretical achievements. The contribution of this work can be separated into three main pillars:

An exhaustive survey of Multimedia Query Languages

This survey contains an overview of 77 Multimedia query languages beginning from the 1980s until now. For every query language a short description is provided. In order to find meaningful clusters of languages that fulfill specific needs, the survey introduces a set of requirements for a) query languages in general (e.g. transitive closure) and b) Multimedia query languages (e.g. spatial operations) in particular. Additionally example queries for every requirement are provided, which allow a simple and exact evaluation of a language.

An extension of Semantic Web query language to Multimedia facilities

With SPARQL-MM I introduce a novel Multimedia query language by extending the de-facto standard language of the Semantic Web with Multimedia facilities. This includes spatial and temporal functions (relational, aggregational, accessor)

and is designed to support several fragment specification standards. The current implementation supports the widely used Multimedia Fragment URIs. Together with the extension I provide a solution for an efficient evaluation of Multimedia queries with SPARQL.

A novel approach for Similarity Measurement in Linked Media

As a third contribution, I propose a similarity metric for Linked Media (Multimedia embedded in a Linked Data environment). This approach is a proper basis for further extensions of SPARQL-MM to semantic image similarity features.

1.3 Overview

This work consists of six Parts. Part I contains an introduction to the topic as well as a summary of the contributions. In Part II the State of the Art in both, Linked Media and Multimedia query languages is described. To give a proper basement for Linked Media, I introduce Semantic Web technologies. This includes the basic model, retrieval and access techniques as well as their extendability. Furthermore I give an overview of the Linked Data movement and the role of Multimedia items in this environment. In order to give an overview on Multimedia retrieval and its requirements, a exhaustive survey of this topic together with a feature listing is outlined.

Part III starts with a description of application scenarios that again show the lackings in current technology regarding Semantic Multimedia. To allow a solid description of the scientific and technical contributions in this thesis, theoretical models for (annotated) Multimedia as well as for the de facto standard query language for the Semantic Web SPARQL are introduced. These are defined in different abstraction layers to cover all the necessary steps afterwards.

In Part IV I describe SPARQL-MM as a Multimedia extension for SPARQL. This includes spatial and temporal relations, aggregations and properties together with its theoretical grounding. As an efficient evaluation of the extensions is obligatory, various optimization steps are described and exhaustively evaluated.

Part V adds semantic Multimedia similarity to the picture by combining common semantic with Multimedia specific distances. The approach is evaluated in comparison to others using A/B testing. In Part IV I discuss the results of the former Chapters and give an outlook to future progression.

Part II

Related Work

Linked Media

"If the future Web will be able to fully leverage the scale and quality of online media, a Web scale layer of structured, interlinked media annotations is needed"

Lyndon Nixon [Nix13]

2.1 The Semantic Web

"In addition to the classic "Web of documents" W3C is helping to build a technology stack to support a "Web of data," the sort of data you find in databases. The ultimate goal of the Web of data is to enable computers to do more useful work and to develop systems that can support trusted interactions over the network. The term "Semantic Web" refers to W3C's vision of the Web of linked data. Semantic Web technologies enable people to create data stores on the Web, build vocabularies, and write rules for handling data. Linked data is empowered by technologies such as RDF, SPARQL, OWL, and SKOS."¹

W3C

In 2001, Tim Berners-Lee described a technology stack for his vision of a Web of Data [BLHL01]. Figure 2.1^{2 3} shows these so called 'Semantic Web Layer Cake'. The

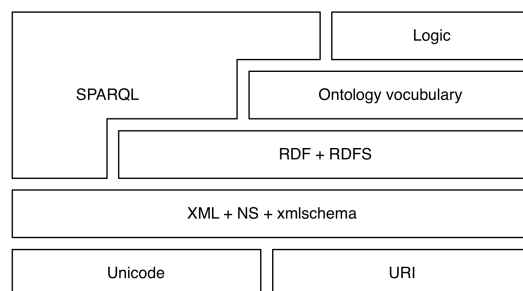


Figure 2.1: Semantic Web Layer cake

¹Taken from <http://www.w3.org/standards/semanticweb/>

²Marob1, Semantic Web Stack, May 2008, Creative Commons Attribution.

³Image URL: <http://en.wikipedia.org/wiki/File:Semantic-web-stack.png>

single layers represent classes of different abstraction, which build upon each other. The basis of this stack is the Unique Resource Identifier (URI) [BLFM05] which ensures the uniqueness of informational and non-informational resources, where informational resources are documents (like JPG files) and non-informational resources denote things in the real world (e.g., a person). The Resource Description Framework (RDF) [KC04, WLC14], the RDF Vocabulary Definition Language (RDFS) [BG04] and the Web Ontology Language (OWL)[MvH04] define a model for describing resources, documents and relations in between. RDF encodes data in the form of subject, predicate and object triples. The subject and object of a triple are both URIs that each identify a resource, or a URI and a string literal respectively. The predicate specifies how the subject and object are related and is also represented by a URI. To add meaningful triple relations, it is necessary to build vocabularies that are themselves expressed in RDF, using terms from RDFS and OWL. Based on these descriptions it is possible to query for data by using the SPARQL Protocol and RDF Query Language (SPARQL)[PS08]. In this Section I will give a deeper insight into the technologies of the Semantic Web. Furthermore I will outline how (Multi-) media data is currently considered in the Web of Data and emphasize what is the current stack lacking.

Resource Description Framework - RDF

The Resource Description Framework is a formal language for describing Web resources and their relationship to each other. It is recommended by the W3C [WLC14] in version 1.1 in 2014. RDF allows to specify logical statements by a set of triples (RDF graph), whereby a triple consists of subject, predicate and object. A triple is the simplest logical expression. An RDF graph is directed and describes a conjunction of triples. Nodes can be of type *IRI* (Internationalized Resource Identifier), Blank Node or RDF Literal. Vertices (relations between nodes) are defined by IRIs. An IRI can be node and vertex at the same time.

Definition 1 (Uniform Resource Identifier (URI)) *A Uniform Resource Identifier (URI) [BLFM05] is defined as ‘a compact sequence of characters that identifies an abstract or physical resource. ... The URI syntax defines a grammar that is a superset of all valid URIs.’ Each URI starts with a schema name whereby the schema definition specifies the syntax of the remaining URI parts. Examples for widely used schemas are HTTP [FGM⁺97] (e.g. `http://www.ietf.org/rfc/rfc2396.txt`) or URN [Moa97] (e.g. `urn:oasis:names:specification:docbook:dtd:xml:4.1.2`).*

Definition 2 (Internationalized Resource Identifier (IRI)) *An Internationalized Resource Identifier (IRI) [DS05] is defined ‘as a complement to the Uniform Resource Identifier (URI). An IRI is a sequence of characters from the Universal Character Set (Unicode/ISO 10646). A mapping from IRIs to URIs is defined, which means that IRIs can be used instead of URIs, where appropriate, to identify resources.’*

Definition 3 (RDF Literal) Let S be the set of all Unicode strings in normal form C like defined in [The14]. Let D be the set of all IRIs describing datatypes. Let N be the set of all non-empty language tags as defined in [PD09]. The set of RDF Literals is defined by $L = (S \times D \times N)$.

Definition 4 (Blank Node) Blank Nodes are local identifiers which are scoped to a specific RDF store and not portable. They are disjoint from IRIs and Literals. They are not part of the RDF abstract syntax, do not follow any specific schema and are dependent on concrete syntax or implementation. Blank Nodes can be replaced by IRIs in a Skolemisation process⁴ for the matter of independent identification.

Definition 5 (RDF Triple) Let I be the set of IRIs. Let L be the set of RDF Literals. Let B be the set of Blank Nodes. Let I, B and L pairwise disjoint sets. A triple t is a member of set T with:

$$T = (I \cup B) \times I \times (I \cup B \cup L)$$

Definition 6 (RDF Graph) The RDF Graph g is a directed, edgelabeled graph, which is defined by a set of triples $T^* \in T$, whereby T is the set of all triples.

Example 1 (RDF triple statements) Given the statement ‘Tom likes Paris, France’, it can be expressed as RDF Graph g by splitting it into triples (subject, predicate, object):

$$g = \{tom \times like \times paris, \\ paris \times part_of \times france, \\ tom \times label \times "Tom", \\ paris \times label \times "Paris" \\ france \times label \times "France"@en\},$$

with

$I = \{tom, like, paris, part_of, france, label\}$,

$L = \{"Tom", "Paris", "France"\}$. For the matter of simplicity Literals are reduced to strings.

In order to provide data in an interoperatable manner there is a need of a standardized serialization format for RDF Graphs. In the past decade there have been various recommendations for this purpose. The most widely used (and a W3C recommendation for primary usage) are Turtle [CP14] and RDF/XML [GS14]. In this thesis all RDF examples are serialized using Turtle format as it is compact, easy to read and quite close to the SPARQL syntax.

Example 2 (Turtle syntax) This Example shows how the RDF graph outlined in Example 1 can be written using Turtle syntax. For the matter of readability and

⁴Skolemisation: <http://www.w3.org/2011/rdf-wg/wiki/Skolemisation>

compactness turtle supports prefixes that are substituted while interpretation. Note that properties and entities are described using Uniform Resource Identifiers (URIs) [BLFM05].

Listing 2.1: A simple RDF example in turtle syntax

```
@prefix vocab: <urn:vocabulary> .  
  
<urn:inst:tom> vocab:like <urn:inst:paris> ;  
             vocab:label "Tom" .  
<urn:inst:paris> vocab:part_of <urn:inst:france> ;  
             voca:label "Praris" .  
<urn:inst:france> vocab:label "France"@en .
```

Vocabularies

"At times it may be important or valuable to ... enrich data with additional meaning, which allows more people (and more machines) to do more with the data."⁵

W3C

Data organization is a crucial part in the process of making data understandable and reusable for external consumers, both human and machine. Common vocabularies that define concepts and relationships therefore play a central role in Linked Data. The terms *vocabulary* and *ontology* are not really deferrable as they are both validly used in literature for the same things. Nevertheless it is common to use *vocabulary* for simple description schemes (e.g. modeled with RDF schema) and *ontology* for more complex ones (e.g. modeled with OWL, used in reasoning cases for data validation, inferencing etc.). In this thesis I consider the two terms as equivalent.

There are many vocabularies used in the area of Linked Data. This is due to the fact that data with different origins cannot always be aligned without losing information. In addition the Web of Data is not restricted to certain knowledge and not supervised by central instance (which correlates the basic idea of the Web). Thus data is heterogeneous and so are the used vocabularies. Nevertheless, as the Web of Data is meant to break up data borders it is recommended to use or at least derive from widely used vocabularies. In this Section I give a short overview on these well known representatives. I divide the Section in two parts, one describing data-modeling vocabularies (that can be seen as construction material for building specific vocabularies), and the other a bunch of domain vocabularies widely used in Linked Data sets.

⁵Taken from <http://www.w3.org/standards/semanticweb/>

Data-modeling Vocabularies:

RDF Schema (RDFS) [BG04] provides a classes and property system with the aim to enable the description of resource groups and its relations. This can be done by defining "domains" and "ranges" for properties and relations. This differs RDFS from other systems that follow an object oriented approach, where entities are defined by a set of properties they may have. Thus RDFS can be labeled as "property centric", which makes it easy to adapt schemas by extending it with new properties without re-defining the existing classes. As an example, one can define a property *creator* with the domain *MediaAsset* and the range *Person*. The two classes can be reused for other relations without changing their "signature". Thus RDFS enables the extension of existing instances while keeping their original form, which is one of the fundamental principles of the Web.

Simple Knowledge Organization System (SKOS) [MB09] defines a standard knowledge organization system using RDF. It is a basic system for taxonomies, classification schemes and thesauri. It is based on RDFS. Its main classes are *Collection*, *Concept* and *ConceptScheme*. SKOS supports broader-narrower relationships as well as matching definitions. In addition it supports multi-labeling of concepts by preferred and alternate labels. As an example, one can define *ConceptScheme:MediaItems* with a top-concept *Concept:MediaAsset*. More specific entities like *Concept:StillImage* or *Concept:MovingImage* can be related to it via broader/narrower relationships. Figure 2.2 shows the example within SKOSJS editor⁶.

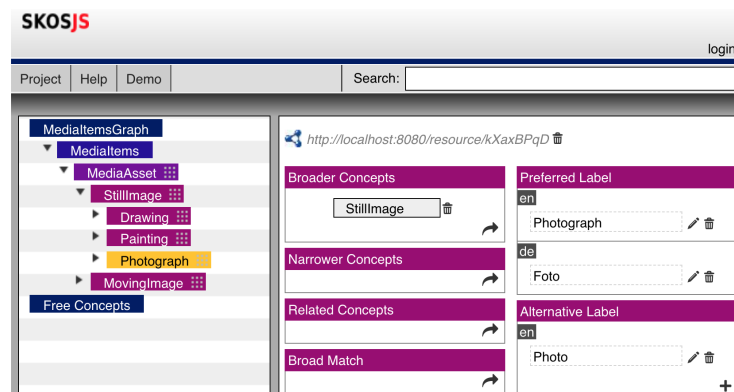


Figure 2.2: SKOS example in SKOSJS

Web Ontology Language (OWL) [MvH04] is a formal description language designed to define complex knowledge about things and their relationships. There are various manifestations of OWL, namely *OWL Lite*, *OWL DL* and *OWL Full*, which represent different complexity levels regarding predicate logic

⁶SKOSJS: <https://github.com/tkurz/skosjs>

and decidability. The language expands and restricts RDFS in classes, predicates and instances with the aim to allow decidability in an open world assumption. As I neither focus on OWL in this thesis nor use OWL for examples and evaluations a more detailed introduction is spared.

Domain Vocabularies:

There are many open vocabularies from various domains which should be reused in terms of interoperability. A list of currently 652 vocabularies (status 22.11.2018) can be found on Linked Open Vocabularies (LOV) registry⁷. In this Section I introduce only ones that are applied within my thesis in examples and evaluation scenarios. This set also overlaps the most widely used vocabularies.

Friend of a Friend (FOAF) [BM07] is a vocabulary to formal describe social networks by a) support an open standard for personal information and b) use RDF as linking mechanism between persons and other data. It is based on RDFS. In FOAF documents one can describe properties about a *Person* (e.g. *name*, *birthday*) and social media details (e.g. *yahooChatId*). By combining FOAF with other vocabulary it is possible to create exhaustive and universal interpretable descriptions of persons and its embeddings.

DCMI Metadata Terms (DCTERMS) [DCM08] is a vocabulary developed and maintained by the Dublin Core Metadata Initiative⁸. Its main aim is a common description of document resources. Thus DCTERMS includes generic properties in the area of document description, provenance, licensing, and versioning. The vocabulary is build on top of RDFS. The example in Listing 3 shows the description of a media resource with some description fields and the regarding author.

Example 3 (DCTERMS and FOAF) *In this example I use DCTERMS together with FOAF. As one can see the vocabulary approach is highly flexible and spares hard restrictions, which makes it a good fit for the open world of the Web.*

⁷LOV: <https://lov.linkeddata.es/dataset/lov/>

⁸DCMI: <http://dublincore.org/>

Listing 2.2: Example usage of DCTERMS and FOAF

```
@base <http://example.org/instance/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix dct: <http://purl.org/dc/terms/> .

<#image1> dct:title "Eiffel tower"@en ;
          dct:description "I love Paris"@en ;
          dct:creator [
            foaf:firstname "Tom" ;
            foaf:lastname "Doe"
          ] .
```

Ontology for Media Resources (MA) is a core vocabulary for the descriptions of media items. In addition the standard contains mappings to a set of common metadata formats (*exif*, *xmp*, *mpeg7* etc.). As this ontology is a central one for the topic of the thesis it is described in detail in Section 2.5.

2.2 Semantic Web Query Languages

"Query languages go hand-in-hand with databases. If the Semantic Web is viewed as a global database, then it is easy to understand why one would need a query language for that data."⁹

W3C

Depending on the underlying data format there are three main categories for Web query languages, as described in [BBFS05], namely XML Query and transformation languages, RDF query languages and Topic Maps query languages. In the case of Semantic Web as described above only the RDF ones are relevant. RDF query languages can be grouped mainly into seven families that differ in aspects like data model, expressivity, support for schema information, and kind of queries. The families are RQL [KAC⁺02], XPath-, XSLT-, and XQuery-based languages (e.g. [Sch04a]), Metalog [Mar04], reactive languages like [Pru04], deductive languages like [DSB⁺05], and, in the sphere of Linked Data, path traversal languages (like SQUIN [Har13] or LDPATH [SBK⁺12]), as well as the SPARQL family with its most common instance SPARQL (SPARQL query language for RDF) [HS13].

The SPARQL query language for RDF (SPARQL) is an extension of RDQL [Sea04] and provides Semantic Web developers with a powerful tool to extract information from large datasets. It is designed to meet the use cases and requirements identified by the RDF Data Access Working Group. SPARQL allows

⁹Taken from <http://www.w3.org/standards/semanticweb/>

expressing queries across diverse data sources, whether the data is represented as RDF. A formal description of SPARQL and its semantics by transform SPARQL into the relational algebra is described in [Cyg05] and [PAG09]. The query language is a syntactically-SQL-like language for querying RDF graphs via pattern matching. It includes features like basic conjunctive patterns, value filters, optional patterns, and pattern disjunction. In addition to the query language itself, the W3C recommendation also specifies a transfer protocol, a description for SPARQL services, and several query result formats. In the next Sections I describe SPARQL, whereby I especially highlight the extendability that is utilized within this thesis. The description is a summary of [ABS⁺15b].

SPARQL Protocol and RDF Query Language

SPARQL defines a standardization for RDF query syntax, semantics and protocol. It allows interoperability on the level of expressing rich queries on RDF datasets. The SPARQL Standard 1.1 Recommendation is separated in 11 parts ¹⁰, whereby the most important ones are the data retrieval language SPARQL 1.1 Query Language [HS13], the data manipulation language SPARQL 1.1 Update [GPP13], the definition of the results formats with their most important representative SPARQL Query Results XML Format (Second Edition) [Haw13], and SPARQL Protocol 1.1 [FWCT13], a means for conveying SPARQL queries and updates to a SPARQL processing service and returning the results via HTTP. In this Section I introduce the SPARQL 1.1 query language by highlighting some details.

SPARQL follows an SQL-like syntax but is based around graph pattern matching. Smaller patterns can be combined to complex graph patterns in various ways. The 4 main types of queries are **SELECT** (which returns a result table), **CONSTRUCT / DESCRIBE** (which returns RDF triples) and **ASK** (which returns a boolean value). Basically, a SPARQL query may consist of one or more of these clauses:

PREFIX allows to shorten URLs.

SELECT / CONSTRUCT / DESCRIBE / ASK is the projection clause. It identifies the return values, mostly variables that are bound within the where clause. Additionally, aggregation functions like **AVG**, **SUM**, etc. or custom ones are often used here.

FROM / FROM NAMED identifies the subgraph that is used to calculate the results. This enables SPARQL not just for querying triples but also quadruples.

WHERE is the selection clause. It identifies the values and bind the variables for the projection. Several constructs are allowed within the where clause, e.g. **OPTIONAL**, **UNION**, **FILTER**, negation, etc.

¹⁰SPARQL 1.1: <http://www.w3.org/TR/sparql11-overview/>

LIMIT / **OFFSET** / **ORDER BY** are sequence modifiers that can be used to change the quantity and the (per default random) order of a result set.

GROUP BY / **HAVING** are used to aggregate results, whereby **HAVING** is similar to **FLTER** in a **WHERE** clause.

For the matter of readability the list of clauses is not complete but includes the widely used ones. Like in SQL, in SPARQL 1.1 subqueries are allowed, too.

Variables in SPARQL are marked by the use of either "?" or "\$" followed by a string of characters; the "?" or "\$" is not part of the variable name. Variables are bound within the **WHERE** clause, the most important pattern, which is a kind of group graph pattern. SPARQL 1.1 defines some functions for filtering and aggregation (e.g. `regex`), which can be extended with custom operations.

Listing 2.3 shows a simple SPARQL query that selects first- and lastname of persons having a lastname that starts with 'A', ascendent ordered by their age. The selection as well as the filtering in SPARQL happens on the **WHERE** block. In lines 6-7 the properties of the person are bound. In line 11 a filter is used to narrow the results. The ordering is defined in line 13, the projection and thus the shape of the result in line 4.

Listing 2.3: A simple SPARQL query

```
1 PREFIX foaf: <http://xmlns.com/foaf/0.1/>
2 PREFIX sample: <http://example.org/sample/>
3
4 SELECT ?firstname ?lastname
5   WHERE {
6     ?p a foaf:Person.
7     ?p foaf:firstname ?firstname.
8     ?p foaf:lastname ?lastname.
9     ?p sample:age ?age.
10
11     FILTER regex( ?lastname , "^A" )
12   }
13 ORDER BY ASC( ?age )
```

Since version 1.1 SPARQL also takes into account the trends towards Linked Data and supports path expressions within patterns (whereby a triple pattern is also a special path expression of length 1). Listing 2.4 shows an example of a property path including an alternative path with an arbitrary length match. Such a fact is not expressible with simple triple patterns, which extend the expressiveness of SPARQL but dramatically decreases optimization facilities.

Listing 2.4: A SPARQL path expression

```
{ ?ancestor (ex:motherOf|ex:fatherOf)+ <#me> }
```

Path expressions also support some forms of limited inferences, for example for RDFS, all types and subtypes of a resource, like outlined in Listing 2.5.

Listing 2.5: Simple inference with SPARQL path expression

```
{ :thing rdf:type/rdfs:subClassOf* ?type }
```

2.3 Extension-Mechanism of SPARQL

SPARQL 1.1 can be extended beyond its specified feature set in various ways and thus be adapted for a broad field of use cases. These can be e.g. scenarios for geo-spatial search, fuzzy matching, document search or, as focused in this thesis, the integration of Multimedia specific functions. This section contains a short summary of four different kind of SPARQL feature range enlargements, namely SPARQL functions, functional predicates, meta extensions and syntax adaptations. This list is ordered by the complexity of the extension pattern.

SPARQL 1.1 Extension Functions

Custom SPARQL functions are the most standard conform way for feature adaption. It is specified in SPARQL 1.1 recommendation [HS13] as “*Extension Functions*”. They can be differentiated in two kinds, which are Filter Function and Producing Function. The first one are mainly used in SPARQL FILTER statements and produce a boolean value. The second one can be used in different statements like BIND or SELECT. Extension Functions are globally identified by IRIs and can be defined for a set of arguments (RDF terms). There is no standardized way to describe how the functions should be evaluated or what is the exact algorithm behind it. A common practice is to use script languages (e.g. javascript) to define this; the script code can thereby be found following concrete IRI paths [Wil07]. The advantage of this approach is that every SPARQL engine can load and execute the code. For more complex operations the script way is not practical; nevertheless it makes functions shareable and thus could be used as fallback mechanism for globally valid extension definitions. Figure 2.6 [ABS⁺15a] shows an example of a geo-spatial filter function.

Listing 2.6: SPARQL Filter Function example

```
FILTER (custom:geoDistance(?placeA,?placeB) < 10)
```


Function Predicates Extension

The approach of functional predicates (also known as magic predicates or property functions) uses triple patterns in order to describe binary functional relations between two RDF terms. The relation itself thereby is the predicate IRI of a pattern. At evaluation time the predicate is replaced by the corresponding (stored) function. Even if it is not covered by the official W3C recommendation, this kind of feature adaption is quite wide-spread, because it does not break the SPARQL grammar and thus can be parsed by every standard parser. A disadvantage of functional predicates is that their integration in existing evaluators and optimizers is a major effort. In addition the approach is limited to binary functions, whereby this issue is often overcome by using RDF lists. Listing 2.7¹¹ shows a full-text search extension implemented as functional predicate which is supported by the RDF4J¹² SPARQL engine.

Listing 2.7: SPARQL Function Predicates example

```
?subj search:matches [  
  search:query "search terms...";  
  search:property my:property;  
  search:score ?score;  
  search:snippet ?snippet  
] .
```

Meta Extension

Meta extensions represent a specific kind of functional predicates, whereby the predicates itself are replaced by standalone SPARQL queries. The advantage of this approach is that there is no need for specific implementation of the function. It hides the complexity of the underlying SPARQL query and lower the barrier to SPARQL for non-experts. Furthermore the result can be generated using the same SPARQL evaluator. A proper basis for the description of such extensions is the SPARQL Inferencing Notation SPIN¹³ (a W3C member submission from 2011)¹⁴, and SPIN Functions in particular. As described in [ABS⁺15a], SPIN Magic Properties are "boxed" queries, which declare new SPARQL functions that determine bindings of the subject and object variables. As the definition of SPIN is rather complex and not part of further investigation within the thesis a more detailed description is spared here.

Language Syntax Extension

Extending the syntax of a query language provides maximum flexibility when aiming very specific use cases. Such language extensions extensions modify the

¹¹SPARQL extension inventory: <https://tinyurl.com/yyvgalk2>

¹²RDF4J: <https://rdf4j.eclipse.org/>

¹³SPIN: <https://spinrdf.org/>

¹⁴SPIN W3C: <https://www.w3.org/Submission/spin-overview/>

basic grammar and thus allow to introduce new keywords and operators. But this approach has the disadvantage, that the queries are not compatible with any existing query evaluator. Prominent SPARQL language extensions are f-SPARQL [CMY10] (an adaption to fuzzy set theory) and SPARQL-ST [PJS11] (an extension to complex geospatial objects and filters). An example of SPARQL-ST is outlined in Listing 2.8 [ABS⁺15a]. It shows the definition of a complex spatial area which is used as SPATIAL FILTER in order to ensure that a given point falls within a polygon. It is obvious that the query substantially differs from classical SPARQL and thus builds a major barrier even for users that are familiar with the basic grammar.

Listing 2.8: SPARQL Syntax Extensions example

```
SELECT * WHERE {
  ?c stt:located_at %g.
  SPATIAL FILTER (inside(%g, GEOM(POLYGON ((
    -75.14 40.88, -70.77 40.88, -70.77 42.35,
    -70.77 42.35, -75.14 42.35,
    -75.14 42.35, -75.14 40.88))))))
}
```

Further examples regarding SPARQL extensions can be found on the SPARQL extension inventory¹⁵.

2.4 The Linked Data Movement

"The Semantic Web is a Web of data - of dates and titles and part numbers and chemical properties and any other data one might conceive of. RDF provides the foundation for publishing and linking your data."¹⁶
W3C

To bootstrap the idea of the Semantic Web, Tim Berners-Lee presented some design issues in 2006 [BL06] that outlined a best practice for exposing, sharing, and connecting pieces of data, information, and knowledge:

1. Use URIs as names for things;
2. Use HTTP URIs so that people can look up those names;
3. When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL);
4. Include links to other URIs so that they can discover more things.

¹⁵Same link as in footnote 11

¹⁶Taken from <http://www.w3.org/standards/semanticweb/>

Following these four principles it is possible to open formerly closed data silos to the Web, present them in a well-defined universal (and thus machine-interpretable) structure and interconnect different datasets in a simple manner. In 2007 the Linking Open Data (LOD) community project was initiated in the W3C [W3C07]. Its goal was to implement the Semantic Web idea by publishing and interlinking datasets following the given design principles. As such, LOD builds a widespread information pool for various linked media issues.

In 2011 a collaboration of major search providers (namely Google, Yahoo and Bing) provided schema.org¹⁷, a collection of schemas for the semantic markup of webpages. This approach is quite similar to Linked Data but does not use RDF and does not allow the use of domain specific ontologies. To bridge the gap between schema.org and Linked Data the Linked Data community has introduced schema.rdfs.org¹⁸ as a complementary effort. Since 2014 the RDF format of schema.org is managed again from the founders, which is an indicator for the increasing acceptance rate of Semantic Web technologies in the industry.

2.5 Media in the Web of Data

The vision of *Linked Media* is described in [Nix13]. The authors propose to follow the Linked Data principles to publish metadata about media resources which can then be interlinked on the Web. In the recent past there were some first approaches that follows this vision like [NBB⁺12], [NMT14] or [FSK15]. In this Section I give an overview on existing standards and techniques for Linked Media, which is an updated summary of the descriptions in [KGD⁺14].

Ontology for Media Resources

The description of digital media resources with metadata properties has a long history in research and industry [D⁺11]. Over the years many standards came up, differing in complexity and completeness, which led to interoperability issues in search, retrieval and annotation. To address this problem, the W3C launched the Media Annotation Working Group, which aims to improve interoperability between Multimedia metadata formats on the Web. They listed relevant formats in the group report, including basic standards like Exchangeable Image File Format (EXIF[Tec02]), Extensible Metadata Platform (XMP [Int05]) or Dublin Core (DC [DCM08]) as well as higher-level description formats like MPEG-7(ISO/IEC 15938). The group analyzed 18 Multimedia metadata formats and 6 container formats and selected a subset of 28 properties as greatest common denominator, making up a core ontology for Multimedia metadata [C⁺12]. Within the recommendation they also developed a mapping table for all standards included and a client-side API [SLPB] to access this metadata information.

¹⁷schema.org: <http://schema.org/>

¹⁸schema.rdfs.org: <http://schema.rdfs.org>

This base properties of the ontology are ones that the majority of the matching vocabularies support. They are split into groups namely:

Identification = {title, hasLanguage, locator}.

Creation = {hasContributor, hasCreator, date, hasRelatedLocation¹⁹}

Content description = {description, hasKeyword, hasGenre, hasRating¹⁹}

Relational = {hasRelatedResource, isMemberOf}

Rights = {copyright, isCopyrightedBy, hasPolicy}

Distribution = {hasPublisher, hasTargetAudience, hasClassification, hasClassificationSystem}

Fragment = {hasFragment, hasNamedFragment}

Technical Properties = {frameWidth, frameHeight, frameSizeUnit, hasCompression, duration, hasFormat, samplingRate, frameRate, averageBitRate, numberOfTracks}

Together with the properties the ontology defines a set of media related classes, which are MediaResource, MediaFragment, Image, Track, AudioTrack, VideoTrack, DataTrack Rating, Agent, Person, TargetAudience, Collection, Location, and Organisation.

With the ontology it is straight forward to describe e.g. an image as outlined in Listing 2.9.

Listing 2.9: Media Ontology example

```

1 @prefix      <http://example.org/instance/> .
2 @prefix ma:  <http://www.w3.org/ns/ma-ont#> .
3 @prefix foaf: <http://xmlns.com/foaf/0.1/> .
4
5 :img1 a ma:Image ;
6     ma:title "Paris" ;
7     ma:description "Me in front of the Eifel Tower"@en ;
8     ma:creator [
9         foaf:name "Tom" .
10    ] .

```

In line 5 the resource is defined as `Image`. In lines 6 and 7 two descriptive properties are used. In the lines 8-10 the example shows how to link to resources described in different schemes, here a creator specified using FOAF.

¹⁹Additional properties are speared here

As the core ontology was initiated to tear down the walls between several metadata standards, properties are extremely limited. To achieve a further more accurate description like how a media object is composed of its parts and what the parts represent, there is a need for more fine grained ontologies like COMM [ATSH09](more or less a re-engineering of MPEG-7 using DOLCE), M3O [SS10] or RICO [BS09], which is a conceptual model and a set of ontologies to mark up Multimedia content embedded in webpages. Even though these higher-level ontologies fulfill many requirements for media annotation, they are not widely accepted precisely because of their complexity, which is a big hurdle for Web users. This list of annotation and metadata models is not complete, but gives an overview of the most important representatives for our purpose.

Publishing Multimedia data and metadata on the Web in a standardized way is a basic task of Linked Media. For the Web of non-Multimedia data, there are several interlinking frameworks trying to detect related and linked resources in different datasets. In [SE09] several frameworks are compared concerning their functionalities. Because the common interlinking methods are used on resources dominated by text, they are commonly not sufficient for standalone Multimedia data. Hence, there are approaches which use media surrounding meta-data [Ste10], e.g., the title of the page, descriptions above or beneath, etc. Also there are aims to aligning Multimedia and events [FTH⁺10] and some other special use cases like the linking of image libraries and semantic resources [HSWW03]. Altogether the Multimedia interlinking vision as described in [BH08], [HTRB09] and [Nix13] is far from being universally implemented.

Semantic Media Annotation

According to the Open Annotation Collaboration²⁰ (a precursor of the W3C Web annotation working group²¹, an annotation associates "one piece of information with one or more other pieces of information". The act of annotating is therefore considered as "a pervasive activity shared by all humanity across all walks of life" [SvdS11]. Consequently a Web annotation associates a Web resource (typically a webpage, a video, an image, etc.) with one or more other Web resources.

In the broadcasting and media production domain, the annotation of media assets is one of the core processes: Annotation allows "extra information to be associated with any existing process artifact and often denotes the step of adding metadata by a single human user to facilitate search" (in [HuON⁺08] the term "process artifact" denotes any type of resource relevant in the annotation process, i.e., a media asset). Usually, when media resources (e.g., video and audio clips, images and photos) are annotated, automatically generated feature extraction approaches are combined with human annotation. Therefore media annotations quite often take over the role of bridging the "Semantic Gap"[SWS⁺00] between the results of low level

²⁰Open Annotation Collaboration (OAC): <http://www.openannotation.org/>

²¹W3C Web annotation working group: <https://www.w3.org/annotation/>

feature extraction and rich semantic meta-information. Low-level features are e.g., the duration of a video or its frame rate, face detection, speaker recognition, shot detection, etc. The identification of objects and persons depicted in a video clip, a summarization of the plot of the video or the provision of background information about the production of the media asset are labeled as higher level information. Semantic media annotation goes beyond traditional media annotation by associating media assets or media fragments with semantically well-defined concepts which are described in a machine readable form (independent of how these semantics are defined). Since in the Web of Data such concepts are embedded in a graph of relations, additional information can be obtained by traversing the edges of the graph. For example, if a media fragment is annotated with the concept identifying the activity of *trip* and the graph includes an edge identifying *trip* as a kind of *journey*, then the media fragment can be said to deal with *journey*.

Media Fragments URI

Linking media does not always involve only the whole resource, but also content fragments. Depending on their media type these fragments can have different manifestations, e.g., a rectangle-segment of an image, a time-slot or region in a video or even a combination of both. There are different approaches how such fragments can be handled for integration in the Web of Data. At the moment one of the most popular is the W3C fragments 1.0 recommendation [TDMP12]. It allows the identification of different kinds of media fragments using URL hash codes. Media fragments support addressing the media along different dimensions:

temporal denotes a specific time range in the original media, such as "starting at second 10, continuing until second 20".

spatial denotes a specific range of pixels in the original media, such as "a rectangle with size (100,100) with its top-left at coordinate (10,10)".

Note that in an extended version of the recommendation there are two more dimensions, namely **track** and **id**, but I omit them because they are just weak specified and not used within my thesis.

Temporal and spatial dimensions can be represented following the standard by using fragment parameters, whereby:

temporal can be defined by parameter **t** and is specified as an interval with a begin time and an end time (**t=1,5**)

spatial can be defined by parameter **xywh** and is specified by a quadrupel representing the *horizontal offset*, the *vertical offset*, *with*, and *height* (**xywh=100,100,150,200**). The offsets are bound to the neutral point on the top-left corner of an image.

Based on this specification I can extend our example for media annotation in Listing 2.9 to spatial fragments. The result is listed in Listing 2.10 and outlines in figure 2.3. As one can see, `:img1` is related to a fragment resource which is further described using *DCTM*.

Listing 2.10: Media Ontology example including Media Fragments

```

1 @prefix      <http://example.org/instance/> .
2 @prefix ma:  <http://www.w3.org/ns/ma-ont#> .
3 @prefix foaf: <http://xmlns.com/foaf/0.1/> .
4 @prefix dct: <http://purl.org/dc/terms> .
5
6 :img1 a ma:Image ;
7 ma:title "Paris" ;
8 ma:description "Me in front of the Eifel Tower"@en ;
9 ma:creator :person1 ;
10 ma:hasFragment :frag1#xywh=100,100,150,200 .
11
12 :frag1#xywh=100,100,150,200 dct:subject :person1 .
13
14 :person1 foaf:name "Tom" .

```

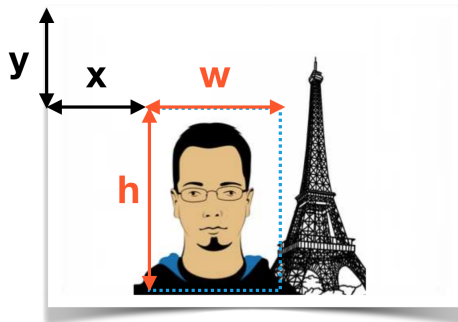


Figure 2.3: Descriptive example of a media fragment

Note, that the fragment resource identifier uses a media fragment and thus includes the information about section size and positioning by itself. This is on the one hand very convenient, as the Media Fragment can be directly interpreted e.g. from Web browsers and do automatic clipping when using it e.g. in a `src` attribute of an html `img` tag. On the other hand using the fragmented resources as a subject of other relations reduces the flexibility when changing fragment-values later. This fact leads sometimes to the need of a more complex annotation model like the Web Annotation Data Model that I am going to describe in the next Section.

Web Annotation Data Model

The Open Annotation Collaboration (OAC), a collaboration of the universities of Illinois, Maryland, Queensland, and the Los Alamos National Laboratory, started its work in 2009 with the aim to provide a resource-centric interoperable annotation environment. In its first phase, OAC focused on the development of a foundational data model and ontology for interoperable scholarly annotation, while in second phase concrete, collaborative small-scale demonstration projects are executed. This phase ended in 2013 and built the basis for the W3C Web Annotation Working Group²², which published three W3C recommendations including the one for the Web Annotation Data Model (WADM).

In the resource-centric baseline WADM, an annotation is a Web resource identified by an HTTP URI that describes an association created between a *body resource* and a *target resource*. The body must be somehow "about" the target for it to be considered the body of an annotation. This model follows the same basic structure as that of W3C Annotea [KK01].

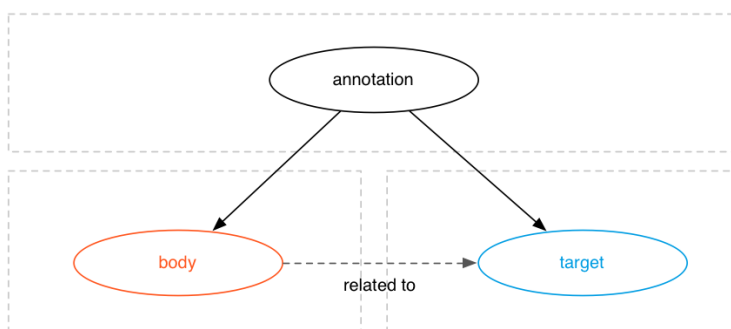


Figure 2.4: Baseline Web Annotation Data Model

In the baseline WADM outlined in Figure 2.4²³, both, the body and the target of the annotation, are identified by an URI and identify any resource on the Web with a representation in any format or language. Resources without representation identify abstract resources that denote a concept.

The model supports the usage of selectors in order to refer to parts of resources as the Target. In the WADM a part of a resource is called segment. A selector can be used to identify the segment from within the resource. Due to the diversity of resources there is a list of 9 selectors supported. The list includes selectors for text, xml, bytestream, etc. As this thesis is about video and image assets the one to mention here are the fragment selector and the SVG selector. The fragment selector is the most widely used selector for parts of resources represented by IRIs. To be clear how to interpret the value of a fragment the selector my refer

²²Web Annotation Working Group: <https://www.w3.org/annotation/>

²³Image taken from <https://www.w3.org/TR/annotation-model/>

to specification (e.g. *namedSection* of HTML [SB02], *mediaFragments* of Image, Video, Audio [TMPD11], etc.). An example describing the occurrence of a person within an image using a media fragment selector in the WADM is outlined in 2.11.

Listing 2.11: Media fragments within the WADM

```

1 @prefix      <http://example.org/instance/> .
2 @prefix oa:  <http://www.w3.org/ns/oa#> .
3 @prefix dct: <http://purl.org/dc/terms> .
4 @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
5
6 :annotation a oa:Annotation ;
7             oa:hasBody :body ;
8             oa:hasTarget :person .
9
10 :body      oa:hasSource :image ;
11           oa:hasSelector :fragment .
12
13 :fragment  a oa:FragmentSelector ;
14           dct:conformsTo "http://www.w3.org/TR/media-fragments/" ;
15           rdf:value "xywh=10,50,100,100" .

```

The SVG selector allows to describe an area using the Scalable Vector Graphics standard (SVG) [DDG⁺11]. In comparison to the current media fragment standard it allows to use more complex shapes, like circle or polygon for segment specification. The selector can be provided as embedded string or external file. To wrap it up, the WADM is a use-case agnostic data model that allows to annotate (parts of) various Web resources with (parts of) (other) Web resources, while being flexible, adaptable and extendable.

Media Annotation Frameworks

On the technical level, there are several software frameworks for semantic annotation. At [UCI⁺06] the authors provide a review of existing annotation frameworks, concluding that there are many systems which provide some of the requirements. But that fully integrated environments are some way off. The technical challenges include the support of Multimedia document formats, the ability to address issues of trust, provenance and access rights, as well as the resolution of storage problems. The W3C Annotea project [KK01], with its emphasis on collaboration, has influenced the development of a number of systems with good user interfaces that are well suited to distribute knowledge sharing. CREAM [HSS03], with its greater emphasis on the deep Web and the annotation of legacy resources, has pushed the development of annotation systems more aimed towards corporate knowledge management.

B. Haslhofer et al. identify a set of annotation requirements that is described in [HJK⁺09]. Several annotation systems are evaluated with respect to these requirements, including e.g., MADCOW [BLL⁺06], Vannotea [SHG⁺06], Multiva-

lent Annotations [PW97] or Debora [NPD⁺00]. B. Haslhofer and his team also investigated requirements that go beyond the State of the Art including the use of a uniform annotation model, uniform fragment identification and the integration with Web architectures. Consequently, the LEMO Annotation Framework is proposed as an approach to, according to an initial evaluation, provide a solution to a wide set of annotation requirements [HJK⁺09], arguing for a linkable (L), extensible (E), Multimedia-enabled (M), open and interoperable (O) architecture.

2.6 Conclusion

In this Section I gave a short introduction into the Semantic Web, also know as Web of Data. This included a definition of RDF graphs, triples statements and syntax. Furthermore I explained vocabularies for both, data-modeling and domain-specific ones. I gave an overview on Semantic Web query languages with a focus on SPARQL and its extendability. As a third step I introduced Linked Data and explained the role and status of media in the Web of Data. It becomes clear that media is already well-received there in cases of description and modeling. There are many media annotations approaches that are all suitable for their use cases. With the Media Fragment URI specification a resource centered approach for fragment descriptions has been established with fits seamless into the current Web ecosystem. So a very fine grained description of media is possible. Obviously the next step would be the re-usage of this well described media. This includes also to manipulate, clip, or merge media items and fragments in order to get new assets that fit specific information needs. To get a clear picture of these requirements I will survey Multimedia specific query languages, including its special features and outline how the Semantic Web stack can be adapted to support semantic media querying.

Multimedia Query Languages

"..leading the user to those documents that will best enable him/her to satisfy his/her need for information."

Stephen E. Robertson [Rob81]

One of the basic functions of a Database Management System is the efficient retrieval of stored data. The special needs of such a retrieval are strongly dependent on a) the stored data (and its underlying representation) and b) the specific use case. In my thesis I focus on a the Web of Data as a global Multimedia store as described in Section 2.5. Therefore the retrieval mechanism will be a mixture of classical Multimedia functionalities and Semantic Web related data querying. Following the standard definition for Information Retrieval in [MR09] I define Multimedia retrieval for media in context as follows:

Multimedia Retrieval on the Web of Data is finding (fragments of) resources of an unstructured nature (text, image, video, etc.) that satisfy an information need.

whereby:

Web of Data means a dataspace of resources, which are represented in interchangeable, common formats, and interconnected by named links. Thus, the Web of Data is an exchange medium for data as well as documents, like described in the vision of the related W3C Data Activity group [W3C13]. The terms *Web of Data* and *Semantic Web* are used as synonyms in this document.

finding means providing a subset of Web resources that meets someones expectations and is human-manageable in presentation form and amount (e.g. ordered list, collage etc.). This task includes the support of suitable ranking methods as well as pre-processing methods from data mining (e.g. clustering).

resources means in this context all things that are addressable via common Web standards. For a seamless integration of Linked Data principles [BL06], information resources (metadata) must be accessible via HTTP protocol; non-information resources (video etc.) may use different (more suitable) protocols. In addition, the fragmentation of resources requires a suitable representation format, e.g. like the Media Fragments URI specification [TDMP12] described in Section 2.5.

unstructured nature means that the resource is not interpretable per se but must be interpreted by experts or specialized machines to extract common understandable structure and features. This task is well supported for texts (e.g. Named Entity Recognition and disambiguation [RT12]) but is resource intensive for Multimedia content. Due to the latest progress in cloud computing (e.g. map-reduce programming model [CLH⁺14]), the decreasing costs and dynamic accessibility of hardware, and the commodity of information extraction tasks provided as Software as a Service, Multimedia analysis is also supported for *big content* and not just affordable for big companies.

information need means an abstract description of the expected subset or list. The more exact the information need is defined the more exact the presented set fits the expected results. The query language can be seen as an instrument for formalizing this need. It is an interface between user needs and the (mostly abstract) Multimedia data and metadata storage layer. The more the language fits use case specific needs, the more adequate it is for the use case.

In this Chapter I give a generic overview on both areas with a special focus on the different kinds of Multimedia retrieval functionalities and the extendability of the most common query language in the Semantic Web called SPARQL. The content of this Chapter includes summarized parts of [ABS⁺15b].

3.1 Survey of Multimedia Query Languages

The current landscape for Multimedia query languages is very broad and includes many different approaches. In this Section I give an overview of the chronological sequence of the investigated query languages in history, which shows some trends due to historical influences. In order to make the topic better understandable, I introduce representatives of different language categories in detail. Based on the historical survey an exhaustive set of requirements for query languages in general and Multimedia query languages in particular is defined. This set accommodates the special needs of users (e.g., in terms of expressiveness or easy to use and understandability) as well as exigencies of the underlying evaluation process in order to have a sound and formal model (e.g., relational completeness or safety) and give the basement for the specific requirement specification in Chapter 4.

3.1.1 Historical Overview

The access to and retrieval of Multimedia data has been the topic of many research projects and articles over more than 40 years. For instance, the activities in storing and retrieving images within databases can be traced back to the late 1970s where first conference contributions (e.g., Data Base Techniques for Pictorial Applications, 1979) introduced the use of relational databases for images by [CF80b]. In general, these early works focused on the annotation and retrieval of images by textual

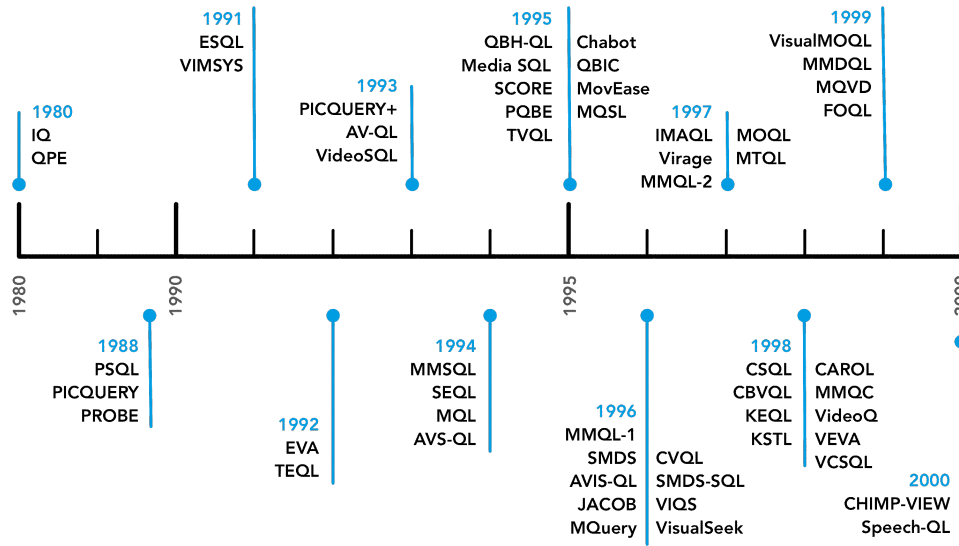


Figure 3.1: Multimedia Query Languages in the years 1980 to 2000

information. For this purpose, the images were described by keywords or textual descriptions and common relational database technologies and their text-based retrieval approaches were used for searching within the pool of annotated images. A substantial survey for text-based image retrieval can be found in [TY84].

In this Chapter, I list and summarize this group in order to have a grounding for requirement definitions described in 3.2. The observed Multimedia query languages are arranged temporally based on their appearance.

3.1.2 Early works in the 80s

Related to Multimedia query languages, the first work in this direction was the *Image Query Language* (IQL) by [CF80b] that focused on retrieval and manipulation of images on a file system. In parallel *Query By Pictorial Example* (QPE) [CF80a] has been originated, which was mainly build for the retrieval in Geographic Information Systems (GIS). This pioneers were followed by a longer period of silence and has been reestablished in 1988 by the languages PICQUERY [JC88], PSQL [RFS88] and PROBE [OM88], which also focused on GIS data and based on known paradigms like relational or object oriented database structures. It has been observable that the early works where mainly focused on spatial data and their appropriate operations.

3.1.3 Extended works between 1990 and 2000

In the 90's the amount of Multimedia query languages literally exploded, which becomes obvious from the timeline of the observed query languages from 1980 until

2000 in Figure 3.1.

The well known and widely used database paradigms are the basis for many query languages also in the 90's. Whereby relational approaches like ESQL [AB91] or MQSL [HK96] utilized the relations for exploiting semantics, object oriented approaches like VIMSYS [GWJ91], EVA [GD92] and MQL [KT94] focused on pattern matching and similarity search. A new aim that came up is the investigation of time series of images, especially in the medicine sector. This lead to languages like TEQL [CITB92], PICQUERY+ [CIB+93] and SEQL [CIT94].

Additional to image, in the mid 90's the era of video retrieval has been originated. It started with exclusive video retrieval query languages like VideoSQL [OT93], AVS-QL[WDG94] or CVQL [KC96] and continued to so calls multi-modal query languages. These try to combine several media types for both query and retrieval types. Examples for such kind of QLs are AV-QL[LG93] and Media SQL [LC95]. At the end of the 90s the multi-modal trend lead to mighty languages e.g. KEQL [CHIT98] and IMAQL [KCCC97] that have been very broad in its application scenarios. A smaller sector focused at the same time on pure audio retrieval, like e.g. QBH-QL [GLCS95].

The 90s have been also the decade of experimental database forms. Therefore it's no wonder that also in this niches Multimedia query languages had been based. There were functional QLs like MMQL [ATS96], logical ones like SMDS [MS96] or CSQL [LC98], rule bases like MQVD [DHK99] and very formal ones like VEVA [GD98]. In parallel to these experiments the first Multimedia query language found its way into the commercial product Virage [BFG+96].

Additionally to appearance, I could also determine some new trends and streams in query formulation and result interaction that have been introduced like relevance feedback (e.g in SCORE [ATY+95]) or fuzzy matching, like in MMQL [ATS96] and FOQL [NRT99]. Also new paradigms appeared like Query-By-Sketch (VideoQ [CCM+97]) or Query-By-Humming (QBH-QL [GLCS95]).

The second halve of the 90s is the era of successor languages that added additional functionality to the basic language. This involves extensions like adding graphical user interfaces or graphical query extensions (like in VisualMOQL [OÖX+99] or VCSQL [LC98]), additional functionality (e.g. PQBE [PS95]), or even the extension to other domains (like KSTL [CHCT98], VCSQL [LC98]).

3.1.4 Works from 2000 until now

After the millennium the engagement regarding Multimedia information retrieval did not decrease. But the rise of the Internet and especially the tremendous growth of Multimedia data within the Social Web changes the requirements. In this Section

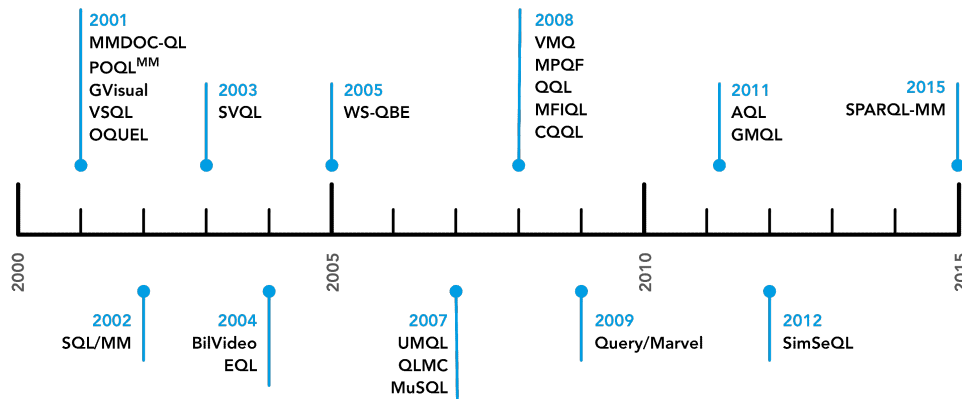


Figure 3.2: Multimedia Query Languages in the years 2001 to 2015

I list the Multimedia query languages from beginning of the century until today, a timeline can be found at 3.2.

Like in the 90's the trend to use various paradigms for Multimedia query languages continues after the millennium to. So there occurred logical Qls like MMDOC-QL [LCH01b], object oriented like POQL^{MM} [HR01] and SQL related, e.g. SQL/MM [ME01b] or SimSeQL [BBZ12].

Due to the advent of XML [BPS97] and its usage for metadata description many Multimedia Qls followed now this data format. This became even more prominent after the standardization of MPEG-7 [MKP02]. Prominent examples for these are SVQL [FKC03, FLR04] (which is a derivation of XQuery [B⁺07]), QLMC [MMSS07] and MPQF [DTG⁺08b], which were not bound to XML metadata but used it for specifying both, input and output parameters.

One major endeavour of the Qls that came up in the first decade of this century was to make Multimedia querying a commodity and thus accessible and manageable even by non-experts. As the topic is complex this lead to a bunch of visual query languages that support spatio-temporal queries (e.g. CHIMP/VIEW [CLS00]), trajectory like VSQL [CHL01] or even very complex Query-By-Example like WS-QBE [SSH05a].

Like in the 90's the developers of Multimedia Qls stayed eager to try out new things. This lead to languages like QQL [Sch08], which was a complete new approach based on quantum logic, or QueryMarvel [JS09], which followed the methodology of comic strips.

A new trend that started after the millennium and continued is the usage of ontologies to describe semantics that are hidden in Multimedia data (e.g. a

specific person within a video doing a specific thing). This led to languages like OQUEL [TS01], the integration of Semantic Web technologies in languages (e.g. in MPQF [DTG⁺08b]) or the extension of existing query languages for semi-structured (semantic) data with Multimedia facilities, like SPARQL-MM [KSK15].

Many languages that I consider in the survey are multi-modal. A new trend that had been there since the early 80's but became suitable for the mass in the 10's is 3 dimensional data. Therefore it is natural that there arose also query languages, like GMQL [WXZ11] in this sector. It is to be expected that more languages will follow in this area.

3.1.5 Detailed View on representatives

As the reader can see, a classification of query languages is not trivial and can be done along various dimensions. In this next Section I took the dimension of *basic paradigm* for categorization and give a deeper insight in popular representatives. The categories I defined are:

- a) languages that extend SQL as the common standard for querying relational databases or follow an SQL-like approach, like WebSSQL [ZMWZ00] or SQL/MM [ME01a],
- b) languages that build or extend query languages for object oriented databases like MOQL [LOS097] or POQL^{MM} [Hen01],
- c) languages that are focusing an XML metadata structure, like MMDOC-QL [LCH01a] or XQuery [B⁺07] (which is not explicitly build for Multimedia),
- d) visual query languages, like MQuery [DC96] (that focus on visual timeline retrieval) or VisualMOQL [OÖX⁺99],
- e) approaches that allow query-by-example, like [Jon07] or WS-QBE [SSH05b], and
- f) languages that try to build a meta-language, which are metadata agnostic and thus can be shared/distributed over several storage backends, like MPQF [DTG⁺08a].

Most of these Multimedia query languages use proprietary metadata models to express descriptive information. Generally, this information is represented by XML instance documents based on a specific XML Schema (such as MPEG-7 [MKP02] or TV-Anytime [GS13]). For this purpose, one also needs to consider query languages that are designed for XML data queried by XQuery [B⁺07]. The main drawback of XML is its limitations in expressing semantic meaning of the content information. This led to the development of RDF, the basis of the Semantic Web. To get a clear picture of each category of Multimedia query languages, I describe one example for every category in more detail.

a) SQL like approaches: MM/SQL

In the early 1990s the SQL (Structured Query Language) community came up with many incompatible extensions (especially for Multimedia) that forced the ISO subcommittee for SQL JTC1/SC32 to regularize such attempts. The proposed standard was immediately known as SQL/MM [ME01a] and meant to integrate Multimedia features to SQL. Like SQL, SQL/MM is a multipart standard that consists of various, mostly independent parts. Part 1 [ISO00a] represents the backbone of the standard and describes, how other parts use SQL's structured, user-defined types required for the specific purpose of each part.

Besides Multimedia functionality, text retrieval plays an important role for media in context. The full-text standard is covered by part 2 [ISO00b] and defines a number of structured user-defined types for storage. This is necessary because full-text in comparison to regular expression matching needs more complex data and query structures for (mostly language specific) tokenization, stemming, lemmatization, and fuzzy matching. In addition, fulltext search may support things like phonetic search (sounds like) and context search (heading, paragraph. etc.). Listing 3.1 shows a sample query using SQL/MM full-text extension on a sample table `documents` that includes a row `document` of type `FULLTEXT`.

Listing 3.1: Example for SQL/MM full-text search

```
SELECT * FROM documents
WHERE document.CONTAINS(
    'dog' IN SAME PARAGRAPH AS
    SOUNDS LIKE "Balu"
) = 1
```

The query combines contextual with phonetic search to retrieve documents that most probably include a dog named "Balu", "Baloo", "Paloo", etc. This type of search can be useful in combination with automatic extraction techniques e.g. speech-to-text.

Part 3 [ISO99] of SQL/MM covers the aspects of spatial data, such as geometry, location and topology. As described in [Sto03], SQL/MM defines a class model for 0- to 2-dimensional geometric objects (like points, lines, polygons or composites) as well as specific functions for spatial data. The spatial part of SQL/MM is mostly driven from geographic information system (GIS) but can be used for non-geographic use cases (e.g. fragment description for still images), too, whereby the reference system is replaced.

Figure 3.3 shows the SQL geometric type hierarchy for SQL/MM, which has been adapted from the geometric model of the OpenGIS Features Specification for SQL [Ope99]. The model differentiates between non-instantiable (supertypes)

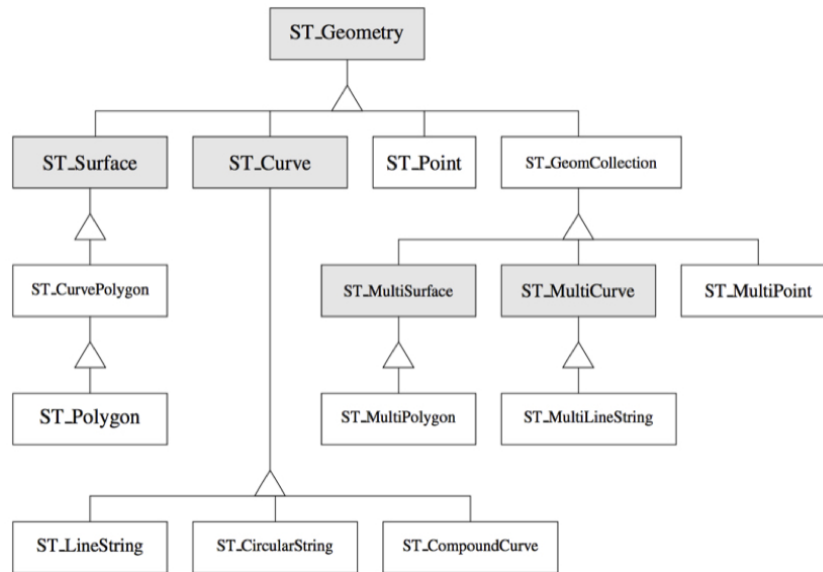


Figure 3.3: SQL/MM geometric type hierarchy

and instantiable types, like `ST_Point`, `ST_Curve`, etc.. There are many functions that can be performed over the spatial data model. They include the creation of new geometric objects out of existing ones, relational operations between objects like intersection or adjacence, and accessor methods that allow the extraction of fundamental information about type instance, e.g. the vertices of a line or the area of a polygone. Listing 3.2 shows a query that uses a spatial description of US counties to determine counties larger than the largest county in California¹.

Listing 3.2: Example for SQL/MM spatial query

```

SELECT c1.county_name
FROM County c1
WHERE ST_Area(c1.geometry) > (
    SELECT max (ST_Area(c.geometry))
    FROM County c, State s
    WHERE s.state_code = c.state_code
    AND s.state_name = 'California'
)

```

The temporal aspects of Multimedia were meant to be represented in part 4 of the SQL/MM standard but are not considered anymore, because *temporal* has a broader scope beyond the Multimedia applications and thus is included in the revised SQL:2011 standard [ISO11], like described in [KM12].

¹Sample is taken from

http://cs.ulb.ac.be/public/_media/teaching/infoh415/spatialnotes.pdf

In part 5 [ISO01] the standard focuses on storage, manipulation and retrieval of still images. The SI_StillImage datatype allows many formats (gif, png, tiff, etc.) for in- and output as well as for internal representation. The type also captures basic information about each image, such as format, dimension, color space, and so forth. Several operations can be applied on SI_StillImage including scaling, rotation, cropping, and shearing. SQL/MM also supports complex feature types, such as SI_ColorHistogram and SI_Texture (for coarseness, contrast, etc.).

In addition to classical Multimedia features, SQL/MM also includes a part 6 about Data Mining [ISO06], but I consider it as out of scope for my thesis.

b) OQL like approaches: MOQL

Object oriented databases combine database capabilities with object-oriented programming capabilities. This type of database management systems has been very popular a few years ago. The effort has been mainly driven by the Object Data Management Group (ODMG) that came up with several specification components including an object model, an object definition language (ODL) and a declarative, nonprocedural language for object oriented querying and updating (OQL) [CBB⁺00]. With MOQL (M for Multimedia) [LOSO97], this query language has been extended to deal with spatial, temporal and presentation properties by introducing new predicates and functions. In comparison to other approaches in the object oriented QL domain, MOQL is suitable for both video and still image retrieval. Most of the extensions of MOQL are placed in the WHERE clause in the form of 3 new expressions, namely *spatial_expression*, *temporal_expressions* and *contains_predicate*. Additionally, MOQL introduces a PRESENT statement that allows to specify how to deal with retrieval objects, especially with different mediatypes that has to be synchronized. I outline MOQL in this Section because it has a clear focus and a user-friendly language design.

Contains predicate

The contains predicate is an relation between an instance of a particular medium type (e.g. an image) and a salient object, which represents an physical object that is *contained* within the medium (e.g. a person). Listing 3.3 [LOSO97] shows a query that aims to retrieve all images in which a person appears.

Listing 3.3: Example for MOQL contains query

```
SELECT m
FROM Images m, Persons p
WHERE m contains p
```

Spatial predicates and functions

Spatial predicates compare spatial properties of spatial objects (such as a region, a point, etc.) with each other. A predicate (e.g. *inside*) can only compare specific types of properties. For example can *nearest* only be applied to two points, whereby *cover* can only apply to a region and a point / line. Spatial functions compute attributes of an spatial object or a set of spatial objects. The query in Listing 3.4 [LOSO97] shows both a spatial predicate *coveredBy* and spatial function *area*.

Listing 3.4: Example for MOQL spatial query

```
SELECT province, forest, area(forest.region)
FROM Forests forest, Provinces province
WHERE forest.region coveredBy province.region
```

Temporal primitives and functions

MOQL supports a set of 13 temporal relations that has been specified in [All83b] and are widely accepted, which are *equal*, *before*, *after*, *meet*, *metBy*, *overlap*, *overlapedBy*, *during*, *include*, *start*, *startedBy*, *finish*, and *finishedBy*. In addition, MOQL supports several so called continuous media functions especially for video objects and their frame character e.g. *firstClip* or *next*. Listing 3.5 [LOSO97] shows a query that returns the last clip in which a person appears from within a video v.

Listing 3.5: Example for MOQL temporal query

```
SELECT lastClip(
    SELECT c FROM v.clips c
    WHERE c contains p
    ORDER BY upperBound( c.timestamp )
)
```

Presentation statement

MOQL allows to integrate all retrieved objects of different media types in a synchronized way by adding a PRESENT clause. These layout consists of a spatial layout, which specifies things like number of images etc., a temporal layout, which allows to specify things like temporal order and total length, and a scenario layout, which allows also the usage of other presentation models or languages. Listing 3.6 [LOSO97] shows a query that presents the result (an image of a car and a video showing the same car) in two different windows simultaneously.

Listing 3.6: Example for MOQL present query

```

SELECT m, v
FROM Images m, videos v
WHERE for all c in (
    SELECT r FROM Cars r WHERE m contains r
) v contains r
PRESENT atWindow(m, (0, 0), (300, 400))
AND atWindow(v, (301, 401), (500, 700))
AND play(v, 10, normal, 30*60) parStart display(m, 0, 20)

```

MOQL mainly focuses spatial and temporal relationships but lacks any kind of similarity or best-match queries. Other object oriented approaches have different focuses, e.g. POQL^{MM} [Hen01], which targets asset similarity based on low-level features (like SQL/MM part 5 [ISO01]).

c) XML-based query Schemas: MMDOC-QL

The emerging of MPEG-7 [MKP02] Multimedia standard and its XML Schema datatypes in the late 1990s triggered attempts for XML-based media retrieval. For expressing audio and visual features, MPEG-7 defines so called Descriptors, for the relation and semantics between these features the standard provides description schemes. Video scenes for example can be formalized by using `SegmentDecompositon` with type `SpatioTemporal`. As most XML query proposals had limitations regarding this type of documents, MMDOC-QL [LCH01a] (Multimedia Document Query Language) was introduced, a language with Multimedia constructs that is based on a logic formalism called path predict calculus. Queries in this calculus are equivalent to the identification of path predicates that are satisfied by the XML tree document. This formalism allows to describe also spatial, temporal and visual datatypes and relationships by utilizing MPEG-7s description of media fragments.

In MMDOC-QL there are 4 clause types:

GENERATE / INSERT / DELETE / UPDATE are building the operation clauses. They are used to describe the logic conclusions in the form of allowed element and path predicates.

PATTERN clause describes the domain constraints of free logical variables (parts of the XML documents) by using regular expressions.

FROM clause defines the source (files).

CONTEXT clause is used to describe logic assertions about document elements in logic formulas (path predicate calculus). Within the calculus the language

uses a logic form of XPath axis-operators with logical variables in the path formula (e.g. DIRECTLY CONTAINING).

Listing 3.7 [LCH01a] shows an example query, whereby the path formula in the CONTEXT clause asserts that element "Segment" with id equal to %id contains element "SpatioTemporalLocator" (where the video objects are located during MediaTime %x). The form of %id is restricted by a pattern. The other lines in the CONTEXT part specifies the selection of %t; the GENERATE clause manages the output of the result as XML element.

Listing 3.7: Example for MMDOC-QL query

```

GENERATE <List>
    <Videoobject>%id</Videoobject>
    <ShowUpTime>%t</ShowUpTime>
</List>
PATTERN {"MR"[0-9]/%id}
    {<region> ... </region>%focus}

FROM    mpef7video.xml

CONTEXT ( ( <Segment> WITH xsi:type="MovingRegionType"
            id=%id AT %movingregion )
CONTAINING
    ( <SpatioTemporalLocator> DIRECTLY CONTAINING
      ( <MediaTime> AT %x ) )
AND MEMBERP (%t %c)
AND OVERLAP ( TRAJECTORY( %movingregion %t ) %focus )
)

```

d) Visual query languages: MQuery

MQuery [DC96] is a visual query language for the domains of simulation and validation, medical timelines and Multimedia visualization. The general framework that was worked out for querying all kind of Multimedia data (images, sounds, long text, video, and timelines). The language has a direct, visual support for all these datatypes and includes the entire range of query operations (insert, retrieve, delete, update). It supports alphanumerical queries, Multimedia results, Multimedia predicates, time-based data, and query nesting. Figure 3.4 [DC96] shows an example of MQuery for *obtaining the sex, age, and doctor of all patients with tumors similar in the shape to the tumor currently being viewed.*

e) Query by example: WS-QBE

The visual database query language QBE (query-by-example) [OÖX⁺99] is a declarative query language. It is based on the relational domain calculus. WS-QBE

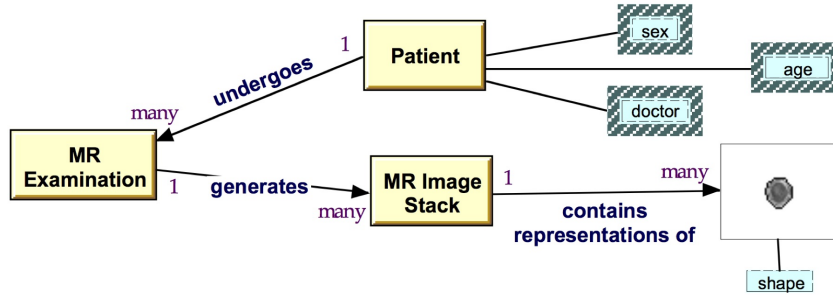


Figure 3.4: MQuery: visual query example

is an extension of QBE and adds fuzzy logic concepts as well as a schema for query-weighting, which enables it for complex similarity queries in the Multimedia domain. WS-QBE builds a core language for Multimedia similarity queries but lacks specific features like spatio-temporal functions and predicates. Result presentation is not considered in the basic approach. Formulating a query in WS-QBE means to fill table skeletons. A query like "Find all oil paintings from a Dutch painter, which are similar to a given image from my digital camera" is formulated by the two tables in Figure 3.1 and 3.5.


painting	id	photo	painter	title	technique
P.		~ 	_painter		oil

Table 3.1: Query-by-example with WS-QBE: 1

artist	id	name	country
	_painter		Netherlands

Figure 3.5: Query-by-example with WS-QBE: 2

The table headings map the underlying database schema. By inserting one or more new tuples the user gives an example that is used for similarity calculation. The entry **P.** is used to indicate, which fields (or tables) belong to the result set.

f) Generic Approaches: MPQF

The query languages I introduced are all strongly bound to the underlying metadata representation and schema. MPQF (MPEG Query Format) has the goal to unify the access to (distributed) Multimedia repositories in a schema agnostic way. The Language specifies precise input and output parameters within XML documents but does not use specific elements that are related to a metadata schema like MPEG-7

(like it is used for example in MMDOC-QL [LCH01a]). An MPQF query always includes a *MpegQuery* root element with two child elements *Management* and *Query*. The management Section provides a means for requesting service-level functionalities, the query Section can either include an input or an output (depending if it is a request or a response). Figure 3.6 [DTG⁺08a] shows the schema diagram of an

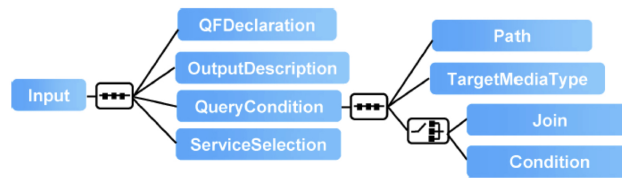


Figure 3.6: MPQF Input Query Format

MPQF Input element. It may contain one or more of the following elements:

QFDeclaration allows the definition of reusable definitions like paths and/or resources (descriptive as well as media resources) that can be referred from other parts of the query.

OutputDescription describes the structure and content representation for result set items. Furthermore it supports set operations like sorting, counting, and paging.

QueryCondition contains the actual filter criteria:

Path is a XPath expression and specifies the granularity of the retrieval, for instance if the process focuses on whole videos or on video fragments.

TargetMediaType contains MIME type descriptions like *audio/mp3* (if the user wants to retrieve audio files in MP3 format).

Join / Condition supports further diversity in filter criteria with arithmetic / boolean expressions, several query types (query-by-media, query-by-freetext, etc.) and joins.

ServiceSelection specifies a set of Multimedia query services where the query should be evaluated.

Figure 3.7 [DTG⁺08a] shows the schema of an MPQF Output element, which may contain one or more of the elements:

GlobalComment is meant for sending general messages such as the service subscription expiration or messages that are valid for the whole result set.

ResultItem element holds a single record of a query result with attributes *record-Number*, *rank*, *confidence* and *originID* and the elements:

Comment is similar to GlobalComment but focus in the specific result item.

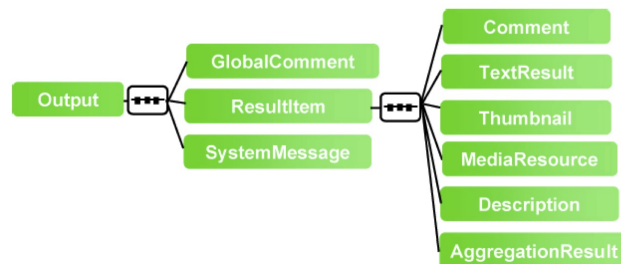


Figure 3.7: MPQF Output Query Format

TextResult element holds the result item as type text.

Thumbnail carries the URL of a thumbnail image.

MediaResource carries the URL of the media resource in the requested format.

Description is a container for any kind of metadata in any format like MPEG-7 or TV-Anytime.

AggregationResult allows schema-valid result aggregation operation (e.g. SUM).

SystemMessages includes special messages regarding the responding system such as warnings or exceptions.

Listing 3.8 [DTG⁺08a] shows an example of a simple MPQF query that combines free-text search and conditions over XML metadata. The aim of the query is to find large images of "Hong Kong" (greater than 1000 pixels in width).

Listing 3.8: Example for MPQF query

```
<MpegQuery>
  <Query>
    <Input>
      <OutputDescription thumbnailUse="true">
        <ReqField typeName="MediaInformationType">
          MediaProfile/MediaFormat/FileSize</ReqField>
        <ReqField typeName="CreationInformationType">
          Creation
        </ReqField>
      </OutputDescription>
      <QueryCondition>
        <TargetMediaType>image/*</TargetMediaType>
        <Condition xsi:type="AND" preferenceValue="0.1">
          <Condition>
            <FreeText>Hong Kong</FreeText>
          </Condition>
          <Condition xsi:type="GreaterThanOrEqual">
            <ArithmeticField typeName="MediaInformationType">
              MediaProfile/MediaFormat/Frame@width
            </ArithmeticField>
            <LongValue>1000</LongValue>
          </Condition>
        </Condition>
      </QueryCondition>
    </Input>
  </Query>
</MpegQuery>
```

Further examples for MPQF queries can be found in [DTG⁺08a].

3.2 Requirements of Multimedia Query Languages

Since Codd has proven the equivalence of relational algebra and relational calculus [Cod72], the term *relational completeness* stands for the expressive power of query languages. However, nowadays Multimedia data and the search therein is on the rise, which demands for a refinement of the principles of query languages in terms of Multimedia specific predicates, fuzziness or vagueness, weighting and similarity. In this context, first principles of query languages have been defined early in 1991 by Heuer et al [HS91]. Further refinements in this direction have been undertaken by Heuer and Saake in [HS00] and by Schulz in [Sch04b]. It has to be noted that some requirements pursue conflictive aims, which is especially true for the general requirements. Therefore, none query language can fulfill all requirements at the

best. The following Subsections will define an entire list of principle requirements for Multimedia query languages in order to get a proper bases for requirement definition in Chapter 4.

3.2.1 Preliminaries

The following nomenclature is valid throughout the set of definitions: Let U be some set and $R \subseteq U \times U$ be a binary relation of U . \mathcal{L} is denoted as query language.

3.2.2 General requirements of query languages

The following set of definitions summarizes a general list of requirements a query language should provide. Those definitions are also valid for other type of query languages in various domains as for instance for Resource Description Framework (RDF) data [HBEV04].

Definition 7 (Transitive closure) *Transitive closure within \mathcal{L} requires that the result elements of an operation are part of the data model. To be more concrete, the transitive closure R^+ of a relation R is the smallest subset $R^+ \subseteq U \times U$ with $R \subseteq R^+ : \forall x, y \in U | R(x, y) \rightarrow R^+(x, y)$ and $\forall x, y, z \in U | R^+(x, y) \wedge R^+(y, z) \rightarrow R^+(x, z)$.*

Definition 8 ((Relational) Completeness, Codd [Cod72]) *\mathcal{L} is relationally complete if, given any finite collection of relations R_1, \dots, R_N in simple normal form, the expressions of the query language permit definitions of any relation definable from R_1, \dots, R_N by predicates. A predicate is a binary expression of the form attribute \star attribute, where \star denotes a binary operators over the domain of the attributes such as $<, >, =, \neq, \leq, \geq$.*

Definition 9 (Ad-Hoc formulation) *\mathcal{L} supports ad-hoc formulation if there is no need for application logic or user program to express queries. In this context, requests to a database should be phraseable by an iterative user interface.*

Definition 10 (Extensibility) *A query language \mathcal{L} must be extensible in terms of new operations especially in regard to the underlying data model (e.g., adding of new media types).*

Definition 11 (Optimizable) *A query language \mathcal{L} must support automatic optimization steps on the basis of an internal (algebraic) representation. This requires a definition of formal semantics. This criteria is diametrically opposed to Extensibility for one and the same part of the language. But considering different parts \mathcal{L} can fulfill both by a certain amount.*

There are more general requirements, which are not considered for the evaluation in this thesis because they are too complex to validate seriously with an affordable effort or are not orthogonal to other requirements and such conditions. For the matter of completeness these are:

Definition 12 (Safety [Hir92, Rev10]) \mathcal{L} is considered safe, if any instance of \mathcal{L} returns a finite set of results on a finite data set.

Definition 13 (Adequacy) \mathcal{L} is called adequate if it uses all concepts of the underlying data model, which complements the closure property: For the closure criteria, a query result must not be outside the data model, for the adequacy criteria the entire data model needs to be exploited.

Definition 14 (Orthogonality) Orthogonality of a query language \mathcal{L} requires that any operation may be used independently of the usage context and therefore supports the combination and nesting of query operations and constructs.

3.2.3 Specific requirements of Multimedia query languages

As illustrated in the Section before, the presented requirements should be fulfilled by any query language in any domain. This includes of course well established systems in the relational, object-relational and object-oriented DBMS world as well as languages in the more novel areas such as XML based DBMS, RDF based DBMS and Multimedia related DMBS (MMDBMS). Nevertheless, any mentioned domain demands further needs in terms of additional and specialized operations, semantic concepts and expressiveness. To cope with such specific requirements for Multimedia query languages, this Subsection summarizes definitions for expressing Multimedia needs. I provide a set of example queries that support us to test if query languages fulfill a specific requirement and thus make the evaluation transparent and reasonable.

Definition 15 (Universal) The universal requirement of a query language \mathcal{L} for Multimedia data demands besides the application independence (see definition 9) criteria also the support for multi-modal Multimedia data types such as video, image, audio and text data.

Query 15

Give me all video/image/audio that fulfills a specific constraint (e.g. tagged with "red car").

Definition 16 (Uncertainty) A query language \mathcal{L} must support operations that base on the concept of partial truth, whose result values represent uncertainty. Related to the well known fuzzy logic approach, which has been originally introduced by Zadeh [Zad65], the following is defined: A fuzzy operation μ is characterized as a function of the reference set U ($u \in U$) to the interval $[0,1]$, $\mu : U \rightarrow [0,1]$. A fuzzy

conjunction \wedge is expressed by the function $\top : [0, 1]^2 \rightarrow [0, 1]$ (*t-norm*) following the axioms *monotony*, *commutativity*, *associativity* and $\top(u, 0) = 0$, $\top(u, 1) = u$. A fuzzy disjunction \vee is expressed by the function $\perp : [0, 1]^2 \rightarrow [0, 1]$ (*t-conorm*) which is *monotone*, *commutative*, *associative* and with a *unity* of 0. The complement is defined as $\bar{\mu}(u) = 1 - \mu(u)$. Important implication ($\varphi \rightarrow \psi$) operations have been defined for instance by Lukasiewicz, Gödel or Goguen. Elements $u \in U$ are called a *truth degrees*.

Query 16: Fuzzy Matching

Give me all media that contains a red car a order it by relevance score.

Node that this query in a fuzzy evaluation will also return items that matches *green motorbike* or even non-matching items and can be only validated together with ordering.

Definition 17 (Spatial Operations) *Spatial Operations can be separated in three pillars, which are relational operations, aggregation operations and accessors. Spatial operations define an operation $\diamond(x, y) \rightarrow [0, 1]$, where \diamond is an operation for instance defined in [Zla07] and $x, y \in U \wedge \text{dom}(x) = \text{dom}(y)$. Spatial relations can be of type topological relational (e.g. contains, overlaps, etc.), directional relational (e.g. right beside) or distance relational (e.g. nearby). Spatial aggregation functions $\alpha(x, y) \rightarrow z$ with $x, y \in U \wedge \text{dom}(x) = \text{dom}(y) = \text{dom}(z)$ creates a spatial (media) fragment out of two fragments, e.g. difference, intersection etc.. Spatial accessors allow access to spatial "metadata", e.g. area, center, boundingBox, etc..*

Query 17b: Spatial Relation

Find me images/videos that show a tree (once given as annotation and once given as example item) left of (right of, etc.) a house.

Query 17b: Spatial Aggregation

Show me image parts that show a dog on top of a bed.

Definition 18 (Temporal Operations) *As spatial operations, temporal operations can also be separated in relations, aggregations and accessors. Temporal relations define an operation $\triangleright(x, y) \rightarrow [0, 1]$, where $x, y \in U \wedge \text{dom}(x) = \text{dom}(y)$. The classical temporal relation model defined by [All83a] contains 13 relations, whereby 12 are pairwise contrary (e.g. after vs. before). Temporal aggregation functions $\alpha(x, y) \rightarrow z$ with $x, y \in U \wedge \text{dom}(x) = \text{dom}(y) = \text{dom}(z)$ creates temporal (media) fragment out of two temporal fragments (e.g. intermediate). Temporal accessors allow access to temporal "metadata", e.g. duration, start, end, etc..*

Query 18a: Temporal Relation

Give me a video where a clip of an horse ridding is followed by (at least Allen's temporal relations are supported) a flock of cows.

Query 18b: Temporal Metadata Access

Give me a clip of a video that contains a red car (given as annotation and as example item) and lasts at least 10 sec.

Definition 19 (Evolution) *In many disciplines like medicine or environmental research the investigation of temporal image series (e.g. satellite images, x-ray) is fundamental. The significant evaluation of a image series S regarding a specific parameter (set) P can be defined by a function $evol(S, P, \epsilon) \rightarrow [0, 1]$, whereby ϵ defines the threshold.*

Query 19: Evolution

Give me image pairs that shows a significant change in (parts of) the infrared image series of the south pole from 1980 to 2016.

Definition 20 (Metadata Operations) *Multimedia Objects often have metadata information in addition to the primary (raw) data. Metadata helps to reduce the semantic gap that is caused by the hidden character of Multimedia (raw) data. There are many different kinds of metadata operations, like:*

- a. *Structural Metadata Operations: Structural metadata includes basic information that describes the appearance of the image itself and not the content. These data is often included already in media codecs (e.g. EXIF [Tec02]).*
- b. *Content-descriptive Metadata Operations: This metadata allows to query for semantics of an image by using well-formed content description. This contains keyword based search as its most simple representative up to queries that use a complex and rich description context.*

Note, I do not differentiate between the specific forms of metadata operations in this survey but consider it as one feature.

Query 20a: Structural Metadata

Give me all video/image/audio that have a file size equal/smaller/larger/... to/than 1024KB.

Query 20b: Keyword

Give me all video/image/audio whose description contains "Summer".

Query 20c: Complex Content

Find all videos where a car of a German car manufacturer drives through a country that belongs to the European Union.

Definition 21 (Media Similarity Operations) *Example based Operation: Defines an operation $\sim_M(x, y) \rightarrow [0, 1]$ where $\forall x' \in U : \sim_M(x, y) \leq \sim_M(x', y)$. This can be enhanced by a value k as follows $\sim_M^k(x, y)$ where $\{x_1 \dots x_k \in U \mid \neg \exists x' \in U \setminus \{x_1 \dots x_k\} \wedge \neg \exists i, 1 \leq i \leq k : \sim_M^k(x_i, y) > \sim_M^k(x', y)\}$. Note, M describes the used metric, $k \in \mathbf{N} \wedge \text{dom}(x) = \text{dom}(y)$.*

Query 21a: Basic Similarity

Give me all image/video whose color spectrum is within a maximum range of 30.

Query 16b: Pseudo-Similarity

Give me all media item that are similar to a complex media item description (e.g. a red car on the right upper corner).

Query 21c: Complex Similarity

Give me all image/video/audio that are similar to the positively selected examples image/video/audio1 and image/video/audio2 but not similar to image/video/audio3.

Definition 22 (Weighting) *A query language \mathcal{L} should support weighting capabilities of operations in order to accentuate the importance of specific functions in context to the overall query evaluation. This also supports query operations like relevance feedback.*

Query 22: Weighting

Return images/videos with red cars (very important), yellow motorcycles (important) and green buses (nice to have).

Definition 23 (Feature Combination) *A query language \mathcal{L} should support the combination of (different) features in order to support complex features. As this feature is very dependent on the supported features to be combined I formulated the example query quite fuzzy.*

Query 22: Feature combination

Return images/videos/audio files which contains A OR B, which bears relation to something similar to B.

Looking at the languages I used for evaluation it is obvious that there are more features that are enabled by Multimedia query languages. But as many are very specific to uses cases, media structure and/or language paradigms I consider only the presented subset. This whole set of requirements used for evaluation is outlined in Table 3.2.3.

3.3 Conclusion

In this Section I gave an exhaustive overview of Multimedia queries from the past decades. I analyzed six representatives of different language types in detail. Based on this survey I specified a list of typical requirements of Multimedia query languages and provided examples queries for each of them. This will be used in later Sections for the evaluation of the Multimedia query language that is developed within this thesis.

ID	Type	Full name
TC	General	Transitive Closure
RC	General	Relational Completeness
AF	General	Ad-Hoc formulation
E	General	Extensibility
O	General	Optimizable
UF	General	User Friendliness
U	Multimedia	Universal
SO	Multimedia	Spatial Operations
TO	Multimedia	Temporal Operations
TE	Multimedia	Temporal Evolution
MO	Multimedia	Metadata Operations
MS	Multimedia	Media Similarity Operations
W	Multimedia	Weighting

Table 3.2: Requirements for Multimedia Query Languages

The survey showed that most languages are very use case specific. In addition many of them do not build on top of well known query languages which lead to a steep learning curve. Furthermore creating and editing the queries is not easy because of the lack of tools and/or missing integration in IDEs. Nevertheless there are suitable candidates like SQL/MM, MOQL and MQuery that use widely known patterns (SQL, OQL, graph-visualization) and contain a set of useful Multimedia features. But the only language that supports RDF/SPARQL is MPQF, whereby this integration happened quite late in the standardization process, and thus only can be used for Metadata Operations and does not really interact with other language components. Therefore I decided to combine both graph pattern features and Multimedia specific requirements in an extension of SPARQL - namely SPARQL-MM.

In the next Section I am going to introduce two real world use cases, which allows me to map the requirements to understandable example queries that can be later used for evaluation. The extension I mentioned above will be specified in Chapter 7.

Part III

Semantic Multimedia

Application Scenarios

In Section 3.2.3 I gave an overview of basic requirements for Multimedia query languages based on a historical survey. In this Chapter I introduce two real world use cases and identify their requirements. The first use case describes an image retrieval use case within a crowdsourcing platform. The second one describes a video snippet retrieval use case within a pool of extreme sports video clips. I describe the requirement analysis process and the use cases in detail. Based on that, I define a set of example queries that are used to verify the fulfillment of requirements in a later Section, whereby I am going to outline the used features of SPARQL-MM. The requirement gathering process is a condensed summary of the work I did in the MICO project and published on a Use Case requirement analysis compendium ¹.

4.1 Requirements gathering

The process I choose was meant to link requirements that has been identified together with use case partners to specific parts of overall systems. The diagram in Figure 4.1² outlines the parts and relations of the process. This model allows to consider both, top-down (abstract requirements) as well as bottom-up (technical feasibility) aspects and thus allows to structure complex scenarios with the outcome of specific requirements mapped to components (e.g. querying).

Showcases (SC) are the main entry point of the requirement analysis process. The showcases represent planned projects of the use case partner. They provide a highlevel description and form the basis for dedicated user stories.

User Stories (US) are derived from showcases and serve as the starting point for requirement analysis. The separation of SC and US allow to collect many possible showcases without going into detail from the beginning. US define requirements without a direct coupling to technology and are formulated in freetext following the scheme: **As a <ROLE>, I want to <GOAL> so that <BENEFIT>**. US are bound to one or more technology enablers. In addition US define involved Datasets.

Technology Enablers (TE) are the technical counterparts of US. This split enables to find best-matching technologies for a user driven requirement by a

¹MICO - Compendium Use Case Requirements Analysis: <https://tinyurl.com/y7bem99p>

²Taken from document mentioned in footnote 1

dedicated State of the Art analysis. In addition it outlines which technological solutions are related to which US and thus allows an analysis of cross US requirements and guarantees an optimal workplan while preventing parallel and/or contrary technological development. TE are mainly specified by functional requirements but may also contain non-functional requirement.

Non-functional requirements (NF) are less US dependent but driven by the overall system architecture (e.g. scalability, extendability, usability, etc.). Thus they define additional boundaries for technological decisions.

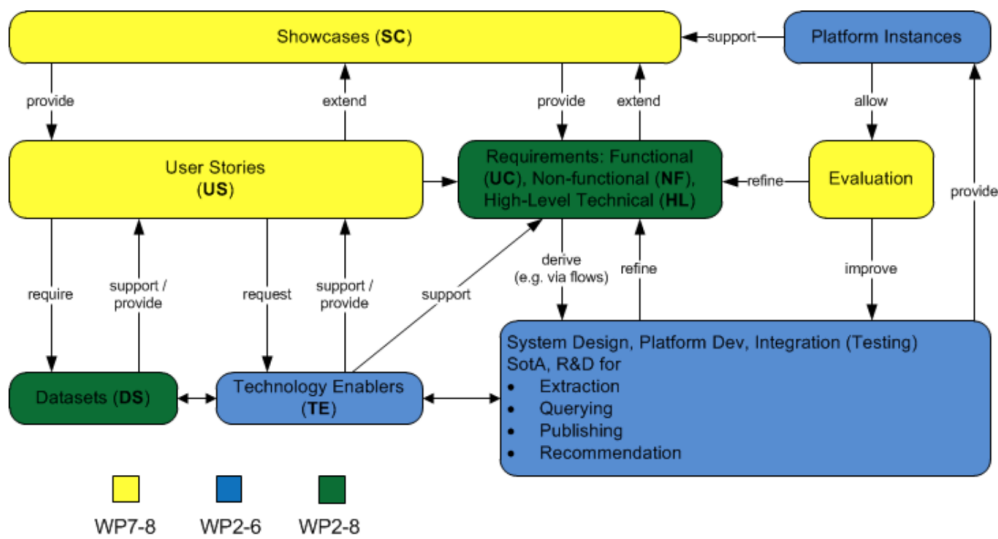


Figure 4.1: Diagram of requirement gathering

In the next two Sections I am going to introduce two showcases that are driven by real-world aims. After a short introduction of the industrial partner and the showcase as such, I will list user stories and the underlying requirements. As this thesis is limited to information retrieval I only consider related technology enablers, namely *Multimedia Query Language* and *Semantic Multimedia Similarity*.

4.2 Image Retrieval

Instance: Zooniverse

Zooniverse is the largest and most popular online citizen science platform. The team is based in the astrophysics department at the University of Oxford and the Adler Planetarium in Chicago. It started in 2007 with the Galaxy Zoo project and now operates over 50 separate projects across many fields of research such as astrophysics, climatology, ecology, biology and history. Each project is built around the idea

that volunteers can access the website and classify data (images, video, audio) by performing basic recognition tasks that cannot be easily performed automatically by computers. The form of the task and size of the dataset depends on each project that normally take between six months and one year to build. Zooniverse receives 3040 proposals for new projects every year but at the moment they are only able to build a small fraction of them. State of December 2018, zooniverse has a community of 1.7 million registered users, which have done 400 million classification tasks.³

Use Case: Snapshot Serengeti

The Snapshot Serengeti tries to collect information about the entire community of large animals in Tanzania's Serengeti National Park and the Ngorongoro Conservation Area. A grid of 225 camera traps continuously produces images of animals. The crowdsourcing task involves identifying 48 various species and their behaviour from camera trap images. It includes pure classification as well as spatial annotation. The manual created metadata is meant to be consumed by researchers in order to get insights in the behavior and coexistence of species. Therefore the use case contains the following user stories related to image retrieval:

- [US-SS-1] As a researcher I need to find images that contain a certain species.
- [US-SS-2] As a researcher I need to find images that contain a certain species and another (unspecified) species.
- [US-SS-3] As researcher I need to find images that does not contain any animal.
- [US-SS-4] As a researcher I need to find images that contain a species A in the background.
- [US-SS-5] As a researcher I need images that contain species A and B next to each other.
- [US-SS-6] As a researcher I need images that contain species A on the left and species B on the right.
- [US-SS-7] As a researcher I need images that contain species A filling out almost the whole image.
- [US-SS-8] As a researcher I need to find images that contain a carnivore next to a herbivore.
- [US-SS-9] As a researcher I need to find images with a species x on the top and a species y on the bottom.
- [US-SS-10] As a researcher I need to find images 2 different species overlapping to 90%.

³The parter description of zooniverse is an upadted version of <https://tinyurl.com/y7bem99p>

Looking at the requirements for Multimedia query languages that I have elaborated in Section 3.2.3 the use case mainly focus on spatial operations (SO) and Metadata Operations (MO). Therefore this use case is used to cover the variety of spatial operations.

4.3 Image/Video Retrieval

Instance: Media Company

This use case is an adaption of the one presented in [KSFG12]. The scenario had to be anonymized. The company is acting as a support, competence and service center to maximise/optimize global media output. A central part of the technical ecosystem is the asset management system where the company collects digital assets that are produced continuously and consumed in many streams including broadcasting, on-demand video, social media, Web presences and print. The content is connected to well structured metadata using named relations and taxonomies. The metadata is generated (semi-) automatic as well as manually and includes both, ordinary asset specific (e.g. image-height, video format, etc.) and semantic relations (e.g. contains person A) on asset and sub-asset level (e.g. video fragment).

Use Case: Content Management Service

The Content Management Service plays a central role within the Media Asset Management Lifecycle. It is driven by archivers and accessed by various kind of user groups. These include managers with the aim to find images for their presentations, people writing articles for specific channels (e.g. print, Web, social media, etc.) and search for embeddable material, people who found a video clip on a 3rd party source and need to access the original high-resolution asset from the asset management system, and many more. As for the Snapshot Serengeti use cases, I identified user stories for media retrieval, whereby I split by image and video specific:

Image Retrieval User Stories

US-MCI-1 As a manager I want to find images with some person on the right and a climber on the left.

US-MCI-2 As a manager I need a action image with fits my main presentation color.

US-MCI-3 As a manager I need a image with a lot of blue sky and a small person on the lower center.

US-MCI-4 As a journalist i need a photo with person A on the upper left and person B on the lower right.

US-MCI-5 As an author i need an image for my blog that shows somehow the same things like on a photo I have but looks different.

Video Retrieval User Stories

US-MCV-1 As a content manager I have to find a video that contains person A moving from the left to the right.

US-MCV-2 As a content manager I have to find a video clip from the a specific challenge where person A performs a backflip and crashes afterwards.

US-MCV-3 As a content manager I have to find a clip that shows person A, person B and an alpine ski driver within 5 seconds.

US-MCV-4 As a cutter I have to find a winner ceremony from the F1 grand prix in hungary which lasts at least 5 seconds and where the persons take least 75% of the image height.

US-MCV-5 As a cutter I need a short scene where a surfer in the right video section is riding a big wave.

As one can see, this use case is broader in case of the requirements for Multimedia query languages. It contains Media Similarity together with weighting (MS + W), Spatial as well as Temporal Operations (SO, TO) and Metadata Operations (MO).

4.4 Conclusion

In this Section I introduced two real world use cases for both image and video retrieval. The use cases include user stories that build the basis to specify a fitting information retrieval query language in later Sections. As all queries can be mapped to the feature set defined in 3.2, I will use this to evaluate the new function set and its adequacy in later Sections. Note, that the requirements I specified in general for Multimedia query languages in Section 3.2.3 won't be not part of the evaluation, as I am only going to extend an existing language by using build-in extension mechanism and thus not change the signature of SPARQL itself. In the next Section I will introduce a theoretical model in order to get a solid basis for the specification, evaluation and optimization of the language.

Basic Model for a Semantic Web Multimedia QL

In this Chapter I will introduce the theoretical model that is used later on for the description of the extension functions and the formalization of the computation and optimization steps. As described in Chapter 4 there are several use cases that a the semantic Multimedia query language has to fulfill:

- Support the retrieval of media items that contains a specific object. This will support queries like *'Return videos that contain an athlete'*. In comparison to existing query languages this should also consider some semantics, e.g. the mentioned query has to support subclass relations like: $\langle windsurfer \rangle \langle isA \rangle \langle athlete \rangle$.
- Provide functions to specify spatial relations between media fragments of the same media item. That enables queries like *'Return images that contain a dog right beside a banana'*.
- Provide operations to specify temporal relations between media fragments of the same media item. That allows for queries like *'Return video scenes that show a lion and a gazelle at the same time'*.
- Support similarity metrics for content item sets. This will allow queries like *'Return images that are similar to a given image'* and thus the typical query patterns *Query-by-Example* and *Query-by-sketch*. The aimed query language has to take into account spatial and temporal relations of fragments in combination with semantic concept similarity.
- Enable free combinations of all the mentioned facilities.

The Multimedia-specific theoretical fundamentals of SPARQL-MM algebra that I am going to describe in the this Sections are based on the DISIMA image model. The model is the basis for all the operations that are supported by SPARQL-MM and therefore adapted accordingly. Furthermore, as the language is an extension of the SPARQL query language, I introduce the the SPARQL algebra as described in [HS13].

5.1 Modeling Multimedia

The DISIMA model aims to describe images and its context on a abstract level. It uses a layer approach where it differs between two blocks:

image block that includes the image and its representations embedded in a descriptive context, and a

salient object block, which is a three layered model including logical objects (e.g. the person Barack Obama), a physical layer representing the fragment of an object and representation layer (in order to have the same decoupling as in the *image block*).

The central unit in the DISIMA model is the *image*. The model in this thesis aims to target video (and should be expendable also to other media types) so it is generalized to *media object* as the central unit. The Definitions 24, 25 and 26 are derived from [OOL⁺97].

Definition 24 *An media object m is defined by a quadrupel $\langle m, R_{(m)}, C_{(m)}, D_{(m)} \rangle$ where,*

- m is the unique (raw) object identifier;
- $R_{(m)}$ is a set of representations of the raw media object in a format such as GIF, JPEG for image, MP4 for video, etc;
- $C_{(m)}$ is the content defined in Definitions 26;
- $D_{(m)}$ is a set of descriptive alpha-numeric data associated with m .

As mentioned the salient object block splits physical and logical description parts as follows:

Definition 25 *A physical salient object is a part of an image and is characterized by a position in the media object space. A logical salient object is an object that is used to give semantics to a physical salient object.*

The matching of physical to logical objects is described by a relation function. Note, that in case of RDF this mapping is done via one or more labeled, directed links. The structure depends on the underlying metadata model that is used.

Definition 26 *Let L be the set of all logical salient objects and P be the set of all physical salient objects. The content of an media object m is defined by a pair $C_{(m)} = \langle P_m, s \rangle$ where:*

- $P_m \subseteq P$;
- $s : P_m \mapsto L$: maps each physical salient object to a logical salient object.

This means that every instance of the media object space (e.g. a rectangular snippet with a specific width, height, vertical and horizontal offset) has a dedicated relation (e.g. *hasSubject*) to an object of the logical salient object space (e.g. the person named 'Tom'). This means upon reversion that the existence of $P_m \in P$ is dependent on the existence $C_{(m)}$.

Definition 27 Let $P_m, P_n \subseteq P$ with $m, n \in M$. Let $n \neq m$, hence:

- $P_m \cap P_n = \emptyset$.

That prevent physical salient media objects to be part of two or more media items. This generic model now allows us to make a definition of physical salient image objects.

Definition 28 Let I be the set of Images with $I \subseteq M$. An physical salient image object $p_i \in P$ with $i \in I$ is characterized by a distinct position (e.g., a set of coordinates) in the image space.

Fragments can have various formats. As a first representative I introduce rectangular physical salient image objects. In order of readability I now use the term *image fragments* and *physical salient image objects* analogously. This extends to *fragments* and *physical salient objects*, too.

Definition 29 Let the physical salient object layer of an image be a cartesian coordinate system on a two-dimensional euclidean plane, whereby the image's top left corner is handled as the origin. The coordinate axis from the left to the right is called X , the axis from the top to the bottom is called Y . Then a rectangular image fragment $p \in P$ can be described using a tuple $\langle \vec{a}, \vec{b} \rangle$, whereby:

- \vec{a} defines a vector $\begin{pmatrix} x_a \\ y_a \end{pmatrix}$ with $x_a, y_a \in \mathbb{R}^+$ that denotes the top left corner of the fragment, and
- \vec{b} defines a vector $\begin{pmatrix} x_b \\ y_b \end{pmatrix}$ with $x_b, y_b \in \mathbb{R}^+$ that denotes the bottom right corner of the fragment.

With this definitions I can define semantic media fragments. Note, that the rectangular fragments also allows the definition of points (whereby $\vec{a} = \vec{b}$) and vertical/horizontal lines, which is sufficient for our purposes.

SPARQL-MM aims to provide (binary) relational operations between fragments. Taking the model described above, a set of relations can be defined as follows:

Definition 30 Let P be the set of all media fragments with $P = \bigcup_{m \in M} P_m$. The set of fragment relations is defined as $R \subseteq P \times P$. This is equivalent to indicator function $\chi_R : P \times P \mapsto \{0, 1\}$.

That means that I can define any relational functions between two media fragments. The following example aims to make the model more clear and to show, how the model can be used for fragment relation.

Example 4 Let P^{rect} be the set of rectangular image fragments. Let $I \subseteq M$ the set of images. For the example I define four fragments corresponding to the one outlined in figure 5.1.

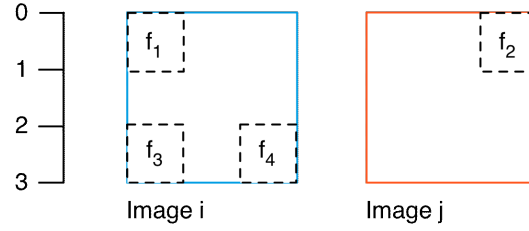


Figure 5.1: Example for rectangular image fragments

- $p_1 = \langle \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix} \rangle$, with $p_1 \in P_1$
- $p_2 = \langle \begin{pmatrix} 2 \\ 0 \end{pmatrix}, \begin{pmatrix} 3 \\ 1 \end{pmatrix} \rangle$, with $p_2 \in P_2$
- $p_3 = \langle \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 1 \\ 3 \end{pmatrix} \rangle$, with $p_3 \in P_1$
- $p_4 = \langle \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \begin{pmatrix} 3 \\ 3 \end{pmatrix} \rangle$, with $p_4 \in P_1$

With $P_1, P_2 \subseteq P^{rect}$ I define $r_{rightBeside} \in R$ as an example function, with $\chi_{r_{rightBeside}} : P^{rect} \times P^{rect} \mapsto \{0, 1\}$

$$\chi_{r_{rightBeside}}(p_n, p_m) = \begin{cases} 1 & \text{for } x_{a_{p_n}} > x_{b_{p_m}} \wedge p_n, p_m \in P_i, i \in I. \\ 0 & \text{in any other case} \end{cases}$$

In words: a rectangular image fragment p_n is right beside a rectangular image

	f_1	f_2	f_3	f_4
f_1	0	0	0	0
f_2	0	0	0	0
f_3	0	0	0	0
f_4	1	0	1	0

Table 5.1: Truth table for rightBeside function on rectangular image fragments

fragment p_m if they left border of p_n is right beside the right border of p_m and both

fragments belong to the same image. The result of the relation function is displayed within the truth table 5.1. Note, that fragment p_4 is right beside both p_1 and p_2 .

The model so far fulfills the needs for spatial fragment operations on images but lacks support for (temporal) video fragments. There has been efforts to extend the DISIMA model to videos described in [CÖO03]. This extension uses a frame based model, that does not fit very well with the later function description, wherefore I take a different approach.

Definition 31 Let T be a set of temporal media objects with $T \subseteq M$. A physical salient temporal object (from now on called temporal fragment) $p_t \in P$ with $t \in T$ is characterized by a distinct subpart in the temporal space.

A temporal fragment can have various formats. For our model it is sufficient to define a temporal interval as subset of the temporal space of a media object. Therefore I propose a interval subset definition:

Definition 32 Let I be an (time) interval $I = [a, b] = \{t \in \mathbb{R} | a \leq t \leq b\}$. $I^* = [a^*, b^*]$ is a subinterval of I if $a \leq a^* \leq b^* \leq b$. I write $I^* \subseteq I$.

With this, a temporal space $p \in P$ can be described by its interval I_t and a temporal fragment as subinterval $I_{p_t} \subseteq I_t$. Note, a temporal instant is also supported by the model as interval $I = [a, a]$.

With temporal and image spaces a definition of a video is straight forward.

Definition 33 A video $v \in V$ (with $V \subseteq M$) is a tuple $\langle S_{(v)}, I_{(v)} \rangle$ with

- $S_{(v)}$ is the image space of the video and corresponds to the physical salient layer of an image, and
- I is an interval $I = [a, b] = \{t \in \mathbb{R} | a \leq t \leq b\}$ representing the temporal space or the time line of the video.

An video fragment in this model is characterized by a distinct position (e.g., a set of coordinates) in the image space and a distinct position in the temporal space (e.g. a temporal interval).

Definition 34 Let V be the set of Videos with $V \subseteq M$. A static physical salient video object $p_v \in P$ with $v \in V$ can be represented by a tuple $p_v = \langle p_{S_{(v)}}, I_{(v)}^* \rangle$ with:

- $p_{S_{(v)}}$ is the a spatial fragment in the image space of v , and
- $I^* \subseteq I$.

Note, that a static physical salient video object can be seen as a video v^* by itself, whereby $S_{(v^*)} = p_{S_{(v)}}$ and $I_{(v^*)} = I_{(v)}^*$

For dynamic physical salient video objects (e.g. a fragment that broadens in time) I have to add an additional definition.

Definition 35 Let V be the set of Videos with $V \subseteq M$. A dynamic physical salient video object $p_v \in P$ with $v \in V$ can be represented by a triple $p_v = \langle p_{S_{(v)}}, I_{(v)}^*, A \rangle$, with:

- $p_{S_{(v)}}$ is the a spatial fragment in the image space of v ,
- $I^* \subseteq I$, and
- A is a set of animations.

A static video fragment can be defined as a dynamic video fragment with $A = \emptyset$.

In order to get the manifestation of an image fragment for a specific timestamp I define a mapping function.

Definition 36 Let P_v the set of video fragments of $v \in V$, and $P_{S_{(v)}}$ the set of image fragments in the image space of v like defined in Definition 34. The function $\sigma : P_v \times \mathbb{R} \mapsto P_{S_{(v)}}$ maps each video fragment to an image fragment for a given timestamp, whereby $\sigma(p_v, t) = \emptyset$ for $t \notin I_{(v)}^*$.

Example 5 To make the definition more concrete I outline an example of an animated video fragment p_v . Therefore I define an animation $a_{xscale} \in A$, which defines a linear scaling and is defined by single value $s \in \mathbb{R}$. The video v in this example is defined by the tuple $\langle \langle 6, 6 \rangle, [0, 3] \rangle$, which means the the image space of v is 6 times 6 and the temporal interval starts at 0 and ends at 3. The fragment f in the example is defined as a triple $\langle \langle (2, 2), (4, 4) \rangle, [1, 2], \{a_{xscale}^1\} \rangle$. Note, that the example uses a rectangular fragment like in Definition 28. Furthermore the animation is indicated with the value 1 (which means a scaling to 100%). The video v and the fragment f are outlined in Figure 29. As defined, a function σ allows to get the image fragment for a given video fragment at any distinct time t . Assuming $t = 1.5$ it is straightforward to calculate the related image fragment using the animation a_{xscale} with the resulting fragment $f_{(1.5)} = \langle (1.5, 2), (4.5, 2) \rangle$. In the figure this is outlined by the (green) line at time 1.5.

As the relation between fragments from Definition 30 is already defined for media fragments (so image and video) I do not have to extend it. With this and the definition of video fragments I can then define also spatio-temporal functions.

Example 6 The aim of the example is to explain, how a spatio-temporal relation $r_{overlap} \in R$ with $\chi_{r_{overlap}} : P_{(v)} \times P_{(v)} \mapsto \{0, 1\}$ can enable checks for spatio-temporal overlap of two video fragments. First I define a overlap function for the spatial domain $r_{imageOverlap}$ analogous to the one in Example 4. So $r_{overlap}$ can be defined

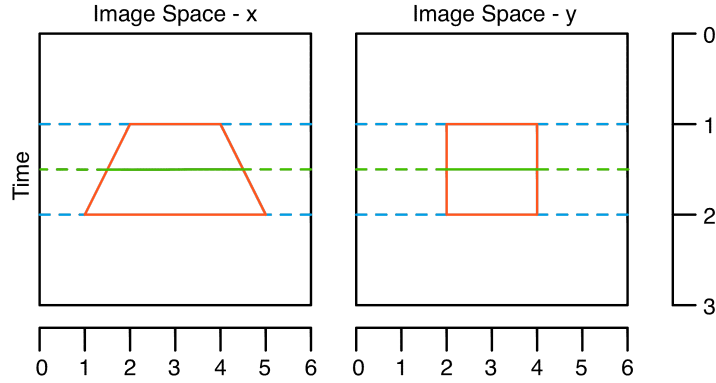


Figure 5.2: Example for an animated video fragment

as:

$$\chi_{r_{overlap}}(p_n, p_m) = \begin{cases} 1 & \exists t \in I : \chi_{r_{imageOverlap}}(p_{in} p_{im}) \neq \emptyset \\ & \text{for } p_{in} = \sigma(p_n, t); \quad p_{im} = \sigma(p_m, t). \\ 0 & \text{in any other case} \end{cases}$$

In this Section I gave a definition of a basic model for the description image and video fragment annotations. I therefore extended an existing model for layered image descriptions (DISIMA) to videos. Furthermore I introduced relation functions for both image and video fragments. This model is used along the following Chapters to give a well theoretical grounding.

5.2 SPARQL Algebra

In this Section I am going to summarize the SPARQL Algebra and its main concepts as defined in [HS13]. The translation algorithm from SPARQL syntax to SPARQL algebra is described in Chapter 8. To keep it simple I just include definitions that are used for later optimization approaches, which are **Basic Graph Patterns** and **Filters**.

5.2.1 SPARQL Abstract Query Syntax

Like described in Section 2, RDF-Terms are defined as a union of the sets of IRIs, Literals and Blank nodes:

Definition 37 (RDF Term) *Let I be the set of all IRIs. Let $RDF-L$ be the set of all RDF Literals. Let $RDF-B$ be the set of all blank nodes in RDF graphs.*

*The set of **RDF Terms**, $RDF-T$, is $I \cup RDF-L \cup RDF-B$.*

In RDF information is described via RDF triples. As SPARQL is meant as a triple pattern language there is a need of placeholders or query variables.

Definition 38 (Query Variable) A *query variable* is a member of the set V where V is infinite and disjoint from $RDF-T$.

Hence I can specify triples patterns as follows:

Definition 39 (Triple Pattern) A *triple pattern* is member of the set: $(RDF-T \cup V) \times (I \cup V) \times (RDF-T \cup V)$

A combination of triple patterns allows to describe subgraph matchings, which leads to the following definition:

Definition 40 (Basic Graph Pattern) A *Basic Graph Pattern* is a set of Triple Patterns.

Note, the empty graph pattern is a basic graph pattern, which is the empty set.

Definition 41 (Join) A *Join*(M_1, M_2) is a conjunctive combination of two sets M_1 and M_2 .

Definition 42 (Filter) A *Filter*(F, M) is the evaluation of a Filter F on a set M , the result is a boolean value.

Having this basic definitions I can go one step further and define, how SPARQL queries are evaluated.

Solution mapping

The solution of a SPARQL SELECT query is a mapping from a set of variables to a set of RDF terms.

Definition 43 (Solution Mapping) A *solution mapping* μ is a partial function $\mu : V \rightarrow RDF-T$. The domain of μ , $dom(\mu)$, is the subset of V where μ is defined. A *solution sequence* is a list of solutions, possibly unordered.

This means that the domain is exactly the set of all variables in the list of triples T . By substituting the empty nodes by URIs or Literals a subset of all triples $T' \in T$ can be defined, so that all triples in $\mu(T')$ exists in the graph that is queried, which is a solution for $BGP(T)$.

Definition 44 (Compatible Mappings) Two solution mappings μ_1 and μ_2 are compatible if, for every variable v in $dom(\mu_1)$ and in $dom(\mu_2)$, $\mu_1(v) = \mu_2(v)$.

This allows a definition of basic operations. For the matter of compactness I only consider Filter and Join here.

Definition 45 (Filter Operation) The *Filter Operation* of Filter F and multiset of solution mappings Ψ is defined as:

$$Filter(\Psi, F) = \{\mu \mid \mu \in \Psi \text{ and } F(\mu) \text{ is an expression that has an effective boolean value of true}\}.$$

Definition 46 (Join Operation) *The Join Operation of two multisets of solution mappings Ψ_1 and Ψ_2 is defined as:*

$$\text{Join}(\Psi_1, \Psi_2) = \{\mu_1 \cup \mu_2 \mid \mu_1 \in \Psi_1, \mu_2 \in \Psi_2 \text{ and } \mu_1 \text{ and } \mu_2 \text{ are compatible}\}.$$

5.2.2 SPARQL query string translation

In order to have a proper basis for the optimization process I will summarize the translation of SPARQL queries in SPARQL algebra expressions as outlined in the standard definition [HS13] in Section 18.2. In order to reduce the complexity I only consider SPARQL query elements, which are used within the optimization process described in a later Section, namely RDF terms, triples, basic graph patterns and filters. This results in the SPARQL algebra graph patterns BGP, Join and Filter. I sketch a simplified version of the translation process:

1. Expand Syntax Forms

SPARQL allows abbreviations for IRIs and triple patterns. In this first step, the abbreviations are expanded. Note, that for the matter of readability I will stay with abbreviations in the examples.

2. Translate Basic Graph Patterns

Groups of triples are translated to Basic Graph Pattern (BGP), so SPARQL syntax triples are transformed as follows:

```
?f2 ex:shows "Alice".
?f1 ex:shows "Bob" .

⇒

BGP (
  ?f2 ex:shows "Alice".
  ?f1 ex:shows "Bob" .
)
```

3. Translate Filters

As I only focus on patterns within one graph, the translation algorithm is quite straight forward:

Data: Let $G :=$ the empty graph pattern. Let $BGPS :=$ the set of Basic Graph Patterns Let $FS :=$ the set of filters.

Result: G

```
1
2 foreach element BGP in BGPS do
3   |  $G := \text{Join}(\text{BGP}, G)$ 
4 end
5
6 foreach element F in FS do
7   |  $G := \text{Filter}(F, G)$ 
8 end
```

4. Simplification Step

Joins of BGP and Z (the empty set) can be simplified to BGP. Additionally nested Joins of BGPs can be converted to one Join containing a set of BGPs. And list of BGPs can be combined to one BGP. The example shows the simplification step:

```
Join (
  BGP (?a :b ?c),
  Join (
    BGP (?c :x ?y),
    Join(
      BGP (?a :d :e),
      BGP (Z)
    )
  )
)
```

\Rightarrow

```
BGP (
  ?a :b ?c.
  ?c :x ?y.
  ?a :d :e.
)
```

Following the outlined translation process allows to transform SPARQL queries straight forward to SPARQL Algebra, which is a proper formal representation and build a solid basis for optimization steps, which I am going to target in Chapter 8.

5.3 Conclusion

In this Chapter I defined a model for a Multimedia query language based on DISIMA. I extended the model from salient image objects to temporal media objects by introducing time intervals. In addition I added mapping functions in order to support objects changing within these intervals. The model is used in later Section as a basis for more abstract definitions of Multimedia specific functions.

As the aim of this work is to extend the de-facto standard query language for the Semantic Web (SPARQL) I introduced an abstract definition of SPARQL query language, its query evaluation process as well as the algorithm to transform query strings to abstract queries.

In the next Chapters I will introduce SPARQL-MM as a extension of SPARQL to handle Multimedia queries and elaborate based on the basic model optimization processes.

Class and Property Model for Extensions

6.1 Design Principles

In this Chapter I define the core ontology that is necessary to describe spatio-temporal properties as well as relational and aggregational functions for SPARQL-MM. The ontology is an extension of the basic model described in [ABS⁺15c] (which is also the backbone for this Section). The ontology is a translation of the basic model described in Chapter 5 in a class model and serves as a pivot vocabulary used for matching existing standards and ontologies. This guarantees a most widely independence of the (abstract) SPARQL-MM function set described in Chapter 7, which makes it adaptable to many use cases even to non yet existing. The Chapter is structured as follows: First I introduce a class model that supports the whole basic model from the former Chapter. Secondly I provide mappings to the existing standards SVG basic shapes and Media Fragment URIs. The third point is an excursion, which proposes an extension of the Multimedia Fragment URIs Standard to more complex shapes and animations in order to support the introduced model as a whole.

6.2 Class Model

There are some vocabularies, which could be used here, complex ones like MPEG7 [MKP02] or very simple ones like Ninsuna [D⁺14]. With this ontology I tried to get the tradeoff between expressiveness and complexity. I am going to present a set of basic classes and properties in text that are necessary to describe the spatio-temporal functions I want to introduce. The ontology classes are aligned to the model in described in Section 5 and covers the theoretical model presented in Section 5.1 but might be adapted to upcoming issues. Figure 6.1 shows the main class model, a formal version can be found in the appendix B.2 (LMO). This Section summarizes the model that has been introduced in [ABS⁺15c].

In order to be aligned to the RDF model I use URIs for class and instant specification. The mapping of namespace prefixes to ontology URIs can be found in appendix B.1. I introduce two classes *lmo:SpatialEntity* and *lmo:TemporalEntity* (yellow) to describe spatial as well as temporal instances and two classes (green) to describe the actual values of the instances, whereby *lmo:Vector* is a superclass

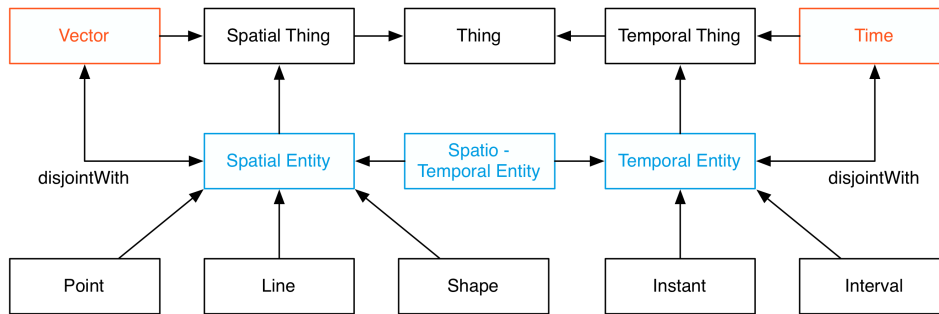


Figure 6.1: SPARQL-MM basic classes and relations

of all classes of multidimensional vectors and *lmo:Time* is a superclass of possible time representations (e.g. Normal Play Time NPT [SRL98]). As the reader can see, they are disjoint with each other. The following part introduces the model classes and a instants using RDF Notation 3 [BLC11]. The meaning of the class can be found in the object of the `rdfs:comment` relation.

Spatial Entity

The class *lmo:SpatialEntity* is defined by the following:

Listing 6.1: class Spatial Entity

```

lmo:SpatialEntity rdf:type owl:Class ;
  rdfs:label "Spatial Entity" ;
  rdfs:comment "A superclass of any spatial entities like
    point, line, polygone, curcle, etc." ;
  rdfs:subClassOf lmo:SpatialThing ;
  owl:disjointWith lmo:Time .
  
```

Temporal Entity

The class *lmo:TemporalEntity* is defined by the following:

Listing 6.2: class Temporal Entity

```

lmo:TemporalEntity rdf:type owl:Class ;
  rdfs:label "Temporal Entity" ;
  rdfs:comment "A superclass of any temporal entity
    like instant, interval, etc." ;
  rdfs:subClassOf lmo:TemporalThing ;
  owl:disjointWith lmo:Time ;
  owl:disjointUnionOf ( lmo:Instant lmo:Interval ) .
  
```

Spatio-Temporal Entity

The class *lmo:SpatioTemporalEntity* is defined as a class with exactly one spatial and exactly one temporal entity:

Listing 6.3: class SpatioTemporalEntity

```

lmo:SpatioTemporalEntity rdf:type owl:Class ;
  rdfs:label "Spatio-Temporal Entity" ;
  rdfs:comment "A class that relates to spatial
                and temporal features" .

rdfs:subClassOf
[ a owl:Restriction ;
  owl:onProperty lmo:hasSpatialEntity ;
  owl:cardinality "1"^^xsd:int
] ,
[ a owl:Restriction ;
  owl:onProperty lmo:hasTemporalEntity ;
  owl:cardinality
    "1"^^xsd:int
] .

lmo:hasSpatialEntity rdf:type owl:ObjectProperty,
                        owl:FunctionalProperty ;
  rdfs:label "hasSpatialEntity" ;
  rdfs:comment "The functional relation between a
                spatio-temporal entity and a spatial entity" ;
  rdf:domain lmo:SpatioTemporalEntity ;
  rdf:range lmo:SpatialEntity .

lmo:hasTemporalEntity rdf:type owl:ObjectProperty,
                            owl:FunctionalProperty ;
  rdfs:label "hasSpatialEntity" ;
  rdfs:comment "The functional relation between a
                spatio-temporal entity and a temporal entity" ;
  rdf:domain lmo:SpatioTemporalEntity ;
  rdf:range lmo:TemporalEntity .

```

The mapping to the abstract model from Chapter 5 is straightforward:

The *lmo:SpatialEntity* maps with the physical salient image object $p_i \in P$ from Definition 28 while the *lmo:TemporalEntity* maps with the physical salient temporal object $p_t \in P$ determined in Definition 31. Following this a *lmo:SpatioTemporalEntity* as a coalescence from spatial and temporal features represents a physical salient video object $p_v \in P$ from Definition 34.

Vector

The class *lmo:Vector* is defined by the following:

Listing 6.4: class Vector

```
lmo:Vector rdf:type owl:Class ;
  rdfs:label "Vector" ;
  rdfs:comment "A superclass for vectors." ;
  rdfs:subClassOf :SpatialThing ;
  owl:disjointWith lmo:SpatialEntity .
```

Time

The class *lmo:Time* is defined by the following:

Listing 6.5: class Time

```
lmo:Time rdf:type owl:Class ;
  rdfs:label "Time" ;
  rdfs:comment "A superclass for any kind of
              time specification." ;
  rdfs:subClassOf :TemporalThing ;
  owl:disjointWith lmo:TemporalEntity .
```

Animation

The class *lmo:Animation* is defined by the following:

Listing 6.6: class Time

```
lmo:Animation rdf:type owl:Class ;
  rdfs:label "Animation" ;
  rdfs:comment "A superclass for all animations." .
```

Animations must be linked to *lmo:SpatioTemporalEntity* via *lmo:animates*, which is defined as follows:

Listing 6.7: property animates

```
lmo:animates rdf:type owl:ObjectProperty ,
              owl:FunctionalProperty ;
  rdfs:label "animates" ;
  rdfs:comment "A property that links a lmo:Animation
              to a lmo:SpatioTemporalEntity." ;
  rdf:domain lmo:Animation ;
  rdf:range lmo:SpatialEntity .
```

Note, that *lmo:animates* is a functional property, so each animation instance does only belong to one and only one entity.

6.3 Instant Model

The class model allows us the definition of several subclasses and properties for non abstract entities. In this Section I give some examples and define a minimal set of spatial and temporal instants as well as animation.

Spatial Instants

Example 7 Figure 6.2 e.g. shows a class *Circle* with properties *hasXY* (describing the center point) and *hasRadius* (describing the radius). Note, that I abstract from real units (e.g. percentage, pixel, etc), which makes the model more flexible for function definition.

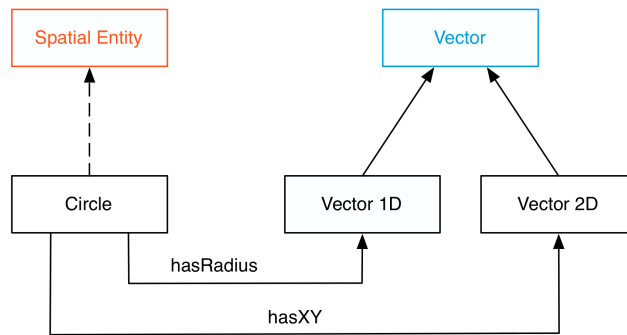


Figure 6.2: Sample object Circle

I defined a minimal set of spatial instants (Appendix B.2), which is inspired by SVG basic shapes¹ and contain:

lmo:Rectangle defines a rectangular shape based on left-upper corner $xy::Vector2D$, $width::Vector1D$, $height::Vector1D$, $rx::Vector1D$, and $ry::Vector1D$.

lmo:Circle defines a circle shape based on a center point $xy::Vector2D$ and a $radius::Vector1D$.

lmo:Ellipse defines a elliptical shape based on center point $xy::Vector2D$ and two radi $radiusX::Vector1D$ and $radiusY::Vector1D$.

lmo:Polygon defines a polygonal shape based on a start point $xy::Vector2D$ and a list of $n \in \mathbb{N}$ additional points (ordered by order number) $xy_i::Vector2D$ whereby $0 < i < n$. The shape is closed by a line from xy_{n-1} to xy .

¹SVG basic shapes: <https://www.w3.org/TR/SVG2/shapes.html>

Temporal Instants

Example 8 The example in Figure 6.3 shows a basic class *Interval*, which has two defined relations to class *Time* (start and endpoint).

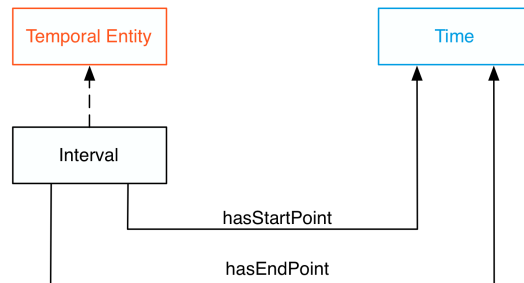


Figure 6.3: Sample object Interval

The minimal temporal instant set contain:

lmo:Instant defines a temporal instant with zero extent or duration based on a *position::Time*.

lmo:Interval defines a temporal interval based on a *start::Time* and an *end::Time*.

Spatio-Temporal Instants

The definition of a spatio-temporal instant is straightforward by linking an resource to exactly one spatial entity, e.g. a circle using the `hasSpatialEntity` relation and exactly one temporal entity e.g. an interval using the `hasTemporalEntity` relation.

Animated Instants

As defined instants of animations may linked to spatio-temporal entities.

Example 9 In Figure 6.4 a circle is animated by a linear scale during an interval.

As mentioned this instant list is minimal but straight forward to expand and thus can be adapted to any new usecases, e.g a extended version of media fragments like specified in a later Section. Therefore the minimal instant set does not contain animation instants.

6.4 Alignment to Existing Models

Like described on Chapter 2 there are some existing ontological models that allows to represent media metadata in RDF, varying in feature sets and granularity. I took 2 standardized variations, which are a) the W3C Media Ontology in combination with the W3C Media Fragment URIs and b)the Open Annotation Specification Model including simple and complex selectors (SVG).

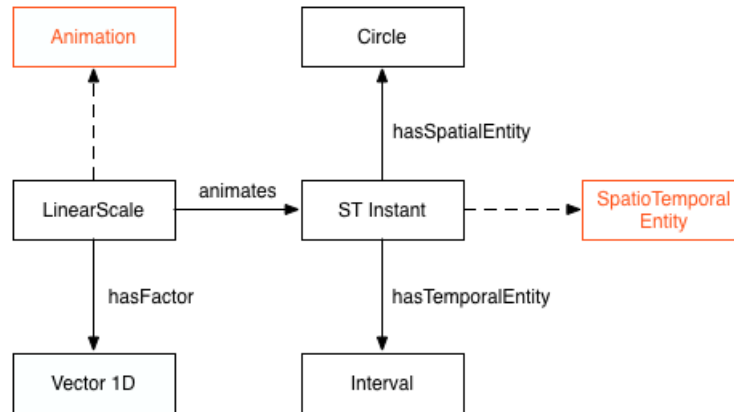


Figure 6.4: Sample animation

Alignment to Media Fragment URIs

Media Fragment URIs (MFU) can be aligned straightforward and is listed in Table 6.1.

MFU property	MMO property	Mapping Rules
xywh	lmo:Rectangle	mapping: $x = xy.x$; mapping: $y = xy.y$; mapping: width = w ; mapping: height = h; mapping: rx = <i>undefined</i> ; mapping: ry = <i>undefined</i> .
t	lmo:Instant	condition: begin time equals end time ; mapping: position = begin time .
t	lmo:Interval	condition: begin time not equals end time ; mapping: start = begin time ; mapping: end = end time .

Table 6.1: Mapping: Media Fragments URI

Alignment to SVG Basic Shapes with SVG Animations

The SVG basic shapes defined in [BRBL⁺18] contain six different graphical elements that are *circle*, *ellipse*, *line*, *polyline*, *polygon* and *rect*. As the basic spatial instant set of MMO is based on SVG the alignment of the models is straightforward and therefore not outlined in detail here.

For animation of SVG shapes a prominent approach is the usage of the *ani-*

mateTransform element². It contains four different types, namely translate, scale, rotate, and skew. As these are the basis for a Media Fragment URI extension proposal I am going to present in the next Section, a mapping/definition of MMO animations is spared here.

6.5 Excursion: Extending Media Fragments URIs

As described in the former Section, Media Fragment URI standard for media section identification convinces of the easy of use and its seamless integration into well known Web infrastructure. Nevertheless they are limited in expressiveness and thus only reflects a subset of the model presented in 5 (namely static spatial fragments). In this Section I propose an extension to support a broader set of fragment descriptions by extending the existing standard. The Section summarizes a work presented to the scientific community during the Linked Media Workshop (LIME) 2016 and published in [KK16].

The standards weaknesses of the current standard I aim to target are:

Imprecise spatial fragments: Spatial regions often cannot be sufficiently specified with rectangles. This fact may cause problems in calculating relations between fragments, e.g. if bounding boxes of spatial objects overlap, whereby the objects itself don't.

Lacking support for moving objects: Spatial regions in videos rarely stay on the same position during longer temporal sections (e.g. actors moving around within a scene). To sufficiently describe such scenarios many short spatial-temporal fragments have to be used, which leads to a big overhead in data transfer and recombination.

6.5.1 Media Fragment URI Extensions

In the following I present how Media Fragment URIs can be extended to various directions.

Shape Extension

Currently Media Fragment URI's spatial dimension is limited to rectangular shapes (*xywh*). An extension to basic geometric shapes, like circles, ellipses, etc. would allow a more fine-grained fragment description. In [ABS⁺15a] I recommended, inspired by SVG Basic Shape specification in [Fer01] four shapes in addition or substitution to *xywh*:

Rectangle: `rect=x,y,w,h[,rx,ry]`

²SVG Animation Transform Element:

<https://svgwg.org/specs/animations/#AnimateTransformElement>

The integers denote x , y , width, height and (optionally) the x and y radius (rx and ry) of the ellipse used to round off the corners of the rectangle respectively.

Circle: `circle=x,y,r`

The integers denote x and y as the center of the circle and r as the radius.

Ellipse: `ellipse=cx,cy,rx,ry`

The integers denote cx , cy (the center of the ellipse) and rx , ry (the radius of the ellipse).

Polygon: `polygon=x1,y1*(,xn,yn)`

The value is composed by $2*n$ comma-separated integers (with $n \in N$). The integers denote $x1$, $y1$ as starting point and xn , yn as points on the polyline that borders the polygon; the polygon is closed.

The value is an optional format `pixel:` or `percent:`, the defaulting format is `pixel`. I give an example for an ellipse fragment in Figure 6.5, all the other shapes work accordingly.



Figure 6.5: Shape Extension: `image.jpg#ellipse=percent:50,52,15,22`

Transformation Extension

Even with this shape extensions the identification of spatial fragments is limited. Additionally, with regard to further extensions for example animations, a proper representation of shape transformation and translation is lacking. To overcome this limitations I introduce four shape transformations:

Translate: `translate=x[,y]` The integers denote x for horizontal and y (optionally) for vertical translation.

Scale: `scale=x[,y]` The integers denote x for horizontal and y (optionally)

for vertical scale.

Rotate: `rotate=a[,x,y]` The integers denote `a` as rotation angle and `x,y` as center of rotation. The default center is denoted by the center of the bounding box of the region to rotate.

Skew: `skew=x[,y]` The integers denote `x` for horizontal and `y` (optionally) for vertical skew.

Transformations in Media Fragment URIs are only considered if one and only one shape is defined. Transformations can be stacked. If a transformation type occurs more than once, only the first value is considered. Like for shapes, the value has an optional format `pixel:` or `percent:`, whereby the defaulting format is pixel. Figure 6.6 shows a transformed shape.



Figure 6.6: Transformation Extension: `image.jpg#rect=230,100,80,55&rotate=25`

Animated Transformation Extension

The static shapes and transformations mainly focus on still images. But spatial fragments often needs to transform over time e.g. for videos or interactive charts. I introduce animated transformations as temporal extension to the static in order to satisfy this need.

Animated Translate: `aTranslate=d1,x1[,y1]*[;dn,xn[,yn]]`

The value is an optional format `pixel:` or `percent:` (defaulting to pixel) plus a semicolon-separated list of comma-separated numbers. The first number of each number set (`d.`) is defined as duration and may be defined in percent (for videos) or milliseconds (for images). The other numbers represent the translation as specified.

Animated Scale: `aScale=d1,x1[,y1]*[;dn,xn[,yn]]`

Analogous to animated translate.

Animated Rotate: `aRotate=d1,r1[,x1,y1]*[;dn,rn[,xn,yn]]`

Analogous to animated translate.

Animated Skew: `aSkew=d1,x1[,y1]*[;dn,xn[,yn]]`

Analogous to animated translate.

Animated Transformations in Media Fragment URIs are only considered if one and only one shape is defined. Animated transformations can be stacked. If an animated transformation type occurs more than once, only the *first* value is considered. Figure 6.7 shows how a spatial fragment is animated over time in both scale and translation. In this case there is no transformation until 45% of the temporal fragment (3.5 seconds overall), in the next 10% of time the shape translates to south-west and scales to 70%. During the remaining time there is no transformation.



Figure 6.7: Animated Transformation Extension: `video.mp4#ellipse=330,100,50,80&aTranslate=0.45,0,0;0.1,-50,50&aScale=0.45,1;0.1,0.7&t=0.5,4`

6.5.2 Related approaches for Media Fragment URI extensions

On <http://github.com/tomayac/dynamic-media-fragments> the author describes, how spatial media fragments *xywh* can be extended to temporal dynamics by stringing together quadruples, whereby each one identifies a rectangular shape. The shapes are equally distributed in time (represented by a temporal fragment or the whole video play time). The approach is aligned with CSS transitions and such fits smoothly into current browser animation implementations. To extend the approach from equal to fixed distribution, the author suggested to extend the quadruples to a micro syntax representing the time in percentage. Another interesting approach is described on <https://github.com/oaubert/mediafragment-prototype>. The author introduces a new fragment parameter *shape*, which represents the spatial dimension and utilizes SVG path definition as values. The main difference to my approach is the fact that shapes are not first class entities (defined by a name-value

pair) but are values of one spatial dimension descriptor. For the temporal dynamic the author introduces a trajectory parameter with an SVG path value, which makes the defined shape follow the given path within a given temporal fragment. The author also suggest to extend both the shape and the trajectory values to basic SVG shapes.

6.5.3 Mapping Media Fragments URI Extensions to the Model

As mentioned above, the LMO fits partly SVG basic shapes. The classes can aligned to the recommended extensions of Media Fragment URIs outlined in Table 6.2.

MFU-EXT property	MMO-EXT property	Mapping Rules
rect	lmo:Rectangle	mapping: x = xy.x ; mapping: y = xy.y ; mapping: width = w ; mapping: height = h ; mapping: rx = rx ; mapping: ry = ry.
circle	lmo:Circle	mapping: x = xy.x ; mapping: y = xy.y ; mapping: radius = r .
ellipse	lmo:Ellipse	mapping: x = xy.x ; mapping: y = xy.y ; mapping: rx = rx ; mapping: ry = ry.
polygon	lmo:Polygon	mapping: x = xy.x ; mapping: y = xy.y ; mapping: x _i = xy _i .x ; mapping: y _i = xy _i .y .

Table 6.2: Mapping: Media Fragments URI Extension

Note, the mapping of the basic version of Media Fragment URIs is analogous to the mapping defined in Section 6.4. As annotations is not considered in this thesis and not part of the LMO, a mapping cannot be done here but may be part of further work.

6.6 Conclusion

In this Chapter I introduced a basic RDF model that covers the theoretical model in Chapter 5. The model is a metamodel for (animated) spatio-temporal media fragments, which enables a standard-independent specification for media fragment relations in the next Chapter. To give an alignment between existing models and the metamodel, I outlined mapping tables to two existing standards, namely SVG basic shapes and Media Fragment URIs. Furthermore I showed how Media Fragment URIs could be extended to broaden the mapping to the metamodel and thus find more acceptance in the Web community. In the next Chapter I am going to introduce SPARQL-MM as an Multimedia extension for querying the Web of Linked Media.

Part IV

Multimedia Extension for SPARQL

Sparql-MM Functions

SPARQL is the de facto standard query language for the Semantic Web. As exhaustively described in former Sections it lacks Multimedia facilities. In this Chapter I will extend SPARQL to spatio-temporal functions.

7.1 Extension Functions

In the following I consider 2 kind of functions, spatial and temporal, that are both again separated in 3 different types, namely relations, aggregations and features. A definition of these 6 types is given in Table 7.1. It has to be mentioned that there might be overlaps between one or more types (e.g. spatio-temporal overlap), which are discussed later. To provide the required functionality a theoretical based model for spatial and temporal relations is necessary.

	Spatial
Relation	Type: <i>SR</i> how 2 spatial objects relate to each other (e.g. A right beside B)
Aggregation	Type: <i>SA</i> how 2 or more spatial objects can be aggregated with each other (e.g. intersection of A and B)
Feature	Type: <i>SF</i> features of spatial objects (e.g. area of A)
	Temporal
Relation	Type: <i>TR</i> how 2 temporal objects relate to each other (e.g. A before B)
Aggregation	Type: <i>TA</i> how 2 or more temporal objects can be aggregated with each other (e.g. intermediate of A and B)
Feature	Type: <i>TF</i> features of temporal objects (e.g. duration of A)

Table 7.1: SPARQL-MM Function Types

7.2 Spatial Relations, Aggregations and Properties

Relations between spatial objects can be separated in three main classes, which are a) topological relations (how two spatial objects relates to each other, e.g. contains), b) directional relations (how a spatial object a relates to a spatial object b e.g. rightBeside), and distance relations (the attributes of the relation itself e.g. nearby). In the following I describe models for topological and directional relations and specify SPARQL-MM functions based on these models. Currently distance relations are not considered, because in comparison to the topological and directional relations they are fuzzy and therefore does not seamlessly integrate into SAPRQL (unless e.g. extending SPARQL to fuzzy logic).

7.2.1 Topological Relations

A standard model to describe relations between spatial objects in a 2 dimensional geometric model is the Dimensionally Extended nine-Intersection Model (DE-9im) [CFvO93]. The model is based on a 3x3 intersection matrix (Clementini-Matrix,

$$DE9IM(a, b) = \begin{bmatrix} \dim(I_a \cap I_b) & \dim(I_a \cap B_b) & \dim(I_a \cap E_b) \\ \dim(B_a \cap I_b) & \dim(B_a \cap B_b) & \dim(B_a \cap E_b) \\ \dim(E_a \cap I_b) & \dim(E_a \cap B_b) & \dim(E_a \cap E_b) \end{bmatrix} \quad (7.1)$$

Figure 7.1: Clementini-Matrix

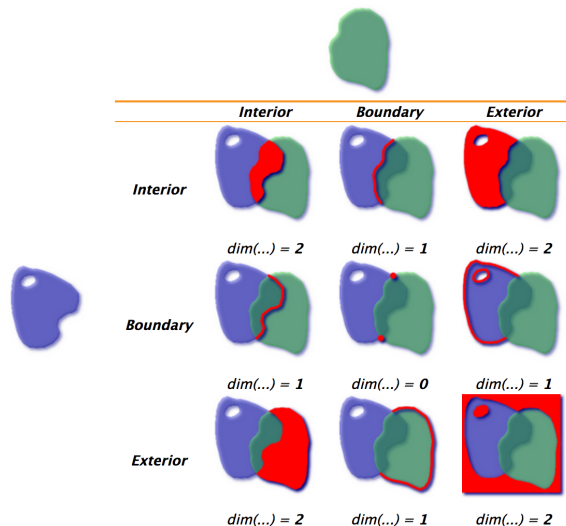


Figure 7.2: Example for DE9im

Figure 7.1), which allows to specify the spatial relation of two geometric objects according to interior (I), boundary (B) and exterior (E). The result of `dim(x)` is the maximum value of all matching intersection pattern, whereby -1 is the value of \emptyset , 0 the dimension of point intersection, 1 the line intersection and 2 the dimension of area intersection. Additionally * represents a wildcard and indicates that the actual value does not have any influence on the current problem.

Figure 7.2¹ shows the matrix and the dimension results for two example shapes. To get a compact string representation it is common to concatenate the pattern values from left-to-top and from top-to-bottom. Hence, the resulting pattern string for the example is 212101212. As the raw model is complex to use, a set of spatial predicates has been defined, which describes well known object relations. Each predicate is mapped to one or more matrix forms / pattern strings, so a clear semantic is bound to former fuzzy natural language terms. Therefore, the spatial predicates build a proper grounding for SPARQL-MM spatial relations. It is obvious that the pattern strongly depend on the dimension of the involved geometric objects. To reduce the amount patterns the resulting dimension is reduced to values {T,F,*}, if it does not change the underlying semantics.

For example, the *contains* predicate is described by the pattern T*****FF*, which means that:

- (a) $\dim[I(a) \cap I(b)]$ is true (a and b has an interior in common)
- (b) $\dim[E(a) \cap I(b)]$ is false (the exterior of a and the interior of b has nothing in common)
- (c) $\dim[E(a) \cap B(b)]$ is false (the boundary of b and the exterior of a has nothing in common)
- (d) all the other fields of the matrix does not matter

The complete list of spatial predicates (together with their DE9im patterns) that are supported by SPARQL-MM are: In the following I list the spatial predicates with their patterns:

equals [T*F**FFF*]

disjoint [FF*FF*****]

touches [FT*****], [F**T*****], [F***T*****]

contains [T*****FF*]

covers [[T*****FF*], [*T*****FF*], [***T**FF*], [****T*FF*]

¹DE9im:

http://postgis.org/documentation/manual-svn/using_postgis_dbmanagement.html

intersects [T*****], [T*****], [***T*****], [****T*****]; logical inversion of disjoint

within [T**F**F***]; within(a,b) = contains(b,a)

coveredBy [T**F**F***], [*TF**F***], [**FT**F***], [**F**TF***]; coveredBy(a,b) = covers(b,a)

crosses [T*T*****] for $\dim(a) < \dim(b)$; [T*****T**] for $\dim(a) > \dim(b)$; [0*****] for $\dim(\text{any})$

overlaps [T*T***T**] for $\dim=0$ or $\dim=2$; [1*T***T**]for $\dim=1$.

7.2.2 Directional Relations

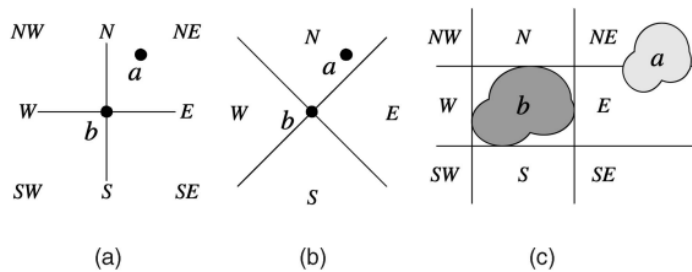


Figure 7.3: Models of directional relations

Like for topological relations, predicates have to be found for directional relations to specify proper functions for SPARQL-MM. Directional relations describe, how a primary object a is placed relative to a reference object b based on a coordinate system (for example, object a is south of object b). There are several models, which describe directional relations in different spaces. Figure 7.3 (taken from [S⁺07]) shows 3 of the most common ones. 7.3a (projection-based model) and 7.3b (cone-based model) are defining relations between punctual objects but can be easily extended to spatial object by approximating an extended representative point (e.g. the centroid). Both models partition the space around the reference object b into a number of mutually exclusive areas. Other models like 7.3c (Projection-based Directional Relation model, PDR) extend the definition to spatial objects, which provides more preciseness and expressivity but increases the number of relations that can be expressed (for PDR there are 511 possible relations), which disqualifies it as basis for directional predicates. For SPARQL-MM I decided to take the projection-based model because a) it is easy to understand for users, b) it allows us to specify intuitive predicate names and c) it can be calculated very efficiently (e.g. by indexing the centroid for any spatial object). To make it even more intuitive, I refrain from using words from the geographical domain (*Modelname*) and replaced it with daily used words (name of *Function*) as follows:

Abbr.	Modelname	Function
W	West	$leftBeside(a, b) = a.x < b.x$
E	East	$rightBeside(a, b) = a.x > b.x$
N	North	$above(a, b) = a.y > b.y$
S	South	$below(a, b) = a.y < b.y$
NW	Northwest	$leftAbove(a, b) = leftBeside(a, b) \wedge above(a, b)$
NE	Northeast	$rightAbove(a, b) = rightBeside(a, b) \wedge above(a, b)$
SW	Southwest	$leftBelow(a, b) = leftBeside(a, b) \wedge below(a, b)$
SE	Southeast	$rightBelow(a, b) = rightBeside(a, b) \wedge below(a, b)$

A further spatial relation function type are distance functions (e.g. nearby), which are not considered in this thesis as they can simulated by using accessors like `getCenter` and mathematical functions provided by SPARQL. Nevertheless, this simulation may be not well optimizable regarding query evaluation and thus can be integrated in SPARQL-MM in further works.

7.3 Temporal Relations, Aggregations and Properties

precedes	meets	overlaps	finished by	contains	starts	equals	started by	during	finishes	overlap-ped by	met by	preceded by
p	m	o	F	D	s	e	S	d	f	O	M	P

Figure 7.4: Allen's 13 basic temporal relations

The standard model for temporal relation was introduced by Allen's interval algebra for temporal reasoning [All83b]. The algebra defines thirteen basic relations between two time intervals, whereby a time point can be interpreted as a interval with duration 0. Figure 7.4² illustrates these relations. It has to be mentioned that 6 pairs of relations are converse, which is represented through upper-lower case letters (e.g. *precedes* p is converse to *preceded by* P).

²Allan: <http://www.ics.uci.edu/~alspaugh/cls/shr/allen.html>

7.4 Spatio-Temporal Property and Function Specification

Based on the foundations in 7.2, the functions for SPARQL-MM are:

Function SR-a: `spatialEquals`

```
xsd:boolean mm:spatialEquals (  
    lmo:SpatialEntity a  
    ,lmo:SpatialEntity b  
)
```

Function SR-b: `disjoint`

```
xsd:boolean mm:disjoint (  
    lmo:SpatialEntity a  
    ,lmo:SpatialEntity b  
)
```

Function SR-c: `touches`

```
xsd:boolean mm:touches (  
    lmo:SpatialEntity a  
    ,lmo:SpatialEntity b  
)
```

Function SR-d: `spatialContains`

```
xsd:boolean mm:spatialContains (  
    lmo:SpatialEntity a  
    ,lmo:SpatialEntity b  
)
```

Function SR-e: `covers`

```
xsd:boolean mm:covers (  
    lmo:SpatialEntity a  
    ,lmo:SpatialEntity b  
)
```

Function SR-f: `intersects`

```
xsd:boolean mm:intersects (  
    lmo:SpatialEntity a  
    ,lmo:SpatialEntity b  
)
```

Function SR-g: within

```
xsd:boolean mm:within (
  lmo:SpatialEntity a
  ,lmo:SpatialEntity b
)
```

Function SR-h: coveredBy

```
xsd:boolean mm:coveredBy (
  lmo:SpatialEntity a
  ,lmo:SpatialEntity b
)
```

Function SR-i: crosses

```
xsd:boolean mm:crosses (
  lmo:SpatialEntity a
  ,lmo:SpatialEntity b
)
```

Function SR-j: spatialOverlaps

```
xsd:boolean mm:spatialOverlaps (
  lmo:SpatialEntity a
  ,lmo:SpatialEntity b
)
```

Some of the methods are overloaded, which means that they are used for temporal as well as spatial relations. Therefore I added a prefix *spatial* to some of them.

Example 10 *The query in Listing 7.1 returns all images where fragments containing Alice and Bob intersect.*

Listing 7.1: Example for SPARQL-MM topological relation function

```
SELECT ?image WHERE {
  ?image ma:hasFragment ?f1 .
  ?image ma:hasFragment ?f2 .
  ?f1 dc:subject 'Alice' .
  ?f2 dc:subject 'Bob' .
  FILTER mm:intersects(?f1, ?f2)
}
```

In addition to the functions I defined a set of topological aggregation functions. Currently only rectangle shapes are considered, so I took a reasonable subset, which has to be extended in the further work. This functions are:

Function SA-a: boundingBox

```
lmo:SpatialEntity mm:boundingBox (
  lmo:SpatialEntity a
  ,lmo:SpatialEntity b
)
```

Function SA-b: intersection

```
lmo:SpatialEntity mm:intersection (
  lmo:SpatialEntity a
  ,lmo:SpatialEntity b
)
```

Example 11 *The query in Listing 7.2 returns a bounding box fragment that includes fragments of Alice and Bob where both intersect.*

Listing 7.2: Example for SPARQL-MM spatial aggregation function

```
SELECT (mm:boundingBox(?f1,?f2) AS ?bbox) WHERE {
  ?image ma:hasFragment ?f1 .
  ?image ma:hasFragment ?f2 .
  ?f1 dc:subject 'Alice' .
  ?f2 dc:subject 'Bob' .
  FILTER mm:intersects(?f1, ?f2)
}
```

It has to be mentioned that the implementation of the two aggregation functions does not have to be delimited to two parameters but as both functions are assoziative, commutative and distributive they can be nested.

Function SR-k: leftBeside

```
xsd:boolean mm:leftBeside (
  lmo:SpatialEntity a
  ,lmo:SpatialEntity b
)
```

Function SR-l: rightBeside

```
xsd:boolean mm:rightBeside (
  lmo:SpatialEntity a
  ,lmo:SpatialEntity b
)
```

Function SR-m: above

```
xsd:boolean mm:above (  
    lmo:SpatialEntity a  
    ,lmo:SpatialEntity b  
)
```

Function SR-n: below

```
xsd:boolean mm:below (  
    lmo:SpatialEntity a  
    ,lmo:SpatialEntity b  
)
```

Function SR-o: leftAbove

```
xsd:boolean mm:leftAbove (  
    lmo:SpatialEntity a  
    ,lmo:SpatialEntity b  
)
```

Function SR-p: rightAbove

```
xsd:boolean mm:rightAbove (  
    lmo:SpatialEntity a  
    ,lmo:SpatialEntity b  
)
```

Function SR-q: leftBelow

```
xsd:boolean mm:leftBelow (  
    lmo:SpatialEntity a  
    ,lmo:SpatialEntity b  
)
```

Function SR-r: rightBelow

```
xsd:boolean mm:rightBelow (  
    lmo:SpatialEntity a  
    ,lmo:SpatialEntity b  
)
```

Example 12 *The query in Listing 7.3 returns all images where Alice appears right beside Bob whereby both may not intersect.*

Listing 7.3: Example for SPARQL-MM spatial directional function

```
SELECT DISTINCT(?image) WHERE {
  ?image ma:hasFragment ?f1 .
  ?image ma:hasFragment ?f2 .
  ?f1 dc:subject 'Alice' .
  ?f2 dc:subject 'Bob' .
  FILTER mm:rightBeside(?f1, ?f2)
  FILTER mm:disjoint(?f1, ?f2)
}
```

Function TR-a: precedes

```
xsd:boolean mm:precedes (
  lmo:TemporalEntity a
  ,lmo:TemporalEntity b
)
```

Function TR-b: meets

```
xsd:boolean mm:meets (
  lmo:TemporalEntity a
  ,lmo:TemporalEntity b
)
```

Function TR-c: overlaps

```
xsd:boolean mm:overlaps (
  lmo:TemporalEntity a
  ,lmo:TemporalEntity b
)
```

Function TR-d: finishedBy

```
xsd:boolean mm:finishedBy (
  lmo:TemporalEntity a
  ,lmo:TemporalEntity b
)
```

Function TR-e: contains

```
xsd:boolean mm:contains (  
    lmo:TemporalEntity a  
    ,lmo:TemporalEntity b  
)
```

Function TR-f: starts

```
xsd:boolean mm:starts (  
    lmo:TemporalEntity a  
    ,lmo:TemporalEntity b  
)
```

Function TR-g: equals

```
xsd:boolean mm>equals (  
    lmo:TemporalEntity a  
    ,lmo:TemporalEntity b  
)
```

Function TR-h: startedBy

```
xsd:boolean mm:startedBy (  
    lmo:TemporalEntity a  
    ,lmo:TemporalEntity b  
)
```

Function TR-i: during

```
xsd:boolean mm:during (  
    lmo:TemporalEntity a  
    ,lmo:TemporalEntity b  
)
```

Function TR-j: finishes

```
xsd:boolean mm:finishes (  
    lmo:TemporalEntity a  
    ,lmo:TemporalEntity b  
)
```

Function TR-k: overlapedBy

```
xsd:boolean mm:overlapedBy (
  lmo:TemporalEntity a
  ,lmo:TemporalEntity b
)
```

Function TR-l: metBy

```
xsd:boolean mm:metBy (
  lmo:TemporalEntity a
  ,lmo:TemporalEntity b
)
```

Function TR-m: precededBy

```
xsd:boolean mm:precededBy (
  lmo:TemporalEntity a
  ,lmo:TemporalEntity b
)
```

Example 13 *The query in Listing 7.4 returns fragment of Alice that are temporal overlapping with fragments of Bob.*

Listing 7.4: Example for SPARQL-MM temoral relation function

```
SELECT ?f1 WHERE {
  ?image ma:hasFragment ?f1 .
  ?image ma:hasFragment ?f2 .
  ?f1 dc:subject 'Alice' .
  ?f2 dc:subject 'Bob' .
  FILTER mm:overlaps(?f1, ?f2)
}
```

Like for topological relations I defined a reasonable set of topological aggregation functions, which are:

Function TA-a: boundingBox

```
lmo:TemporalEntity mm:boundingBox (
  lmo:TemporalEntity a
  ,lmo:TemporalEntity b
)
```


Function TA-b: intersection

```
lmo:TemporalEntity mm:intersection (
  lmo:TemporalEntity a
  ,lmo:TemporalEntity b
)
```

Function TA-c: intermediate

```
lmo:TemporalEntity mm:intermediate (
  lmo:TemporalEntity a
  ,lmo:TemporalEntity b
)
```

It has to be mentioned that TA-a and TA-b can be nested as they are assoziativ, commutative and distrubtive, whereas TA-c is restricted to two parameters, because it is not assoziativ.

Example 14 *The query in Listing 7.5 returns the temporal intersection of fragment of 2 overlapping fragments, whereby one contains Alice and the second one contains Bob.*

Listing 7.5: Example for SPARQL-MM temporal aggregaion function

```
SELECT (mm:intersection(?f1, ?f2) AS ?intersection) WHERE{
  ?image ma:hasFragment ?f1 .
  ?image ma:hasFragment ?f2 .
  ?f1 dc:subject 'Alice' .
  ?f2 dc:subject 'Bob' .
  FILTER mm:overlaps(?f1, ?f2)
}
```

Now I introduce three different kinds of accessor functions:

Spatial Accessor Features (SF):

In order to support pixels and percent, I introduce lmo:unitNumber, which is defined as:

Listing 7.6: Defintion: lmo:unitNumber

```
lmo:unitNumber = [ unit ":" ] 1*number
number         = INTEGER | DECIMAL
unit           = %x70.69.78.65.6C    ; "pixel"
               / %x70.65.72.63.65.6E.74 ; "percent"
```

Function SF-a: getArea

```
lmo:unitNumber mm:getArea (  
  lmo:SpatialEntity entity  
)
```

This function returns the area of a spatial entity as unit number. If the property is not a spatial entity, null is returned.

Function SF-b: getBoundingBox

```
lmo:Rectangle mm:getBoundingBox (  
  lmo:SpatialEntity entity  
)
```

This function returns the rectangular bounding box for a spatial entity. If the property is not a spatial entity, null is returned.

Function SF-c: getXY

```
lmo:Point mm:getXY (  
  lmo:SpatialEntity entity  
)
```

This function returns the left upper point of the bounding rectangle of a spatial entity. If the property is not a spatial entity, null is returned.

Function SF-d: getHight

```
lmo:unitNumber mm:getHight (  
  lmo:SpatialEntity entity  
)
```

This function returns the height of the bounding box for a spatial entity as unit number. If the property is not a spatial entity, null is returned.

Function SF-e: getWidth

```
lmo:unitNumber mm:getWidth (  
  lmo:SpatialEntity entity  
)
```

This function returns the width of the bounding box for a spatial entity as unit number. If the property is not a spatial entity, null is returned.

Function SF-f: getCenter

```
lmo:Point mm:getCenter (
  lmo:SpatialEntity entity
)
```

This function returns the center of the spatial entity as point. If the property is not a spatial entity, null is returned.

Example 15 *The query in Listing 7.7 returns fragments containing Alice plus its width.*

Listing 7.7: Example for SPARQL-MM spatial accessor function

```
SELECT ?f1, (mm:getWidth(?f1) AS ?width) WHERE {
  ?image ma:hasFragment ?f1 .
  ?f1 dc:subject 'Alice' .
}
```

Temporal Accessor Features (TF):

Currently SPARQL-MM only supports Normal Play Time (NPT) as specified in the Media Fragment URI standard.

Function TF-a: getDuration

```
xsd:decimal mm:getDuration (
  lmo:TemporalEntity entity
)
```

This functions returns the duration of a temporal entity as decimal. If the property is not a spatial entity, null is returned.

Function TF-b: getStart

```
xsd:decimal mm:getStart (
  lmo:TemporalEntity entity
)
```

This functions returns the start time of a temporal entity as decimal. If the property is not a spatial entity, null is returned.

Function TF-c: getEnd

```
xsd:decimal mm:getEnd (
  lmo:TemporalEntity entity
)
```

This functions returns the end time of a temporal entity as decimal. If the property is not a spatial entity, null is returned.

Example 16 *The query in Listing 7.8 returns fragments containing Alice that starts at 20 seconds or later.*

Listing 7.8: Example for SPARQL-MM temporal accessor function

```
SELECT ?f1 WHERE {
  ?image ma:hasFragment ?f1 .
  ?f1 dc:subject 'Alice' .
  FILTER(10 >= mm:getStart(?f1))
}
```

In order to get information about the general information regarding media fragments and `lmo:unitNumbers` I define Generic Accessor Features (GF):

Function GF-a: `isMediaFragment`

```
xsd:boolean mm:isMediaFragment (
  xsd:string entity
)
```

This functions returns true if the string can be parsed to a `MediaFragment`, false otherwise.

Function GF-b: `isMediaFragmentURI`

```
xsd:boolean mm:isMediaFragmentURI (
  xsd:string entity
)
```

This functions returns true if the string can be parsed to a `MediaFragmentURI`, false otherwise.

Function GF-c: `hasTemporalFragment`

```
xsd:boolean mm:hasTemporalFragment (
  xsd:string entity
)
```

This functions returns true if the string can be parsed to a `MediaFragment`(URI) and has a `Temporal Fragment`, false otherwise.

Function GF-c: hasSpatialFragment

```
xsd:boolean mm:hasSpatialFragment (  
  xsd:string entity  
)
```

This functions returns true if the string can be parsed to a MediaFragment(URI) and has a Spatial Fragment, false otherwise.

Function GF-d: toPixel

```
xsd:decimal mm:toPixel (  
  lmo:unitNumber number,  
  xsd:decimal conversionNumber (OPTIONAL)  
)
```

This functions returns the unitNumber converted to pixels based in conversion number. If no conversion number is given, only values for pixel units are returned, null otherwise.

Function GF-e: toPixel

```
xsd:decimal mm:toPixel (  
  lmo:SpatialFragment shape,  
  lmo:Rectangle conversionShape (OPTIONAL)  
)
```

This functions returns the Spatial Fragment converted to pixels based in conversion shape. If no conversion shape is given, only values for pixel fragments are returned, null otherwise.

Function GF-f: toPercent

```
xsd:decimal mm:toPercent (  
  lmo:unitNumber number,  
  xsd:decimal conversionNumber (OPTIONAL)  
)
```

This functions returns the unitNumber converted to percent based in conversion number. If no conversion number is given, only values for percent units are returned, null otherwise.

Function GF-g: toPercent

```
xsd:decimal mm:toPercent (
  lmo:SpatialFragment shape,
  lmo:Rectangle conversionShape (OPTIONAL)
)
```

This functions returns the Spatial Fragment converted to percent based in conversion shape. If no conversion shape is given, only values for percent fragments are returned, null otherwise.

Example 17 *The query in Listing 7.9 returns all media fragments containing Alice.*

Listing 7.9: Example for SPARQL-MM general function

```
SELECT ?f1 WHERE {
  ?f1 dc:subject 'Alice' .
  FILTER mm:isMediaFragmentURI(?f1)
}
```

7.5 Conclusion

In this Section I defined relational, aggregational and accessor functions for spatial and temporal operations in SPARQL. All of the described functions are part of the Multimedia extension for SPARQL called SPARQL-MM. As a basis for the function definition I took well known temporal and spatial relation models, which has been explained in detail in this Section, too. In addition I took the requirements gathered in Section 4 as a minimum baseline that has to be fulfilled by the extension. The comparison of SPARQL plus SPARQL-MM extension with the requirement table outlined in Table 3.2.3 shows that some of the features are already overlapping.

Spatial Operations (SO) are supported by topological and relational functions as well as spatial aggregators and accessors. Directional relations can be an extension in the future.

Temporal Operations (TO) are introduced with SPARQL-MM. They include functions aligned to the temporal algebra, and, like spatial operations, aggregators and accessors.

Metadata Operations (MO) are matched by the basic SPARQL standard as it allows many ways to select and filter metadata represented in RDF.

The current function set of SPARQL-MM misses functions for *Multimedia similarity operations (MS)* and the possibility to weight results (*Weighting (W)*). This will be discussed in Chapter 10.

The function set of SPARQL-MM extends SPARQL to Multimedia operations. As Multimedia repositories may include a large set of media assets with many spatial and temporal annotations on each, this may lead to data sets with a huge amount of RDF triples. It is obvious that data access in a reasonable amount of time is one major goal for such media information systems. In the next Section I will investigate how the optimization of SPARQL-MM queries can be integrated in existing SPARQL query optimization processes.

Optimization

8.1 SPARQL Filter Optimization

It is obvious that a performant evaluation of SPARQL queries is mandatory to keep the performance of retrieval systems on a high level. With the introduction of SPARQL-MM's media fragment filter functions and the naive reference implementation provided in [KSK15], it becomes clear that common pattern based optimization algorithms fail regarding (high selective) filters, which is the motivation for this Chapter - an optimization of SPARQL-MM filter evaluations.

The optimization task can be split in two steps:

1. Minimize cost for filter evaluation by using specific indexes for spatio-temporal media fragments, and
2. Optimize SPARQL query evaluation plans.

To target the first aim I describe a spatio-temporal index and discuss how it fits the current SPARQL-MM function set and how it can be used for further adoptions like complex shapes and animations. For the second aim I show, where common optimization algorithms produces non-optimal query plans and propose an extension to filter aware cost calculation in order to overcome this problem.

8.1.1 Spatio-temporal Indexes

An index for optimizing the evaluation of SPARQL-MM filters can be divided in two parts, a) an optimal access of all fragments of a single media item and b) support for evaluating the actual filters, namely all spacial, temporal and general relations described in Chapter 7. The first part can be solved straight forward by mapping media assets to a numeric value (e.g. integer) and use a common index like *Hash* or *B-Tree*. For the second part there is a need for a multi-dimensional index, which has been elaborated from both areas Multimedia and geographical information systems (GIS). In the case of SPARQL-MM the index has to support partial match queries as well as range queries in two dimensions regarding spatial shapes and in three dimensions regarding spatio-temporal fragments. In literature there are two main types of multidimensional indexes, which are *hash-like* structures (e.g. GRID Files) and *tree* structures, whereby the first one are mainly used in the field of classical Data Warehousing. The second are widespread in GIS and Multimedia indexing. They have a lot of different approaches while the most prominent representative are R-tree and its derivatives [BS12].

R-tree

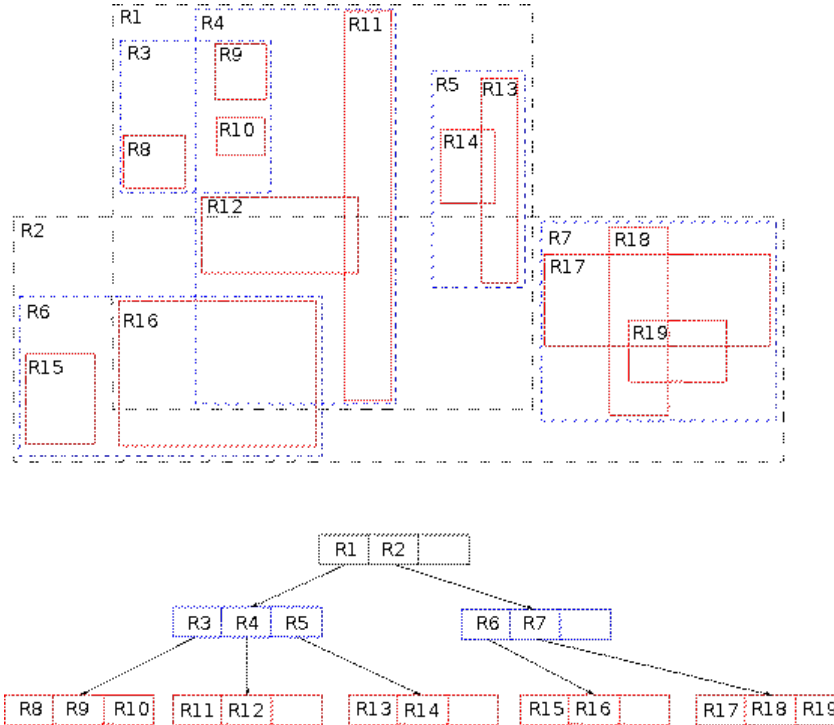


Figure 8.1: R-tree example

An R-tree [Gut84] is a balanced index structure for multidimensional data. The structure allows an efficient region and interval search for geometric objects. The R-tree is a dynamic structure, which allows insertion, updating and deletion of data. Like B-tree, R-tree consists of one root, non-leaf and leaf-nodes; the actual data is held in leaves. In the other nodes it keeps minimum bounding rectangles (MBRs), which contain all underlying data. The nodes have a minimum and a maximum capacity, so the tree is height balanced and depth d of an R-tree with a minimum capacity m and a maximum capacity M is limited to

$$[\log_M N - 1] \leq d \leq [\log_m N - 1] \quad (8.1)$$

for N data elements. An example of an R-tree can be found in Figure 8.1¹. The rectangles in orange (solid lines) represent the data objects, two-dimensional rectangles in this case. The blue (dashed) lines are the corresponding MBRs.

The algorithm in Listing 1 shows how to find all rectangles containing a point q . Search in R-trees starts on the root, checks the closest MBRs in each level while pruning other branches and thus limits the amount of total comparisons.

¹R-tree example: <https://de.wikipedia.org/wiki/Datei:R-tree.svg>

So the complexity can be assumed as $\mathcal{O}(\log_m N)$ in average and $\mathcal{O}(N)$ in worst case.

Algorithm 1: Find all MBRs containing a point

Data: Let N be a set of all nodes.

Let n_r the root node. Let q be a point.

Result: *result* as set of MBRs containing q .

```

1
2 result =  $\emptyset$ ;
3 SearchQ(result,  $n_r$ );
4
5 Function SearchQ(result,  $n$ ):
6
7   if  $n$  is a leaf then
8     | result = result  $\cup$   $n$ 
9   else
10    | foreach child  $n' \in N$  of  $n$  do
11      | | if rectangle of  $n'$  contains  $q$  then
12        | | SearchQ(result,  $n'$ )
13      | end
14
```

R-tree and SPARQL-MM

For R-trees there exist efficient algorithms for the main general types of spatial queries, regarding range, topology, nearest-neighbor, and join [MNPT10]. So all topological (spatial) queries (**covers**, **contains**, etc.) are supported. Directional queries as defined in SPARQL-MM can also be solved efficiently by transforming it to topological queries. Let the image space of i be defined by a rectangle $(0, 0, W, H)$, let $a = (x_a, y_a, w_a, h_a)$ and $b = (x_b, y_b, w_b, h_b)$ fragments in i . The following mappings can be used for transformation:

$$\begin{aligned} \text{rightBeside}(a, b) &= \text{contains}((x_b + w_b, 0, W, H), a) , \\ \text{above}(a, b) &= \text{contains}((0, 0, W, y_b), a) . \end{aligned}$$

As mentioned in a former Section all other directional functions can be transformed or combined to these. Regarding combined (spatio-temporal) functions, the same index structure can be used by adding a third dimension (time) to it. So all SPARQL-MM functions on basic shapes proposed by Media Fragment URIs are supported. As mentioned in [BS12] "R-Trees can also be extended to support extra features like support for storage of details about moving injects. Thus the R-Tree index structure can be modified to provide [a more complex] temporal support" in further optimization efforts. In the next Section I will focus on SPARQL optimization, the second step for building high- performant query evaluators for SPARQL-MM.

8.1.2 SPARQL optimization approaches

To illustrate, why complex SPARQL queries may lead to non-optimal query execution times, I start with a simple example, outlined in Listing 8.1. The query is a SPARQL representation of:

Select all images of a birthday party before 2017 that show Alice, Bob and Charlie, whereby Alice is right beside Charlie and Alice and Bob do not intersect.



Figure 8.2: Sample image: Alices Birthday 2012

In line 5 one of the fragments is related to the result image. In lines 8-10 it is defined what (or which person) each fragment represents. In line 7 and 11 the results are filtered to images before 2017. The lines 12 and 13 uses SPARQL-MM functions to specify the spatial relations between the fragments. Figure 8.2 shows a example result that fits the query.

Listing 8.1: SPARQL Example for Optimization

```

1 PREFIX ex:<http://example.org/>
2 PREFIX ma:<http://www.w3.org/ns/ma-ont#>
3 PREFIX mm:<http://linkedmultimedia.org..function#>
4 SELECT ?image WHERE {
5     ?image ma:hasFragment ?f2.
6     ?image ex:date ?date.
7     ?image ex:concept "Birthday".
8     ?f1 ex:shows "Alice".
9     ?f2 ex:shows "Bob".
10    ?f3 ex:shows "Charlie".
11    FILTER ex:before(?date, "2017")
12    FILTER mm:rightBeside(?f1, ?f3)
13    FILTER mm:disjoint(?f1, ?f2)
14 }
```

This query is translated into SPARQL algebra using the translation algorithm described in Section 5.2.2. For the matter of readability and because the query does

not use abbreviations for triple patterns I skip the first part of the algorithm "*Expand Syntax forms*" and keep the prefixes. In addition I prepone the "*Simplification Step*" and join all BGPs, which results in the query outlined in Listing 8.2.

Listing 8.2: Query in SPARQL Algebra

```
SELECT ?image WHERE {
  BGP (
    (?image, ma:hasFragment, ?f2).
    (?image, ex:date, ?date).
    (?image, ex:concept, "Birthday").
    (f1?, ex:shows, "Alice").
    (f2?, ex:shows, "Bob").
    (f3?, ex:shows, "Charlie")
  )
  FILTER (ex:before, (?date, "2017"))
  FILTER (mm:rightBeside, (?f1, ?f3))
  FILTER (mm:disjoint, (?f1, ?f2))
}
```

As a next step, I "*Translate Filters*" one by one following the ordering of the original query and transform the SELECT into a Projection, which results in the Abstract Syntax Tree (AST), outlined in Listing 8.3.

Listing 8.3: Query in SPARQL Algebra

```
Projection (?image,
  Filter (mm:disjoint(?f1, f2?),
    Filter (mm:rightBeside(?f1, f3?),
      Filter ( date < "2017",
        BGP (
          (?image, ma:hasFragment, ?f2).
          (?image, ex:date, ?date).
          (?image, ex:concept, "Birthday").
          (f1?, ex:shows, "Alice").
          (f2?, ex:shows, "Bob").
          (f3?, ex:shows, "Charlie")
        )
      )
    )
  )
)
```

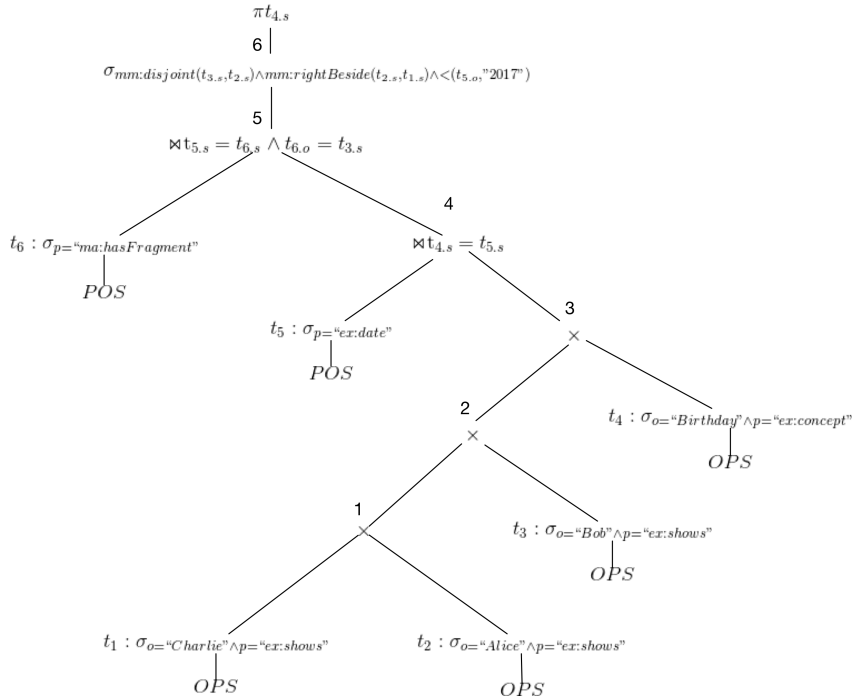


Figure 8.3: Non-optimal evaluation tree

8.1.2.1 Idea of SPARQL-MM-aware optimizer

A naive executor would evaluate this query as given from inside to outside. The *triple* set within the *BGP* would be joined in a row before the filter is executed on the whole join result. The query evaluation tree for this scenario is outlined in Figure 8.3. The execution may lead to non optimal (non minimal) intermediate results (e.g. when the triples have a different level of selectivity). Using the knowledge of pattern and filter selectivity can increase the execution time a lot. Assuming we have a selectivity function for filters *sel* that maps each filter to a numeric value. Additional assuming for the give filter set $filters = \{leftBeside, rightBeside, disjoint, <\}$ we get the ordering $sel_{rightBeside} = sel_{leftBeside} < sel_{disjoint} < sel_{<}$. And assuming the filter calculation per node is equal for any filter within *filters*. Then a rearrangement of the abstract syntax tree may lead to smaller join sizes and thus to a more efficient query evaluation. Figure 8.4 shows the hypothetical progression of join-tables sizes, whereby the orange line depicts the non-optimized and the blue line the optimized evaluation. In the next Sections I introduce an approach that enables an automatic rebuilt of SPARQL ASTs based on triple and filter selectivity.

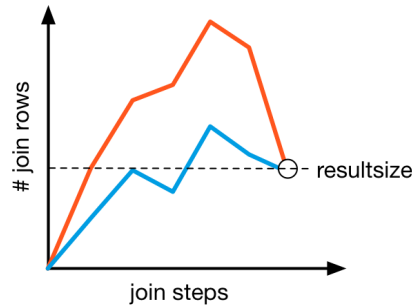


Figure 8.4: Hypothetical progression of join-tables sizes

8.1.2.2 SPARQL optimization approaches

As mentioned, query planing is a crucial task in order to reduce the size of intermediate results and thus leads to a query evaluation process with lower resource consumption (regarding time, memory, etc), which is also known as cost. The optimization process consists of two main steps: a) the representation of the SPARQL query in a suitable format, and b) the calculation of cost in order to estimate the resource consumption (mostly time) for query evaluation. For both steps various approaches have been elaborated. For SPARQL query representation most of the existing works uses graphs to represent triple patterns and its relationships as edges and nodes, like [LWYX10]. As the SPARQL extension that I describe within this thesis is not extending triple patterns but utilizes the SPARQL build-in extension concept of FILTERS, this is not a suitable model. [SC15] extended this concept to other SPARQL expressions like GRAPHS and FILTERS and such qualifies as a good basis for SPARQL-MM optimization efforts. This approach, a further development of [SSB⁺08] and [NW08], is using basic triple pattern as graph-nodes, the edges connect nodes that share at least one variable. A more detailed description is given in the next Sections.

Regarding cost estimation models there are two main strategies, heuristics with pre-computed statistics and heuristics without pre-computed statistics [DM16]. Whereby the first summarizes the data of the search corpus (by e.g. histograms), the second are based in observations of representative RDF datasets. This leads to the fact that statistical approaches like [NW08] have higher cost but provide (in most cases) more exact results. But also heuristic based algorithms, like [TSF⁺12], have been evaluated with good results and even outperformed many other approaches. And beside cost heuristic optimizers are less complex in development and maintenance and can be distributed in multi-computer environments without additional effort.

8.1.3 Optimizing SPARQL

After the given examination of SPARQL query optimizers and its strategies I select the one described by [SC15] as basis. This approach is mainly heuristic based but produces very good query execution plans in comparison to others with a minimal planing effort. Additionally it is suitable for SPARQL-MM because is straight-forward to extend regarding filters. This query execution approach uses the Extended SPARQL query triple pattern Graph (ESG) as a basic representation format for query optimization. My contribution to the approach is the extension of the cost model regarding filters and the influence of the filter costs in the overall cost model.

The planning can be separated into three steps:

1. translating SPARQL to an abstract ESG representation,
2. calculating estimated costs for ESG nodes and vertices, and
3. searching the most cost efficient plan.

Extended SPARQL query pattern graph

The ESG used by [SC15] represents a more abstract query model, which is represented by vertices V (query expressions like bgp, filter etc.) and edges E . An edge links two vertices, if they share at least one variable.

Definition 47 An ESG vertex $v \in V$ is defined as

$$v = (exp, type, cost_{model}, cost), \quad (8.2)$$

whereby exp refers to the abstract syntax expression and $type \in Type$ outlines the type of the expression. In my approach I only consider 2 main types, with $Type = \{T, F\}$, whereby T corresponds to *triple* (as part of the BGP) and F to *filter* in the SPARQL algebra. The $cost_model$ is an abstract model that allows to calculate the estimated $cost$ (the selectivity) of the vertex, depending on the type and heuristics/statistics.

Definition 48 A vertex cost model $v.cost_{model}$ is formalized as follows:

$$v.cost_model = \begin{cases} (S, P, O, N_V, N_F, C_F) & \text{if } v.type = T \\ (F, T_F) & \text{if } v.type = F \end{cases} \quad (8.3)$$

S,P,O,G refer to basic ground terms in quads (an extension of triples to contexts) and can be URI, Literal or Variable. F is a filter identifier (URI or Literal). N_V denotes the number of variables appearing in filters, N_F the number of related filters and C_F the aggregated cost of these filters.

Definition 49 Let F be a set of filters related to triple T . Let n be the cardinality of F . The aggregated costs C_F for all filters $f \in F$ are defined as follows:

$$C_F = \frac{1}{n^2} \times \prod_{i=1}^n \text{cost}(f_n) \quad (8.4)$$

The cost of a single filter is in the range $]0, 1[$. So the aggregated filter cost of a triple cost decrease if more filters are related. The factor enforces this so triples that are related to more filters result in significant lower cost. Note that the heuristic cost model for filters is my extension of the basic approach. The filter selectivity function cost that I will introduce in a later Section is based on heuristics and statistics.

Definition 50 An edge $e \in E$ links two vertices v_1 and v_2 if they share at least one variable and is defined as

$$e = (v_1, v_2, \text{type}, \text{vars}, \text{cost}_{\text{model}}, \text{cost}) \quad (8.5)$$

The $\text{type} \in \{\text{'uni - directional'}, \text{'bi - directional'}\}$ denotes if an edge is uni- or bi-directional. vars is a list of shared variables between the nodes connected via the edge. cost is the selectivity for executing v_1, v_2 . The $\text{cost}_{\text{model}}$ itself is defined as:

$$e.\text{cost}_{\text{model}} = (J_{\text{type}}, N_{\text{share}}) \quad (8.6)$$

whereby J_{type} is the type of BGP joining type N_{share} is the number of shared variables.

Cost estimation for ESG

The heuristic cost-model is based on the one presented in [SC15]. I removed heuristics that are considering named graphs and added H3*, which consider filter selectivity.

H1: The cost for executing query triple patterns is ordered as: $c(s, p, o) \leq c(s, ?, o) \leq c(?, p, p) \leq c(s, p, ?) \leq c(?, ?, o) \leq c(s, ?, ?) \leq c(?, p, ?) \leq c(?, ?, ?)$

H2: A triple pattern that is related to more filters has higher selectivity and cost less.

H3: A triple pattern that has more variables appearing in filters has higher selectivity and less cost.

H3*: The selectivity of triple patterns that are related to filters are influenced by the aggregated filter costs. The cost model is outlined in the next Section.

H6: The position of the join variable of two vertices influences the join selectivity. The ordering is hereby: $p \bowtie o < s \bowtie p < s \bowtie o < o \bowtie o < s \bowtie s < p \bowtie p$, whereby s, p, o refers to the join variable.

H7: Edges whose vertices share more variables are more selective. This is build on the estimation that vertices, which share more variables result in smaller join tables.

The exact calculation of the edge cost is defined in [SC15].

In this basic approach filters are considered to decrease triple pattern costs without taking into account the selectivity of the specific filter function. The assumption in this thesis is that a cost calculation for filters can lead to better query plans as high selective filters on a proper place can decrease join sizes on a early stage in query evaluation.

8.2 Considering Filters for SPARQL query optimization

The idea of considering the selectivity of filters for SPARQL query optimization leads to more performant evaluation is based on the assumption that filters a highly selective and such decreases the size of (intermediate) triple-join tables. In order to validate this assumption I made an experiment with the COCO (Common Objects in Context) dataset², a large-scaled object detection, segmentation and caption dataset. The set contains over 330K images with 5 captions (out of about 80 categories) per image in average.

8.2.1 Experimental proof of selectivity assumption

For the experiment 10.000 randomly chosen images (with 71.937 annotations) are transferred to a RDF representation using Media Fragment URIs (MFUs), DC Terms and SKOS. This very simple model has been selected with a purpose as it produces a small amount of triples and such can be seen as a lower baseline for the proportion of MFUs and other URLs.

The model of segment annotations is outlined in listing 8.4, whereby the `category_id` refers to a specific COCO category; the category model is shown in listing 8.5.

Listing 8.4: COCO segment annotation model

```
[{
  "image_id"    : int,
  "category_id": int,
  "bbox"       : [x,y,width,height],
  "score"      : float
}]
```

²COCO dataset: <http://cocodataset.org/>

Listing 8.5: COCO segment annotation model

```
[{
  "id"          : int ,
  "name"       : string ,
  "supercatory": string ,
}]
```

The transformation that turns the COCO data to RDF is described in algorithm 2. The annotations of all images are transformed to Media Fragment URIs using a common base url, the id of the COCO dataset and the bounding boxes of the annotations (transformed to regional fragment hashes). Each annotation results in two triples, which are the relation between the image and the fragment and the relation between the fragment and the category. In addition the used categories and its super concepts are translated to RDF using basic SKOS relations, namely *broader* and *prefLabel*.

Algorithm 2: Transforming COCO data to RDF

Data: A set of image annotations I,
 A set of categories C

Result: A set of RDF triples

```
1 triples = {};
2 foreach i in I do
3   hash = "#xywh=" i.bbox.x "," i.bbox.y "," i.bbox.width ","
          i.bbox.height"
4   triples.add( <i.image_id> ma:hasFragment <i.image_id+hash> )
5   triples.add( <i.image_id+hash> dct:subject <i.category_id> )
6 end
7 foreach c in C do
8   triples.add( <c.id> skos:prefLabel "c.name" )
9   triples.add( <c.id> skos:broader <c.supercatory> )
10  triples.add( <c.supercatory> skos:prefLabel "c.supercatory" )
11 end
```

Example 18 Listing 8.6 shows an example transformation of an image including one segment annotation to a category skateboard represented in JSON format. Note that I use the base-URLs <http://example.org/image>; prefix *exi* (for image) and <http://example.org/category>, prefix *exc* for category.

Listing 8.6: COCO segment annotation model

```
//the image
{
  "image_id"    : 1,
  "category_id": 1,
  "bbox"       : [10,20,30,40],
  "score"      : 1.0
}
```

Function	Class in $func_{sparql-mm}$	Selectivity $\times 10^5$
touches	a	0.001
leftAbove / rightBelow	b	0.004
covers	c	0.004
above / below	d	0.009
intersects	e	0.010
leftBeside / rightBeside	f	0.016
disjoint	g	0.036

Table 8.1: Results of the function selectivity experiment

```
//the category
{
  "id"          : 1,
  "name"        : "skateboard",
  "supercategory": "sports",
}

//the RDF result
exi:1 ma:hasFragment exi:1#xywh=10,20,30,40 .
exi:1#xywh=10,20,30,40 dct:subject exc:1 .
exc:1 skos:prefLabel "skateboard" .
exc:1 skos:broader exc:sports .
exc:sports skos:prefLabel "sports" .
```

The experimental transformation results in 215.991 triples and 153.988 nodes. In a next step I calculated the selectivity for a subset of Sparql-MM spatial functions, which contain often used functions.

Definition 51 *The selectivity of a function sel_{func} is defined as the cardinality of all possible results of σ_{func} divided through the cross-product of all nodes,*

$$sel_{func} = \frac{|\sigma_{func}(nodes \times nodes)|}{|nodes \times nodes|}, \quad (8.7)$$

The result of the selectivity calculation is listed in Table 11. It shows that SPARQL-MM functions a) are highly selective and thus reduces the cardinality of join lists a lot and b) differ in selectivity.

8.2.2 Filters and Edge costs

The result of the experiment allows to order SPARQL-MM functions based on their selectivity and this results allow to specify heuristic **H3*** in more detail:

H3:** The selectivity of triple patterns that are related to SPARQL-MM filters are influenced by the aggregated filter costs. The ordering of SPARQL-MM function set classes (outlined in Table 11) $func_{sparql-mm} = \{a, b, c, d, e, f, g\}$ regarding filter costs thereby is $a < b < c < d < e < f < g$. Let F be a set of Filter functions of one query. The calculation of the cost of a filter function $f \in F$ is based on the order number in the set of Filters $0 \leq ord(x) < |F|, x \in F$. The normalized cost function $cost(f)$ is defined as

$$cost(f) = \frac{ord(f)}{|F| - 1} \quad (8.8)$$

I reduced the set of all functions in SPARQL-MM to a reasonable subset in order to keep readability. In addition I consider the selectivity of all other functions O as higher than any SPARQL-MM function, so that I can adapt the ordering like $a < .. < g < o$ for any $o \in O$. It is obvious that the approach is robust against the integration of additional functions and their specific selectivity.

8.2.3 Query plan search

With the definitions in the former Section the cost of vertices and edges can be calculated straightforward with Algorithm 3.

Algorithm 3: Cost calculation of vertices

Data: Let $ESG = \langle V, E \rangle$ a tuple of vertices V and edges E .

Let $V = TP \cup F$ be the union of a triple patterns TP and filters F .

Result: A sorted list of filters and triples patterns; costs assigned

```

1 F = sort(F, FilterCostComparator);
2 foreach filter  $f_i$  in  $F$  do
3   |  $f_i.cost = \frac{i+1}{|F|+1}$ ;
4 end
5
6 TP = sort(TP, HeuristicTripleComparator);
7 foreach vertex  $tp_i$  in  $TP$  do
8   |  $tp_i.cost = \frac{i}{|TP|-1}$ ;
9 end
10
11 Function HeuristicTripleComparator( $v_1, v_2$ ):
12   | Get basic triple patterns  $p_1(S_1, P_1, O_1)$  and  $p_2(S_2, P_2, O_2)$  of  $v_1$  and  $v_2$  ;
13   | Get the position  $i_1, i_2$  of  $p_1$  and  $p_2$  in PATTERNS list ; Get aggregated
14   | filter cost  $C_{F_{v_1}}, C_{F_{v_2}}$  ; Cost model  $m_1 = v_1.model, m_2 = v_2.model$  ;
15   | if  $i_1 \neq i_2$  then
16     | return  $i_1 > i_2$  ? 1 : -1;
17   | else if  $m_1.C_F \neq m_2.C_F$  then
18     | return  $(m_1.C_F > m_2.C_F)$  ? 1 : -1;
19   | else if  $m_1.N_F \neq m_2.N_F$  then
20     | return  $(m_1.N_F < m_2.N_F)$  ? 1 : -1;
21   | else if  $m_1.N_V \neq m_2.N_V$  then
22     | return  $(m_1.N_V < m_2.N_V)$  ? 1 : -1;
23   | else
24     | return 0;
25
25 Function FilterCostComparator( $f_1, f_2$ ):
26   | return  $(sel(f_1) < sel(f_2))$  ? 1 : -1;

```

The algorithm adds cost to every vertex in the ESG. In line 1 the filter patterns are sorted based in their selectivity using the comparator in lines 25f. In lines 2 - 4 the costs for filters are set to values of $]0, 1[$. In line 6 the triple patterns are sorted with the comparator defined in lines 11 - 22 based on the heuristics listed in Section 8.1.3, namely H1 (pattern ordering), H3*(aggregated filter cost), H2(number of related filters), and H3 (number of common variables). In lines 6 - 8 the cost for triples are set.

Together with the edge costs a second algorithm can be used to search for an optimal query plan. Like in the basic paper, I use a greedy algorithm starting at the vertex with smaller cost and searching linked vertices recursively. This approach tries to get a query plan, which evaluates triple pattern and filters that are highly selective and thus reduces the data space as fast as possible. As the algorithm is not testing and comparing any possible execution plan regarding cost, it may not always lead to optimal results, but it produces a good tradeoff between plan computation time and evaluation performance optimization (as the query plan computation must be counted as a part of the overall evaluation process). The pseudo code in Algorithm 4 explains the search procedure in detail.

Algorithm 4: Search Algorithm for cost efficient query plan

Data: Let $ESG = \langle V, E \rangle$ a tuple of vertices V and edges E .

Let $V = TP \cup F$ be the union of a triple patterns TP and filters F .

Result: A sorted list of vertices S that can be evaluated in line

```

1 S = [];
2 do
3   t = MinOfRest (TP) ;
4   S.push( TP.remove( t ) ) ;
5   S.push( F.removeAll( GetFilters (F,S)) ) ;
6   RecalculateCost(t,E) ;
7 while V.size ≠ 0
8
9 Function MinOfRest(TP):
10 | return t ∈ T with t.cost ≤ x.cost for all x ∈ T;
11
12 Function GetFilters(F,S):
13 | F_sel = [];
14
15 | Let N_s be the set of all nodes in s, s ∈ S ;
16 | Let N_S = ∪ N_s for all s ∈ S ;
17 | foreach filter f in F do
18 |   | Let N_f be the set of all nodes in f ;
19 |   | if N_f ⊆ N_S then
20 |   |   | F_sel = F_sel ∪ f;
21 |   | end
22 | end
23
24 | F = F \ F_sel ;
25 | return F_sel;
26
27 Function RecalculateCost(t,E):
28 | foreach e ∈ E do
29 |   | if e.n1 == t then
30 |   |   | e.n2.cost *= e.cost;
31 |   | else if e.n2 == t then
32 |   |   | e.n1.cost *= e.cost;
33 | end

```

The algorithm creates a sorted list of vertices (triple patterns and filters), which represents a good (in best case optimal) candidate for evaluation from first to last. Lines 3 - 6 are repeated until all vertices are contained in the result set S . In line 3 and 4 the triple pattern with the lowest cost out of all remaining patterns is selected and added to the result. In line 5 all filters that are applicable are selected and added to the result via the greedy algorithm in lines 12 - 25. After that the costs

are recalculated in line 6 by the algorithm in lines 27 to 33. The recalculation is done for all edges that are related to the vertex selected in line 3.

8.3 Conclusion

Without query optimization SPARQL query evaluation may be inefficient regarding cost. In environments with a bigger amount of data this may lead to long running queries and thus to a inadequate user experience. In this Chapter I introduced a novel query plan optimization approach, which take into account cost estimation heuristics based on the nature of triple patterns. The approach uses a basic model called Extended SPARQL query triple pattern Graph (ESG). I extended the basic algorithm described in [SC15] with heuristics of SPARQL-MM filters that has been found by investigating a set of annotated images of the COCO data set. The current implementation only considers a subset of SPARQL-MM functions but can be easily extended to the full function set and additionally to any heuristics about other well-known filter functions. In the next Chapter I will further explain the approach using a step-by-step example. Furthermore I will compare the cost plans and its evaluation performance to the basic algorithm.

Evaluation

In the former Chapter I introduced a algorithm to calculate cost efficient query plans for SPARQL queries. In this Chapter I will evaluate the approach, starting with a step-by-step example of a plan calculation. After this I sketch a evaluation environment that is used for a comparison of non-optimized and restructured query plans. This environment is the basis for a set of tests that are done with a fixed set of SPARQL-MM queries. These tests are executed and discussed in this Chapter, too.

9.1 Example

In this Section I exercise a query optimization approach using the algorithm that is described in the former Chapter. The example query is already known from Listing 8.1 but repeated in Listing 9.1 for the matter of readability.

Listing 9.1: SPARQL Example for Optimization (repetition)

```

PREFIX ex:<http://example.org/>
PREFIX ma:<http://www.w3.org/ns/ma-ont#>
PREFIX mm:<http://linkedmultimedia.org..function#>
SELECT ?image WHERE {
    ?image ma:hasFragment ?f2.
    ?image ex:date ?date.
    ?image ex:concept "Birthday".
    ?f1 ex:shows "Alice".
    ?f2 ex:shows "Bob".
    ?f3 ex:shows "Charlie".
    FILTER ex:before(?date, "2017")
    FILTER mm:rightBeside(?f1, ?f3)
    FILTER mm:disjoint(?f1, ?f2)
}

```

9.1.1 Example: Translate SPARQL to ESG

The SPARQL patterns are translated into SPARQL algebra already outlined in Figure 8.3. This intermediate result is then transformed into the ESG by transforming Triples and Filters in nodes and linking the nodes that share at least one variable with edges. A graphical representation of the resulting tree can be found in 9.1,

whereby triple nodes are represented as circles (blue) and filter nodes as squares (orange).

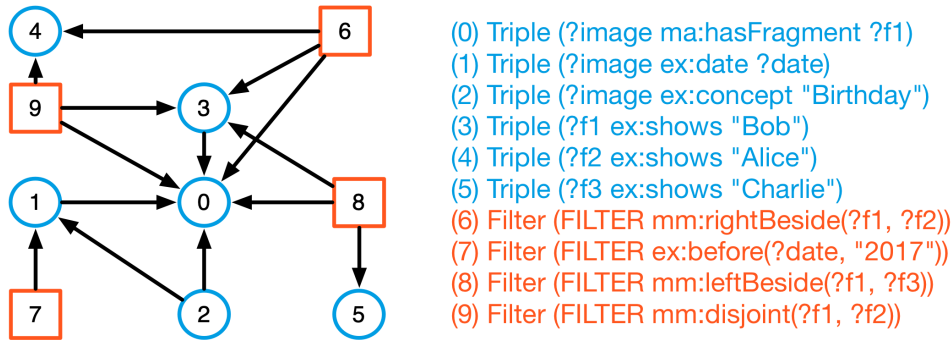


Figure 9.1: Example ESG

9.1.2 Example: Calculate costs for nodes and vertices

The next step is the calculation of estimated costs for edges and vertices. Like described in the former Chapter the costs are based on heuristics and regarding filters take influence to each other. In order to emphasize the difference the basic approach (without filter selectivity) and the extension elaborated in this thesis the two results are shown next to each other in Listings 9.2 and 9.3. In case of triples, the preprocessed results contain the type of pattern, the number of related filters, the number of variables (appearing in filters), and the calculated (start-)cost. In case of edges, cost are only calculated and considered in the algorithm if the two related nodes are of type triple. Note, in the listing the triples and filters are already ordered regarding their costs. In addition edge cost do not differ in both approaches and therefore just listed once.

Listing 9.2: Example for ESG start costs

```

1 (3) ?f1 ex:shows "Alice". (_po), NV=1, NF=2, cost=0.0
2 (4) ?f2 ex:shows "Bob". (_po), NV=1, NF=1, cost=0.2
3 (5) ?f3 ex:shows "Charlie". (_po), NV=1, NF=1, cost=0.4
4 (2) ?image ex:concept "Birthday". (_po), NV=0, NF=0, cost=0.6
5 (0) ?image ma:hasFragment ?f2. (_p_), NV=1, NF=1, cost=0.8
6 (1) ?image ex:date ?date. (_p_), NV=1, NF=1, cost=1.0
7
8 (6) FILTER ex:before(?date, "2017")
9 (7) FILTER mm:rightBeside(?f1, ?f3)
10 (8) FILTER mm:disjoint(?f1, ?f2)

```

Listing 9.3: Example for ESG start costs considering filter selectivity

```

1 (3) ?f1 ex:shows "Alice". (_po), NV=1, FC=0.0, NF=2, cost=0.0
2 (5) ?f3 ex:shows "Charlie". (_po), NV=1, FC=0.0, NF=1, cost=0.2
3 (4) ?f2 ex:shows "Bob". (_po), NV=1, FC=0.5, NF=1, cost=0.4
4 (2) ?image ex:concept "Birthday". (_po), NV=0, FC=1.0, NF=0, cost=0.6
5 (0) ?image ma:hasFragment ?f2. (_p_), NV=1, FC=0.5, NF=1, cost=0.8
6 (1) ?image ex:date ?date. (_p_), NV=1, FC=1.0, NF=1, cost=1.0
7
8 (7) FILTER mm:rightBeside(?f1, ?f3) cost=0.0
9 (8) FILTER mm:disjoint(?f1, ?f2) cost=0.5
10 (6) FILTER ex:before(?date, "2017") cost=1.0

```

The extended model in Listing 9.3 now contains aggregated filter cost FC based on the cost of the related filters, which are also added to the model (cost lines 8 - 10). It is obvious that the cost (and thus the order) differs for vertices in the compared results. So the triples (4) and (5) (lines 2 and 3) switch. This is due to the most selective filter (7), rightBeside (line 8), which turns the cost of triple (5) to 0.2. Note that triple (0) keeps its place even though the filter cost are lower than in triple (2). This is due to the more selective triple pattern, which is the most relevant sort factor.

9.1.3 Example: Find most cost efficient plan

Applying the greedy search algorithm of the last Chapter on the sorted triple patterns allows us to generate the evaluation plan straight forward. Triples (3) and (5) are selected first. The first filter (7) can be executed. Adding triple (4) enabled the execution of the next filter (8). As the edge cost are also part of the algorithm, triple (0) is added before (2). And in order to execute the last filter (6) triple (1) is added. In Listing 9.4 the outcome of ordered vertices is described. This ordering can be translated straight-forward to a SPARQL AST, which is shown in Listing 9.5.

Listing 9.4: Sorted vertices in ESG

```

(3) ?f1 ex:shows "Alice". (_po), NV=1, FC=0.0, NF=2, cost=0.2
(5) ?f3 ex:shows "Charlie". (_po), NV=1, FC=0.0, NF=1, cost=0.0
(7) FILTER mm:rightBeside(?f1, ?f3) cost=0.0
(4) ?f2 ex:shows "Bob". (_po), NV=1, FC=0.5, NF=1, cost=0.1
(8) FILTER mm:disjoint(?f1, ?f2) cost=0.5
(0) ?image ma:hasFragment ?f2. (_p_), NV=1, FC=0.5, NF=1, cost=0.05
(2) ?image ex:concept "Birthday". (_po), NV=0, FC=1.0, NF=0, cost=0.15
(1) ?image ex:date ?date. (_p_), NV=1, FC=1.0, NF=1, cost=0.25
(6) FILTER ex:before(?date, "2017") cost=1.0

```

Listing 9.5: Optimized Query in SPARQL Algebra (repetition)

```

Filter ( date < "2017",
  Join (
    BGP (
      Triple (?image ex:date ?date),
      Triple (?image ex:concept "Birthday")
      Triple (?image ma:hasFragment ?f2)
    ),
    Filter ( mm:disjoint(?f1, ?f2),
      Join (
        BGP (?f2 ex:shows "Bob"),
        Filter (mm:rightBeside(?f1, ?f3),
          BGP (
            ?f3 ex:shows "Charlie".
            ?f1 ex:shows "Alice".
          )
        )
      )
    )
  )
)

```

9.2 Evaluation Environment

In this Section I am going to describe the evaluation of the optimization algorithm based on a real world dataset. After defining the evaluation scenario and the evaluation dataset, I will define a set of example queries and compare evaluation iterations with different optimization settings. To proof the concept I created a test scenario using the COCO dataset that has been already described in Chapter 8. The test set contains 40.504 images having 291.000 annotations overall (out of 80 categories). Transforming it to RDF results in 332.094 nodes and 583.772 triples. The resource identifiers for categories follow a simple URL schema and use the COCO id (e.g. `cat:1`) for categories and the unique name for supercategories (e.g. `cat:sports`). A complete list of categories (identifiers, labels, and supercategories) can be found in Appendix B.3.

In the following I define a set of example queries of varying complexity. All of them include SPARQL-MM filters in order to get a difference between the basic optimization approach and the extension elaborated above. For the queries I only describe the selection, as for evaluation the projection is just a count. In addition I skip the prefix part for the matter of compactness. The used prefixes are listed in Appendix B.1.

Evaluation Query 1: book right beside bottle

Listing 9.6: Evaluation Query 1

```
?i ma:fragment ?f1.  
?f1 dc:subject cat:84. //book  
?f2 dc:subject cat:44. //bottle  
FILTER mm:rightBeside(?f1, ?f2)
```

Evaluation Query 2: what is in the middle of a elephant on the right and a zebra on the left

Listing 9.7: Evaluation Query 2

```
?f2 dc:subject cat:22. //zebra  
?f3 dc:subject cat:24. //elephant  
?i ma:fragment ?f2.  
?i ma:fragment ?f1.  
?f1 dc:subject ?c.  
FILTER mm:rightBeside(?f3, ?f1)  
FILTER mm:rightBeside(?f3, ?f2)  
FILTER mm:rightBeside(?f1, ?f2)
```

Evaluation Query 3: book right beside bottle and touches potted plant

Listing 9.8: Evaluation Query 3

```
?i ma:fragment ?f1.  
?f1 dc:subject cat:84. //book  
?f2 dc:subject cat:44. //bottle  
?i ma:fragment ?f2.  
?i ma:fragment ?f3.  
?f3 dc:subject cat:64. // plant  
FILTER mm:rightBeside(?f1, ?f2)  
FILTER mm:touche(?f1, ?f3)
```

Evaluation Query 4: a dog below an umbrella catching a frisbee

Listing 9.9: Evaluation Query 3

```
?f1 dc:subject cat:18. //dog
?f2 dc:subject cat:28. //umbrella
?f3 dc:subject cat:34. //frisbee
mm:above(?f2, ?f1)
mm:covers(?f1, ?f3)
```

Evaluation Query 5: a dog below an umbrella catching a frisbee, another dog is right beside

Listing 9.10: Evaluation Query 5

```
?f1 dc:subject cat:18. //dog
?f2 dc:subject cat:28. //umbrella
?f4 dc:subject cat:34. //frisbee
mm:above(?f2, ?f1)
mm:covers(?f1, ?f4)
?f3 dc:subject cat:18 //dog
mm:rightBeside(?f3, ?f1)
```

Figure 9.2 shows an image that matches the query. The annotations are marked with colored rectangles (cat:18 \equiv dog \equiv orange, cat:34 \equiv frisbee \equiv blue, cat:28 \equiv umbrella \equiv white)



Figure 9.2: Example result image for Query 5

Evaluation Query 6: a couch right beside a chair, bowl on a table, tv above a table

Listing 9.11: Evaluation Query 6

```
?f1 dc:subject cat:63. //couch
?f2 dc:subject cat:62. //chair
?f3 dc:subject cat:51. //bowl
?f4 dc:subject cat:67. //table
?f5 dc:subject cat:72. //TV
?f6 dc:subject cat:67. //table
?i ma:fragment ?f1.
?i ma:fragment ?f3.
?i ma:fragment ?f5.
FILTER mm:rightBeside(?f1, ?f2)
FILTER mm:covers(?f6, ?f3)
FILTER mm:above(?f5, ?f4)
```

Evaluation Query 7: bottle touches a bottle right beside and below a spoon

Listing 9.12: Evaluation Query 7

```
?f1 dc:subject cat:44. // bottle
?i ma:fragment ?f1.
?f2 dc:subject cat:44. // bottle
?f3 dc:subject cat:50. // spoon
FILTER mm:above(?f3, ?f1)
FILTER mm:rightBeside(?f1, ?f3)
FILTER mm:touches(?f1, ?f2);
```

Evaluation Query 8: bottle touches a bottle right beside and below a spoon, a sink contains a cup

Listing 9.13: Evaluation Query 8

```
?f1 dc:subject cat:44. //bottle
?i ma:fragment ?f3.
?f3 dc:subject cat:50. //spoon
?f2 dc:subject cat:44. //bottle
?f4 dc:subject cat:81. //sink
?i ma:fragment ?f4.
?f5 dc:subject cat:47. //cup
FILTER mm:touche(?f2, ?f1)
FILTER mm:rightBeside(?f2, ?f3)
FILTER mm:above(?f3, ?f2)
FILTER mm:covers(?f4, ?f5)
```

Evaluation Query 9: a bottle touches a vase and someone with a tie is right beside a fridge

Listing 9.14: Evaluation Query 9

```
?i ma:fragment ?f1.
?f1 dc:subject cat:44. //bottle
?f2 dc:subject cat:86. //vase
?i ma:fragment ?f3.
?f3 dc:subject cat:32. //tie
?f4 dc:subject cat:82. //fridge
FILTER mm:touche(?f1, ?f2)
FILTER mm:rightBeside(?f3, ?f4)
```

Evaluation Query 10: a bowl on an oven, a sink is placed above both

Listing 9.15: Evaluation Query 10

```
?f1 dc:subject cat:51. //bowl
?f2 dc:subject cat:81. //sink
?f3 dc:subject cat:79. //oven
FILTER mm:above(?f2, ?f3)
FILTER mm:above(?f2, ?f1)
FILTER mm:covers(?f3, ?f1);
```

9.3 Results

The evaluation results listed below are aligned to the following format. The join sizes of the evaluation steps are drawn on a line chart. Thereby the dotted (black) line with the x symbol represents the unoptimized evaluation (**BASE**). The dashed (blue) line with the triangle symbol represents the optimized evaluation without the filter extension elaborated in this thesis (**SONG**). The solid (orange) line with the square symbol represents the results of the extended optimization approach described above (**KURZ**). In addition to the chart, the underlying numbers are outlined in a table for every single query. In order to get a basis for comparison I calculated the **SUM** of all steps for every approach and the percentage of the summation values regarding the unoptimized (**BASE %**) respectively the (basic) optimized results (**SONG %**). The most important values are marked bold and orange/blue. In queries 6-10 I skipped the **BASE** values because they are far beyond the optimized values. Therefore I confine the comparison to **SONG %**. For the matter of readability the actual evaluation plans (ordered triples/filters) are not listed here but in Appendix B.4.

Evaluation Query 1: book right beside bottle

The query contains just one filter and 3 triples, whereby the join of two is necessary to evaluate the filter. This does not give much space for optimization, so the sizes of join tables in summation does not differ much in size, nevertheless it is slightly lower. (Evaluation: Figure 9.3, Table 9.1; Query Plans: Table B.3)

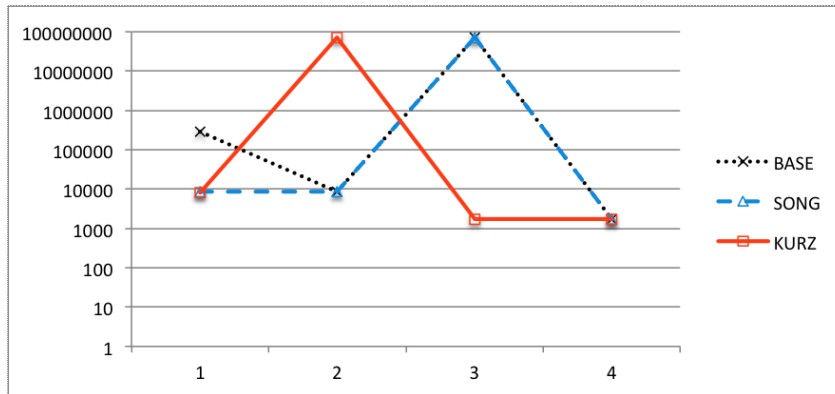


Figure 9.3: Join Evaluation Graph: Query 1

	BASE	SONG	KURZ
1	291.846	8.559	8.381
2	8.559	8.559	71.732.979
3	71.732.979	71.732.979	1.737
4	1.737	1.737	1.737
SUM	72.035.121	71.751.834	71.744.834
BASE %	100	99,607	99,597
SONG %	100,395	100	99,990

Table 9.1: Evaluation Results: Query 1

Evaluation Query 2: what is in the middle of a elephant on the right an a zebra on the left

This query does not focus on images but on fragment results. It contains 3 filters of the same kind so an advantage of KURZ regarding SONG was scarcely to be expected. But the improvement regarding BASE shows that both optimizations perform well. (Evaluation: Figure 9.4, Table 9.2; Query Plans: Table B.4)

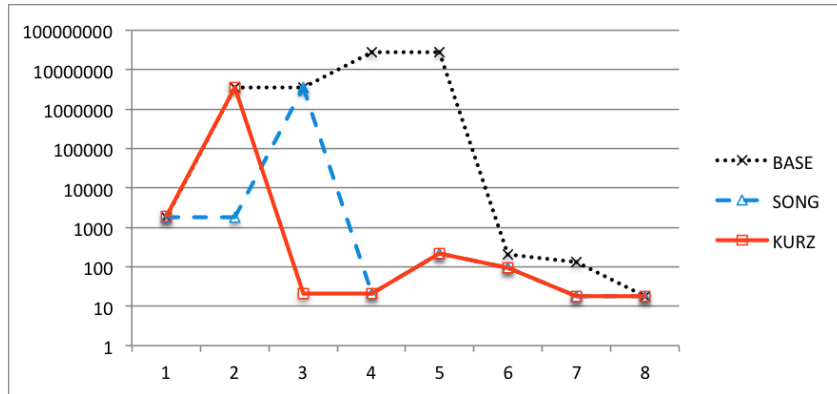


Figure 9.4: Join Evaluation Graph: Query 2

	BASE	SONG	KURZ
1	1.862	1.862	1.885
2	3.509.870	1.862	3.509.870
3	3.509.870	3.509.870	21
4	27.191.125	21	21
5	27.185.470	221	221
6	208	97	97
7	133	18	18
8	18	18	18
SUM	61.398.556	3.513.969	3.512.151
BASE %	100	5,723	5,720
SONG %	1.747,270	100	99,948

Table 9.2: Evaluation Results: Query 2

Evaluation Query 3: book right beside bottle and touches potted plant

This query contains two filters differing in selectivity. Therefore a early evaluation of the `touches` filter enables a substantial reduction of join size in step four of KURZ. This results in lower summation value regarding SONG. (Evaluation: Figure 9.5, Table 9.3; Query Plans: Table B.5)

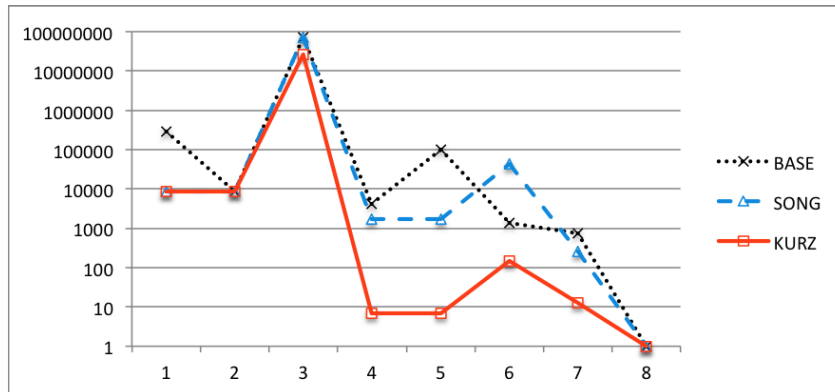


Figure 9.5: Join Evaluation Graph: Query 3

	BASE	SONG	KURZ
1	291.846	8.559	8.559
2	8.559	8.559	8.559
3	71.732.979	71.732.979	26.336.043
4	4189	1737	7
5	101.098	1.738	7
6	1.401	42.923	146
7	725	257	13
8	0	0	0
SUM	72.140.797	71.796.752	26.353.334
BASE %	100	99,523	36,530
SONG %	100,479	100	36,705

Table 9.3: Evaluation Results: Query 3

Evaluation Query 4: a dog below an umbrella catching a frisbee

In this query I skipped the fragment binding, so a evaluation of BASE results in a much bigger join size regarding the optimizations. This shows that early filter evaluation can massively reduce this number. The comparison of SONG and KURZ shows that again higher selective filters on a early stage reduces join sizes. Note, the chart looks quite similar for both approaches because of logarithmic scale, the values in the table makes the advantage more obvious. (Evaluation: Figure 9.6, Table 9.4; Query Plans: Table B.6)

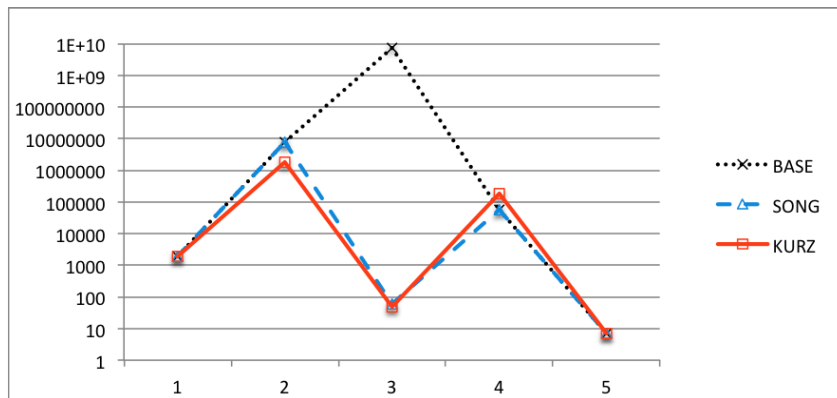


Figure 9.6: Join Evaluation Graph: Query 4

	BASE	SONG	KURZ
1	1.952	1.952	1.952
2	7.767.008	7.767.008	1.825.120
3	7.262.152.480	61	47
4	57.035	57.035	187.013
5	7	7	7
SUM	7.269.978.482	7.826.063	2.014.139
BASE %	100	0,108	0,028
SONG %	92.894,454	100	25,736

Table 9.4: Evaluation Results: Query 4

Evaluation Query 5: a dog below an umbrella catching a frisbee, another dog is right beside

This query is an extension of Query 4. The additional condition is just appended and not optimized because the filter is less selective than the other. Note, the query is already manually "optimized" by not just appending filters at the end of the query but moving them to a more adequate position. If I would not do so, the **BASE** approach will lead to much bigger summation values. (Evaluation: Figure 9.7, Table 9.5; Query Plans: Table B.7)

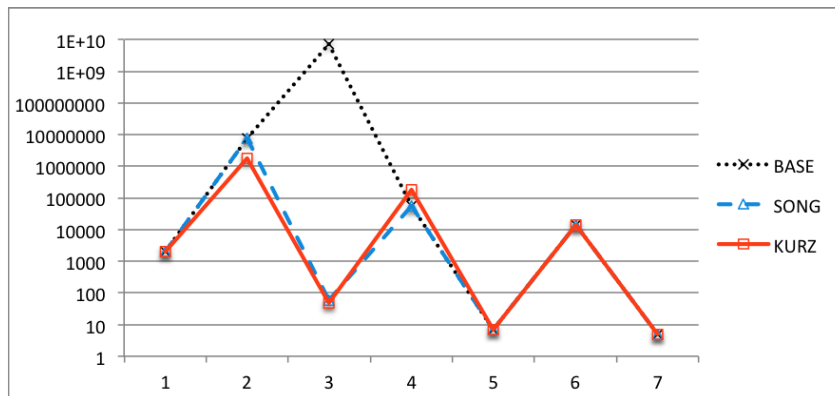


Figure 9.7: Join Evaluation Graph: Query 5

	BASE	SONG	KURZ
1	1.952	1.952	1.952
2	7.767.008	7.767.008	1.825.120
3	7.262.152.480	61	47
4	57.035	57.035	187.013
5	7	7	7
6	13.664	13.664	13.664
7	5	5	5
SUM	7.269.992.151	7.839.732	2.027.808
BASE %	100	0,108	0,0279
SONG %	92.732,661	100	25,866

Table 9.5: Evaluation Results: Query 5

As one can see, more complex query return in huge join sizes for non optimized evaluation plans, so for the next examples I skip **BASE** results.

Evaluation Query 6: a couch right beside a chair, bowl on a table, tv above a table

This query is quite of a complexity and contains six fragments that are narrowed by three filters. Not all fragments are bound to the media asset, so the unbound ones are increasing the joined sets and a typical sawtooth pattern appears. Again a given selective filter (*covers*) is prepared in KURZ, while SONG uses the *rightBeside* filter first. In this case this does not immediately results in less join-rows. Nevertheless re-ordering of triple patterns outperforms SONG on later stages. (Evaluation: Figure 9.8, Table 9.6; Query Plans: Table B.8)

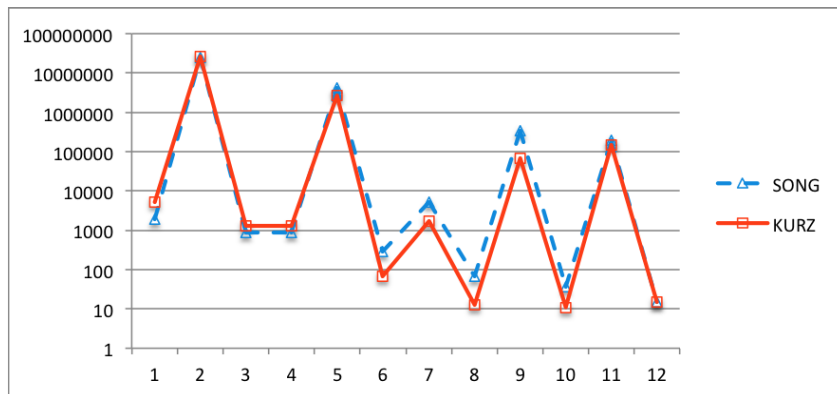


Figure 9.8: Join Evaluation Graph: Query 6

	SONG	KURZ
1	1.927	5.244
2	25.307.291	25.800.480
3	876	1.278
4	876	1.278
5	4.309.920	2.628.846
6	293	66
7	5.317	1.758
8	66	13
9	346.104	68.172
10	37	11
11	194.028	144.463
12	15	15
SUM	30.166.750	28.651.624
SONG %	1	94,977

Table 9.6: Evaluation Results: Query 6

Evaluation Query 7: bottle touches a bottle right beside and below a spoon

The evaluation graph of this query almost overlap for both plans. The advantage of KURZ in this case is caused by the execution of two filters in a order that leads to faster result reduction (steps 4 and 5). Namely *above* is preferred against *rightBeside*. Depending on the underlying index systems, database may execute both filters at one time, so this advantage of the approach is just a minor one. (Evaluation: Figure 9.9, Table 9.7; Query Plans: Table B.9)

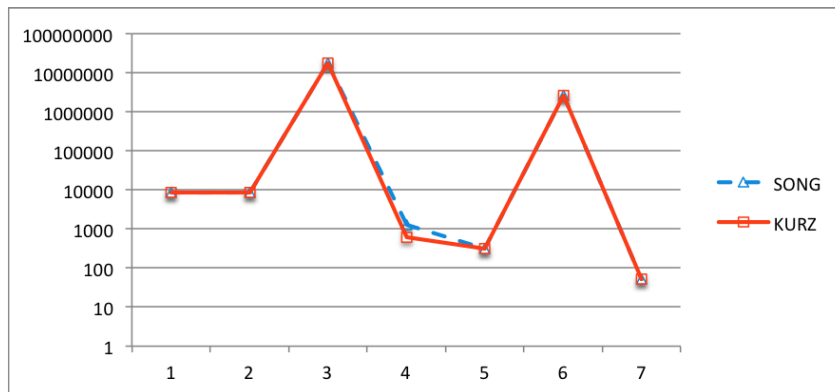


Figure 9.9: Join Evaluation Graph: Query 7

	SONG	KURZ
1	8381	8381
2	8384	8384
3	17.866.304	17.866.304
4	1245	602
5	320	320
6	2.681.920	2.681.920
7	51	51
SUM	20.566.605	20.565.962
SONG %	1	99,996

Table 9.7: Evaluation Results: Query 7

Evaluation Query 8: bottle touches a bottle right beside and below a spoon, a sink contains a cup

This query leads to quite similar plans regarding filters. In detail they differ in two steps (8 and 9), where two triple patterns are permuted due to the effect of related filter costs. Even if KURZ lies on front of SONG, this result is mainly due to the ordering of the base query and thus should not be counted as a major plan enhancement. (Evaluation: Figure 9.10, Table 9.8; Query Plans: Table B.10)

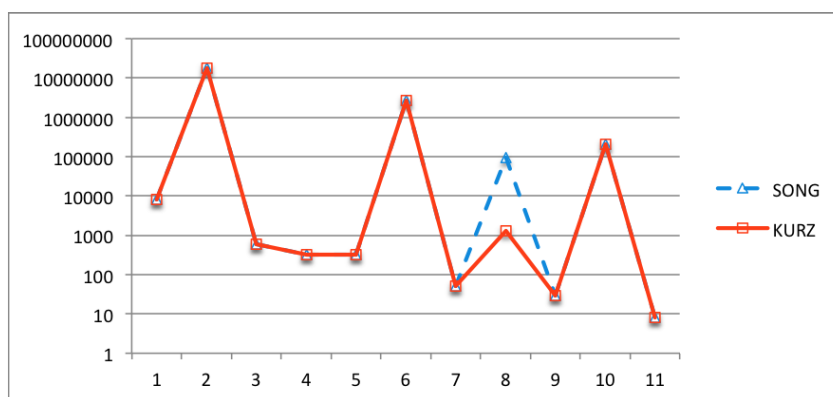


Figure 9.10: Join Evaluation Graph: Query 8

	SONG	KURZ
1	8.381	8.381
2	17.859.911	17.859.911
3	602	602
4	320	320
5	320	320
6	2.681.920	2.681.920
7	51	51
8	96.951	1.324
9	30	30
10	211.050	211.050
11	8	8
SUM	20.859.544	20.763.917
SONG %	1	99,541

Table 9.8: Evaluation Results: Query 8

Evaluation Query 9: a bottle touches a vase and someone with a tie is right beside a fridge

In this query the evaluation plan of SONG outperforms KURZ regarding the given metric. This is due to the fact that considering filter costs makes the plan search algorithm even more greedy. The first filter (*touches*) is evaluated earlier (step 2 against 3), which has a positive effect. But the aim to evaluate the second filter (*rightBeside*) as early as possible leads to a different triple pattern order and (in this case) to a worse result. (Evaluation: Figure 9.11, Table 9.9; Query Plans: Table B.11)

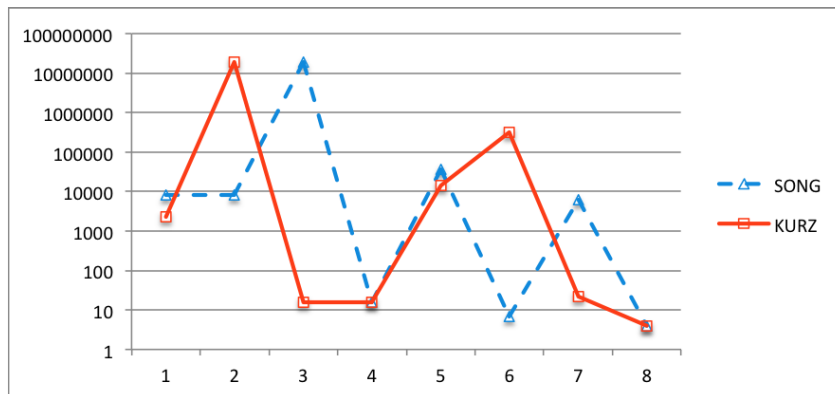


Figure 9.11: Join Evaluation Graph: Query 9

	SONG	KURZ
1	8.381	2.267
2	8.384	18.999.727
3	19.006.528	16
4	16	16
5	36.048	14.208
6	7	309.912
7	6.216	22
8	4	4
SUM	19.065.584	19.326.172
SONG %	1	101,367

Table 9.9: Evaluation Results: Query 9

Evaluation Query 10: a bowl on an oven, a sink is placed above both

This query has a lower complexity but a high amount of filters regarding triple patterns (3:2). Each triple is of pattern `_po` and each variable is used in exactly two filters, so for **SONG** the triples have exactly the same cost. In **KURZ** the cost for triple patterns differs, because the selectivity of filters is considered. This leads to an early evaluation of the most selective filter *covers* and thus to smaller join sizes. (Evaluation: Figure 9.12, Table 9.10; Query Plans: Table B.12)

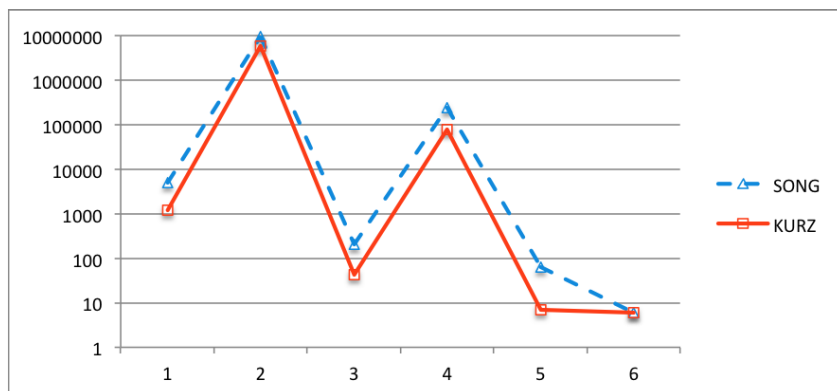


Figure 9.12: Join Evaluation Graph: Query 10

	SONG	KURZ
1	4.920	1.175
2	9.352.920	5.781.000
3	205	42
4	240.875	79.842
5	63	7
6	6	6
SUM	9.598.989	5.862.072
SONG %	1	61,069

Table 9.10: Evaluation Results: Query 10

9.4 Conclusion

In this Chapter I described the functionality of the algorithm introduced in Chapter 8 by a complete optimization walk-through based on a step by step example. It shows how the algorithm works and execution plans for evaluation are created. After that I sketched a evaluation scenario based on the Microsoft COCO data set by transforming the data to RDF using well known ontologies. In addition I created a set of evaluation queries varying in complexity. I compared three query evaluations, namely **BASE** (a naive non optimized query plan), **SONG** (a query plan optimized by the algorithm I used as basis for my work) and **KURZ** (a plan created with the filter optimization approach presented within this thesis). The results show that **KURZ** creates plans with lower join sizes and thus enables a more efficient query execution. In the next Chapters I am going to introduce and evaluate a novel approach regarding semantic Multimedia similarity.

Part V

Semantic Multimedia Relations

Semantic Distance of Media Fragments

SPARQL-MM helps to reduce the gap between common requirements for Multimedia retrieval and the opportunities which are provided by Semantic Web technologies and the corresponding query mechanism. Nevertheless, like I outlined in Chapter 3, media similarity measurement is a major part in the Multimedia retrieval process. In this Chapter I will introduce an approach that combines common (graph-based) semantic concept distance with spatial fragment distances. In order to give a basis for the work, I am going to give a short introduction to similarity concepts. After that I will introduce a metric for spatial fragment distance. The developed approach is formally described and evaluated by user tests.

10.1 Semantic Distance

In this Section I give an overview of semantic concept distance measurements. This is just a small overview as I take the approaches as given in order to use them in combination with a self-defined fragment distance. Therefore this Section does not claim to be exhaustive but introductory. The Section is based on [SK11b] and extended by a text similarity approaches survey of [GF13].

The (semantic) distance of two documents, terms or concepts indicates whether and how strong or weak these two are related to each other - the stronger the semantic relationship, the shorter is the semantic distance. A distance thereby is defined as a length of the shortest path between 2 points. In order to calculate this distance a proper metric has to be defined.

Well-known metrics for concept similarity can be separated in three main categories namely string-based, topological or knowledge-based, and statistical or corpus-based. Character- or Term-based Metrics like defined by Levenstein [Lev66], Winkler [Win90], or Needleman [NW70] are used to calculate a minimal edit distance between concept labels. But they mostly do not lead to adequate results in concept similarity due to known reasons like disambiguity of homonyms, difference in multi-language environments, etc. Therefore I focus on topological and corpus-base metrics.

Topological-based semantic distance

Topological semantic distance calculation is based on one or more ontologies. In simplified terms, an ontology can be seen as a directed, weighted graph. The weighting thereby is defined by the semantic meaning of the edge. The approaches can be split into those which calculate the distance of ontological concepts and those which aims to calculate the similarity between instances. The following example does not introduce a specific metric (because there are tons of it) but give a first insight into topological distance measures. The graph outlined in figure 10.1 shows a rather simple ontology graph for the description of people and their profession. To keep it simple I spared the labels, directions and weightings for edges.

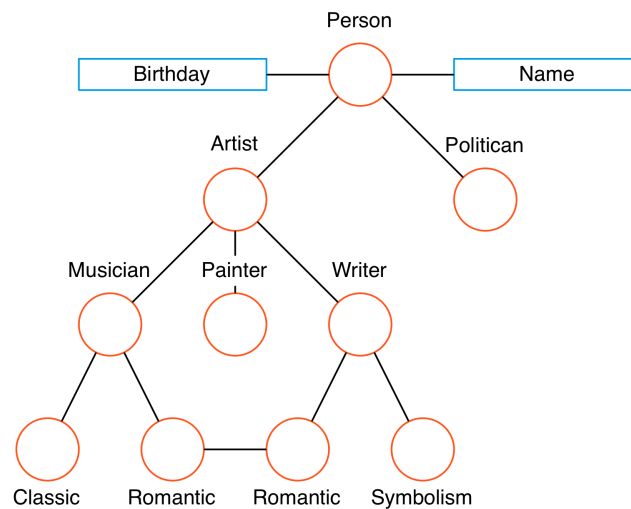


Figure 10.1: Simplified ontology example graph

Distance of ontological concepts

A common distance metric in a graph is the shortest path, whereby it is sufficient to count the number of edges on this path. The shortest path can be efficiently computed using e.g. the Dijkstra algorithm. Using this, I can infer from the sample graph, that a *painter* is more related to a *musician* (distance 2) than to a *politician* (distance 3). With the same metric and graph I get the information that the two concepts of *romantic* are stronger related than e.g. the concepts *classic* and *symbolism*. This is due to how the ontology graph is constructed, which is the major disadvantage of pure ontological distance metrics.

Distance of ontological instances

Taking into account instances of ontological concepts allows to differentiate between things that share the same concept and thus leads to more fine-grained results.

Given three instances of the example ontology namely *mozart::musician-classic*, *traktl::writer-symbolism* and *josef_II::politician*. With a shortest-path algorithm the distance of the instances is four in all cases. Extending the approach to instance-specific values like year of birth, this distance differs, as the euclidian distance of *mozart* and *josef_II* for this property is 15, whereby *mozart* and *traktl* have a distance of 131. Hence, the distance between *mozart* and *josef_II* is lower - they are more similar to each other. Considering more/other attribute values (e.g. birthplace, which have *mozart* and *traktl* in common) can change the relational distance to a high amount. In addition to attributes, current algorithms take also into account the number of relations between concept instances.

The complexity of both approaches is the selection and weighting of relations and attributes. Furthermore the metrics of attributes have to be defined with caution. There is a broad range of algorithm which mainly deal with this problem of weighting and selection, like e.g. DiShIn (Disjunctive Shared Information) [CS11], which takes into account common successors, or LDSD (Linked Data Semantic Distance) [Pas10], which uses direct and distinct links as weighting indicator.

Corpus-based semantic distance

Corpus-based semantic distance metrics use a statistical model based on an a-priori defined set of textual content (which is called corpus). The advantage of this approaches is that the model can be trained in advance and thus the main amount of computation time and resources is consumed in advance. This leads to better runtime performances. The number of statistic algorithms is high, therefore I only consider three basic techniques, which build the basis to many other algorithms. Additionally, as the thesis is grounded in the area of Semantic Web, I describe the Normalized Semantic Web Distance, which is kind of a hybrid metric.

Latent Semantic Analysis

The LSA (Latent-Semantic-Analysis)[LD97], a popular technique of corpus-based similarity and basis for a set of extensions (e.g. GLSA [MLFR05]), rests on the assumption that semantic similar terms are places on similar locations within texts. It is based on word-counts within sentences or paragraphs that are calculated in huge text-corpora which are stored in a matrix [$word \times |occurrences_{textpart}|$]. The basic concept of the approach thus is to map words to vectors of real numbers which is known as word embeddings. It uses singular value decomposition(SVD) to minimize the amount of columns to significant ones and thus reduce the dimensions of the word vectors. The comparison of words is done by cosine-similarity. Like any corpus-based approach the quality of results is strongly related to the characteristic of the underlying text-corpus.

Word2Vec and distance measurement

Word2Vec is a group of models of deep neuronal networks that are trained to reconstruct (linguistic) context of words. Based on a textual corpus the output of the training is a vector space where each single word in the corpus is represented by a high-dimensional vector. Like LSA, Word2Vec is based on word-embeddings and uses dimension-reduction and other mathematical processes to minify the word vectors to a usable size. Experiments show that Word2Vec models automatically organize concepts and learn implicitly the relationships between them [MSC⁺13] which leads to similar representations, like outlined in a high-level Figure 10.2¹. As the model transforms terms into high-dimensional, relational vectors it supports well-know arithmetic operations. Hence, simple formulas like $Paris - France + Italy = ?$ leads to meaningful results (*Rome* in this case).

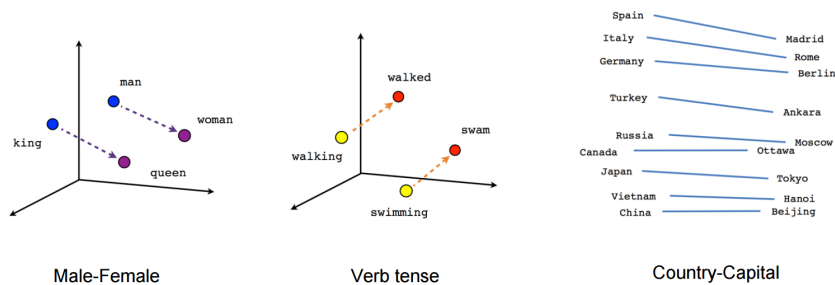


Figure 10.2: Implicit semantic term relationships in Word2Vec models

Normalized Google-Distance

The normalized google distance (NGD) [Kol09] is based on the number of documents returned by the Google search engine for a set of keywords in a given corpus. It assumes that keywords that have a similar meaning tend to occur in same documents. The distance metric is defined as:

$$NGD(x, y) = \frac{\max\{\log f(x), \log f(y)\} - \log f(x, y)}{\log N - \min\{\log f(x), \log f(y)\}} \quad (10.1)$$

whereby M is the total amount of documents in the corpus. $f(x)$ and $f(y)$ are the number of corpus hits for keywords x and y ; $f(x, y)$ is the number of documents that are returned for a combined search. The "closer" two keywords are, the lower is the NGD. The metric uses Google processing algorithms which already includes a lot of (pre-) processing for search terms and corpus documents, so it is expected to lead to more precise results than pure term based techniques like e.g. tf/idf [BYRN99].

¹Term-relationships in Word2Vec:

<https://www.tensorflow.org/tutorials/representation/word2vec>

Normalized Semantic Web Distance

Normalized Semantic Web Distance (NSWD) [DNBG⁺16] aims to reuse NGD principles and adapt them to graph-awareness. The basis concept thereby tries to lower the distance of two concepts if they are used to describe the same things. This metric is defined as:

$$NSWD_{\lambda}(x, y) = \frac{\max\{\log |V_{\lambda}(x)|, \log |V_{\lambda}(y)|\} - \log |V_{\lambda}(x) \cap V_{\lambda}(y)|}{\log |V| - \min\{\log |V_{\lambda}(x)|, \log |V_{\lambda}(y)|\}} \quad (10.2)$$

in a given knowledge graph V, T (nodes V , triples $T \subseteq V \times P \times V$) with $V_{\lambda}(x) = V_{in}(x) \cup V_{out}(x) \cup V_{all}(x)$, whereby $V_{in}(x)$ are nodes linking to x , $V_{out}(x)$ are nodes linked to by x , and $V_{all}(x)$ are nodes that link to x or that y links to. Examples show that the NSWD outperform NWD in most of the test cases and is more correct regarding *Semantic Awareness* (e.g. robust against synonyms).

10.2 Spatio-temporal Fragment Distance

In Chapter 6 I introduced how to specify spatio-temporal fragments for media assets. Together with semantic concepts these fragments allow a rich and detailed description for media assets. The similarity of concepts that appear in the asset promises a good basis for media similarity. Nevertheless similar concepts themselves are a just a vague indicator as they miss the spatial-temporal position and relationship which is a major indicator for a visual similarity for images and videos. In this Section I introduce an approach, how both concept and positional similarity can be combined in order to enhance the quality of an image similarity algorithm. Temporal positioning for video and audio is not part of this Section. In the following I introduce a concept for a fragment based media similarity.

Idea

Figure 10.3 shows a typical example for an annotated image. The thesis is, that a similarity approach which only considers the concepts will perform weaker than an approach which uses the information that is given by the fragment identifiers in addition. The main idea in my approach is inspired by [EU05] and selects a most significant fragment out of the set of fragment annotations for each image. Therefore a metric has to be defined that considers the size and the centrality of every single fragment and allows to define an order on the set of media fragments within one image. The center of most significant fragment defines the fragment fixpoint. The similarity of two fragments regarding position is then calculated using the cosine similarity of fragments relative center vectors (the vectors from the fixpoint to the fragment centers) like outlined in Figure 10.4. The bold rectangles are thereby the most significant fragments with their centers defining the fragment fixpoints. Together with semantic concept similarity, this allows to define a distance metric between linked media fragments - the Linked Media Fragment Distance.

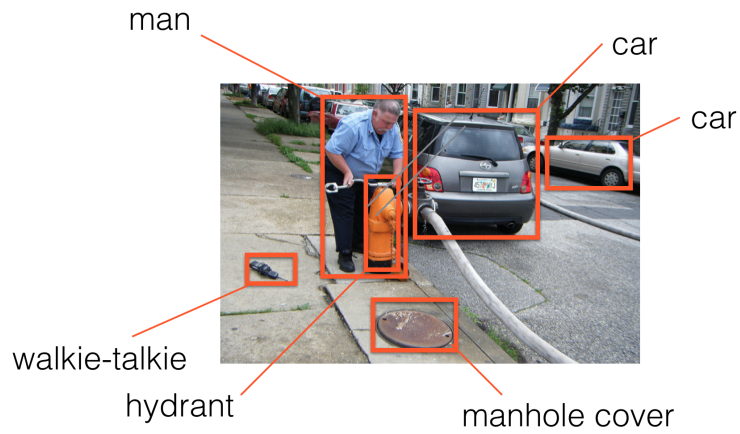


Figure 10.3: Media Similarity Evaluation: Example of an annotated image

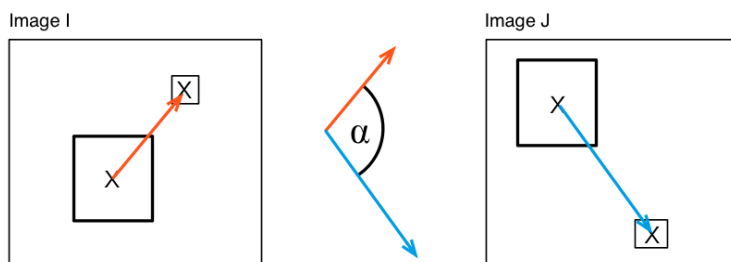


Figure 10.4: Linked Media Fragment Distance: Idea

Approach

Distance measurements for fragments are focusing the distribution of the fragments based on a root fragment. The identification of the root fragment is based in the Normalized Area in combination with the prominence and the position, which is calculated using the Normalized Center Distance. I base on the definitions give in Chapter 5 for Salient Image objects and fragments.

Definition 52 *The center vector $\vec{c}_{(f)}$ of a rectangular fragment $f \in P$, with $p = \langle \vec{a}, \vec{b} \rangle$ is defined as*

$$\vec{c}_{(f)} = \left(\frac{x_a + x_b}{2}, \frac{y_a + y_b}{2} \right) \quad (10.3)$$

Definition 53 *The width of a rectangular image fragment $f \in P$, with $f = \langle \vec{a}, \vec{b} \rangle$ is defined as a function $width : P \mapsto \mathbb{R}^+$:*

$$width(f) = x_b - x_a \quad (10.4)$$

The *height* function is defined analogously.

With these definitions I am able to define the Normalized Center Distance. Note, in the following definitions I consider the visible image ι in the same way as a fragment on the physical salient object layer of an image $i \in I$ with $i = \langle (0, 0), (w(i), h(i)) \rangle$, whereby $w : I \mapsto \mathbb{R}^+$ defines that maximum visible value of X , and $h : I \mapsto \mathbb{R}^+$ defines that maximum visible value of Y .

Definition 54 *The normalized center distance (NCD) for a fragment f_i regarding an image i is defined as*

$$NCD(i, f_i) = \frac{\|\vec{c}_{(\iota)} - \vec{c}_{f_i}\|}{\sqrt{w(i)^2 + h(i)^2}} \quad (10.5)$$

The center distance is the (euclidean) distance between two center vectors. The normalization enables a comparison of images regardless the actual size.

Definition 55 *The normalized area NA for a fragment f_i regarding an image i is defined as*

$$NA(i, f_i) = \frac{width(f_i) * height(f_i)}{width(\iota) * height(\iota)} \quad (10.6)$$

Like for center distance, normalization allows to ignore differences in image sizes.

Definition 56 *Normalized Fragment Significance NFS describes the significance of an image fragment f_i within an image i and is defined as*

$$NFS(i, f_i) = NA(i, f_i) * (1 - NCD(i, f_i)) \quad (10.7)$$

So a fragment which has the same dimensions as the image (and thus is centered by default) has a NSF of 1. As a center of a fragment is not necessarily part of the visible part of the physical salient object layer, the NSF may be lower than 0.

Definition 57 Let $F_{(i)}$ be the set of all fragments of an image i . The most significant fragment $f^*(i) \in F_{(i)}$ is a fragment that fulfills the following condition:

$$NFS(i, f^*(i)) \geq f_{n(i)} \quad \text{for } f_{n(i)} \in F_{(i)}, n \in \mathbb{N} \quad (10.8)$$

This allows the identification of the most significant fragment. It can be used to identify a fragment fixpoint for the calculation of similarity of fragments regarding relative positioning.

Definition 58 A fragment fixpoint $\vec{\phi}_{(i)}$ for an image i is defined as the center of the most significant fragment $f^*(i)$.

This fixpoint is the basis for all further calculations. It enables to describe the relative position of all fragments regarding f^* with relative center vectors.

Definition 59 Let $f_{(i)}$ a fragment of image i . The relative center vector $\vec{\psi}_{f_{(i)}}$ is defined as the displacement of the center vector $\vec{c}_{f_{(i)}}$ of the fragment and the fragment fixpoint of $\vec{\phi}_{(i)}$:

$$\vec{\psi}_{f_{(i)}} = \vec{\phi}_{(i)} - \vec{c}_{f_{(i)}} \quad (10.9)$$

These vectors represent the disposition of all fragments in relation to the most significant fragment. A well known metric like the cosine similarity can be used to compare fragments of different images based in their relation to their fixpoints. This leads to the Spatial Fragment Similarity.

Definition 60 With the taken definitions the Spatial Fragment Similarity (SFS) of the fragments $f_{(i)}, f_{(j)}$ of two images i, j regarding their fixpoints can be calculated as follows:

$$SFS(f_{(i)}, f_{(j)}) = \frac{\vec{\psi}_{f_{(i)}} * \vec{\psi}_{f_{(j)}}}{\|\vec{\psi}_{f_{(i)}}\| * \|\vec{\psi}_{f_{(j)}}\|} \quad (10.10)$$

As the cosine similarity creates values in the interval $[-1, 1]$ it can be easily normalized in order to use it for comparison. Note, that this metric does not consider vector length. Other metrics like Euclidean or Manhattan distance would consider also vector length but must be normalized e.g. by using the dimensions of the regarding f^* . As the actual metric is exchangeable, evaluation with other metrics may be part of further work.

10.3 Semantic fragment similarity

The Normalized Spatial Fragment Similarity and Semantic Concept Similarity can be combined in order to get more accurate result in image similarity use cases. In

this section I describe an algorithm for the calculation of a combined similarity value. For the matter of efficiency, the approach is split in the two steps preparation (can be done once for every dataset) and runtime (things that are done during every query execution). The preparation can be split in two main issues, whereby the first issue handles things that are necessary for *Semantic Similarity* and the second one cares about the preprocessing for fragment similarity.

Enabling Semantic Similarity

As described above there are various metrics that enables semantic concept similarity measurement. In order to allow a combination this values has to be normalized. Beside this constraint any metric can be used. Let A be a set of annotations. For the following algorithm I consider a function $sim : A^2 \mapsto [0, 1]$ that returns a similarity value between 0 and 1 for two annotations as given. To access the annotation a_f for fragment $f \in F$ (only 1 to 1 relationship is considered) I take a function $anno : A \mapsto F$ as given.

Enabling Fragment Similarity

To guarantee a performant query execution the *Normalized Fragment Significance* for all fragments, the most significant fragment, and thus the fragment fixpoint for each image $i \in I$ can be calculated in advance. Therefore I consider the set of fragments F_i for an image i as ordered by *NFS* descending. In addition the relative center vectors for each fragment per image can be set, so I take a function $vector : F \mapsto \mathbb{R}^2$ (return the relative center vector for an Fragment) as given.

Algorithm

The image similarity can be calculated using the functions above.

Algorithm 5: Fragment Similarity Algorithm

Data: A set of fragments F_i for an image i
 A set of fragments F_j for an image j

```

1 imgSim = sim( $f_{i_0}, f_{j_0}$ );
2 foreach  $f_i$  in rest( $F_i$ ) do
3   fragSim = 0;
4   foreach  $f_j$  in rest( $F_j$ ) do
5     fragSim = max(fragSim, sim( $f_i, f_j$ ) * fragSim( $f_i, f_j$ ))
6   end
7   imgSim += fragSim;
8 end
9 normImgSim = imgSim /  $|F_i|$ ;
10 return normImgSim;
```

In line 1 the `imgSim` is initially set with the semantic similarity if the most sign fragments. For each fragment in F_i except the most significant the algorithm calculates

the maximum fragment similarity to a set $rest(F_j)$ (all fragments of F_j except the most significant one) in lines 2-8 and adds it to the `imgSim`. The result is normalized in line 9 and returned. Note, that the function *fragSim* is defined as

$$fragSim(f_i, f_j) = \frac{1 + SFS(f_i, f_j)}{2}. \quad (10.11)$$

It returns a normalized value of the *SFS* in the interval $[0, 1]$.

10.4 Conclusion

In this section I aim to describe a concept for semantic image similarity considering fragment positioning. I gave an overview on semantic distance metrics including topological- as well as corpus-based approaches. I presented the idea of spatial fragment similarity and described it in a formal way. In addition I presented an algorithm to compute *Semantic Fragment Similarity*. In the next section I will test the approach on a big image set and evaluate it with human testers.

Evaluation

In this Chapter I outline an evaluation experiment in order to test the quality of the Semantic Fragment Similarity approach that is described in Chapter 10.

11.1 Evaluation Environment

The experiment setup focuses on the comparison of two similarity algorithms which are:

Semantic Media Similarity This algorithm considers semantic concepts that are attached to an image. The concepts are hierarchically related, whereby the maximal supported depth is 1. The algorithm produces a similarity value between 0 and 1 for a pairwise set of images for similarity calculation, whereby 1 is a good match and 0 indicates no similarity.

```

Data: A set of fragments  $F_i$  for an image i
         A set of fragments  $F_j$  for an image j
1 imgSim = 0;
2 foreach  $f_i$  in  $F_i$  do
3   fragSim = 0;
4   foreach  $f_j$  in  $F_j$  do
5     fragSim = max(fragSim,  $sim(f_i, f_j)$ )
6   end
7   imgSim += fragSim;
8 end
9 normImgSim = imgSim /  $|F_i|$ ;
10 return normImgSim;

```

whereby the Concept Similarity function sim for a pair of image annotations x and y is defined as

$$sim(x, y) = \begin{cases} 1 & \text{if } x = y \\ 0.25 & \text{if } parent(x) = parent(y) \\ 0 & \text{otherwise} \end{cases} \quad (11.1)$$

Semantic Fragment Similarity The algorithm itself is described in Chapter 10 and results in a value between 0 and 1 for a pairwise set of images, whereby 1 is a good match and 0 indicates no similarity. In this experiment we use the algorithm described in Formula 11.1 as concept similarity function.

As testset I use the COCO dataset ¹ with 40504 images with 291875 annotations in total, like already described in Chapter 8. Because the algorithm depends on set of annotations, I filtered the set to all images that includes at least 3 annotations, which results in 27815 images. In order to get a critical mass of overlapping evaluation results, I reduced the testset to 35 randomly chosen images. In this set the average number of annotations is 10.48 (Median: 9, StdDev: 7.5).

Evaluation System

As testers I choose a group of volunteers, which have been confronted with a series of 6 different images from the testset. Figure 11.1 shows the user interface for one image. On the left side you can see the main image, on the right side there is a list of 8 similar images ordered by similarity value. The testers are not aware, which algorithm built the results (Semantic Fragment Similarity (FRAG) or Semantic Media Similarity (MEDIA)). For each image selected by the algorithm the tester has to choose a value out of *Very Similar*, *Similar*, *Not Similar* or (for safety reasons in case the image can not be loaded for any case) *No Image*. The testers have been informed about the test and the algorithms before and told that purely visual indicators like color, size, etc. should not be considered for the selection. But they should focus on the concepts that appear and how they are spatially related.

The tests produced 21 complete results which lead to 3.6 evaluation results per image in average, which means 1.8 results per image and algorithm (21 testers * 6 images per test / 2 algorithms / 35 images overall = 1.8). In order to get interpretable results, I reduced the image set to those, which got at least 1 evaluation for each algorithm. This leads to 22 images with 101 evaluation results overall. That lifts the number of evaluations per image and algorithm to 2.3 in average (with a standard deviation of 1.95).

11.2 Results

Figure 11.2 shows the average percentage of the options that has been selected by the users. The results are split in two groups:

All include results from all 8 images that are presented as similar.

Top include only values from the top 3 images.

It is obvious that *FRAG* outperforms *MEDIA*, because it produces less false positives (*not similar*) but more true positives (*similar* and *very similar*). In the Top group the results show even more the benefit of Semantic Fragment Similarity. This is due to the fact that *FRAG* always provides the exact same image as first match (which *MEDIA* does not). In Figure 11.3 I listed the sum of selected option values for each image, whereby *not similar* counts 1, *similar* counts 2 and *very similar*

¹COCO dataset: <http://cocodataset.org>

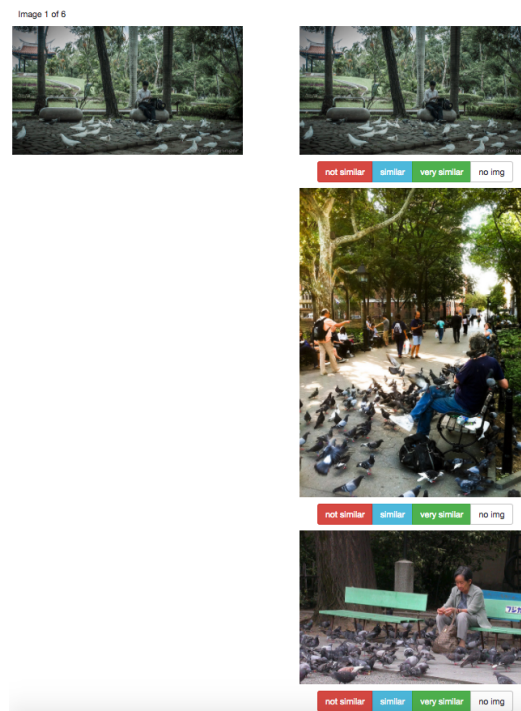


Figure 11.1: Test UI for Similarity Metrics Evaluation

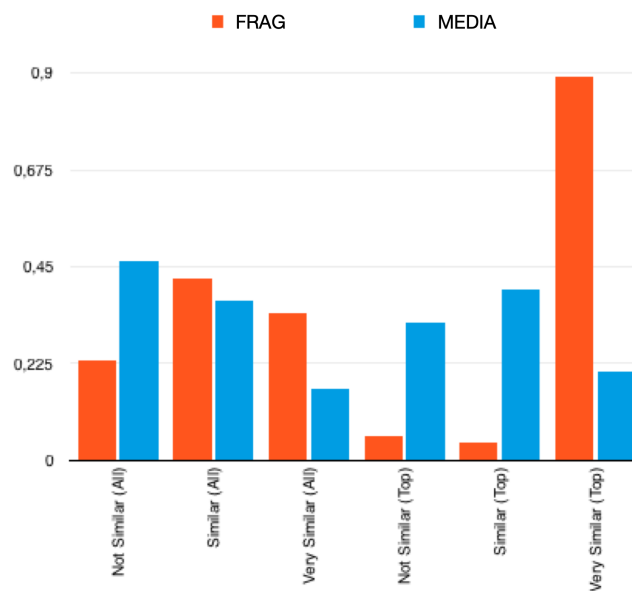


Figure 11.2: Similarity Metrics Evaluation: Options selected in AVG

counts 3. It shows that in most of the cases *FRAG* outperforms *MEDIA*. This becomes even more obvious, when I calculate the weighted sum that takes into account the ranking position of the image presented as similar (Figure 11.4). The weighting thereby divides the value by the position number [1,6]. The outliers

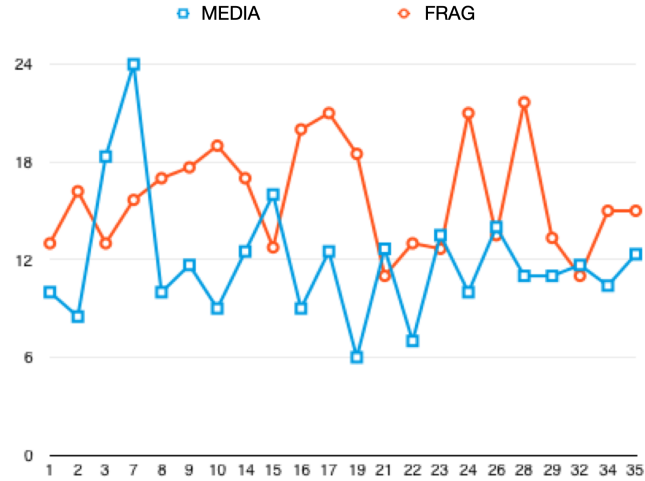


Figure 11.3: Similarity Metrics Evaluation: Sum of values per image

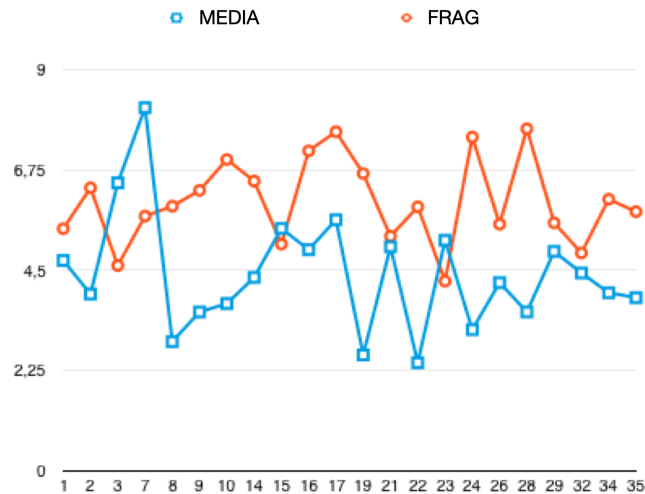


Figure 11.4: Similarity Metrics Evaluation: Weighted sum of values per image

in the evaluation, namely images 3 and 7, can be explained by annotations that does not describe the image very well. In addition the current approach does not consider the area of fragments (except for the most significant one), which leads to down-rating of important (big) fragments and up-rate of tiny fragments just because of their positioning. This may be improved in a further extension of the approach.

11.3 Conclusion

In this Chapter I tested the Semantic Fragment Similarity that I introduced in Chapter 10 using a dedicated test scenario. It used A/B testing, whereby testers are confronted with one image and a set of similar images. The algorithm thereby is unknown to the testing person. The images are a subset of the COCO data set of annotated images. The test results show that the algorithm, which uses Semantic Fragment Similarity as metric, produces better results in most cases.

Part VI

Summary

Résumé

In this thesis I elaborated the integration of Multimedia in the Web of Data and identified a substantial gap, namely the lack of adequate information retrieval functionality. The main aim of the work was, to minimize this gap by investigating and evaluating proper approaches for Multimedia querying in the Semantic Web. In this chapter I will recapitulate the 3 main Parts, which are a) a requirement analysis for Multimedia query languages based on an exhaustive survey, b) the definition of a query language extension for SPARQL, the de-facto standard query language for the Semantic Web, and c) the elaboration and evaluation of a Media Similarity approach based on related semantic concepts and fragment distribution. In this Section I will summarize the work and discuss outstanding and further work for each Part.

12.1 Conclusion

In the survey I provided an overview of Multimedia query languages arising in the last four decades. This includes more than 70 instants varying in basic concepts and use case requirements. To give a deeper insight, I selected a smaller subset that I introduced in detail including a dedicated feature set and usage examples. Based on that I defined a set of requirements for Multimedia query languages, which is separated in 2 parts, namely general and specific demands. The outcome is a list of seven well defined Multimedia features. It includes support for media types per se, spatial and temporal operations, temporal evolution, metadata operations, media similarity functions and result weighting. This set builds a proper basis for the work in subsequent Chapters, namely the adaption of SPARQL to Multimedia facilities.

In order to find valuable extensions for SPARQL I did a dedicated use case analysis which results in 20 example queries that are mapped to the feature requirements of the former Chapter. As basic model I extended the DISIMA image model to video, spatio-temporal fragments, and animations. The well defined definitions provide a solid grounding for a more higher level class and instant model, which has been described using the ontology language OWL. It includes a basic set of spatial shapes (rectangle, circle, etc.), classes for the description of temporal instants and intervals as well as a set of properties that enable a combination of both in order to describe spatio-temporal fragments. This allows to straightforwardly define a set of 53 spatio-temporal functions as SPARQL filters. I called this extension SPARQL-MM. It is grounded on well known algebraic models

for topological, directional and temporal relations.

As the acceptance and usability of query languages strongly rely on performant execution, I described and evaluated a query plan optimization approach based on an existing heuristic algorithm. The contribution of the thesis is the integration of filter selectivity in the existing cost model. The idea behind the approach is that query parts, which are related to high selective filters should be executed in early evaluation phases. Selectivity is thereby based on the reduction factor of the regarding filter function. The evaluation is implemented on a set of more than 40.000 annotated images; the test queries are varying in complexity and filter usage. It shows that the developed extension outperforms the basic algorithm in most of the cases when using SPARQL Multimedia filter functions.

With SPARQL-MM a substantial set of user scenarios can be addressed. While structural as well as complex metadata are covered by SPARQL naturally or can be handled by specific extensions (e.g. the fulltext-search extension in Apache Marmotta¹), SPARQL-MM supports spatio-temporal access, operations and relations. Nevertheless there still remains a gap regarding Multimedia similarity functionality. To overcome this, I described a novel media similarity metric, which considers both, semantic (concept) similarity and the distribution of image fragments. The idea behind is that the place and size of interesting image parts (which often coincide with annotated fragments) have a high impact in the perception regarding image similarity. Therefore the approach aims to identify the most significant image part and involves the spatial relation to all other (annotated) fragments in the process of similarity calculation. The results have been evaluated by an A/B test, where an image together with a ordered set of similar images are shown to users. Without a hint to the underlying algorithm, they had to decide if and on which degree the images are similar. The results show that the metric outperforms pure semantic similarity in many cases.

Note, that in this thesis I did not consider temporal evolution as it was not part of the use cases. But as visual time-series analysis is a feature that is often used, it could be part of further aims.

¹Fulltext-search in Apache Marmotta:

<https://marmotta.apache.org/kiwi/sparql-full-text.html>

12.2 Further Work

The survey of Multimedia query languages targets to be complete in case of instants. Nevertheless the current description only summarizes the whole set and tries to identify streams and differences of the languages in both, features and date of publication. The work which has done here provides a proper basis for further efforts like a detailed compilation of query languages. As the detailed summary of every single query language is done but did not made it into the thesis for the matter of space, this would be a valuable scientific contribution. In addition I only extracted a set of the most important features, which can be extended to more specific use cases. Together with a complete feature matrix (that identifies and scores functions for all query languages) such efforts would provide a major benefit for scientists working in the field of Multimedia retrieval. It would allow to match feature requests directly to existing languages and thus reduce the amount of efforts that aim to build things from scratch, but lead to more iterative proceedings by using extension mechanisms of already existing works.

The feature set of SPARQL-MM is rather complete in case of spatio-temporal functions but can be extended in many ways, like temporal evolution, media similarity and result weighting. Especially the last one could be done quite straight forward based on existing work regarding fuzzy SPARQL evaluation. In addition an integration of special functions for metadata operations, e.g. fulltext search, or time-code translations would be a further step in the direction of a comprehensive query language for Multimedia.

The optimization algorithm for SPARQL-MM, which has been described within the thesis is currently just implemented in order to allow the evaluation process. Hence, it is not fully included in existing SPARQL interpreters, like RDF4J or Apache Marmotta. The main work to do in this case is the implementation of optimal indexes for media fragments (e.g. R-Trees), which have performant and flexible update operations while keeping the evaluation time of SPARQL-MM functions short. This would improve the acceptance of SPARQL-MM even for higher scaling data sources. Looking at the actual optimization approach, it could be adapted to a wider range of SPARQL filters, which would improve also the evaluation efficiency for non media-specific queries. In addition it could consider non just heuristic selectivity ordering but be extended to statistic values of the search space. A valid starting point for this could be literal filter functions, e.g. regex.

As SPARQL-MM is an extension for SPARQL, so a description of the functionalities using standardized patters would be useful. But a well defined description mechanism for filter functions is missing and thus should be part of further standardization processes. Currently there are discussions regarding several

refinements of the recommendation in the SPARQL 1.2 Community Group², so a contribution in this direction would be rational.

The Multimedia similarity metric described above can be seen as a first investigation and thus as a proof on concept, which shows that considering fragments and their distribution in the image space have a major impact. As discussed, simple improvements like taking into account the fragment area would even enforce this fact. As the approach is not exclusively bound to semantic similarity, it can also be used for novel technologies like as input feature for Machine Learning algorithms.

As one can see, the thesis narrows the gap between the Web of Data and Multimedia. Thus reaches, together with existing standards for media annotation, its goal to turn media items to full citizens of the Web of Data. Nevertheless it is just a starting point and so I will close the thesis with a cite of Alan Turing, one of the fathers of theoretical computer science and artificial intelligence:

"We can only see a short distance ahead,
but we can see plenty there that needs to be done."

Alan Turing (1912-1954)

²SPARQL 1.2 Community Group: <https://www.w3.org/community/sparql-12/>

Directory for Publications

A.1 Books and Articles

Semantic enhancement for media asset management systems

Authors:

Thomas Kurz, Georg Güntner, Violeta Damjanovic, Sebastian Schaffert, and Manuel Fernandez

Published in:

Multimedia Tools and Applications, Volume 70, Issue 2, Pages 949-975, May 2014.

Contribution:

Thomas Kurz is the main author of the publication and contributed in the following area: idea, description of related work, semantic enhancement, the description of the Linked Media Framework (LMF) and the section about the smart media pool. Together with Schaffert, Kurz is the main code contributor of the LMF. Güntner participated in the in chapter about media annotation frameworks and outlook, whereby Damjanovic gave input in the section about engines and tools for media enrichment. Fernandez participated in the chapter about semantic video search. In addition he enabled the research by providing real-world data sets. Güntner and Schaffert supported Kurz in research methodology and scientific writing.

Authors (<i>italic = main</i>)	Contributions in the Publication
<i>Kurz</i>	Abstract
<i>Kurz</i> , Güntner, Schaffert	1. Introduction
<i>Kurz</i> , Güntner	2. Related work
<i>Kurz</i> , Damjanovic	3. Semantic enhancement
<i>Kurz</i> , Schaffert	4. LMF - Linked Media Framework
<i>Kurz</i> , Fernandez	5. A smart media pool
<i>Güntner</i> , Kurz	6. Conclusion and outlook

Relation to dissertation:

The article took a central role in the basic research that leads to this thesis, hence the introduction part is used partly in the motivation section in chapter 1. The sections about related work (web of data, media on the web, etc.) are partly integrated in updated versions in the in chapter 2 (Linked Media). The publication is listed in the bibliography as [KGD⁺14].

State of the Art in Cross-Media Analysis, Metadata Publishing, Querying and Recommendations

Authors:

Patrick Aichroth, Johanna Björklund, Florian Stegmaier, *Thomas Kurz*, and Grant Miller

Published in:

Mico - Media in Context, Technical Report: Vol.1, ISBN 978-3-902448-43-9, August, 2015.

Contribution:

Thomas Kurz is the editor of the volume. His main contribution to the report is Chapter 4 (Multimedia Querying) where he is the main author. Kai Schlegel contributed as Research Assistant partly to the State of the Art Section about SPARQL extensions (4.2.2).

Authors (<i>italic = main</i>)	Contributions in the Publication
<i>Kurz</i>	4. Multimedia Querying
<i>Kurz</i>	4.1 Multimedia Query Languages
<i>Kurz</i> , Schlegel	4.2 Semantic Web Query Languages
<i>Kurz</i>	4.2.1 SPARQL Protocol and RDF Query Language
<i>Schlegel</i> , <i>Kurz</i>	4.2.2 SPARQL Extensions

Relation to dissertation:

The introduction of Chapter 4 as well as the detailed view on query languages (4.1) is partly reused in Chapter 3 of the thesis. The description of Semantic Web query languages in Section 4.2 has found its way in Section 2.2. The part where Schlegel contributed to is just used as scaffolding but has been completely rewritten. The publication is listed in the bibliography as [ABS⁺15b].

Specifications and Models for Cross-Media Extraction, Metadata Publishing, Querying and Recommendations - Version I

Authors:

Patrick Aichroth, Henrik Björklund, Johanna Björklund, Kai Schlegel, *Thomas Kurz*, and Grant Miller

Published in:

Mico - Media in Context, Technical Report: Vol.2, ISBN 978-3-902448-44-6, October, 2015.

Contribution:

Thomas Kurz and Henrik Björklund are the editors of the volume. Kurz’s main contribution to the report is Chapter 6 (SPARQL-MM Query Model) where he is the single author.

Authors (<i>italic = main</i>)	Contributions in the Publication
<i>Kurz</i>	6. SPARQL-MM Query Model

Relation to dissertation:

The class and property model defined in Section 6.2 of the report is the basis for the model described in Sections 6.2 and 6.3 of the thesis. The report Section 6.2 is the basis for the SPARQL-MM function definitions in Chapter 7. The publication is listed in the bibliography as [ABS⁺15c].

Specifications and Models for Cross-Media Extraction, Metadata Publishing, Querying and Recommendations - Version II

Authors:

Patrick Aichroth, Johanna Björklund, Kai Schlegel, *Thomas Kurz*, and Thomas Köllmer

Published in:

Mico - Media in Context, Technical Report: Vol.4, ISBN 978-3-902448-46-0, December, 2015.

Contribution:

Thomas Kurz is the editor of the volume. His main contribution of the report is Chapter 4 (Specifications and Models for Cross-media Querying) where he is the single author.

Authors (<i>italic = main</i>)	Contributions in the Publication
<i>Kurz</i>	4. Specifications and Models for Cross-media Querying

Relation to dissertation:

Section 4.2 is describing a set of SPARQL extensions. It contains a summary of GeoSparql Extension ¹ (4.2.2), extensions of SPARQL-MM functions (4.2.3), as well as a description of SPIN (4.2.4). In addition it shows an extension of Media Fragment URIs in Section 4.2.1, which is the basis for [KK16] and thus not used directly within the thesis. Hence, 4.2.3 is used in Chapter 7 and 4.2.4 is partly used in Section 2.2. The publication is listed in the bibliography as [ABS⁺15a].

¹GeoSPARQL Extension <https://www.w3.org/2011/02/GeoSPARQL.pdf>

Enabling Technology Modules: Final Version

Authors:

Patrick Aichroth, Johanna Björklund, Emanuel Berndl, *Thomas Kurz*, and Thomas Köllmer

Published in:

Mico - Media in Context, Technical Report: Vol.5, ISBN 978-3-902448-47-7, December, 2016.

Contribution:

Thomas Kurz is the editor of the volume. His main contribution to the report is Chapter 5 (Enabling Technology Modules for Cross-media Querying) where he is the main author. He is the single author of Section 5.1, where he listed the final version of SPARQL-MM extensions and described the reference implementation. Section 5.2 is a joint work of the Salzburg Research Knowledge and Media Technology group, namely Thomas Kurz, Sebastian Schaffert, Sergio Fernandez and Jakob Frank and introduced a path based Semantic Web query language. Section 5.3 is a contribution of Kurz about Semantic Media Similarity.

Authors (<i>italic = main</i>)	Contributions in the Publication
<i>Kurz</i>	5.1 SPARQL-MM Extensions
<i>Kurz</i> , Schaffert, Fernandez, Frank	5.2 Linked Data Information Retrieval
<i>Kurz</i>	5.3 Semantic Media Similarity

Relation to dissertation:

Section 5.3 about Semantic Media Similarity is an early work that is continued, refined and evaluated in Chapters 10 and 11 of the thesis. The publication is listed in the bibliography as [ABB⁺16].

Smarte Annotationen: Ein Beitrag zur Evaluation von Empfehlungen für Annotationen

Authors:

Sandra Schön and *Thomas Kurz*

Published in:

Linked Media Lab Reports, Issue 4, ISBN 978-3-902448-31-6, October, 2011.

Contribution:

Sandra Schön and Thomas Kurz are the authors of the book, whereby Schön is the main author. Kurz contributed Chapter 5 about similarity metrics of semantic

annotations. Within this, Schön took over the part of an editor and proof reader.

Authors (<i>italic = main</i>)	Contributions in the Publication
<i>Kurz</i> , Schön	5. Vorschläge zur Beurteilung von Nähe und Abweichungen von Annotatonen (p43-46)

Relation to dissertation:

The only relevant part for the thesis is Chapter 5 about similarity metrics of semantic annotations, which builds a basis for Section 10.1. The part is not only translated but adapted, enhanced and updated to the most recent State of the Art. The publication is listed in the bibliography as [SK11a].

A.2 Proceeding Papers

Lifting Media Fragment URIs to the next level

Authors:

Thomas Kurz and Harald Kosch

Published in:

Proceedings of the 4th International Workshop on Linked Media, ESWC2016, May, 2016.

Contribution:

Thomas Kurz is the main author of the paper. Harald Kosch supported the work by conceptual discussions and research methodology. The work has been inspired by works of and talks with people within the Linked Media research field, namely Thomas Steiner and Olivier Aubert.

Authors (<i>italic = main</i>)	Contributions in the Publication
<i>Kurz</i>	Abstract
<i>Kurz</i> , Kosch	1. Introduction
<i>Kurz</i>	2. Media Fragment URIs 1.0
<i>Kurz</i> , Kosch	3. Media Fragment URI Extensions
<i>Kurz</i>	4. Related approaches
<i>Kurz</i>	5. Styling Media Fragments
<i>Kurz</i>	6. Conclusion

Relation to dissertation:

The paper is basis for the excursion in Section 6.5 in the thesis. The publication is listed in the bibliography as [KK16].

Enabling access to Linked Media with SPARQL-MM

Authors:

Thomas Kurz, Kai Schelegel, and Harald Kosch

Published in:

Proceedings of the 24th International Conference on World Wide Web (WWW), May, 2015.

Contribution:

Thomas Kurz is the main author of the paper. Kai Schlegel contributed in the State of the Art part about SPARQL query language. In addition he participated with conceptual discussions and proof reading. Harald Kosch as PhD supervisor gave input to the Section about concepts of Multimedia Query Languages.

Authors (<i>italic = main</i>)	Contributions in the Publication
<i>Kurz</i>	Abstract
<i>Kurz</i> , Kosch, Schlegel	1. Introduction
<i>Kurz</i> , Schlegel	2. Linked Media
<i>Kurz</i> , Kosch	3. Concepts of Multimedia Query Languages
<i>Kurz</i>	4. Introduction to SPARQL-MM
<i>Kurz</i>	5. Using SPARQL-MM
<i>Kurz</i> , Kosch	6. Conclusion and further work

Relation to dissertation:

The paper is a summary of work which has been done in the Mico project² and described in technical reports mentioned above. It is partly used in Chapters 3, 6, and 7 of the thesis. The publication is listed in the bibliography as [KSK15].

²Mico project: <https://www.mico-project.eu/>

Supplementary Material

B.1 Prefixes

Prefix	Value
mm	http://linkedmultimedia.org/sparql-mm/ns/2.0.0/function#
lmo	http://linkedmultimedia.org/sparql-mm/ns/2.0.0/ontology#
foaf	http://xmlns.com/foaf/0.1/
dct	http://purl.org/dc/terms/
ma	http://www.w3.org/ns/ma-ont#
dc	http://purl.org/dc/elements/1.1/
oa	http://www.w3.org/ns/oa#
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs	http://www.w3.org/2000/01/rdf-schema#
owl	http://www.w3.org/2002/07/owl#
xsd	http://www.w3.org/2001/XMLSchema#
skos	http://www.w3.org/2004/02/skos/core#
ex	http://example.org/

Table B.1: Prefix-Table

B.2 Linked Media Fragment Ontology (LMO)

Listing B.1: Linked Media Fragment Ontology

```

@prefix : <http://linkedmultimedia.org/sparql-mm/1.0.0/ontology#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@base <http://linkedmultimedia.org/sparql-mm/ns/2.0.0/ontology#> .

<http://linkedmultimedia.org/sparql-mm/ns/2.0.0/ontology#>
  rdf:type owl:Ontology .

:hasEndPoint rdf:type owl:ObjectProperty ;
  rdfs:label "hasEndPoint" ;
  rdfs:comment "A shape has a two dimensional Vector as endpoint." ;
  rdfs:domain :Line ;
  rdfs:subPropertyOf :hasVector_2D .

:hasEndTime rdf:type owl:ObjectProperty ;
  rdfs:label "hasEndTime" ;
  rdfs:comment "A time class has a certain Time as endpoint." ;
  rdfs:domain :Interval ;
  rdfs:subPropertyOf :hasTime .

:hasHeight rdf:type owl:ObjectProperty ;
  rdfs:label "hasHeight" ;
  rdfs:comment "A shape has a one dimensional vector as height." ;
  rdfs:domain :Rectangle ;
  rdfs:subPropertyOf :hasVector_1D .

:hasRadius rdf:type owl:ObjectProperty ;
  rdfs:label "hasRadius" ;
  rdfs:comment "A shape has a one dimensional vector as radius." ;
  rdfs:domain :Circle ;
  rdfs:subPropertyOf :hasVector_1D .

:hasStartPoint rdf:type owl:ObjectProperty ;
  rdfs:label "hasStartPoint" ;
  rdfs:comment "A shape has a two dimensional Vector as startpoint." ;
  rdfs:domain :Line ;
  rdfs:subPropertyOf :hasVector_2D .

:hasStartTime rdf:type owl:ObjectProperty ;
  rdfs:label "hasStartTime" ;
  rdfs:comment "A time class has a certain Time as startpoint." ;
  rdfs:domain :Interval ;
  rdfs:subPropertyOf :hasTime .

:hasTime rdf:type owl:ObjectProperty ;
  rdfs:label "hasTime" ;
  rdfs:comment "A time class has a certain Time." ;

```

```
rdfs:domain :Instant ;
rdfs:range :Time .

:hasVector rdf:type owl:ObjectProperty ;
rdfs:label "hasVector" ;
rdfs:comment "A superclass of any relation between
    Thing and Vector." ;
rdfs:range :Vector .

:hasVector_1D rdf:type owl:ObjectProperty ;
rdfs:label "hasVector 1D" ;
rdfs:comment "A supertype for any relation between Thing an one
    dimensional Vectors." ;
rdfs:range :Vector_1D ;
rdfs:subPropertyOf :hasVector .

:hasVector_2D rdf:type owl:ObjectProperty ;
rdfs:label "hasVector 2D" ;
rdfs:comment "A supertype for any relation between Thing an two
    dimensional Vectors." ;
rdfs:range :Vector_2D ;
rdfs:subPropertyOf :hasVector .

:hasWidth rdf:type owl:ObjectProperty ;
rdfs:label "hasWidth" ;
rdfs:comment "A shape has a one dimensional vector as width." ;
rdfs:domain :Rectangle ;
rdfs:subPropertyOf :hasVector_1D .

:hasXY rdf:type owl:ObjectProperty ;
rdfs:label "hasXY" ;
rdfs:comment "A shape has a two dimensional Vector." ;
rdfs:domain :Circle ,
:Point ,
:Rectangle ;
rdfs:subPropertyOf :hasVector_2D .

:hasSpatialEntity rdf:type owl:ObjectProperty ,
owl:FunctionalProperty ;
rdfs:label "hasSpatialEntity" ;
rdfs:comment "The functional relation between a spatio-temporal
    entity and a spatial entity" ;
rdf:domain :SpatioTemporalEntity ;
rdf:range :SpatialEntity .

:hasTemporalEntity rdf:type owl:ObjectProperty ,
owl:FunctionalProperty ;
rdfs:label "hasSpatialEntity" ;
rdfs:comment "The functional relation between a spatio-temporal
    entity and a temporal entity" ;
rdf:domain :SpatioTemporalEntity ;
rdf:range :TemporalEntity .

:animates rdf:type owl:ObjectProperty ,
```

```
owl:FunctionalProperty ;
rdfs:label "animates" ;
rdfs:comment "A property that links a :Animation
    to a :SpatialTemporalEntity." ;
rdf:domain :Animation ;
rdf:range :SpatialEntity .

:Circle rdf:type owl:Class ;
rdfs:label "Circle" ;
rdfs:subClassOf :Curved ;
rdfs:comment "A circle is defined by a two dimensional vector
    (center) and a one dimensional vector (radius)." .

:Curved rdf:type owl:Class ;
rdfs:label "Curved" ;
rdfs:subClassOf :Shape ;
owl:disjointWith :Polygon ;
rdfs:comment "A superclass for all curved shapes." .

:Instant rdf:type owl:Class ;
rdfs:label "Instant" ;
rdfs:subClassOf :TemporalEntity ;
owl:disjointWith :Interval ;
rdfs:comment "An Instant is defined by a Time." .

:Interval rdf:type owl:Class ;
rdfs:label "Interval" ;
rdfs:subClassOf :TemporalEntity ;
rdfs:comment "An Interval is defined by 2 Times (start and end)." .

:Line rdf:type owl:Class ;
rdfs:label "Line" ;
rdfs:subClassOf :SpatialEntity ;
rdfs:comment "A line is defined by 2 two dimensional vectors
    (start and endpoint)." .

:NPT rdf:type owl:Class ;
rdfs:label "NPT" ;
rdfs:subClassOf :Time ;
rdfs:comment "Normal Play Time (NPT) like described in:
    Real Time Streaming Protocol (RTSP). IETF RFC 2326, April 1998.
    Available at http://www.ietf.org/rfc/rfc2326.txt." .

:Point rdf:type owl:Class ;
rdfs:label "Point" ;
rdfs:subClassOf :SpatialEntity ;
rdfs:comment "A Point is defined by 1 two dimensional vector." .

:Polygon rdf:type owl:Class ;
rdfs:label "Polygon" ;
rdfs:subClassOf :Shape ;
rdfs:comment "A superclass for all polygonal shapes." .

:Rectangle rdf:type owl:Class ;
```



```
    rdfs:label "Rectangle" ;
    rdfs:subClassOf :Polygon ;
    rdfs:comment "A Rectangle is defined by a two dimensional
        vector (left-upper point) and 2 one dimensional vectors
        (width and height)." .

:SMPTE rdf:type owl:Class ;
    rdfs:label "SMPTE" ;
    rdfs:subClassOf :Time ;
    rdfs:comment "SMPTE RP 136 Time and Control Codes for
        24, 25 or 30 Frame-Per-Second Motion-Picture Systems." .

:Shape rdf:type owl:Class ;
    rdfs:label "Shape" ;
    rdfs:subClassOf :SpatialEntity ;
    rdfs:comment "A superclass for all shapes." .

:SpatialEntity rdf:type owl:Class ;
    rdfs:label "Spatial Entity" ;
    rdfs:subClassOf :SpatialThing ;
    owl:disjointWith :Time ;
    rdfs:comment "A superclass of any spatial entities like
        point, line, polygone, curcle, etc." .

:SpatialThing rdf:type owl:Class ;
    rdfs:label "Spatial Thing" ;
    rdfs:comment "A superclass for any spatial thing." .

:TemporalEntity rdf:type owl:Class ;
    rdfs:label "Temporal Entity" ;
    rdfs:subClassOf :TemporalThing ;
    owl:disjointWith :Time ;
    rdfs:comment "A superclass of any the temporal entity like
        instant, interval, etc." ;
    owl:disjointUnionOf (
        :Instant
        : Interval
    ) .

:TemporalThing rdf:type owl:Class ;
    rdfs:label "Temporal Thing" ;
    rdfs:comment "A superclass for any temporal thing." .

:Animation rdf:type owl:Class ;
    rdfs:label "Animation" ;
    rdfs:comment "A superclass for all animations." .

:SpatioTemporalEntity rdf:type owl:Class ;
    rdfs:label "Spatio-Temporal Entity" ;
    rdfs:comment "A class that relates to spatial and temporal
        features" ;

rdfs:subClassOf
    [ a owl:Restriction ;
```

```

    owl:onProperty :hasSpatialEntity ;
    owl:cardinality "1"^^xsd:integer
  ] ,
  [ a owl:Restriction ;
    owl:onProperty :hasTemporalEntity ;
    owl:cardinality "1"^^xsd:integer
  ] .

:Time rdf:type owl:Class ;
  rdfs:label "Time" ;
  rdfs:subClassOf :TemporalThing ;
  rdfs:comment "A superclass for any kind of time specification." .

:UTC rdf:type owl:Class ;
  rdfs:label "UTC" ;
  rdfs:subClassOf :Time ;
  rdfs:comment "Coordinated Universal Time (UTC) like defined in:
    Real Time Streaming Protocol (RTSP). IETF RFC 2326, April 1998.
    Available at http://www.ietf.org/rfc/rfc2326.txt." .

:Vector rdf:type owl:Class ;
  rdfs:label "Vector" ;
  rdfs:subClassOf :SpatialThing ;
  rdfs:comment "A superclass for vectors." .

:Vector_1D rdf:type owl:Class ;
  rdfs:label "Vector 1D" ;
  rdfs:subClassOf :Vector ;
  rdfs:comment "A one dimensional vector." .

:Vector_2D rdf:type owl:Class ;
  rdfs:label "Vector 2D" ;
  rdfs:subClassOf :Vector ;
  rdfs:comment "A two dimensional vector." .

[ rdf:type owl:AllDisjointClasses ;
  owl:members ( :Line
    :Point
    :Shape
  )
] .
[ rdf:type owl:AllDisjointClasses ;
  owl:members ( :NPT
    :SMPTE
    :UTC
  )
] .

```

B.3 Categories in Optimization Evaluation

Table B.2: Categories in Optimization Evaluation

CATEGORY	LABEL	SUPERCATEGORY
cat:13	"stop sign"	cat:outdoor
cat:27	"backpack"	cat:accessory
cat:56	"broccoli"	cat:food
cat:39	"baseball bat"	cat:sports
cat:55	"orange"	cat:food
cat:24	"zebra"	cat:animal
cat:61	"cake"	cat:food
cat:19	"horse"	cat:animal
cat:25	"giraffe"	cat:animal
cat:36	"snowboard"	cat:sports
cat:48	"fork"	cat:kitchen
cat:41	"skateboard"	cat:sports
cat:59	"pizza"	cat:food
cat:7	"train"	cat:vehicle
cat:4	"motorcycle"	cat:vehicle
cat:8	"truck"	cat:vehicle
cat:58	"hot dog"	cat:food
cat:64	"potted plant"	cat:furniture
cat:21	"cow"	cat:animal
cat:49	"knife"	cat:kitchen
cat:54	"sandwich"	cat:food
cat:67	"dining table"	cat:furniture
cat:15	"bench"	cat:outdoor
cat:11	"fire hydrant"	cat:outdoor
cat:65	"bed"	cat:furniture
cat:35	"skis"	cat:sports
cat:46	"wine glass"	cat:kitchen
cat:20	"sheep"	cat:animal
cat:90	"toothbrush"	cat:indoor
cat:34	"frisbee"	cat:sports
cat:79	"oven"	cat:appliance
cat:80	"toaster"	cat:appliance
cat:84	"book"	cat:indoor
cat:33	"suitcase"	cat:accessory
cat:51	"bowl"	cat:kitchen
cat:52	"banana"	cat:food
cat:2	"bicycle"	cat:vehicle
cat:1	"person"	cat:person
cat:18	"dog"	cat:animal
cat:32	"tie"	cat:accessory
cat:3	"car"	cat:vehicle
cat:37	"sports ball"	cat:sports
cat:43	"tennis racket"	cat:sports
cat:53	"apple"	cat:food
cat:88	"teddy bear"	cat:indoor
cat:9	"boat"	cat:vehicle

cat:63	"couch"	cat:furniture
cat:47	"cup"	cat:kitchen
cat:82	"refrigerator"	cat:appliance
cat:40	"baseball glove"	cat:sports
cat:89	"hair drier"	cat:indoor
cat:87	"scissors"	cat:indoor
cat:10	"traffic light"	cat:outdoor
cat:77	"cell phone"	cat:electronic
cat:72	"tv"	cat:electronic
cat:14	"parking meter"	cat:outdoor
cat:22	"elephant"	cat:animal
cat:28	"umbrella"	cat:accessory
cat:74	"mouse"	cat:electronic
cat:81	"sink"	cat:appliance
cat:44	"bottle"	cat:kitchen
cat:86	"vase"	cat:indoor
cat:5	"airplane"	cat:vehicle
cat:73	"laptop"	cat:electronic
cat:16	"bird"	cat:animal
cat:75	"remote"	cat:electronic
cat:57	"carrot"	cat:food
cat:60	"donut"	cat:food
cat:78	"microwave"	cat:appliance
cat:50	"spoon"	cat:kitchen
cat:31	"handbag"	cat:accessory
cat:38	"kite"	cat:sports
cat:70	"toilet"	cat:furniture
cat:42	"surfboard"	cat:sports
cat:62	"chair"	cat:furniture
cat:76	"keyboard"	cat:electronic
cat:23	"bear"	cat:animal
cat:85	"clock"	cat:indoor
cat:17	"cat"	cat:animal
cat:6	"bus"	cat:vehicle

B.4 Optimized query plans

	SONG	KURZ
1	?f1 dc:subject cat:84.	?f2 dc:subject cat:44.
2	?i ma:fragment ?f1.	?f1 dc:subject cat:84.
3	?f2 dc:subject cat:44.	FILTER mm:rightBeside(?f1, ?f2)
4	FILTER mm:rightBeside(?f1, ?f2)	?i ma:fragment ?f1.

Table B.3: Evaluation: Queryplan 1

	SONG	KURZ
1	?f2 dc:subject cat:22.	?f3 dc:subject cat:24.
2	?i ma:fragment ?f2.	?f2 dc:subject cat:22.
3	?f3 dc:subject cat:24.	FILTER mm:rightBeside(?f3, ?f2)
4	FILTER mm:rightBeside(?f3, ?f2)	?i ma:fragment ?f2.
5	?i ma:fragment ?f1.	?i ma:fragment ?f1.
6	FILTER mm:rightBeside(?f3, ?f1)	FILTER mm:rightBeside(?f3, ?f1)
7	FILTER mm:rightBeside(?f1, ?f2)	FILTER mm:rightBeside(?f1, ?f2)
8	?f1 dc:subject ?c.	?f1 dc:subject ?c.

Table B.4: Evaluation: Queryplan 2

	SONG	KURZ
1	?f1 dc:subject cat:84.	?f1 dc:subject cat:84.
2	?i ma:fragment ?f1.	?i ma:fragment ?f1.
3	?f2 dc:subject cat:44.	?f3 dc:subject cat:64.
4	FILTER mm:rightBeside(?f1, ?f2)	FILTER mm:touces(?f1, ?f3)
5	?i ma:fragment ?f2.	?i ma:fragment ?f3.
6	?i ma:fragment ?f3.	?i ma:fragment ?f2.
7	FILTER mm:touces(?f1, ?f3)	FILTER mm:rightBeside(?f1, ?f2)
8	?f3 dc:subject cat:64.	?f2 dc:subject cat:44.

Table B.5: Evaluation: Queryplan 3

	SONG	KURZ
1	?f1 dc:subject cat:18.	?f1 dc:subject cat:18.
2	?f2 dc:subject cat:28.	?f4 dc:subject cat:34.
3	FILTER mm:above(?f2, ?f1)	FILTER mm:covers(?f1, ?f4)
4	?f4 dc:subject cat:34.	?f2 dc:subject cat:28.
5	FILTER mm:covers(?f1, ?f4)	FILTER mm:above(?f2, ?f1)

Table B.6: Evaluation: Queryplan 4

	SONG	KURZ
1	?f1 dc:subject cat:18.	?f1 dc:subject cat:18.
2	?f2 dc:subject cat:28.	?f4 dc:subject cat:34.
3	FILTER mm:above(?f2, ?f1)	FILTER mm:covers(?f1, ?f4)
4	?f4 dc:subject cat:34.	?f2 dc:subject cat:28.
5	FILTER mm:covers(?f1, ?f4)	FILTER mm:above(?f2, ?f1)
6	?f3 dc:subject cat:18.	?f3 dc:subject cat:18.
7	FILTER mm:rightBeside(?f3, ?f1)	FILTER mm:rightBeside(?f3, ?f1)

Table B.7: Evaluation: Queryplan 5

	SONG	KURZ
1	?f1 dc:subject cat:63.	?f6 dc:subject cat:67.
2	?f2 dc:subject cat:62.	?f3 dc:subject cat:51.
3	FILTER mm:rightBeside(?f1, ?f2)	FILTER mm:covers(?f6, ?f3)
4	?i ma:fragment ?f1.	?i ma:fragment ?f3.
5	?f3 dc:subject cat:51.	?f5 dc:subject cat:72.
6	?i ma:fragment ?f3.	?i ma:fragment ?f5.
7	?i ma:fragment ?f5.	?i ma:fragment ?f1.
8	?f5 dc:subject cat:72.	?f1 dc:subject cat:63.
9	?f4 dc:subject cat:67.	?f4 dc:subject cat:67.
10	FILTER mm:above(?f5, ?f4)	FILTER mm:above(?f5, ?f4)
11	?f6 dc:subject cat:67.	?f2 dc:subject cat:62.
12	FILTER mm:covers(?f6, ?f3)	FILTER mm:rightBeside(?f1, ?f2)

Table B.8: Evaluation: Queryplan 6

	SONG	KURZ
1	?f1 dc:subject cat:44.	?f1 dc:subject cat:44.
2	?i ma:fragment ?f1.	?i ma:fragment ?f1.
3	?f3 dc:subject cat:50.	?f3 dc:subject cat:50.
4	FILTER mm:above(?f3, ?f1)	FILTER mm:rightBeside(?f1, ?f3)
5	FILTER mm:rightBeside(?f1, ?f3)	FILTER mm:above(?f3, ?f1)
6	?f2 dc:subject cat:44.	?f2 dc:subject cat:44.
7	FILTER mm:touches(?f1, ?f2)	FILTER mm:touches(?f1, ?f2)

Table B.9: Evaluation: Queryplan 7

	SONG	KURZ
1	?f2 dc:subject cat:44.	?f2 dc:subject cat:44.
2	?f3 dc:subject cat:50.	?f3 dc:subject cat:50.
3	FILTER mm:above(?f3, ?f2)	FILTER mm:above(?f3, ?f2)
4	FILTER mm:rightBeside(?f2, ?f3)	FILTER mm:rightBeside(?f2, ?f3)
5	?i ma:fragment ?f3.	?i ma:fragment ?f3.
6	?f1 dc:subject cat:44.	?f1 dc:subject cat:44.
7	FILTER mm:touches(?f2, ?f1)	FILTER mm:touches(?f2, ?f1)
8	?f4 dc:subject cat:81.	?i ma:fragment ?f4.
9	?i ma:fragment ?f4.	?f4 dc:subject cat:81.
10	?f5 dc:subject cat:47.	?f5 dc:subject cat:47.
11	FILTER mm:covers(?f4, ?f5)	FILTER mm:covers(?f4, ?f5)

Table B.10: Evaluation: Queryplan 8

	SONG	KURZ
1	?f1 dc:subject cat:44.	?f2 dc:subject cat:86.
2	?i ma:fragment ?f1.	?f1 dc:subject cat:44.
3	?f2 dc:subject cat:86.	FILTER mm:touches(?f1, ?f2)
4	FILTER mm:touches(?f1, ?f2)	?i ma:fragment ?f1.
5	?f3 dc:subject cat:32.	?f4 dc:subject cat:82.
6	?i ma:fragment ?f3.	?i ma:fragment ?f3.
7	?f4 dc:subject cat:82.	FILTER mm:rightBeside(?f3, ?f4)
8	FILTER mm:rightBeside(?f3, ?f4)	?f3 dc:subject cat:32.

Table B.11: Evaluation: Queryplan 9

	SONG	KURZ
1	?f1 dc:subject cat:51.	?f3 dc:subject cat:79.
2	?f2 dc:subject cat:81.	?f1 dc:subject cat:51.
3	FILTER mm:above(?f2, ?f1)	FILTER mm:covers(?f3, ?f1)
4	?f3 dc:subject cat:79.	?f2 dc:subject cat:81.
5	FILTER mm:covers(?f3, ?f1)	FILTER mm:above(?f2, ?f3)
6	FILTER mm:above(?f2, ?f3)	FILTER mm:above(?f2, ?f1)

Table B.12: Evaluation: Queryplan 10

Bibliography

- [AB91] Rafiul Ahad and Amit Basu. ESQL: a query language for the relation model supporting image domains. In *Proceedings of the Seventh International Conference on Data Engineering*, pages 550–559, Kobe, Japan, 1991. IEEE. (Cited on page 32.)
- [ABB⁺16] Patrick Aichroth, Johanna Björklund, Emanuel Berndl, Thomas Kurz, and Thomas Köllmer. Enabling Technology Modules: Final Version. Technical report, Media in Context - MICO, December 2016. (Cited on page 172.)
- [ABS⁺15a] Patrick Aichroth, Johanna Björklund, Kai Schlegel, Thomas Kurz, and Thomas Köllmer. Specifications and Models for Cross-Media Extraction, Metadata Publishing, Querying and Recommendations - Final Version. Technical report, Media in Context - MICO, December 2015. (Cited on pages 18, 19, 20, 78 and 171.)
- [ABS⁺15b] Patrick Aichroth, Johanna Björklund, Florian Stegmaier, Thomas Kurz, and Grant Miller. State of the Art in Cross-Media Analysis, Metadata Publishing, Querying and Recommendations. Technical Report Volume 1, Salzburg Research, August 2015. (Cited on pages 16, 30 and 170.)
- [ABS⁺15c] Patrick Aichroth, Johanna Björklund, Henrik Björklund, Kai Schlegel, Thomas Kurz, and Grant Miller. Specifications and Models for Cross-Media Extraction, Metadata Publishing, Querying and Recommendations - Version I. Technical Report ISBN 978-3-902448-44-6, Media in Context - MICO, November 2015. (Cited on pages 71 and 171.)
- [All83a] James F. Allen. Maintaining Knowledge about Temporal Intervals. *Communications of the ACM*, 26(11):832–843, November 1983. (Cited on page 47.)
- [All83b] J.F. Allen. Maintaining Knowledge About Temporal Intervals. *Communications of the ACM*, 26(11):832–843, 1983. (Cited on pages 38 and 91.)
- [ATS96] Hiroshi Arisawa, Takashi Tomii, and Kiril Salev. Design of multimedia database and a query language for video image data. In *Proceedings of the Third IEEE International Conference on Multimedia Computing and Systems*, pages 462–467, Hiroshima, Japan, 1996. IEEE. (Cited on page 32.)

- [ATSH09] Richard Arndt, Raphaël Troncy, Steffen Staab, and Lynda Hardman. *COMM: A Core Ontology for Multimedia Annotation*, volume 2nd Edition, pages 403–422. Springer, 2009. (Cited on page 23.)
- [ATY⁺95] Y. Alp Aslandogan, Chuck Thier, Clement T. Yu, Chengwen Liu, and Krishnakumar R. Nair. Design, Implementation and Evaluation of SCORE (a System for COntent based REtrieval of Pictures). In *Proceedings of the Eleventh International Conference on Data Engineering (ICDE)*, pages 280–287, Taipei, Taiwan, 1995. IEEE Computer Society. (Cited on page 32.)
- [B⁺07] Scott Boag et al. XQuery 1.0: An XML Query Language. Technical report, W3C, 2007. (Cited on pages 33 and 34.)
- [BBFS05] James Bailey, François Bry, Tim Furche, and Sebastian Schaffert. Web and semantic web query languages: A survey. In *Reasoning Web*, volume 3564 of *Lecture Notes in Computer Science*, pages 35–133. Springer, 2005. (Cited on page 15.)
- [BBZ12] Petra Budikova, Michal Batko, and Pavel Zezula. Query language for complex similarity queries. In *East European Conference on Advances in Databases and Information Systems*, pages 85–98. Springer, 2012. (Cited on page 33.)
- [BFG⁺96] Jeffrey R. Bach, Charles Fuller, Amarnath Gupta, Arun Hampapur, Bradley Horowitz, Rich Humphrey, Ramesh C. Jain, and Chiao-Fe Shu. Virage image search engine: An open framework for image management. In *Storage and Retrieval for Still Image and Video Databases IV, San Diego/La Jolla, CA, USA, January 28 - February 2, 1996*, pages 76–87, 1996. (Cited on page 32.)
- [BG04] Dan Brickley and R. V. Guha. RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation, 2004. (Cited on pages 10 and 13.)
- [BH08] Tobias Bürger and Michael Hausenblas. Interlinking Multimedia - Principles and Requirements. *Proceedings of the First International Workshop on Interacting with Multimedia Content on the Social Semantic Web, co-located with SAMT 2008*, December 2008. (Cited on page 23.)
- [BL06] Tim Berners-Lee. Linked Data: Design Issues. <http://www.w3.org/DesignIssues/LinkedData.html>, 2006. (Cited on pages 20 and 29.)
- [BLC11] Tim Berners-Lee and Dan Connolly. Notation3 (N3): A readable RDF syntax. Technical report, W3C, 2011. (Cited on page 72.)

- [BLFM05] T. Berners-Lee, R. Fielding, and R. Masinter. Uniform Resource Identifier (URI): Generic Syntax. RFC 3986, Network Working Group, January 2005. (Cited on pages 10 and 12.)
- [BLHL01] Tim Berners-Lee, James Hendler, and Ora Lassila. The Semantic Web. *Scientific American*, 284(5):34–43, 2001. (Cited on pages 3 and 9.)
- [BLL⁺06] Paolo Bottoni, Stefano Levialdi, Anna Labella, Emanuele Panizzi, Rosa Trinchese, and Laura Gigli. MADCOW: a visual interface for annotating web pages. In *AVI '06: Proceedings of the working conference on Advanced visual interfaces*, pages 314–317, New York, NY, USA, 2006. ACM Press. (Cited on page 27.)
- [BM07] Dan Brickley and Libby Miller. FOAF vocabulary specification. Technical report, FOAF project, May 2007. (Cited on page 14.)
- [BPL⁺14] Werner Bailer, Chris Poppe, WonSuk Lee, Martin Höffernig, and Florian Stegmaier. Metadata API for media resources 1.0. W3C recommendation, W3C, March 2014. <http://www.w3.org/TR/2014/REC-mediaont-api-1.0-20140313/>. (Cited on page 4.)
- [BPS97] Tim Bray, Jean Paoli, and C. M. Sperberg-McQueen. Extensible markup language (XML). *World Wide Web Journal*, 2(4):27–66, 1997. (Cited on page 33.)
- [BRBL⁺18] Amelia Bellamy-Royds, Bogdan Brinza, Chris Lilley, Dirk Schulze, David Storey, and Eric Willigers. Scalable Vector Graphics (SVG) 2. W3C recommendation, W3C, October 2018. (Cited on page 77.)
- [BS09] Tobias Bürger and Elena Paslaru Bontas Simperl. A Conceptual Model for Publishing Multimedia Content on the Semantic Web. In *SAMT*, volume 5887 of *Lecture Notes in Computer Science*, pages 101–113. Springer, 2009. (Cited on page 23.)
- [BS12] Lakshmi Balasubramanian and M Sugumaran. A state-of-art in r-tree variants for spatial indexing. *International Journal of Computer Applications*, 42(20):35–41, 2012. (Cited on pages 107 and 109.)
- [BYRN99] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1st edition, May 1999. (Cited on page 150.)
- [C⁺12] Pierre-Antoine Champin et al. Ontology for media resources 1.0. W3C recommendation, W3C, February 2012. <http://www.w3.org/TR/2012/REC-mediaont-10-20120209/>. (Cited on pages 4 and 21.)
- [CBB⁺00] R. G. Cattell, Douglas K. Barry, Mark Berler, Jeff Eastman, David Jordan, Conn Russell, Olaf Schadow, Torsten Stanienda, and Fernando

- Velez. *The Object Data Standard: ODMG 3.0*. Morgan Kaufmann, the morgan edition, 2000. (Cited on page 37.)
- [CCM⁺97] Shih-Fu Chang, William Chen, Horace J. Meng, Hari Sundaram, and Di Zhong. Videoq: An automated content based video search system using visual cues. In *Proceedings of the Fifth ACM International Conference on Multimedia*, MULTIMEDIA '97, pages 313–324, New York, NY, USA, 1997. ACM. (Cited on page 32.)
- [CF80a] Ning-San Chang and King-Sun Fu. Query-by-Pictorial-Example. *IEEE Transactions on Software Engineering*, 6:519–524, 1980. (Cited on page 31.)
- [CF80b] Ning-San Chang and King Sun Fu. A relational database system for images. In *Pictorial Information Systems*, pages 288–321. Springer, 1980. (Cited on pages 30 and 31.)
- [CFvO93] Eliseo Clementini, Paolino Di Felice, and Peter van Oosterom. 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. (Cited on page 88.)
- [CHCT98] Wesley W. Chu, Chih-Cheng Hsu, Alfonso F. Cardenas, and Ricky K. Taira. Knowledge-Based Image Retrieval with Spatial and Temporal Constructs. *IEEE Transactions on Knowledge and Data Engineering*, 10(6):872–888, 1998. (Cited on page 32.)
- [CHIT98] Wesley W. Chu, Chih-Cheng Hsu, Ion Tim Ieong, and Ricky K. Taira. Content-based image retrieval using metadata and relaxation techniques. In *Multimedia Data Management*, pages 149–190. McGraw-Hill, 1998. (Cited on page 32.)
- [CHL01] Arbee L. P. Chen Chia-Han Lin. Motion event derivation and query language for video databases. In *Proceeding of the SPIE Conference on Storage and Retrieval for Media Databases*, pages 209–214, San Jose, CA, USA, 2001. IS & T. (Cited on page 33.)
- [CIB⁺93] Alfonso F. Cardenas, Ion Tim Ieong, Roger Barker, Ricky K. Taira, and Claudine M. Breant. The Knowledge-Based Object-Oriented PIC-QUERY+ Language. *IEEE Transactions on Knowledge and Data Engineering*, 5:644–657, 1993. (Cited on page 32.)
- [CIT94] Wesley W. Chu, Ion T. Ieong, and Ricky K. Taira. A semantic modeling approach for image retrieval by content. *The VLDB Journal*, 3:445–477, 1994. (Cited on page 32.)
- [CITB92] Wesley W. Chu, Ion Tim Ieong, Ricky K. Taira, and Claudine M. Breant. A Temporal Evolutionary Object-Oriented Data Model and

- Its Query Language for Medical Image Management. In *Proceedings of the 18th International Conference on Very Large Data Bases*, pages 53–64, Vancouver, Canada, 1992. Morgan Kaufmann Publishers Inc. (Cited on page 32.)
- [CLH⁺14] Shih Yeh Chen, Chin Feng Lai, Ren Hung Hwang, Han Chieh Chao, and Yueh Min Huang. A multimedia parallel processing approach on gpu mapreduce framework. In *Ubi-Media Computing and Workshops (UMEDIA), 2014 7th International Conference on*, pages 154–159, July 2014. (Cited on page 30.)
- [CLS00] K. Selçuk Candan, Eric Lemar, and V. S. Subrahmanian. View management in multimedia databases. *VLDB J.*, 9(2):131–153, 2000. (Cited on page 33.)
- [CMY10] Jingwei Cheng, Z.M. Ma, and Li Yan. f-SPARQL: A Flexible Extension of SPARQL. In *Database and Expert Systems Applications*, volume 6261 of *Lecture Notes in Computer Science*, pages 487–494. Springer Berlin Heidelberg, 2010. (Cited on page 20.)
- [Cod72] Edgar F. Codd. Relational Completeness of Data Base Sublanguages. In R. Rustin, editor, *Data Base Systems*, volume 6, pages 65–98. Prentice Hall, Englewood Cliffs, NJ, 1972. (Cited on pages 44 and 45.)
- [CÖO03] Lei Chen, M. Tamer Özsu, and Vincent Oria. Modeling video data for content based queries: Extending the DISIMA image data model. In *9th International Conference on Multi-Media Modeling, MMM 2003, Taiwan, January 7-10, 2003, Proceedings*, pages 169–189, 2003. (Cited on page 63.)
- [CP14] Gavin Carothers and Eric Prud’hommeaux. RDF 1.1 turtle. W3C recommendation, W3C, February 2014. <http://www.w3.org/TR/2014/REC-turtle-20140225/>. (Cited on page 11.)
- [CS11] Francisco M Couto and Mário J Silva. Disjunctive shared information between ontology concepts: application to gene ontology. *Journal of biomedical semantics*, 2(1):5, 2011. (Cited on page 149.)
- [Cyg05] R. Cyganiak. A relational algebra for SPARQL. Technical report, Digital Media Systems Laboratory, HP Laboratories Bristol, 2005. (Cited on page 16.)
- [CYS17] Paolo Ciccarese, Benjamin Young, and Robert Sanderson. Web annotation data model. W3C recommendation, W3C, February 2017. <https://www.w3.org/TR/2017/REC-annotation-model-20170223/>. (Cited on page 4.)

- [D⁺11] Violeta Damjanovic et al. Semantic Enhancement: The Key to Massive and Heterogeneous Data Pools. In *Proceeding of the 20th International IEEE ERK (Electrotechnical and Computer Science) Conference 2011, Portoroz, Slovenia*, September 2011. (Cited on pages 4 and 21.)
- [D⁺14] Van Deursen et al. Experiencing standardized media fragment annotations within html5. *Multimedia Tools and Applications*, 70(2):827–846, 2014. (Cited on page 71.)
- [DC96] John D. N. Dionisio and Alfonso F. Cardenas. MQuery: A visual query language for multimedia, timeline and simulation data. *Journal of Visual Languages and Computing*, 7(4):377–401, 1996. (Cited on pages 34 and 40.)
- [DCM08] DCMI Usage Board. DCMI Metadata Terms. <http://dublincore.org/documents/dcmi-terms/>, January 2008. (Cited on pages 14 and 21.)
- [DDG⁺11] Erik Dahlström, Patrick Dengler, Anthony Grasso, Chris Lilley, Cameron McCormack, Doug Schepers, and Jonathan Watt. Scalable vector graphics (svg) 1.1. *World Wide Web Consortium Recommendation*, 16, 2011. (Cited on page 27.)
- [DHK99] Cyril Declair, Mohand-Said Hacid, and Jacques Kouloumdjian. A database approach for modeling and querying video data. In *Proceedings of the 15th International Conference on Data Engineering, Sydney, Australia, March 23-26, 1999*, pages 6–13, 1999. (Cited on page 32.)
- [DM16] Drashty R Dadhaniya and Ashwin Makwana. Survey paper for different sparql query optimization techniques. *Multi-disciplinary Journal of Scientific Research & Education*, 2(8), 2016. (Cited on page 113.)
- [DNBG⁺16] Tom De Nies, Christian Beecks, Frédéric Godin, Wesley De Neve, Grzegorz Stepień, Dörthe Arndt, Laurens De Vocht, Ruben Verborgh, Thomas Seidl, Erik Mannens, et al. A distance-based approach for semantic dissimilarity in knowledge graphs. In *2016 IEEE Tenth International Conference on Semantic Computing (ICSC)*, pages 254–257. IEEE, 2016. (Cited on page 151.)
- [DS05] M. Duerst and M. Suignard. Internationalized Resource Identifiers (IRIs). RFC 3987, Network Working Group, January 2005. (Cited on page 10.)
- [DSB⁺05] Stefan Decker, Michael Sintek, Andreas Billig, Nicola Henze, Peter Dolog, Wolfgang Nejdl, Andreas Harth, Andreas Leicher, Susanne Busse, José Luis Ambite, Matthew Weathers, Gustaf Neumann, and

- Uwe Zdun. TRIPLE - an RDF Rule Language with Context and Use Cases. In *Rule Languages for Interoperability*, 2005. (Cited on page 15.)
- [DTG⁺08a] M. Döllner, R. Tous, M. Gruhne, K. Yoon, M. Sano, and I.S. Burnett. The MPEG Query Format: Unifying Access to Multimedia Retrieval Systems. *IEEE Multimedia*, 15, 2008. (Cited on pages 34, 42 and 44.)
- [DTG⁺08b] Mario Döllner, Ruben Tous, Matthias Gruhne, Kyoungro Yoon, Masanori Sano, and Ian S Burnett. The MPEG Query Format: On the way to unify the access to Multimedia Retrieval Systems. *IEEE Multimedia*, 15(4):82–95, 2008. (Cited on pages 33 and 34.)
- [EU05] Boris Epshtein and Shimon Ullman. Identifying semantically equivalent object fragments. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 2–9. IEEE, 2005. (Cited on page 151.)
- [FBH17] Sebastián Ferrada, Benjamin Bustos, and Aidan Hogan. Imgpedia: a linked dataset with content-based analysis of wikimedia images. In *International Semantic Web Conference*, pages 84–93. Springer, 2017. (Cited on page 5.)
- [Fer01] Jon Ferraiolo. Scalable vector graphics (SVG) 1.0 specification. W3C recommendation, W3C, September 2001. <http://www.w3.org/TR/2001/REC-SVG-20010904>. (Cited on page 78.)
- [FGM⁺97] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, and T. Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1. RFC 2068, Network Working Group, January 1997. (Cited on page 10.)
- [FKC03] Nastarn Fatemi, Omar Abou Khaled, and Giovanni Coray. An XQuery Adaptation for MPEG-7 Documents Retrieval. In *Proceedings of the XML Conference and Exposition*, Philadelphia, PA, USA, 2003. deepX Ltd. (Cited on page 33.)
- [FLR04] Nastaran Fatemi, Mounia Lalmas, and Thomas Rölleke. How to retrieve multimedia documents described by MPEG-7. In *Proceedings of the 2nd ACM SIGIR Semantic Web and Information Retrieval Workshop*, ACM Press., New York, NY, USA, 2004. (Cited on page 33.)
- [FSK15] Sergio Fernández, Sebastian Schaffert, and Thomas Kurz. MICO - Towards Contextual Media Analysis. In *Proceedings of the 24th international conference on World Wide Web (WWW2015)*, Florence, May 2015. (Cited on page 21.)
- [FTH⁺10] André Fialho, Raphael Troncy, Lynda Hardman, Carsten Saathoff, and Ansgar Scherp. What’s on this evening? Designing User Support for

- Event-based Annotation and Exploration of Media. In *Proceedings of the Workshop on Recognising and Tracking Events on the Web and in Real Life (located at SETN 2010)*, Athens, Greece, May 2010. (Cited on page 23.)
- [FWCT13] Lee Feigenbaum, Gregory Todd Williams, Kendall Grant Clark, and Elias Torres. SPARQL 1.1 Protocol, 2013. (Cited on page 16.)
- [GD92] F. Golshani and N. Dimitrova. *Design and Specification of EVA: a language for multimedia database systems*, pages 356–362. Springer Vienna, Vienna, 1992. (Cited on page 32.)
- [GD98] Forouzan Golshani and Nevenka Dimitrova. A Language for Content-Based Video Retrieval. *Multimedia Tools and Applications*, 6:289–312, 1998. (Cited on page 32.)
- [GF13] Wael H Gomaa and Aly A Fahmy. A survey of text similarity approaches. *International Journal of Computer Applications*, 68(13):13–18, 2013. (Cited on page 147.)
- [GLCS95] Asif Ghias, Jonathan Logan, David Chamberlin, and Brian C. Smith. Query by humming: Musical information retrieval in an audio database. In *Proceedings of the Third ACM International Conference on Multimedia '95, San Francisco, CA, USA, November 5-9, 1995.*, pages 231–236, 1995. (Cited on page 32.)
- [GPP13] Paul Gearon, Alexandre Passant, and Axel Polleres. SPARQL 1.1 Update, 2013. (Cited on page 16.)
- [GS13] Alberto Gil Solla and Rafael G. Sotelo Bovino. *TV-Anytime*. X.media.publishing. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. (Cited on page 34.)
- [GS14] Fabien Gandon and Guus Schreiber. RDF 1.1 XML syntax. W3C recommendation, W3C, February 2014. <http://www.w3.org/TR/2014/REC-rdf-syntax-grammar-20140225/>. (Cited on page 11.)
- [Gut84] Antonin Guttman. *R-trees: a dynamic index structure for spatial searching*, volume 14 of 2. ACM, 1984. (Cited on page 108.)
- [GWJ91] Amarnath Gupta, Terry E. Weymouth, and Ramesh Jain. Semantic Queries with Pictures: The VIMSYS Model. In *Proceedings of the 17th International Conference on Very Large Data Bases*, pages 69–79, Barcelona, Spain, 1991. Morgan Kaufmann Publishers Inc. (Cited on page 32.)

- [Har13] Olaf Hartig. SQUIN: a traversal based query execution system for the web of linked data. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, pages 1081–1084, 2013. (Cited on page 15.)
- [Haw13] Sandro Hawke. SPARQL Query Results XML Format (Second Edition), 2013. (Cited on page 16.)
- [HBEV04] Peter Haase, Jeen Broekstra, Andreas Eberhart, and Raphael Volz. A Comparison of RDF Query Languages. In *Proceedings of the 3rd International Semantic Web Conference (ISWC)*, pages 502–517, Hiroshima, Japan, 2004. Springer, LNCS 3298. (Cited on page 45.)
- [Hen01] Robbert Günter Henrich Andreas. POQL^{MM}: A Query Language for Structured Multimedia Documents. *Proceedings 1st International Workshop on Multimedia Data and Document Engineering (MDDE'01)*, pages 22–229, 2001. (Cited on pages 34 and 39.)
- [Hir92] Yoram Hirshfeld. Safe Queries in Relational Databases with Functions. In *Proceedings of the 5th Workshop on Computer Science Logic*, pages 173–183, Berne, Switzerland, 1992. Springer-Verlag, LNCS 626. (Cited on page 46.)
- [HJK⁺09] Bernhard Haslhofer, Wolfgang Jochum, Ross King, Christian Sadilek, and Karin Schellner. The LEMO annotation framework: weaving multimedia annotations with the web. *Int. J. on Digital Libraries*, 10(1):15–32, 2009. (Cited on pages 27 and 28.)
- [HK96] Na'el Hirzalla and Ahmed Karmouch. A multimedia query specification language. In *Multimedia Database Systems*, pages 160–184. Springer, 1996. (Cited on page 32.)
- [HR01] Andreas Henrich and Günter Robbert. POQL^{MM}: A Query Language for Structured Multimedia Documents. In *Proceedings 1st International Workshop on Multimedia Data and Document Engineering (MDDE'01)*, pages 17–26, Lyon, France, July 2001. (Cited on page 33.)
- [HS91] Andreas Heuer and Marc H. Scholl. Principles of Object-Oriented Query Languages. In *Proceedings of the 4th GI Conference on Database Systems for Office, Engineering, and Scientific Applications (BTW)*, pages 178–197, Kaiserslautern, Germany, 1991. Springer. (Cited on page 44.)
- [HS00] Andreas Heuer and Gunter Saake. *Datenbanken: Konzepte und Sprachen*. mitp, 2000. 704 pages, ISBN: 978-3826606199. (Cited on page 44.)

- [HS13] Steve Harris and Andy Seaborne. SPARQL 1.1 Query Language, 2013. (Cited on pages 4, 15, 16, 18, 59, 65 and 67.)
- [HSS03] Siegfried Handschuh, Steffen Staab, and Rudi Studer. Leveraging metadata creation for the semantic web with cream. In *Proceedings of KI 2003: Advances in Artificial Intelligence: 26th Annual German Conference on AI, KI 2003, Hamburg, Germany*, volume 2821, pages 19–33, Berlin, September 2003. Springer. (Cited on page 27.)
- [HSWW03] Laura Hollink, Guus Schreiber, Jan Wielemaker, and Bob Wielinga. Semantic Annotation of Image Collections. In *proceedings of the KCAP'03 Workshop on Knowledge Capture and Semantic Annotation*, Florida, October 2003. (Cited on page 23.)
- [HTRB09] M. Hausenblas, R. Troncy, Y. Raimond, and T. Bürger. Interlinking Multimedia: How to Apply Linked Data Principles to Multimedia Fragments. In *WWW 2009 Workshop: Linked Data on the Web (LDOW2009)*, Madrid, Spain, 2009. (Cited on page 23.)
- [HuON⁺08] Lynda Hardman, Željko Obrenovic, Frank Nack, Brigitte Kerhervé, and Kurt Piersol. Canonical processes of semantically annotated media production. *Multimedia Systems*, 14(6):327–340, June 2008. (Cited on page 23.)
- [Int05] International Press Telecommunications Council. “IPTC Core” Schema for XMP Version 1.0 Specification document, 2005. (Cited on page 21.)
- [ISO99] ISO/IEC. Information technology – Database languages – SQL Multimedia and Application Packages – Part 3: Spatial. ISO 13249-3:1999, International Organization for Standardization, Geneva, Switzerland, 1999. (Cited on page 35.)
- [ISO00a] ISO/IEC. Information technology – Database languages – SQL multimedia and application packages – Part 1: Framework. ISO 13249-1:2000, International Organization for Standardization, Geneva, Switzerland, 2000. (Cited on page 35.)
- [ISO00b] ISO/IEC. Information technology – Database languages – SQL multimedia and application packages – Part 2: Full-Text. ISO 13249-2:2000, International Organization for Standardization, Geneva, Switzerland, 2000. (Cited on page 35.)
- [ISO01] ISO/IEC. Information technology – Database languages – SQL multimedia and application packages – Part 5: Still Image. ISO 13249-5:2001, International Organization for Standardization, Geneva, Switzerland, 2001. (Cited on pages 37 and 39.)

- [ISO06] ISO/IEC. Information technology – Database languages – SQL multimedia and application packages – Part 6: Data mining. ISO 13249-6:2006, International Organization for Standardization, Geneva, Switzerland, 2006. (Cited on page 37.)
- [ISO11] ISO/IEC. Information technology – Database languages – SQL – Part 11: Information and Definition Schemas (SQL/Schemata). ISO 9075-11:2011, International Organization for Standardization, Geneva, Switzerland, 2011. (Cited on page 36.)
- [JC88] Thomas Joseph and Alfonso F. Cardenas. Picquery: A high level query language for pictorial database management. *IEEE Transactions on Software Engineering*, 14:630–638, 1988. (Cited on page 31.)
- [Jon07] Yosi Mass Jonathan Mamou. A Query Language for Multimedia Content. In *Proceeding of the Multimedia Information Retrieval workshop*, 2007. (Cited on page 34.)
- [JS09] Jing Jin and Pedro Szekely. Querymarvel: A visual query language for temporal patterns using comic strips. In *Proceedings of the 2009 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 207–214. IEEE Computer Society, 2009. (Cited on page 33.)
- [KAC⁺02] Gregory Karvounarakis, Sofia Alexaki, Vassilis Christophides, Dimitris Plexousakis, and Michel Scholl. RQL: a declarative query language for RDF. In *Proc Intl World Wide Web Conf WWW*, pages 592–603, 2002. (Cited on page 15.)
- [KC96] Tony C.T. Kuo and Arbee L.P. Chen. A content-based query language for video databases. In *Proceeding of the Third IEEE International Conference on Multimedia Computing and Systems*, pages 209–214, Hiroshima, Japan, 1996. IEEE Computer Society. (Cited on page 32.)
- [KC04] Graham Klyne and Jeremy J. Carroll. Resource Description Framework (RDF): Concepts and Abstract Syntax. Technical report, W3C, 2004. (Cited on page 10.)
- [KCCC97] Jia-Ling Koh, Arbee L. P. Chen, Paul C. M. Chang, and James C. C. Chen. A Query Language and Interface for Integrated Media and Alphanumeric Database Systems. In *Proceedings of the 8th International Conference on Database and Expert Systems Applications (DEXA)*, pages 508–518, London, UK, 1997. Springer-Verlag. (Cited on page 32.)
- [KGD⁺14] Thomas Kurz, Georg Güntner, Violeta Damjanovic, Sebastian Schaffert, and Manuel Fernandez. Semantic enhancement for media asset management systems. *Multimedia Tools and Applications*, pages 1–27, 2014. 10.1007/s11042-012-1197-7. (Cited on pages 4, 21 and 169.)

- [KK01] José Kahan and Marja-Ritta Koivunen. Annotea: an open RDF infrastructure for shared Web annotations. In *WWW '01: Proceedings of the 10th international conference on World Wide Web*, pages 623–632, New York, NY, USA, 2001. ACM Press. (Cited on pages 26 and 27.)
- [KK16] Thomas Kurz and Harald Kosch. Lifting media fragment uris to the next level. In *LIME/SemDev@ ESWC*, 2016. (Cited on pages 78, 171 and 173.)
- [KM12] Krishna Kulkarni and Jan-Eike Michels. Temporal features in SQL:2011. *ACM SIGMOD Record*, 41(3):34, October 2012. (Cited on page 36.)
- [Kol09] Peter Kolb. Experiments on the difference between semantic similarity and relatedness. In *Proceedings of the 17th Nordic Conference of Computational Linguistics (NODALIDA 2009)*, pages 81–88, 2009. (Cited on page 150.)
- [KSFG12] Thomas Kurz, Sebastian Schaffert, Manuel Fernandez, and Georg Günthner. Adding Wings to Red Bull Media: Search and Display semantically enhanced Video Fragments. In *Proceeding of the 21th World Wide Web Conferenece (WWW2012)*, Demo Track, Lyon, France, 2012. (Cited on page 56.)
- [KSK15] Thomas Kurz, Kai Schlegel, and Harald Kosch. Enabling access to Linked Media with SPARQL-MM. In *Proceedings of the 24nd international conference on World Wide Web (WWW2015) companion (LIME15)*, 2015. (Cited on pages 34, 107 and 174.)
- [KT94] Shu-Chen Kau and J.C.R. Tseng. MQL—a query language for multimedia database. In *Proceedings of the 5th IEEE COMSOC International Workshop on Multimedia Communications*, pages 5/1/1–5/1/6, Kyoto, Japan, 1994. (Cited on page 32.)
- [LC95] Chih-Chin Liu and Arbee LP Chen. The design and implementation of the vega multimedia database system. *Journal of Information Science and Engineering*, 11, 1995. (Cited on page 32.)
- [LC98] Wen-Syan Li and K. Selçuk Candan. SEMCOG: A hybrid object-based image database system and its modeling, language, and query processing. In *Proceedings of the Fourteenth International Conference on Data Engineering, Orlando, Florida, USA, February 23-27, 1998*, pages 284–291, 1998. (Cited on page 32.)
- [LCH01a] Peiya Liu, Amit Chakraborty, and Liang H Hsu. A Logic Approach for MPEG-7 XML Document Queries. In *Proceedings of Extreme Markup Languages®*, 2001. (Cited on pages 34, 39, 40 and 42.)

- [LCH01b] Peiya Lui, Amit Charkraborty, and Liang H. Hsu. A Logic Approach for MPEG-7 XML Document Queries. In *Proceedings of the Extreme Markup Languages*, Montreal, Canada, 2001. (Cited on page 33.)
- [LD97] Thomas K Landauer and Susan T Dumais. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, 104(2):211, 1997. (Cited on page 149.)
- [Lev66] Vladimir I Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710, 1966. (Cited on page 147.)
- [LG93] Thomas D. C. Little and Arif Ghafoor. Interval-based conceptual models for time-dependent multimedia data. *IEEE Trans. Knowl. Data Eng.*, 5(4):551–563, 1993. (Cited on page 32.)
- [LOSO97] John Z. Li, M. Tamer Özsu, Duane Szafron, and Vincent Oria. MOQL: A Multimedia Object Query Language. In *Proceedings of the third International Workshop on Multimedia Information Systems*, pages 19–28, Como Italy, 1997. (Cited on pages 34, 37 and 38.)
- [LWYX10] Chang Liu, Haofen Wang, Yong Yu, and Linhao Xu. Towards efficient sparql query processing on rdf data. *Tsinghua science and technology*, 15(6):613–622, 2010. (Cited on page 113.)
- [Mar04] M. Marchiori. Towards a People's Web: Metalog. *IEEE/WIC/ACM International Conference on Web Intelligence (WI'04)*, 2004. (Cited on page 15.)
- [MB09] Alistair Miles and Sean Bechhofer. SKOS Simple Knowledge Organization System Reference. <http://www.w3.org/TR/2009/REC-skos-reference-20090818/>, August 2009. (Cited on page 13.)
- [ME01a] Jim Melton and Andrew Eisenberg. SQL Multimedia and Application Packages (SQL/MM). *SIGMOD Rec.*, 30(4):97–102, December 2001. (Cited on pages 34 and 35.)
- [ME01b] Jim Melton and Andrew Eisenberg. SQL Multimedia Application packages (SQL/MM). *ACM SIGMOD Record*, 30(4):97–102, December 2001. (Cited on page 33.)
- [MKP02] J.M. Martinez, R. Koenen, and F. Pereira. MPEG-7: The Generic Multimedia Content Description Standard, part 1. *IEEE Multimedia*, 9, 2002. (Cited on pages 33, 34, 39 and 71.)
- [MLFR05] Irina Matveeva, G Levow, Ayman Farahat, and Christian Royer. Generalized latent semantic analysis for term representation. In *Proc. of RANLP*, 2005. (Cited on page 149.)

- [MMSS07] Jonathan Mamou, Yosi Mass, Michal Shmueli-Scheuer, and Benjamin Sznajder. A Query Language for Multimedia Content. In *Proceedings of the 30th Annual International ACM SIGIR Conference*, pages 71–82, Amsterdam, The Nederland, 2007. (Cited on page 33.)
- [MNPT10] Yannis Manolopoulos, Alexandros Nanopoulos, Apostolos N Papadopoulos, and Yannis Theodoridis. *R-trees: Theory and Applications*. Springer Science & Business Media, 2010. (Cited on page 109.)
- [Moa97] R. Moats. URN Syntax. RFC 2141, Network Working Group, May 1997. (Cited on page 10.)
- [MR09] Christopher D. Manning and Prabhakar Raghavan. *An Introduction to Information Retrieval*. Cambridge University Press, 2009. (Cited on page 29.)
- [MS96] Sherry Marcus and V. S. Subrahmanian. Foundations of multimedia database systems. *Journal of the ACM*, 43:474–523, 1996. (Cited on page 32.)
- [MSC⁺13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of Words and Phrases and their Compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013. (Cited on page 150.)
- [MvH04] Deborah L. McGuinness and Frank van Harmelen. OWL Web Ontology Language Overview. W3c recommendation, World Wide Web Consortium, February 2004. (Cited on pages 10 and 13.)
- [NBB⁺12] Lyndon J. B. Nixon, Matthias Bauer, Cristian Bara, Thomas Kurz, and John Pereira. Connectme: Semantic tools for enriching online video with web content. In *I-SEMANTICS (Posters & Demos)*, volume 932 of *CEUR Workshop Proceedings*, pages 55–62. CEUR-WS.org, 2012. (Cited on page 21.)
- [Nix13] Lyndon Nixon. The importance of linked media to the future web. In *Proceedings of the 22nd international conference on World Wide Web companion*, pages 455–456. International World Wide Web Conferences Steering Committee, 2013. (Cited on pages 9, 21 and 23.)
- [NMT14] Lyndon Nixon, Vasileios Mezaris, and Jan Thomsen. Seamlessly interlinking tv and web content to enable linked television. In *ACM Int. Conf. on Interactive Experiences for Television and Online Video (TVX 2014), Adjunct Proceedings, Newcastle Upon Tyne, UK*, 2014. (Cited on page 21.)

- [NPD⁺00] David M. Nichols, Duncan Pemberton, Salah Dalhoumi, Omar Larouk, Claire Belisle, and Michael Twidale. DEBORA: Developing an Interface to Support Collaboration in a Digital Library. In *ECDL*, volume 1923 of *Lecture Notes in Computer Science*, pages 239–248. Springer, 2000. (Cited on page 28.)
- [NRT99] Surya Nepal, M. V. Ramakrishna, and James A. Thom. A Fuzzy Object Query Language (FOQL) for Image Databases. In *Proceedings of the Sixth International Conference on Database Systems for Advanced Applications (DASFAA)*, pages 117–124, Washington, DC, USA, 1999. IEEE Computer Society. (Cited on page 32.)
- [NW70] Saul B Needleman and Christian D Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453, 1970. (Cited on page 147.)
- [NW08] Thomas Neumann and Gerhard Weikum. Rdf-3x: a risc-style engine for rdf. *Proceedings of the VLDB Endowment*, 1(1):647–659, 2008. (Cited on page 113.)
- [NW18] Klinsukon Nimkanjana and Suntorn Witosurapot. A simple approach for enabling sparql-based temporal queries for media fragments. In *Proceedings of the 2018 7th International Conference on Software and Computer Applications*, pages 212–216. ACM, 2018. (Cited on page 5.)
- [OM88] Jack A. Orenstein and Frank A. Manola. PROBE Spatial Data Modeling and Query Processing in an Image Database Application. *IEEE Transactions on Software Engineering*, 14:611–629, 1988. (Cited on page 31.)
- [OOL⁺97] Vincent Oria, M. Tamer Özsu, Ling Liu, Xiaobo Li, John Z. Li, Youping Niu, and Paul J. Iglinski. Modeling Images for Content-Based Queries: The DISIMA Approach. In *Proceedings of the 2nd International Conference of Visual Information Systems*, pages 339–346, San Diego, California, 1997. (Cited on page 60.)
- [OÖX⁺99] Vincent Oria, M. Tamer Özsu, Bing Xu, L. Irene Cheng, and Paul Iglinski. Visualmoql: The DISIMA visual query language. In *IEEE International Conference on Multimedia Computing and Systems, ICMCS 1999, Florence, Italy, June 7-11, 1999. Volume I*, pages 536–542, 1999. (Cited on pages 32, 34 and 40.)
- [Ope99] Open GIS Consortium, Inc. OpenGIS simple features specification for SQL. *OpenGIS Project Document 99*, 49:49–99, 1999. (Cited on page 35.)

- [OT93] Eitetsu Oomoto and Katsumi Tanaka. OVID: Design and Implementation of a Video-Object Database System. *IEEE Transactions on Knowledge and Data Engineering*, 5:629–643, 1993. (Cited on page 32.)
- [PAG09] Jorge Pérez, Marcelo Arenas, and Claudio Gutierrez. Semantics and complexity of SPARQL. *ACM Transactions on Database Systems*, 34(3):1–45, August 2009. (Cited on page 16.)
- [Pas10] Alexandre Passant. dbrec - Music Recommendations Using DBpedia. In *Proceedings of the 9th International Semantic Web Conference (ISWC 2010)*, pages 209–224. Springer, 2010. (Cited on page 149.)
- [PD09] A. Phillips and M. Davis. Tags for Identifying Languages. RFC 4646, Network Working Group, September 2009. (Cited on page 11.)
- [PJS11] Matthew Perry, Prateek Jain, and Amit P. Sheth. SPARQL-ST: Extending SPARQL to Support Spatiotemporal Queries. In *Geospatial Semantics and the Semantic Web*, volume 12 of *Semantic Web and Beyond*, pages 61–86. Springer US, 2011. (Cited on page 20.)
- [Pru04] Eric Prud'hommeaux. Algae RDF Query Language. <https://www.w3.org/2004/05/06-Algae/>, 2004. (Cited on page 15.)
- [PS95] Dimitris Papadias and Timos Sellis. A Pictorial Query-By-Example Language. *Journal of Visual Languages Computing*, 6(1):53–72, 1995. (Cited on page 32.)
- [PS08] Eric Prud'hommeaux and Andy Seaborne. SPARQL Query Language for RDF. W3C Recommendation, 2008. (Cited on page 10.)
- [PW97] Thomas A. Phelps and Robert Wilensky. Multivalent Annotations. In *ECDL*, volume 1324 of *Lecture Notes in Computer Science*, pages 287–303. Springer, 1997. (Cited on page 28.)
- [Rev10] Peter Revesz. Safe Query Languages. In *Introduction to Databases*, pages 555–570. Texts in Computer Science, Springer, 2010. (Cited on page 46.)
- [RFS88] Nick Roussopoulos, Christos Faloutsos, and Timos Sellis. An Efficient Pictorial Database System for PSQL. *IEEE Transactions on Software Engineering*, 14:639–650, 1988. (Cited on page 31.)
- [Rob81] Stephen E Robertson. The methodology of information retrieval experiment. *Information retrieval experiment*, 1:9–31, 1981. (Cited on page 29.)
- [RT12] Giuseppe Rizzo and R Troncy. NERD: A Framework for Unifying Named Entity Recognition and Disambiguation Extraction Tools. *EACL 2012*, pages 73–76, 2012. (Cited on page 30.)

- [S⁺07] Spiros Skiadopoulos et al. A family of directional relation models for extended objects. *Knowledge and Data Engineering, IEEE Transactions on*, 19(8):1116–1130, 2007. (Cited on page 90.)
- [SB02] Peter Stark and Mark Baker. The 'application/xhtml+xml' Media Type. RFC 3236, January 2002. (Cited on page 27.)
- [SBK⁺12] Sebastian Schaffert, Christoph Bauer, Thomas Kurz, Fabian Dorschel, Dietmar Glachs, and Manuel Fernandez. The Linked Media Framework: Integrating and Interlinking Enterprise Media Content and Data. *Proceedings of the 8th International Conference on Semantic Systems - I-SEMANTICS '12*, 2012. (Cited on page 15.)
- [SC15] Fuqi Song and Olivier Corby. Extended query pattern graph and heuristics-based sparql query planning. *Procedia Computer Science*, 60:302–311, 2015. (Cited on pages 113, 114, 115, 116 and 123.)
- [SCdS13] Robert Sanderson, Paolo Ciccarese, and Herbert Van de Sompel. Open Annotation Data Model. Community Draft, Open Annotation Collaboration, February 2013. <http://www.openannotation.org/spec/core/>. (Cited on page 4.)
- [Sch04a] Sebastian Schaffert. *Xcerpt: A Rule-Based Query and Transformation Language for the Web*. PhD thesis, University of Munich, 2004. (Cited on page 15.)
- [Sch04b] Nadine Schulz. *Formulierung von Nutzerpräferenzen in Multimedia-Retrieval-Systemen*. Doctoral thesis, Otto-von-Guericke-University, Magdeburg, 2004. (Cited on page 44.)
- [Sch08] Ingo Schmitt. QQL: A DB & IR Query Language. *VLDB J.*, 17(1):39–56, 2008. (Cited on page 33.)
- [SE09] F. Scharffe and J. Euzenat. Alignments for data interlinking. <http://melinda.inrialpes.fr>, 2009. (Cited on page 23.)
- [Sea04] Andy Seaborne. RDQL - A Query Language for RDF (Member Submission), 2004. (Cited on page 15.)
- [SHG⁺06] Ronald Schroeter, Jane Hunter, Jonathon Guerin, Imran Khan, and Michael Henderson. A Synchronous Multimedia Annotation System for Secure Collaboratories. In *e-Science*, page 41. IEEE Computer Society, 2006. (Cited on page 27.)
- [SK11a] Sandra Schön and Thomas Kurz. *Linked Media Interfaces -Graphical User Interfaces for Search and Annotation*, volume 4 of *Linked Media Lab Reports of the "Salzburg NewMediaLab – The Next Generation"*. Christoph Bauer, Georg Güntner and Sebastian Schaffert, Salzburg, 2011. (Cited on page 173.)

- [SK11b] Sandra Schön and Thomas Kurz. *Smarte Annotationen (German Edition)*. Salzburg Research Forschungsgesellschaft, October 2011. (Cited on page 147.)
- [SLPB] Florian Stegmaier, WonSuk Lee, Chris Poppe, and Werner Bailer. API for Media Resources 1.0. W3C Working Draft. 12 July 2011. <http://www.w3.org/TR/2011/WD-mediaont-api-1.0-20110712/>. (Cited on page 21.)
- [SPM⁺16] Konstantinos Stravoskoufos, Euripides GM Petrakis, Nikolaos Mainas, Sotirios Batsakis, and Vasilis Samoladas. Sowl ql: querying spatio-temporal ontologies in owl. *Journal on Data Semantics*, 5(4):249–269, 2016. (Cited on page 5.)
- [SRL98] Henning Schulzrinne, A. Rao, and R. Lanphier. RFC2326 - Real Time Streaming Protocol (RTSP), April 1998. (Cited on page 72.)
- [SS10] Carsten Saathoff and Ansgar Scherp. Unlocking the Semantics of Multimedia Presentations in the Web with the Multimedia Metadata Ontology. In *Proceedings of the World Wide Web Conference 2010 (WWW2010)*, 2010. (Cited on page 23.)
- [SSB⁺08] Markus Stocker, Andy Seaborne, Abraham Bernstein, Christoph Kiefer, and Dave Reynolds. Sparql basic graph pattern optimization using selectivity estimation. In *Proceedings of the 17th international conference on World Wide Web*, pages 595–604. ACM, 2008. (Cited on page 113.)
- [SSH05a] Ingo Schmitt, Nadine Schulz, and Thomas Herstel. WS-QBE: A QBE-Like Query Language for Complex Multimedia Queries. In *Proceedings of the Eleventh International Multi-Media Modelling Conference (MMM)*, pages 222–229, Melbourne, Australia, 2005. IEEE Computer Society. (Cited on page 33.)
- [SSH05b] Ingo Schmitt, Nadine Schulz, and Thomas Herstel. WS-QBE: A QBE-Like Query Language for Complex Multimedia Queries. *11th International Multimedia Modelling Conference*, 2005. (Cited on page 34.)
- [Ste10] Thomas Steiner. SemWebVid - Making Video A First Class Semantic Web Citizen and a First Class Web Bourgeois. In *Proceedings of the ISWC 2010 Posters and Demonstrations Track*, Shanghai, 2010. Thomas Steiner. (Cited on page 23.)
- [Sto03] Knut Stolze. SQL/MM Spatial: The Standard to Manage Spatial Data in Relational Database Systems. In *Proceedings of the Database Systems for Business, Technology and Web (BTW)*, pages 115–122, Leipzig, Germany, 2003. GI. (Cited on page 35.)

- [SvdS11] R. Sanderson and H. van de Sompel. Open annotation: Beta data model guide. <http://www.openannotation.org/spec/beta/>, 2011. (Cited on page 23.)
- [SWS⁺00] Arnold W. M. Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain. Content-Based Image Retrieval at the End of the Early Years. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22:1349–1380, December 2000. (Cited on page 23.)
- [TDMP12] Raphaël Troncy, Davy Van Deursen, Erik Mannens, and Silvia Pfeiffer. Media Fragments URI 1.0 (basic). W3C recommendation, W3C, September 2012. (Cited on pages 24 and 29.)
- [Tec02] Technical Standardization Committee on AV & IT Storage Systems and Equipment. Exchangeable image file format for digital still cameras: Exif Version 2.2. Technical Report JEITA CP-3451, Japan Electronics and Information Technology Industries Association, April 2002. (Cited on pages 21 and 48.)
- [The14] The Unicode Consortium. Unicode Normalization Forms. Technical Report Version 7.0.0, Unicode Consortium, Mountain View, CA, 2014. (Cited on page 11.)
- [TMPD11] Raphael Troncy, Erik Mannens, Silvia Pfeiffer, and Dany Van Deursen. Media Fragments URI 1.0. Technical report, W3C, 2011. (Cited on page 27.)
- [TS01] Christopher Town and David Sinclair. Ontological Query Language for Content Based Image Retrieval. In *Proceedings of the IEEE Workshop on Content-based Access of Image and Video Libraries (CBAIVL)*, pages 75–80, Kauai, HI , USA, 2001. IEEE Computer Society. (Cited on page 34.)
- [TSF⁺12] Petros Tsialiamanis, Lefteris Sidirourgos, Irimi Fundulaki, Vassilis Christophides, and Peter Boncz. Heuristics-based query optimisation for sparql. In *Proceedings of the 15th International Conference on Extending Database Technology*, pages 324–335. ACM, 2012. (Cited on page 113.)
- [TY84] Hideyuki Tamura and Naokazu Yokoya. Image database systems: A survey. *Pattern recognition*, 17(1):29–43, 1984. (Cited on page 31.)
- [UCI⁺06] Victoria Uren, Philipp Cimiano, Jose Iria, Siegfried Handschuh, Maria Vargas-Vera, Enrico Motta, and Fabio Ciravegna. Semantic annotation for knowledge management: Requirements and a survey of the state of the art. *Journal of Web Semantics*, 4(1):14–28, 2006. (Cited on page 27.)

- [W3C07] W3C. Linking Open Data. <http://tinyurl.com/LinkingOpenDataProject>, August 2007. (Cited on pages 4 and 21.)
- [W3C13] W3C. W3C Data Activity. <http://www.w3.org/2013/data/>, 2013. (Cited on page 29.)
- [WDG94] Ron Weiss, Andrzej Duda, and David K. Gifford. Content-based access to algebraic video. In *Proceedings of the International Conference on Multimedia Computing and Systems, ICMCS 1994, Boston, Massachusetts, USA, May 14-19, 1994*, pages 140–151, 1994. (Cited on page 32.)
- [Wil07] Gregory Todd Williams. Extensible SPARQL functions with embedded Javascript. In *Proceedings of 3rd ESWC Workshop on Scripting for the Semantic Web (SFSW07)*, volume 248 of *CEUR Workshop Proceedings ISSN 1613-0073*, June 2007. (Cited on page 18.)
- [Win90] William E Winkler. String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage. In *Proceedings of the Section on Survey Research*, pages 354–359. ERIC, 1990. (Cited on page 147.)
- [WLC14] David Wood, Markus Lanthaler, and Richard Cyganiak. RDF 1.1 concepts and abstract syntax. W3C recommendation, W3C, February 2014. <http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/>. (Cited on page 10.)
- [WXZ11] Zongda Wu, Guandong Xu, and Yanchun Zhang. GMQL: A graphical multimedia query language. *Knowledge-Based Systems*, 2011. (Cited on page 34.)
- [Zad65] Lotfi A. Zadeh. Fuzzy Sets. *Information and Control*, 8(3):338–353, 1965. (Cited on page 46.)
- [Zla07] Jordan Zlatev. Spatial semantics. *The Oxford handbook of cognitive linguistics*, pages 318–350, 2007. (Cited on page 47.)
- [ZMWZ00] Changqing Zhang, Weiyi Meng, Z Wu, and Zhongfei Zhang. WebSSQL - A Query Language for Multimedia Web Documents. In *IEEE Advances in Digital Libraries 2000 - ADL2000*, page 10. IEEE Computer Society, 2000. (Cited on page 34.)