OSLO AND AKERSHUS
UNIVERSITY COLLEGE
OF APPLIED SCIENCES

Matteo Antoniazzi

Mapping the TORCH Ontology to Schema.org

Master thesis 2015
Master in Library and Information Science
Oslo and Akershus University College of Applied Sciences,
Department of Archivistics, Library and Information Science

Abstract

Schema.org is an ontology supported by the biggest search engines, which allows to add structured semantic markup to the HTML of web pages. This kind of markup is fairly easy to implement and has become popular among web developers. It is also becoming more and more important for the strategies of search engines and some organizations are starting to use it to expose large sets of data. Schema.org can be considered as a component of the Semantic Web, which is an extension of the Web, where ontologies, interoperability, and linked open data are important concepts. Together with established data standards, they allow machines to be able to understand the semantics, i.e. the meaning, of data. The wish to explore ontologies through an experimental project was the main motivations of this thesis. The experiment is a mapping of the TORCH ontology to Schema.org that has been evaluated with a small use case in order to test if the mapping could be applied to the markup of a web page made with Schema.org. The purpose is to provide a useful preparatory study for the TORCH project, currently in progress at Oslo and Akershus University College of Applied Sciences, and involving transformation of metadata into Linked Data and extraction and mapping of cultural heritage data.

Sammendrag

Schema.org er en ontologi som er støttet av de største søkemotorene. Den tillater å legge til semantisk strukturert *markup* i HTML-koden i nettsider. Denne type markup er ganske enkel å implementere og den har blitt populær blant web-butviklere. Den holder på å bli viktigere og viktigere for søkemotorene sine strategier fremover, og noen organisasjoner begynner å bruke den for å tilgjengeliggjøre store datasett. Schema.org er ansett som en komponent til den Semantiske Veven, hvor ontologier, interoperabilitet og åpne data er sentrale konsepter. Sammen med etablerte datastandarder, disse gjør at maskiner kan forstå semantikken, dvs. meningen, bak data. Ønsken om å undersøke ontologier gjennom et eksperimentelt prosjekt var den største motivasjon for denne oppgaven. Eksperimentet består av en mapping av TORCH ontologi til Schema.orgMappingen har blitt evaluert med en liten case for å se om mappingen kunne bli brukt for markup av en nettside med Schema.org. Formålet er å komme med en nyttig forstudie for TORCH-prosjekt. Prosjektet er i gang ved Høgskolen i Oslo og Akershus, og består av trasnformasjon av metadata til linked data og ekstraksjon og mapping av data innenfor kulturarv.

Oslo and Akershus University College of Applied Sciences,
Department of Archivistics, Library and Information Science
Oslo 2015

CONTENTS

LIST OF FIGURES

LIST OF LISTINGS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Data on the World Wide Web has largely been unstructured, produced in human-readable forms, and rising exponentially until a point where it is impossible to quantify, some say virtually infinite. Trying to organize the content of the Internet has revealed itself not feasible by humans, and a need to have more machine-readable data has manifested itself. At the same time keyword searching in a full text web of linked documents has become the context in which search engines, particularly Google, have become very important gatekeepers of information.

We are developing standards and finding ways to convey not only information about form and content, but also semantic information that could be understandable to machines. This constitutes the idea behind the Semantic Web, an extension of the current Web, where machine understand the meaning of data. A key concept of this vision is the ontology, a way to organize information in a knowledge domain through shared sets of classes, properties and inferences, that can be mapped to other ontologies thus establishing what Berners-Lee called a giant global graph of data. Schema.org is an ontology proposed by the biggest search engines as a way of structuring data on web pages directly in the HTML code, so that it is easier for search engines to understand the data. They can also use these data, making it more visible, and enhancing the user experience.

The mapping of an ontology in the cultural heritage domain, the TORCH ontology, to Schema.org will be the object of this thesis.

In order to give the reader the necessary context, an initial presentation of the concepts and the technologies underlying the semantic web and the use of ontologies will be given. Along with this, there will also be a brief presentation of the TORCH project and the TORCH ontology. There will also be a more in-depth discussion about Schema.org, its relation to the Semantic Web, its use, structure and data model.

Another aspect that will be presented is an overview of existing mappings of other ontologies to Schema.org. This part will not go in-depth in the details of these ontologies nor the projects in which they are used, but will focus on their work with the mappings to Schema.org, exposing the methods, the motivations and the challenges encountered.

The chapter 4 will present the proposal for a mapping of the TORCH ontology to Schema.org. An analysis of the mapping along with the challenges and a small evaluation will be also presented.

The main motivation behind the thesis was the desire to make an experimental project. Another motivation was also the wish to better understand ontologies and their use on the semantic web. This resulted in the idea of developing a manual mapping between the TORCH ontology and Schema.org. Another purpose is to constitute a starting point for later work of the TORCH project, if it will be considered useful to map the TORCH ontology to Schema.org.

1.1 Research questions

- How can the TORCH ontology be mapped to Schema.org?
- What are the challenges in the mapping process?
- How can the mapping be used in a real-world use case?

CHAPTER 2

METHODOLOGY

This thesis consists of a preparatory study, that has mainly an experimental character. A combination of *action research* and *case study* may be used to frame the methodology. In addition to this, a more qualitative, in-depth analysis of the two ontologies has been carried out.

Action research "is conducted with the primary intention of solving a specific, immediate and concrete problem in the local setting, [. . .] its major goal is to seek a solution to a given problem. [. . .] Action research does develop solutions that are of practical value to the people or organisations with whom the researcher is working with as it is an interactive process. [. . .] The major strength of the method is in its in-depth and first hand understanding the research gets" (Franklin & Anselimo, 2011, p. 255). The problem that is the object of the research is the mapping of the two ontologies, and the mapping proposal aims to be the solution for the problem.

A case study " is an in-depth investigation of an individual, group, institution or phenomenon within its real life context especially where phenomena and context have a slim difference. [. . .] The main purpose of the case study is to determine factors and relationships among the factors that resulted in the behavior under study. It does use purposive sampling in selecting the cases to involve in the study (2011, p. 255). The objects of the case study are the TORCH project and Schema.org.

The main purpose of this study is to propose a mapping between the TORCH ontology and Schema.org. The mapping has been carried out manually at the schema-level, i.e. between classes and properties of the two ontologies. The TORCH project is still ongoing, and some parts are still work in progress, and this constitutes a motivation for a manual mapping. Working with a small dataset, there weren't enough instances to justify an instance-level mapping approach, i.e. the mapping of instances that populate the classes of an ontology. This motivated also the decision of a manual approach. Later on in the project automatic mapping methods on large-scale datasets may be tried and evaluated.

The first step in carrying out the mapping was to examine the starting ontology, and to lay out a representation of the ontology, in order to have an overview of all classes and properties. This was done with the Protégé ontology editor software. The software allows for a tree-like view of an ontology's hierarchy. Each class and property was then examined to search for candidate Schema.org terms. Schema.org website was used to explore the hierarchical view of the classes and find candidate terms for the mapping. The candidate terms were then analyzed on their Schema.org pages, in order to see definition, use and constraints. Then a spreadsheet was laid out, containing mapping proposals, in order to give a readable and clear view of the whole mapping in one place.

The relationships between the terms were established through the use of some OWL and RDF Schema properties. These properties have also been discussed in more depth in section 3.7, because other projects make use of them to map their ontologies with Schema.org. The mapping, along with problems and challenges encountered, is presented and discussed in chapter 4.

Regarding the literature used in the thesis, there is not so much academic research that directly concerns Schema.org and mappings involving Schema.org. This may be due to the fact that Schema.org is a new initiative, started in 2011. There exists documentation about mapping efforts, although often it consists of blog posts, Wiki pages or discussions on message boards. Moreover, a consistent part of the primary sources in this thesis has been found on the world wide web and is not academic material. These sources are mostly reports, Wiki pages, technical blog posts, conference presentations and websites, usually authored by developers or other technical staff and documenting particular initiatives. The need for these sources comes from the fact that often these are the only forms of documentation about these projects. In the case of Schema.org, most of the more in-depth documentation is online. What makes these sources useful and trustworthy is that they contain first-hand information from the developers, who have a particularly deep and precise knowledge of the subject.

There is some literature about ontology mapping, starting approximately from the beginning of the 2000s. This is a branch that is rooted in computer science, so a lot of research is of high complexity and based on finding and evaluating automatic methods for mappings, which is not the aim of this thesis. Because of this, just some of the most general concepts and definitions will be presented below.

Because there are quite many links to projects, files and other things that are not cited, but just referenced, it was chosen to put the majority of these in footnotes, to avoid having too many URLs in the text.

2.1 Evaluation methodology

An important part of the methodology is the mapping evaluation. The nature of this thesis is that of an introductory study that paves the way for possible future developments of the TORCH project. To establish if this could be a good way to proceed for the project, e.g. in order to obtain visibility on search engines, some form of evaluation of the mapping is needed.

This part is also the most challenging, for various reasons. Firstly, because it is quite experimental, there are not many documented attempts to map an ontology to Schema.org, so it is challenging to find a valid evaluation method to follow. The projects involving Schema.org often do not contain enough documentation to establish if and how they have evaluated their mappings.

The second reason is because there are many different variables that regulate if and how markup will be displayed by search engines in their search results. In fact, only the search engines has the final say on what to display in the search results, however right the mapping is or however meaningful the data fed to the search engine is. Because of this, it is problematic to find measurable evaluation criteria to follow.

Another reason is that the mapping has been developed manually, and a test has been carried out on a small amount of data, so it's difficult to find some numerical values that would allow one to draw some credible conclusions.

However, a few different methods were considered, and some were tried out, as explained in more detail in section 4.1.

CHAPTER 3

BACKGROUND

This chapter contains the theoretical standpoint of the work and the background concepts that are important to better understand the research problem.

3.1 The Semantic Web

The amount of data on the World Wide Web has started rising exponentially almost since its very beginning. Today it's almost impossible to quantify its size and the amount of information on it. Because of this there has been a need to describe, represent and organize this information, in order to be able to search and retrieve what is relevant for our information needs. As Tim Berners-Lee, the man behind the vision of the World Wide Web, states, "traditional knowledge representation systems typically have been centralized, requiring everyone to share exactly the same definition of common concepts such as "parent" or "vehicle". But central control is stifling, and increasing the size and scope of such a system rapidly becomes unmanageable" (2001).

The incredible success of the World Wide Web can first and foremost be attributed to its simplicity. In fact, less and less technical expertise is required to add content to it, and access to it has become easier and cheaper. This is why information on the World Wide Web has largely been presented in human rather than machine-readable format. The consequences of this are that data has become heterogeneous and its size virtually infinite. This requires that, sooner or later, we also need machines to understand and process the data, because the amount of data is too large and humans can no longer process everything by themselves, nor exploit the full potential of what these data can offer.

The most basic instrument we have for making information on the Web machine-readable, is metadata. The simplest definition of metadata is suggested by its name: data about data, or information about information. Using a more elaborate definition, metadata is "structured information that describes, explains, locates, or otherwise makes it easier to retrieve, use, or manage an information resource" (NISO, 2004, p. 1). Metadata is "machine understandable information about web resources or other things" (Berners-Lee, 1997), but the challenge is that machines can quite easily read and exchange textual information and numbers, but they cannot understand the semantics, that is the meaning of the data itself, like humans do.

This is why the field of semantics has become more and more relevant to information retrieval during the last 15 years, since the time Tim Berners-Lee envisioned the Semantic Web (1998). Berners-Lee saw the need for more complex forms of metadata, capable of conveying not only information about form and content, but also semantic information that could be understandable to machines. From this need derives the concept of "a Web of actionable information — information derived from data through a semantic theory for interpreting the symbols. The semantic theory provides an account of "meaning" in which the logical connection of terms establishes interoperability between systems" (Shadbolt, Hall, & Berners-Lee, 2006, p. 96). The Semantic Web is an extension of the World Wide Web. In the Semantic Web "information is given well defined meaning, better enabling computers and people to work in cooperation" (Berners-Lee et al., 2001). We have seen the transformation of the Internet from a Net of machines to a Web of documents, that is slowly becoming a Graph of things, and in the hopes of its supporters it has the potentiality to become a "Giant Global Graph" of connected data (Berners-Lee, 2007). In particular, in the Web of documents, search engines became very important technologies (see section 3.3).

Legg (2007, p. 407) explains: "[t]he Semantic Web vision is to develop technology to facilitate retrieval of information via meanings, not just spellings. For this to be possible [. . .] Semantic Web applications will have to draw on some kind of shared, structured, machine-readable conceptual scheme". This was already in the thoughts of Berners-Lee in the first years after the theorization of the Semantic

Web. “The challenge [. . .] is to provide a language that expresses both data and rules for reasoning about the data and that allows rules from any existing knowledge representation system to be exported onto the Web” (Berners-Lee et al., 2001).

3.1.1 Semantic Web standards

There are some standards that constitute the base of the Semantic Web. One of the most important is RDF (Resource Description Framework). “RDF is a framework for expressing information about resources” (Manola, Miller, & McBride, 2014, § 1). It is developed to be a common framework for expressing information in a machine-readable format, thus allowing exchange of data between machines without loss of data. A key factor in this exchange is that data is made available in a format that expresses RDF data. RDF has in fact different serializations: Turtle, JSON-LD (JavaScript Object Notation for Linked Data; see also section 3.6.2), RDFa (Resource Description Framework in Attributes; see also section 3.6.2), N-Triples and RDF/XML. They are different ways of express the same data. The data are formed in so called *triples*, one of the most important concepts in the Semantic Web, because it constitutes the most basic architecture for the structure of data. Triples are formed by a *subject*, a *predicate* and an *object*, or *object*, *attribute* and *value* (Legg, 2007, p. 419). These three elements are usually expressed by *URIs* (Uniform Resource Identifiers) that univocally identify a resource, or *Literals*, i.e. text strings, dates, numbers. A URI can for example look like
`<http://dbpedia.org/resource/Leonardo_da_Vinci>`,
`<http://www.wikidata.org/entity/Q1731>`, or
`<http://xmlns.com/foaf/0.1/Person>`,
while a predicate can be like: `<http://xmlns.com/foaf/0.1/knows>`.

Through this architecture it is possible to express relationships inside the same dataset, e.g. the following triple means that Leonardo da Vinci is a person:

```
(s) <http://dbpedia.org/resource/Leonardo_da_Vinci>
(p) <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
(o) <http://xmlns.com/foaf/0.1/Person>
```

The following example expresses the fact that a person knows another person:

```
(s) <http://identi.ca/user/45563>
(p) <http://xmlns.com/foaf/0.1/knows>
(o) <http://id.loc.gov/authorities/names/no2011070118>
```

Both examples show how it is possible to combine URIs from different datasets. Among other things one can do, is specifying that some individuals from different datasets refer to the same thing, e.g.

```
(s) <http://dbpedia.org/resource/Leonardo\_da\_Vinci>
(p) <http://www.w3.org/2002/07/owl#sameAs>
(o) <http://viaf.org/viaf/24604287/>
```

This gives machines more information about the resource that's described, helping its semantic understanding of the data (Manola et al., 2014, § 1-3). However, RDF isn't expressive enough to express all the complex relationships data on the Semantic Web may contain (Legg, 2007).

RDFS (RDF Schema) is one of several RDF vocabularies. It is an extension of RDF that enables the definition of the data structure of RDF data. It allows to declare classes, properties, and instances and to add range and domain constraints to properties (Manola et al., 2014, § 4).

Schema.org is also mentioned by the W3C as a RDF vocabulary that enables to add "machine-readable information to Web pages, [...] enabling them to be displayed in an enhanced format on search engines or to be processed automatically by third-party applications" (Manola et al., 2014, § 2). However, "RDFS is still too logically simple to express a great deal of what is said on the Web" (Legg, 2007, p. 431-432).

OWL (Web Ontology Language), recently updated to version 2, "is an ontology language for the Semantic Web with formally defined meaning. OWL 2 ontologies provide classes, properties, individuals, and data values and are stored as Semantic Web documents. OWL 2 ontologies can be used along with information written in RDF, and [...] are primarily exchanged as RDF documents" (W3C OWL Working Group, 2012, § Abstract). OWL goes beyond RDF and RDFS in terms of logical expressivity, resulting in a much more complex logic (Legg, 2007, p. 432-434). OWL has three sublanguages: OWL Lite, OWL DL and OWL Full. OWL DL is apparently the most used by developers, because it seems to be a good compromise between the simpler OWL Lite and the much more expressive and difficult to implement OWL Full (2007, p. 432-434). Legg mentions also how "OWL, although currently the flagship ontology of the W3C group, is not problem-free; a case in point is its complex verbosity" (2007, p. 434).

SPARQL (SPARQL Protocol and RDF Query Language) is the query language used for data conforming to the RDF data model. It is able to retrieve and manipulate RDF data through queries consisting of triple patterns (“SPARQL,” 2015; DuCharme, 2011, p. 1-5). Data providers that offer Linked Data usually provide SPARQL endpoints, which are machine-friendly interfaces towards a knowledge base. Endpoints are web services that accept SPARQL queries, that are usually made in an automated way in the infrastructure of an application. However providers often provide web forms to directly make manual queries and get answers on a web page (DuCharme, 2011, p. 13-16).

An overview of the so called Semantic Web Stack, showing how the components that have just been described, as well as others, add up to the structure of the Semantic Web, is visible in fig. 3.1.

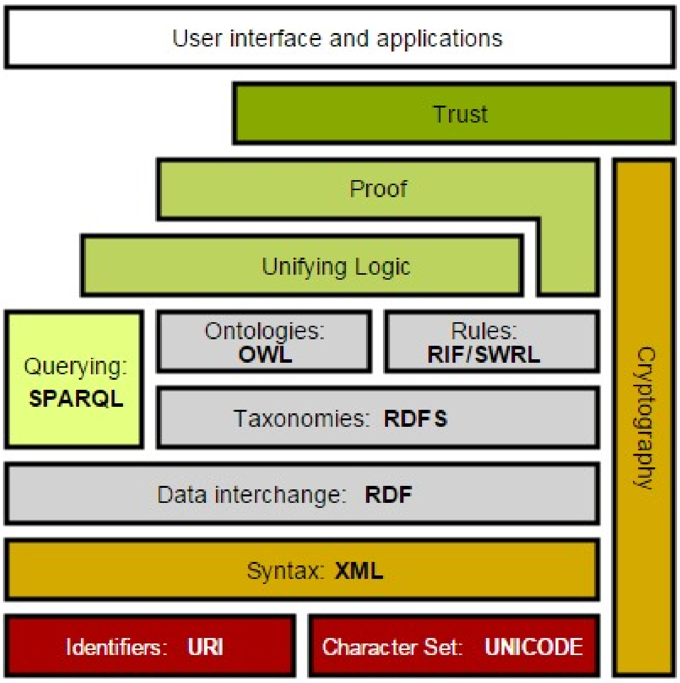


Figure 3.1: The Semantic Web Stack (“Semantic Web,” 2015, § Standards)

These standards and formats provide the tools for the correct functioning of the Semantic Web. Moreover, Berners-Lee (2006) introduced the concept of Linked Data, which is related with the Semantic Web because it constitutes some best practices that, according to its supporters, could make the vision of the Semantic Web to be fully realized. They consist of four principles: use URIs as names for things; use HTTP URIs; provide useful information with these URIs, using RDF and SPARQL; link to other URIs for discovery. They are coupled with a checklist of 5 steps one can follow to be sure one’s data is Linked Data compliant: it should be made available on the web as Open Data; it should be made available as machine-readable structured data; in a non-proprietary format; it should use RDF and SPARQL to identify things; it should link to other people's data.

3.2 Ontologies

One of the central concepts in Semantic Web technology is the ontology. The term ontology is borrowed from the branch of philosophy dealing with the study of which things exist and their categorization. Adapting the concept to the field of Artificial Intelligence, Gruber defined it as "an explicit specification of a conceptualization" (1993, p. 1). This means that one commits to a given conceptualization of a domain, usually made with a set of classes, properties and inferences. This formalizes how one describes and reasons about this conceptualization. An ontology describes a shared vocabulary concerning some specific domain where the assumptions regarding each term in the vocabulary are made explicit. Legg (2007, p. 407) defines an ontology as "a machine-readable theory of the most fundamental concepts or "categories" required in order to understand information pertaining to any knowledge domain".

"An ontology typically provides a vocabulary that describes a domain of interest and a specification of the meaning of terms used in the vocabulary. Depending on the precision of this specification, the notion of ontology encompasses several data/conceptual models, for example, classifications, database schemas, fully axiomatized theories " (Ontology Matching, 2015). The most typical kind of ontology for the Web is composed by a taxonomy and a set of inference rules. The taxonomy defines classes of objects and relations among them (properties). Inference rules apply to properties and classes and give further power to the ontology. E.g., a *gender* property applies only to the class *person.* Classes, subclasses and relations among entities are a very powerful tool for Web use. One can express a large number of relations among entities by assigning properties to classes and allowing subclasses to inherit such properties. E.g., a subclass *teacher* will inherit the property *gender.* "The computer doesn't truly "understand" any of this information, but it can now manipulate the terms much more effectively in ways that are useful and meaningful to the human user" (Berners-Lee et al., 2001). Ontologies have found a role in many different applications. On the web they can for example provide common access to information in a neutral format, facilitate ontology-based semantic search, serve as an exchange language between different applications (Schlenoff, 2009, pp. 180-181).

Based on McGuinness' model for ontology expressivity (see fig. 3.2) Legg (2007, p. 427-438), divides ontologies on the Semantic Web in three groups: thesaurus ontologies (e.g. Dublin Core[1]), ontology with the expressivity of Description Logic (e.g. RDFS, OWL) and those with the expressivity of first-order logic (e.g. Cyc). In fig. 3.2, on the left side of the diagonal line there are the simplest forms of expressivity, while on the right side the most complex, those who represent first-order logic. The line represents properties and relations that can be reasoned over formally, meaning that one can run a reasoning engine only on ontologies that have been enough expressivity to be placed on the right side. Reasoning means that ontologies have the ability to reason over information and find logical errors, or even provide additional information that was not previously formally represented, something that is called inference (Schlenoff, 2009, pp. 180-181).

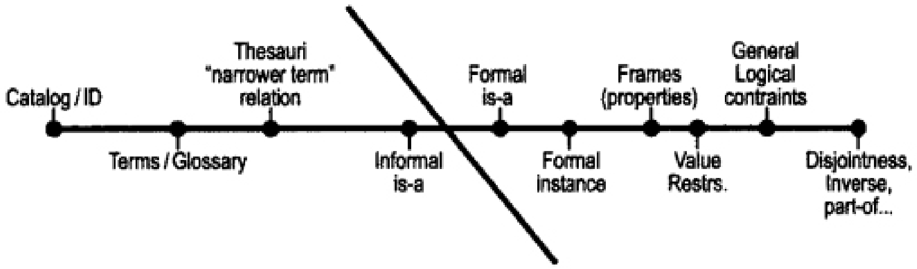


Figure 3.2: McGuinness' model for ontology expressivity (Legg, 2007, p. 428)

Another categorization that is useful to mention is that between three categories of ontologies: upper, mid-level and domain ontologies. Upper ontologies are general ontologies that have the goal of describing all fields of knowledge through abstract concepts. Mid-level ontologies, sometimes called utility ontologies, provide more concrete representations of common, general concepts. Domain ontologies are ontologies whose aim is to describe one particular field of knowledge in detail. Ontologies on a higher level can serve as a starting point to build more detailed domain ontologies (Legg, 2007, p. 439; Semy, Pulvermacher, & Obrst, 2004, p. 2-4). Schema.org can be regarded as a general purpose mid-level ontology (see also section 3.6.1).

After having defined standards and ontologies, it is worth mentioning another concept which underlies the whole infrastructure of the Semantic Web: interoperability. Interoperability is defined as "the ability of multiple systems with different hardware and software platforms, data structures, and interfaces to exchange data with minimal loss of content and functionality" (NISO, 2004, p. 2). Maybe the fact that data structures and interfaces are being standardized with the use of ontologies, RDF and SPARQL can lower the barriers for interoperability and alleviate the chal-

[1] http://dublincore.org/

lenges of data heterogeneity. Connected to this is what Pretorius (2004) says about information sharing and reuse being one of the advantages of using ontologies to capture conceptualizations. Tools that base the organization of their information on the same ontology can easily share their knowledge stores, since they use the same explicit definitions to describe things. Reusing information becomes easier, as one can use existing knowledge stores by committing to the ontology they are described with, or by finding mappings from that ontology to other relevant ontologies one wants to use.

The problem of interoperability can be also viewed from another perspective: "Ontologies tend to be put everywhere. They are viewed as the silver bullet for many applications [...] However, in open or evolving systems, such as the semantic web, different parties would, in general, adopt different ontologies. Thus, just using ontologies, lik-e just using XML, does not reduce heterogeneity: it raises heterogeneity problems at a higher level" (Ontology Matching, 2015).

3.2.1 Challenges to the Semantic Web

The ideas underlying the visions of the Semantic Web and Linked Data are interesting and ambitious, however there seems to be some technical and social challenges to the full development of this framework. Legg (2007, p. 413-415) points out some challenges in the use of ontologies in the Semantic Web. E.g. a challenge is inferential tractability, meaning that it is impossible for systems to make inferences about such a vast and virtually infinite amount of data as the world wide web. Another challenge is the high probability that automatic systems have to make contradictory statements about information on the web. The third challenge is connected to the rapid changeability of information on the web, with the high rate of change and unpredictability. Along with these technical challenges there are also some political challenges.

Also Barbera (2013, §. 92-97) reasons about the issue, and focuses on some technical and social challenges. The technical ones are related to the size and heterogeneity of the datasets and lack of versioning of RDF graphs. The social ones are related to the way we think about data and how we model it, which is restricted by some technical constraints and by our personal view of the world. Another issue is that of the Open World Assumption, that states that a statement cannot be considered false just because there is nothing explicitly stating that it is true. This creates a problem in that many systems are designed to operate, as well as people is used to thinking in closed worlds.

More on the technical side, there is an interesting report made in April 2014, which analyzes the structure and content of the crawlable subset of the Linked Open Data (LOD) cloud (approximately 1000 datasets containing 649 vocabularies describing 8 million resources) and examines to which extent the Linked Data best practices are followed. One of the most interesting results is that 56.11% of the datasets link to at least one other dataset. It is demonstrated that "a relatively small number of datasets that set RDF links pointing at many other datasets, while many datasets only links to a few other datasets" (Schmachtenberg, Bizer, & Paulheim, 2014, p. 15). The remaining datasets are only targets of RDF links, or are isolated. However, the positive note is that "the overall number of Linked Datasets on the Web has grown significantly since 2011" (2014, p. 15).

Shirky (2005) criticizes the use of ontologies as a way of trying to enforce a structure on something that is by nature unstructured. Part of the reason of the critic towards the ontology approach is that every ontology will prove to be inadequate, because it seems improbable that experts can know the needs of all the users *a priori*. Moreover, categorizations can be biased by one's own view of the world, constraints, or the context of use of the information. Another reason is that the Web is "a radical break with previous categorization strategies", so it requires a new way of categorization. For these reasons Shirky supports the idea of common tagging and the so called folksonomies, or social tagging (collections of tags attached to resources by website users) as instruments underlying new forms of categorization and ontologies.

A similar idea has been promoted in several papers. It's the case with Aberer et al. (2004) who expand on semantic interoperability speaking of *emergent semantics* "as an emergent phenomenon constructed incrementally, and its state at any given point in time depends on the frequency, the quality and the efficiency with which negotiations can be conducted to reach agreements on common interpretations within the context of a given task" Aberer et al. (2004, p. 3). They also speak about how global semantic self-organized structures can emerge from local interactions, information and decisions, yet not without challenges. Mika (2005) derives a generic, abstract model of semantic-social networks. Built on the concept of emergent semantics, "a tripartite graph of person, concept and instance associations, extends the traditional concept of ontologies (concepts and instances) with the social dimension" Mika (2005, p. 1). The research of Lin and Davis (2010) is based on the extraction of an ontology from Flickr and Citeulike tags. Wu, Zubair, and Maly (2006) has developed an experimental framework for automatic generation of a hierarchical ontology from a set of tags, using clustering techniques. Laniado, Eynard, and Colombetti (2007) tries to incorporate the Wordnet[2] thesaurus with the Delicious[3] folksonomy in order to build a semantically meaningful hierarchy that can assist users in navigation and resource discovering. Another example is the recent attempt by García-Silva, García-Castro, García, and Corcho (2015), where the authors extract the vocabulary from social tagging systems and reinforce its semantics with data taken from the Linked Open Data cloud.

Papers like these, as well as much other research, fuel the debate between proponents of bottom-up approaches, such as folksonomies and those who are supporters of top-down approaches, such as controlled vocabularies, that have been the traditional way of indexing in libraries. Also Legg (2007, p.433) discusses this and describes the debate as "the divide within the Semantic Web community between application-oriented pragmatism and formal correctness".

3.3 Semantic markup and the role of search engines

As discussed in section 3.1, because data on web pages was largely unstructured, made for human consumption and heterogeneous, there was a great need to find information with some satisfying grade of pecision, so new actors came in the Internet market in the middle of the 1990s: search engines. This enterprises started to organize that data, crawling web pages, indexing the collected information and displaying it to users that were searching for those data through user interfaces ("Web search engine," 2015). Some of these enterprises, such as Google, have become large tech corporations where search isn't the only activity anymore. To date, Google is the most popular search engine, with an overwhelming (nothing short of monopolistic, one could say) global market share of 91,8% in the first quarter of 2015 (Net Market Share, 2015). The reasons of this success are several. Maybe the most important one has been the innovative *PageRank* ranking algorithm, that allowed Google to organize information better than the competitors, the clarity of interface and results, the speed (Legg, 2013, p.137-139). These factors made the company to grow, so that it reached an immense computational capacity to index

[2] https://wordnet.princeton.edu/
[3] https://delicious.com/

massive amounts of data[4]. However, Google has always been good at finding things based on full text search via word spellings, not really via the meanings of search terms. "We've always believed that the perfect search engine should understand exactly what you mean and give you back exactly what you want" (Singhal, 2012), so Google clearly has always had the ambition to understand the semantics of data. The fact that this also means that Google collects and analyzes data about its users' search behavior, its monopolistic position and the fact that there's the danger of biased search results – a so called *filter bubble*[5] – fuels an interesting discussion.

In order to understand the semantics of web pages, there's a need for "some kind of shared, structured, machine-readable *conceptual scheme*, as Legg (2007, p. 407) already noticed before the launch of Schema.org. A way to do this is markup, i.e. embedding extra metadata in the HTML source code, which is easy for search engines to parse. The challenge, related to what discussed in section 3.2.1, is that "[m]ost Web applications sit on top of databases that contain lots of structured data. However, the trip this data takes from the database into HTML is often lossy, as the structured data is converted to HTML for display. Maybe a human can read your field labels on the page to understand your metadata, but that meaning is lost on machines" (Ronallo, 2012, § HTML5). It is clear that the problem has a semantic dimension, combined with an infrastructural challenge: how to keep the structure so that it makes sense to machines?

HTML has always had `<meta>` tags to indicate description, keywords, author, title. However, these are no longer supported, because the keywords tag especially came to be considered harmful because of inappropriate use by spammers (Barker, 2012). Likewise, Dublin Core (DC), an important metadata format used to describe web resources, seems to have been ignored by search engines, according to some researchers. DC, in its simplest version, is very simple and it has very few elements. This produces some weaknesses, in that mapping from any other richer vocabulary to DC will result in a loss of data, which constitutes a challenge for interoperability (Beall, 2004, p. 40-41). The fact that the standard was ignored by search engines is also demonstrated by Kenning Arlitsch and Patrick S. O'Brien (2012), for what concerns the indexing of Institutional Repositories. One of the conclusions of this research is that "Dublin Core schema works "poorly for journal papers" because it does not include adequate fields for citation data and because it is interpreted

[4] It seems that Google's index is "well over 100,000,000 gigabytes", and that the company spent over one million computing hours to build its infrastructure(Google, n.d.)

[5] a concept introduced by Eli Pariser in the 2011 book *The filter bubble: What the Internet is hiding from you*

inconsistently" (2012, p. 72). DC is discussed further in section 3.8.

Jeffrey Beall (2014, p. 12) comes to the harsh conclusion that "[i]n the end, here's what really killed Dublin Core: Schema.org [. . .] The launch essentially made DC obsolete, at least in the context of the Web environment. It was a rejection of the Dublin Core metadata standard by the industry the standard's proponents most wanted to impress, a fatal blow if there ever was one".

Schema.org comes in the picture as a shared vocabulary that allows the semantic structuring of HTML. This kind of markup, a kind of structured semantic data, seems to be picking up on the Web and may have some advantages for search and retrieval, information presentation, information integration, personalization, reusability and interoperability (Khalili & Auer, 2013, p. 1-2). On the other hand, being Google a commercial enterprise, someone speculates about what advantages the free of charge data markup will bring to the company, which basically would be more advertising revenues (Tennison, 2011b). Of course Google's approach is a crowdsourcing approach: content creators are solicited to structure their own data with a commonly recognized format and vocabulary, supported by search engines. On the other hand, search engines have the advantage of dispose of large amounts of meaningful, structured data for free. The potential payoff for using /s/ for the content creators is that it will "help you to surface your content more clearly or more prominently in search results. Not every type of information in schema.org will be surfaced in search results [. . .] but over time you can expect that more data will be used in more ways" (Schema.org, n.d.-a). This may contribute to enhance the user experience with rich snippets as well, which may produce more visits to websites that use structured data. What Schema.org is not supposed to do directly is enhancing the ranking of web pages (Google Webmasters, 2012).

In 2012, Bing and Google made a big step into the Semantic Web, as they launched respectively Satori and the Knowledge Graph (KG), "an intelligent model [. . .] that understands real-world entities and their relationships to one another: things, not strings" (Singhal, 2012). Regarding the KG, the graph is partially powered by linked data from the collaborative ontology Freebase[6], acquired by Google in 2010, Wikipedia, Schema.org data and other sources. One of the most important source is search data itself.

The graph consists of hundreds of millions of entities. These entities are not documents on the web, but rather constructed information about real world objects and

[6] https://www.freebase.com/

concepts, and its data model is no longer an entity database, but a graph. “This requires search engines to redesign many of the previously used algorithms for entity databases. New relevancy detection and entity ranking mechanisms are needed. Traditionally used ranking algorithms such as PageRank need to be reformulated (Ahmet Uyar & Farouk Musa Aliyu, 2015, p. 197-198). So it is clear that the task for search engines is huge.

The KG enhances Google search results in three main ways: by disambiguation of search queries, by displaying key facts, and by explorative search suggestions (Singhal, 2012; “Knowledge Graph,” 2015). Currently, information from the KG is displayed on a panel on the right side of the search results (see also section 3.6.3). Barbera (2013, p. 98) defines it as “a sort of Closed Enterprise linked data cloud”, as “search engines do not want to make their entity types and relationships public” (Ahmet Uyar & Farouk Musa Aliyu, 2015, p. 198), though there are plans for a point of access to it in the form of a new API that will replace the Freebase API. The new API will return name, subtitle, image, description and URL of matches, and will use Schema.org as vocabulary with a JSON-LD syntax (Freebase, 2015). This is very recent information and may indicate that Schema.org is getting more and more important in Google’s “semantic” strategy.

So the possibility of a central part of Schema.org in the KG may contribute to bring it to a central role inside the Semantic Web. This seems to reinforce the statement that the KG “has a “things not strings” approach which resonates very well with the schema.org ontology. So perhaps it is here that we will see the semantic web approach and schema.org begin to bring benefits to the majority of web users” (Barker, 2012).

The potentiality of Schema.org in the context of the Semantic Web lies in the fact that most of the technologies are already well established and can function together. E.g. Frederick Giasson (2014) demonstrates how to set up a framework that is able to query an ontology and utilize the mappings from this ontology to Schema.org to dynamically generate the mark up in the HTML code.

3.4 Ontology Mapping

Ontology mapping, matching or alignment “finds correspondences between semantically related entities of the ontologies. These correspondences can be used for various tasks, such as ontology merging, query answering, data translation, or for navigation on the semantic web. Thus, matching ontologies enables the knowledge and data expressed in the matched ontologies to interoperate" (Shvaiko & Euzenat, 2013).

Mappings may be done either by hand or automatically. Manual mapping should be done on smaller, simpler ontologies, because it becomes impractical for bigger and more complex ontologies (Rahm, 2011, p. 3). "Fully or semi-automated mapping approaches have been examined by several research studies, e.g., analyzing linguistic information of elements in ontologies, treating ontologies as structural graphs, applying heuristic rules to look for specific mapping patterns, and machine learning techniques (Mao, 2007, p. 931).

Most approaches focus on 2-way or pairwise schema matching where two related ontologies are matched with each other. "The result of pairwise schema matching is usually an equivalence mapping containing the identified semantic correspondences, i.e., pairs of semantically equivalent schema elements" (Rahm, 2011, p. 4). Usually one can only determine approximate mappings with algorithms, after which a human has to go through the correspondences to adjust them so that a correct mapping can be achieved (2011, p. 4).

There are two main approaches: metadata-based and instance-based. Metadata-based is the most common and considers "characteristics of schema or ontology elements such as their names, comments, data types, as well as structural properties". The instance-based approach determines "the similarity between schema elements from the similarity of their instances", and they are used primarily for matching large ontologies (2011, p. 6).

Suchanek, Abiteboul, and Senellart (2011, p. 4-6) call this two approaches schema and instance level, and propose a method to combine the two approaches, due to the fact that ontology landscape has changed and there are many very large and complex ontologies, and one must be able to make this to approaches work together in a complementary way.

3.5 The TORCH project and the TORCH ontology

TORCH is an ongoing project by the *information systems based on metadata* research group at Oslo and Akershus University College. The project focuses on three different case studies where library and broadcasting data from two different datasets is being annotated, extracted, mapped, disambiguated and interlinked. The first case study deals with the mapping of MARC records from the Norwegian National Bibliography to other relevant ontologies in the library community, such as BIBFRAME[7], FRBRoo[8] and Schema.org. The second case study deals with the automated extraction of metadata from semi-structured textual data coming from the archives of the Norwegian Broadcasting Corporation (NRK). The third activity includes the “interlinking of shared entities across the two domains” (Tallerås, Massey, Husevåg, Preminger, & Pharo, 2014, p. 250-252).

The aim of the second case study is extracting entities, such as people, places, events, and the relationships between each other, from the textual data. This will be done first in a smaller representative dataset, through manual annotations. The entities will be given relationships with each other and they will be disambiguated by the annotators. Then an algorithm that will automatically extract entities and relationships will be developed. Currently the project is in the annotation phase and the algorithm is under development (2014, p. 253).

The data consist of so-called SIFT-records[9], free text fields containing descriptions of TV and radio programs broadcasted by NRK since the early 1990s. In order to structure and describe the extracted entities and the relationships between them, it was decided to create an ontology (TORCH ontology), which was partly inspired by the BBC Programmes ontology[10], EBU-Core[11] and Schema.org itself. Its functions are to be the target of the automated extraction and to support the annotation (2014, p. 253).

The TORCH ontology is a domain ontology that contains classes and relationships in the broadcasting domain. “The ontology builds on hierarchies of classes and properties, realized through the RDF Schema properties `rdfs:subPropertyOf` and `rdfs:subClassOf` (e.g. `FictionalCharacter rdfs:subClassOf Person rdfs:subClassOf Agent`)” (2014, p. 257-258).

There exist several versions of the ontology, a “full” version and an “annotation” version. “[T]he exposure to test-annotators at an early stage further encouraged the development of a project-specific ontology. The ontology is designed for efficient

[7] http://bibframe.org/

[8] http://www.cidoc-crm.org/frbr_inro.html

[9] Searching In FreeText, see also http://dl.acm.org/citation.cfm?id=810936

[10] http://www.bbc.co.uk/ontologies/po

[11] http://www.ebu.ch/metadata/ontologies/ebucore/

and consistent annotation by reducing complexity and the intuitive naming of classes and properties" (2014, p. 257). This version was used as source for the mapping in this thesis.

In figs. 3.3 and 3.4 it is possible to see the hierarchical structure of the ontology, visualized with the Protégé ontology editor software.

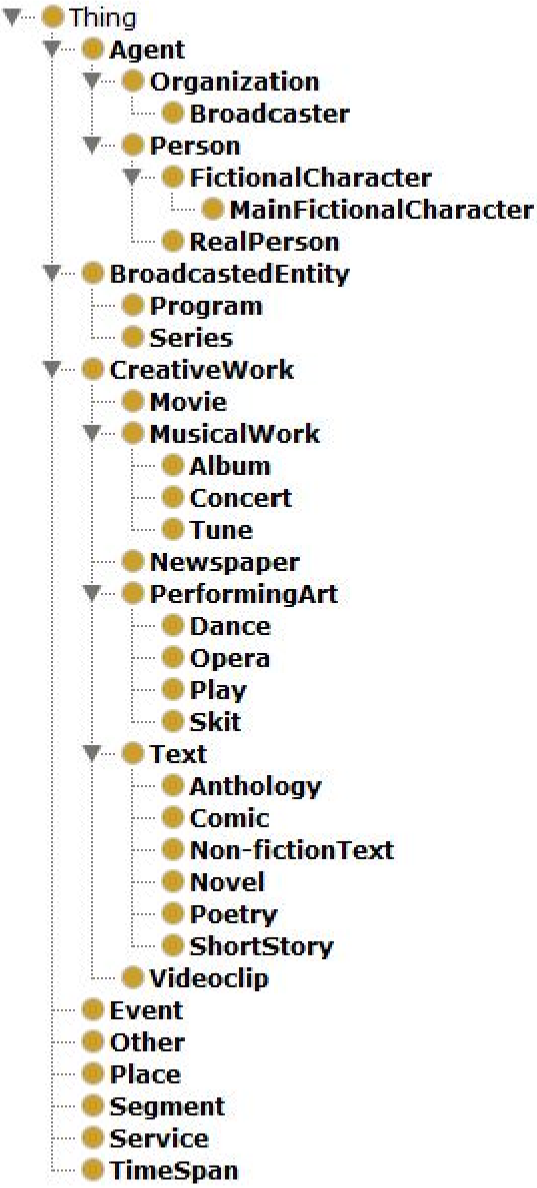
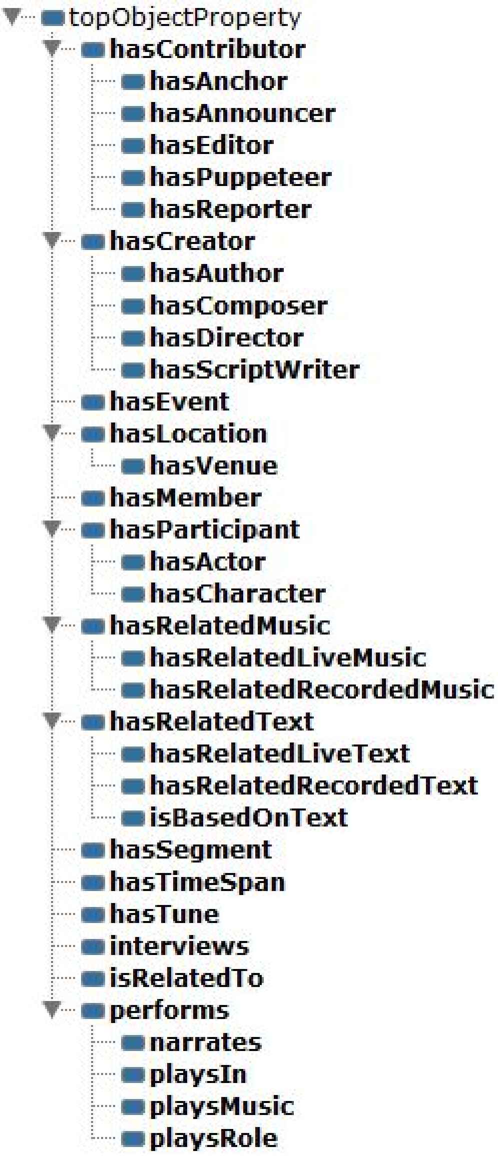


Figure 3.3: The classes of the TORCH ontology

An example of the manual annotation's results is showed in fig. 3.5.

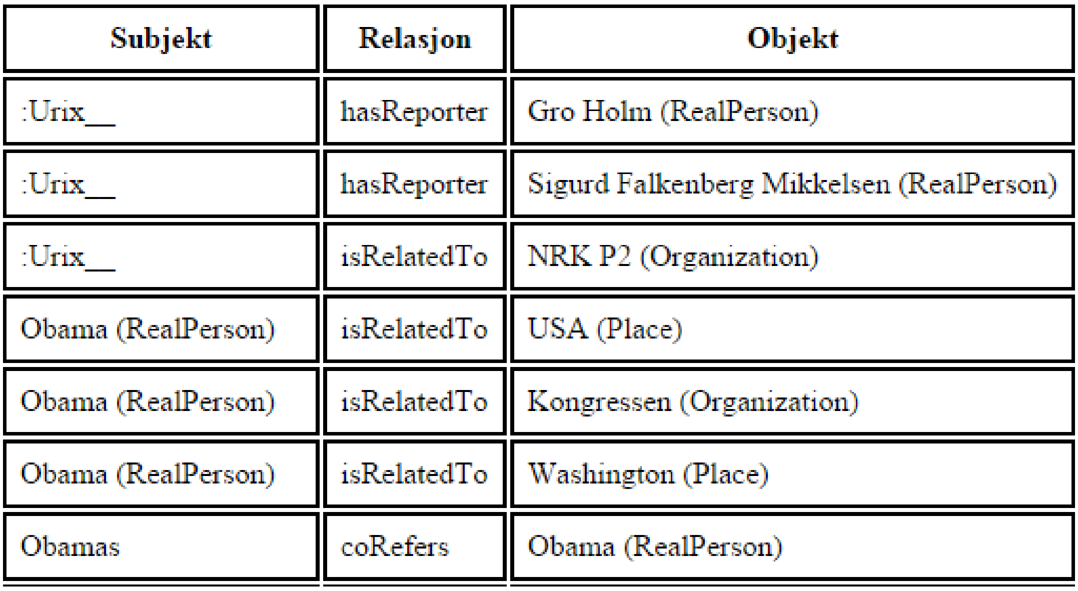
By looking at the figure, one can see that several kinds of relations exist between entities. On the one hand, there are the relations defined by the triple Subject-Relation-Object, such as "Obama" `isRelatedTo` "USA". On the other hand one



- topObjectProperty
  - hasContributor
    - hasAnchor
    - hasAnnouncer
    - hasEditor
    - hasPuppeteer
    - hasReporter
  - hasCreator
    - hasAuthor
    - hasComposer
    - hasDirector
    - hasScriptWriter
  - hasEvent
  - hasLocation
    - hasVenue
  - hasMember
  - hasParticipant
    - hasActor
    - hasCharacter
  - hasRelatedMusic
    - hasRelatedLiveMusic
    - hasRelatedRecordedMusic
  - hasRelatedText
    - hasRelatedLiveText
    - hasRelatedRecordedText
    - isBasedOnText
  - hasSegment
  - hasTimeSpan
  - hasTune
  - interviews
  - isRelatedTo
  - performs
    - narrates
    - playsIn
    - playsMusic
    - playsRole

Figure 3.4: The properties of the TORCH ontology

can see other relations, i.e. “Gro Holm” is an instance of the class `RealPerson`, and “Washington” is an instance of the class `Place`, etc. This kind of properties are defined by the `rdf:type` property (Manola et al., 2014, § 3.3). Moreover, one can also see that there is a linking of different mentions of the same entity through the special property `coRefers` used to disambiguate two instances. E.g. “Obamas” (genitive form of the name) `coRefers` “Obama”. The property `isRelatedTo` is present in the ontology, and it is problematic to express with Schema.org. `coRefers` is not present in the ontology examined, it is a special property used for the purposes of the annotation.



Subjekt	Relasjon	Objekt
:Urix__	hasReporter	Gro Holm (RealPerson)
:Urix__	hasReporter	Sigurd Falkenberg Mikkelsen (RealPerson)
:Urix__	isRelatedTo	NRK P2 (Organization)
Obama (RealPerson)	isRelatedTo	USA (Place)
Obama (RealPerson)	isRelatedTo	Kongressen (Organization)
Obama (RealPerson)	isRelatedTo	Washington (Place)
Obamas	coRefers	Obama (RealPerson)

Figure 3.5: Example of the manual annotation in the TORCH project

3.6 Schema.org

Schema.org is a joint initiative launched in 2011 by the three largest search engines, Google, Yahoo! and Bing, (and later joined by the russian search engine Yandex) with the aim of helping search engines interpret information on web pages in order to create richer search results. This has some more advantages for different stakeholders: “[w]ith schema.org, webmasters have a single place to go to learn about markup for a wide selection of item types, search engines get structured information that helps improve search result quality, and users end up with better search results and a better experience on the web” (Schema.org, n.d.-a).

The idea behind Schema.org is that content creators can enrich their HTML code with extra Schema.org metadata, making human readable information double up as machine readable information, or what Google calls *structured data*. Search engines can also use this metadata to provide information for rich snippets (Guha, 2011). A rich snippet is the representation of small samples of marked-up information in search results. Rich snippets can enrich a search result, showing a picture, review ratings, part of the multimedia content or some other content out of the page (Goel, Gupta, & Hansson, 2009a; Goel, Gupta, & Hansson, 2009b).

To date, Google and Bing are the only two search engines explicitly using rich snippets directly in their result lists. Yahoo! and Yandex don’t appear to enrich results in result lists, while all, except Yandex, have some sort of display of information from their Knowledge Graph, displaying information about different items special result panels (see an example from Google's Knowledge Graph in fig. 3.11).

Google is supporting a few types of items in rich snippets: `Product`, `Recipe`, `Review`, `Event` `SoftwareApplication`, `Article`, `Video`, `Person` and `Organization` (Google, 2015a; Smarty, 2013). Bing is supporting Breadcrumbs, Businesses and organizations, Events, People, Products and offers, Recipes and Reviews, also thanks to Schema.org. Although Schema.org is not a central part of the supported metadata in Bing, it is understood by the search engine, which displays its search results very similarly to Google, i.e. displaying rich snippets (Bing, n.d.).

However, although not all the search engines support rich snippets or use Schema.org for visual enrichment of search results, they support it as a shared vocabulary that is useful in giving semantic meaning to the data they index. Another thing to point out is that what the search engines are supporting and how they are using the content of markup is changing quite rapidly.

3.6.0.1 Employment of Schema.org

There are also some research that shows that Schema.org and in particular the RDFa and Microdata syntaxes, along with Microformats, "have all three found considerable adoption on the Web". The research is based on a Web crawl from 2012 and consisting of 3 billion HTML pages taken from about 40 million websites (Bizer et al., 2013, p. 17). Almost 50% of the top 10,000 websites use structured data, and overall, the most used format is Microformats, followed by RDFa and Microdata (2013, p. 19-20). However, the richness of the data is very limited, because a quite small set of properties are recurring the most, so many pages are described with a few properties and types. The use of RDFa is mostly related to Facebook's Open Graph Protocol (see section 3.8), while Microdata is the most used syntax with Schema.org (2013, p. 31).

Google said that in 2013, 2 years after the launch of Schema.org 15% of the crawled pages and over 5 million web sites used the markup. The fact that also important publishing platforms like Drupal and WordPress are adopting it will make it much easier for content publishers to integrate it in their workflow, and this probably will contribute to a quick growth of Schema.org (Zaino, 2013).

A much smaller research conducted in March 2014 for Google, that analyzed search results for tens of thousands of keywords and about half a million domains, shows some interesting results. Only 0.3% of the analyzed domains included Schema.org markup. However, Google enhances search results with Schema.org markups in more than 36% of keyword queries. Besides, pages with Schema.org integrations rank better by an average of four positions compared to pages without Schema.org integrations (Searchmetrics, 2014, p. 1-8).

3.6.1 The Schema.org data model

Schema.org's current version is 2.0 and it has been published on 15 May 2015. The data model is available as a summary that describes all the terms – types and properties[12] in one page. It is also available as a hierarchy of all types, shown on one page[13]. Every type and property has its own page with more details (see section 3.6.1). Moreover, the whole vocabulary is also available in RDFa/RDFS[14] and N-Triples[15] formats, while the types are available in JSON[16].

The developers of Schema.org say that "the data model used is very generic and derived from RDF Schema (Schema.org, 2014). Schema.org draws also "from the decades of work in the database and knowledge representation communities, from projects such as Jim Gray's SDSS Skyserver[17], Cyc[18] and from ongoing efforts such as dbpedia.org and linked data" (Guha, 2011)[19]. Microformats[20], FOAF[21], GoodRelations[22], OpenCyc[23], rNews[24] are also mentioned as inspiration for Schema.org (Khalili & Auer, 2013, p. 428; Schema.org, n.d.-a). "In creating schema.org, one of our goals was to create a single place where webmasters could go to figure out how to mark up their content, with reasonable syntax and style consistency across types. This way, webmasters only need to learn one thing rather than having to understand different, often overlapping vocabularies. [. . .] Many terms in schema.org came through collaborations, and we acknowledge these on the schema.org site rather

[12] available at https://schema.org/version/2.0/

[13] available at https://schema.org/docs/full.html

[14] https://schema.org/version/2.0/schema.rdfa

[15] https://schema.org/version/2.0/schema.nt

[16] https://schema.org/docs/tree.jsonld

[17] http://skyserver.sdss.org/dr1/en/skyserver/paper/

[18] http://www.cycfoundation.org/

[19] some of the source of Schema.org can be found here: http://www.w3.org/wiki/WebSchemas/SchemaDotOrgSources#Automotive_Ontology_Working_Group

[20] http://microformats.org/

[21] http://www.foaf-project.org/

[22] http://www.heppnetz.de/projects/goodrelations/

[23] http://sw.opencyc.org/

[24] http://dev.iptc.org/rNews-1-Introduction-to-rNews

than by making our markup more complex" (Schema.org, n.d.-a).

The need for simplicity motivates also the fact that Schema.org aims to be what can be described as a general purpose mid-level ontology, i.e. an ontology that neither tries to be too abstract or all-encompassing nor describe a knowledge field in depth. Isaac and Baker (2015, p. 35) defines it as a broadly scoped but semantically lightweight vocabulary. As Ronallo (2012) states, "[i]t is unrealistic for them to try to support every vocabulary in use. Schema.org is an attempt to define a broad, Webscale, shared vocabulary focusing on popular concepts. [...] A central goal of having such a broad schema all in one place is to simplify things for mass adoption and cover the most common use cases." Schema.org developers remark this also by stating that "[t]he type hierarchy presented on this site is not intended to be a 'global ontology' of the world. It only covers the types of entities for which we [...] think we can provide some special treatment for, through our search engines, in the near future" (Schema.org, 2014).

Understandably, the vocabulary does seem to have an inclination towards search engine and commercial use cases, both in the kinds of concepts that are selected and in the hierarchical structure (Ronallo, 2012). This may be already quite clear just by looking at the top types of the hierarchy in fig. 3.6.

3.6.1.1 The Schema.org ontology

The Schema.org data model has two main components:

- an ontology. It is the vocabulary describing:
  - *types* – i.e. classes – the things that can be described. Types start with a capital letter and are CamelCase;
  - *properties* – attributes and relationships between types. Properties start with a lower case and are CamelCase;
  - constraints – how properties and types should be used;
- a syntax to express the information defined by the ontology. In Schema.org's case three different syntaxes are permitted, as discussed in section 3.6.2.

Items, i.e. resources being described, are classified as specific types, in order to identify what kind of things items are. Each type has a set of properties that define characteristics that may be used to describe items. Properties are indicated on every type's page, and can be unique for a type or inherited from broader types that are higher up in the hierarchy. As an example, one can look at https://schema.org/MedicalScholarlyArticle. This type has only one own property `publicationType`, and it inherits most of its other properties from its broader types `Article`, `CreativeWork` and `Thing`.

As shown in the hierarchical list of types[25], there are two separate hierarchies: one defining the datatypes, where the most generic type is `DataType`, and one defining the things that can be described, where the most generic type is `Thing`. Enumerations are special kinds of types. Enumerations can only take limited sets of values, e. g. `BookFormatType`, which may have the `EBook`, `Hardcover` or `Paperback` types as values. The concept is very similar to the one of enumeration in XML Schema[26]. Enumerations where lists of values are large, such as countries, can be external, and therefore can be referred to through external URIs, e.g. from DBpedia (Brickley, 2012).

The top types in the ontology, i.e. the subtypes of `Thing`, are shown in fig. 3.6.

Properties and constraints for every type are are listed on each type's page. On the single pages there's a indication of how many domains make use of the type. There's also the possibility to monitor the open issues related to the type on the Schema.org Github pages[27].

With the exclusion of the datatypes, every type is a subtype of `Thing` and inherits its properties: `additionalType`, `alternateName`, `description`, `image`, `mainEntityOfPage`, `name`, `potentialAction`, `sameAs` and `url`.

An overview of the use of most properties is provided on the type's page through an overview in form of a table (see example in fig. 3.7). Each property name has the same meaning when found in any type in the vocabulary (Ronallo, 2012, § What is Schema.org?).

As showed in fig. 3.7, properties may expect other types as values. E.g. the property `illustrator` expects another type, `Person`, while the property `isbn` expects the type `Text`, which is indicating a text string datatype. So the correct way to express these two properties would be as showed in fig. 3.8. In particular, the book's illustrator is a person whose name is Carl Beckston. So, following Schema.org's indications

[25] see https://schema.org/docs/full.html

[26] see http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/datatypes.html#dt-enumeration

[27] https://github.com/schemaorg/schemaorg

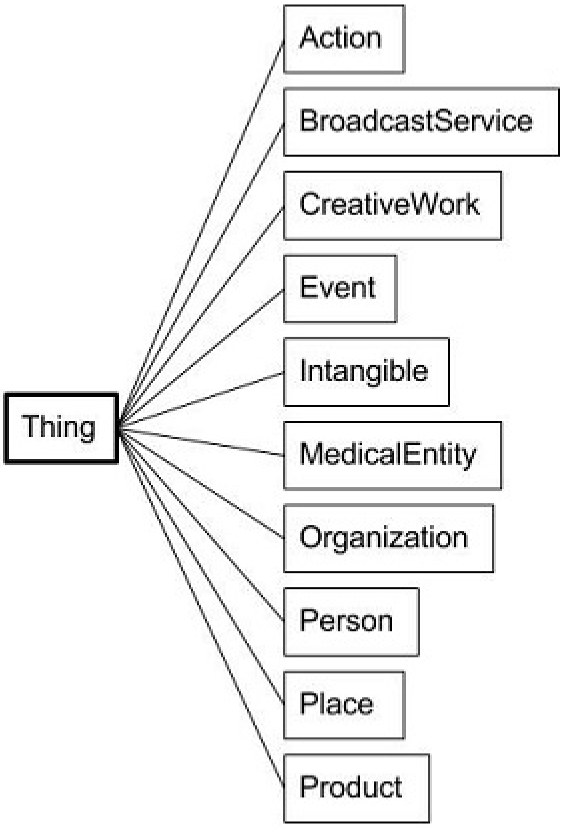
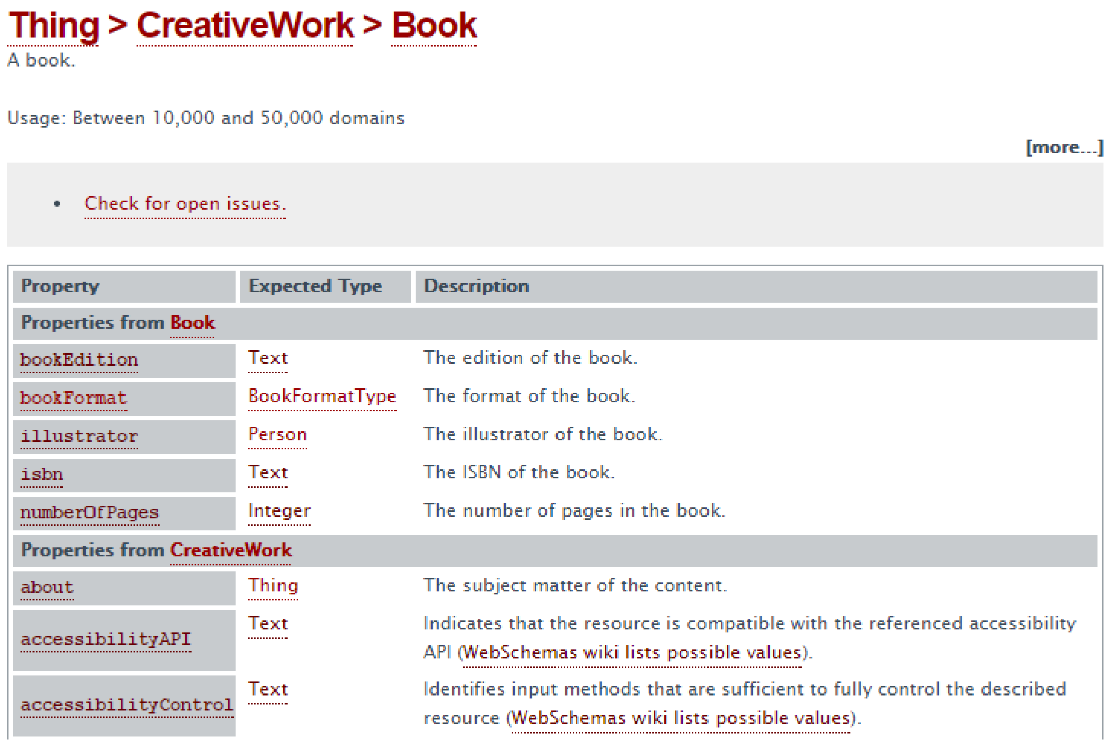


Figure 3.6: The top types in the Schema.org ontology

strictly, a `Person` type should be embedded in the `Book` type, and then the name could be indicated in a text string. However the developers of Schema.org expect a lot of non-compliant data, and they explain that "where we expect a property value of type Person, Place, Organization or some other subClassOf Thing, we will get a text string. In the spirit of "some data is better than none", we will accept this markup and do the best we can" (Schema.org, 2014, § Conformance). As also noted by Barker and Campbell (2014, p. 3),"Schema.org is designed to be forgiving to the extent that if information [. . .] is provided as a plain text string, an attempt should be made to interpret it". Probably this choice goes in the direction of making Schema.org more simple and accessible to content creators.

Every property is described in detail on its own page (see example in fig. 3.9). A description and the expected values and types for the property are always provided. Additionally, when relevant, some properties can also require some other details, such as:

- outdated properties replaced by the property;
- inverse-properties;



Thing > CreativeWork > Book

A book.

Usage: Between 10,000 and 50,000 domains

[more...]

- Check for open issues.

Property	Expected Type	Description
Properties from Book		
bookEdition	Text	The edition of the book.
bookFormat	BookFormatType	The format of the book.
illustrator	Person	The illustrator of the book.
isbn	Text	The ISBN of the book.
numberOfPages	Integer	The number of pages in the book.
Properties from CreativeWork		
about	Thing	The subject matter of the content.
accessibilityAPI	Text	Indicates that the resource is compatible with the referenced accessibility API (WebSchemas wiki lists possible values).
accessibilityControl	Text	Identifies input methods that are sufficient to fully control the described resource (WebSchemas wiki lists possible values).

Figure 3.7: Example of the way properties are indicated on a `Book` type's page (excerpt, taken from https://schema.org/Book)

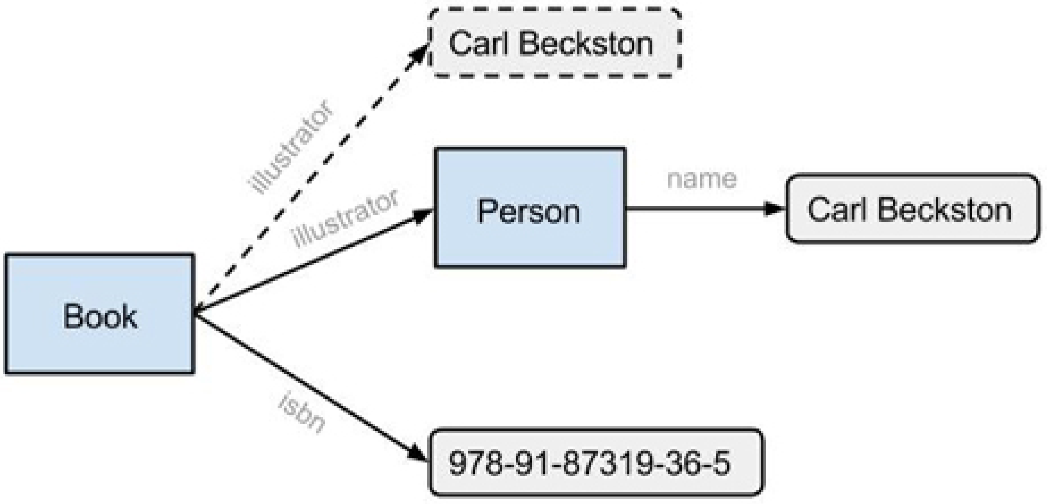
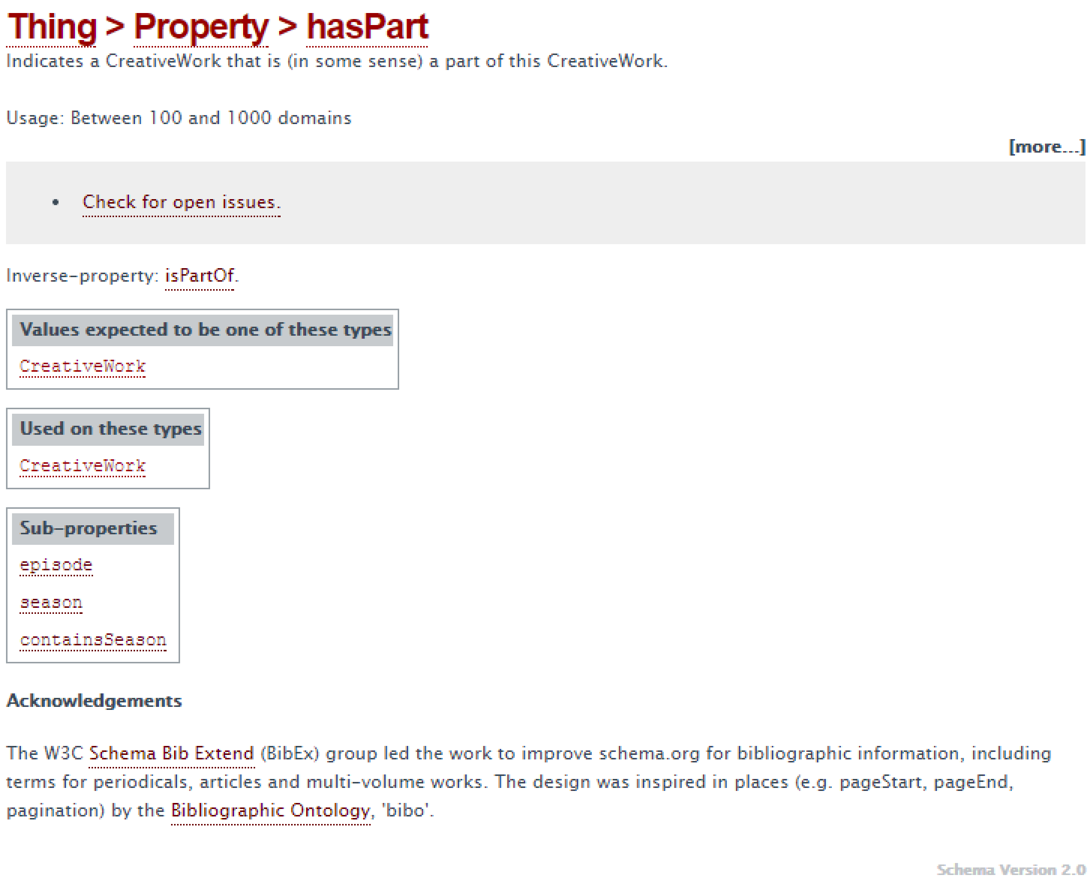


Figure 3.8: Some details about a `Book` type showing two different methods to indicate an `illustrator` property

- sub- or super-properties;
- usage guidelines;

- notes or acknowledgements.



Thing > Property > hasPart

Indicates a CreativeWork that is (in some sense) a part of this CreativeWork.

Usage: Between 100 and 1000 domains

[more...]

- Check for open issues.

Inverse-property: isPartOf.

Values expected to be one of these types
CreativeWork

Used on these types
CreativeWork

Sub-properties
episode
season
containsSeason

Acknowledgements

The W3C Schema Bib Extend (BibEx) group led the work to improve schema.org for bibliographic information, including terms for periodicals, articles and multi-volume works. The design was inspired in places (e.g. pageStart, pageEnd, pagination) by the Bibliographic Ontology, 'bibo'.

Schema Version 2.0

Figure 3.9: Example of the https://schema.org/hasPart property page

3.6.2 Syntaxes

As commented above, Schema.org supports different syntaxes. "[S]tructured data on the web can and should accommodate multiple encodings: we hope to emphasize this by accepting both microformat encoding and RDFa encoding. Each encoding has its pluses and minuses, and the debate is a fine intellectual exercise, but it detracts from the real issues" (Goel et al., 2009b). Moreover, the support for the JSON-LD format was introduced later on in the development of Schema.org (Brickley, 2013).

To better understand how the syntaxes look like, some examples from the markup of the same thing with different syntaxes will be shown. All the examples are taken from https://schema.org/RadioSeries. In order to be able to see the difference between markup and plain HTML, an example of code without the use of Schema.org is also shown in listing 1.

```
<div>
   <a href="http://www.bbc.co.uk/programmes/b006qykl">In Our Time</a>,
   <a href="http://www.bbc.co.uk/programmes/b03ggc19">Ordinary Language Philosophy</a>, episode
     ↪  604,
 broadcast on BBC Radio 4 at 09:00, 7th of November 2013, available online from 09:45 on the same
   ↪  day.
</div>
```

Listing 1: Example of markup of the type https://schema.org/RadioSeries using plain HTML without Schema.org markup

3.6.2.1 Microdata

One of the syntaxes used in Schema.org is Microdata. Microdata is an HTML5 specification that allows the markup of structured data in HTML documents. Thhe natural development of this is having a shared vocabulary that facilitates the use and cooperation between content creators. Microdata defines three new HTML attributes:

- `itemscope` – it requires no value and indicates that a new thing is being described;
- `itemtype` – indicates the type of item;
- `itemprop` – defines properties and values (Ronallo, 2012, § So what is Microdata?).

3.6.2.2 RDFa Lite 1.1

RDFa is a W3C standard that enables embedding RDF in HTML. There exist different versions of the specification. The full RDFa 1.1 syntax does have an high expressivity, while its subset RDFa Lite 1.1 is fairly simple.

> “[I]mplementing RDFa has proven to be overly complex for most Web developers. Google has supported RDFa in some fashion since 2009, and over that time has discovered a large error rate in the application of RDFa by webmasters. Simplicity is a central reason for the development of Microdata and the search engines preferring it over RDFa. In part a reaction to greater adoption of Microdata, a simplified profile of RDFa has been created. RDFa Lite 1.1 provides simpler authoring guidelines that mirror more closely the

```
<div itemscope itemtype="http://schema.org/RadioSeries">
  <a itemprop="url" href="http://www.bbc.co.uk/programmes/b006qykl">
   <span itemprop="name">In Our Time</span></a>,
  <div itemprop="episode" itemscope itemtype="http://schema.org/RadioEpisode">
    <a itemprop="url" href="http://www.bbc.co.uk/programmes/b03ggc19">
     <span itemprop="name">Ordinary Language Philosophy</span></a>,
    episode <span itemprop="position">604</span>, broadcast on
    <div itemprop="publication" itemscope itemtype="http://schema.org/BroadcastEvent">
      <div itemprop="publishedOn" itemscope itemtype="http://schema.org/BroadcastService">
        <a itemprop="url" href="http://www.bbc.co.uk/radio4"><span itemprop="name">BBC Radio 4</span></a>
      </div> at
      <span itemprop="startDate"
        content="2013-11-07T09:00:00+01:00">09:00, 7th of November 2013</span>
    </div>,
    available online from
    <div itemprop="publication" itemscope itemtype="http://schema.org/OnDemandEvent">
      <span itemprop="startDate"
       content="2013-11-07T09:45:00+01:00">09:45 on the same day</span>.
    </div>
  </div>
</div>
```

Listing 2: Example of markup of the type https://schema.org/RadioSeries using the Microdata syntax

> syntax of Microdata" (Ronallo, 2012, § A Short History of Structured Data in HTML).

Also the W3C specifies that RDFa Lite is not a complete solution for advanced data markup tasks, but that it works for the needs of most web masters, that can learn the specification quite quickly (Sporny, Bazaar, & Inc., 2015, § Abstract). Speaking of libraries and similar institutions, RDFa has the potential to expose their data in an effective way: "RDFa gives libraries as data publishers the opportunity to share their data in a structured way, using HTML without additional weighty infrastructure and how useful that data can be to others in creating their own graphs" (Coombs, 2015a). Coombs (2015a) from OCLC shows through the manipulation of WorldCat data how powerful the linking and exposing of data using RDFa, SPARQL and other Linked Data technologies could be.

RDFa Lite consists of five simple attributes; `vocab`, `typeof`, `property`, `resource`, and `prefix`. The first three are showed in the next code example below. the attribute `resource` is used to identify a thing described on a page, that others can link to. In a similar way than RDF, `prefix` specifies a prefix and a namespace for an external vocabulary, so that one can use external terms to describe his own things. E.g. `prefix="ov:`

http://open.vocab.org/terms/" allows the use of something like My favorite animal is the <span property="ov:preferredAnimal">Koala</span>. One interesting thing is that RDFa allows the value of a property attribute to be a list of values, and this is particularly useful when using different vocabularies. E.g. in <p property="dc:creator schema:creator">John Doe</p> both Dublin Core and Schema.org properties are used (see also section 3.7.3 for other examples). In this way our vocabulary can be richer and machines can disambiguate the data in a better way. There is little difference between Microdata and RDFa, and this is showed in listings 2 and 3. However, the wider Linked Data community is based upon RDF and therefore more closely aligned with RDFa. Moreover, " RDFa brings some of this semantic web thinking to the markup of web pages, hence the overlap with Schema.org" (Barker, 2012).

```
<div vocab="http://schema.org/" typeof="RadioSeries">
  <a property="url" href="http://www.bbc.co.uk/programmes/b006qykl">
   <span property="name">In Our Time</span></a>,
  <div property="episode"  typeof="RadioEpisode">
    <a property="url" href="http://www.bbc.co.uk/programmes/b03ggc19">
     <span property="name">Ordinary Language Philosophy</span></a>,
    episode <span property="position">604</span>, broadcast on
    <div property="publication"  typeof="BroadcastEvent">
      <div property="publishedOn"  typeof="BroadcastService">
        <a property="url" href="http://www.bbc.co.uk/radio4"><span property="name">BBC Radio
         ↪  4</span></a>
      </div> at
      <span property="startDate"
        content="2013-11-07T09:00:00+01:00">09:00, 7th of November 2013</span>
    </div>,
    available online from
    <div property="publication"  typeof="OnDemandEvent">
      <span property="startDate"
       content="2013-11-07T09:45:00+01:00">09:45 on the same day</span>.
    </div>
  </div>
</div>
```

Listing 3: Example of markup of the type https://schema.org/RadioSeries using the RDFa Lite syntax

Enthusiastic supporters of Microdata and RDFa often have heated discussions about their various merits. In implementation terms, apart from Microdata being more complex for defining multiple Types, there is little to choose between them, as you will see from the examples below. It must be pointed out that the wider Linked Data community is based upon RDF and therefore more closely aligned with RDFa.

3.6.2.3 JSON-LD

The support for this syntax was introduced in Schema.org in 2013. The motivation given from Schema.org developers is that "there are often cases when data is exchanged in pure JSON or as JSON within HTML. [...] JSON-LD provides mechanisms for interpreting structured data in JSON that promotes interoperability with other data formats [...] and improves the flow of information between JSON and other environments" (Schema.org, 2013).

The way of using JSON-LD for marking up HTML is different than Microdata and RDFa. As showed in listing 4, JSON is not embedded in the HTML tags along with the text as the other formats (*inline markup*). On the contrary, JSON can be inserted as a single block of structured data within the `<head>` element of a web page (*single block markup*). Besides being a good format for interoperability, the use of the JSON syntax has some other advantages, such as being able to separate the markup from the actual tag structure of the body, which, in case of changes to the page, lessens the risk of breaking the structure of the HTML and the RDF triples in the page. Moreover, it is a quite straightforward task to add an independent JSON block to an existing webpage at a later stage. The disadvantages are related to the fact that the markup is not easily human readable, not being alongside the text. Besides, it is more difficult for search engines to verify that the information in the JSON block doesn't contain misleading information (Purohit & Harrison, 2014).

3.6.3 Schema.org real world examples

Recipes are one of the best developed and supported things in Schema.org, so it is useful to show some examples from this. An example of enriched search results can be seen in fig. 3.10. Here it is possible to see how some results present some enriched information made possible by the use of Schema.org markup. E.g. the website minimalistbaker.com uses the following code to indicate the total cooking time: `time itemprop="totalTime" datetime="PT1H5M">1 hour 5 mins</time>`. On the https://schema.org/totalTime page, the property `totalTime` is described as "[t]he total time it takes to prepare and cook the recipe", that the property expects some `Duration` value, and that this value should be indicated in ISO 8601[28] duration standard. The code `datetime="PT1H5M"` is the machine-readable format of the duration expressed with the ISO 8601 standard, which shows as "1 hr 5 mins" in

[28] see http://en.wikipedia.org/wiki/ISO_8601

```
<script type="application/ld+json">
{
  "@context": "http://schema.org",
  "@type": "RadioSeries",
  "episode": {
    "@type": "RadioEpisode",
    "position": "604",
    "publication": [
      {
        "@type": "BroadcastEvent",
        "publishedOn": {
          "@type": "BroadcastService",
          "url": "http://www.bbc.co.uk/radio4"
        },
        "startDate": "09:00, 7th of November 2013"
      },
      {
        "@type": "OnDemandEvent",
        "startDate": "09:45 on the same day"
      }
    ],
    "url": "http://www.bbc.co.uk/programmes/b03ggc19"
  },
  "url": "http://www.bbc.co.uk/programmes/b006qykl"
}
</script>
```

Listing 4: Example of markup of the type https://schema.org/RadioSeries using the JSON-LD syntax

the enriched search result.

Similarly, the website Epicurious[29] uses Schema.org to show the amount of calories in the recipe directly in the search result, using the following code:

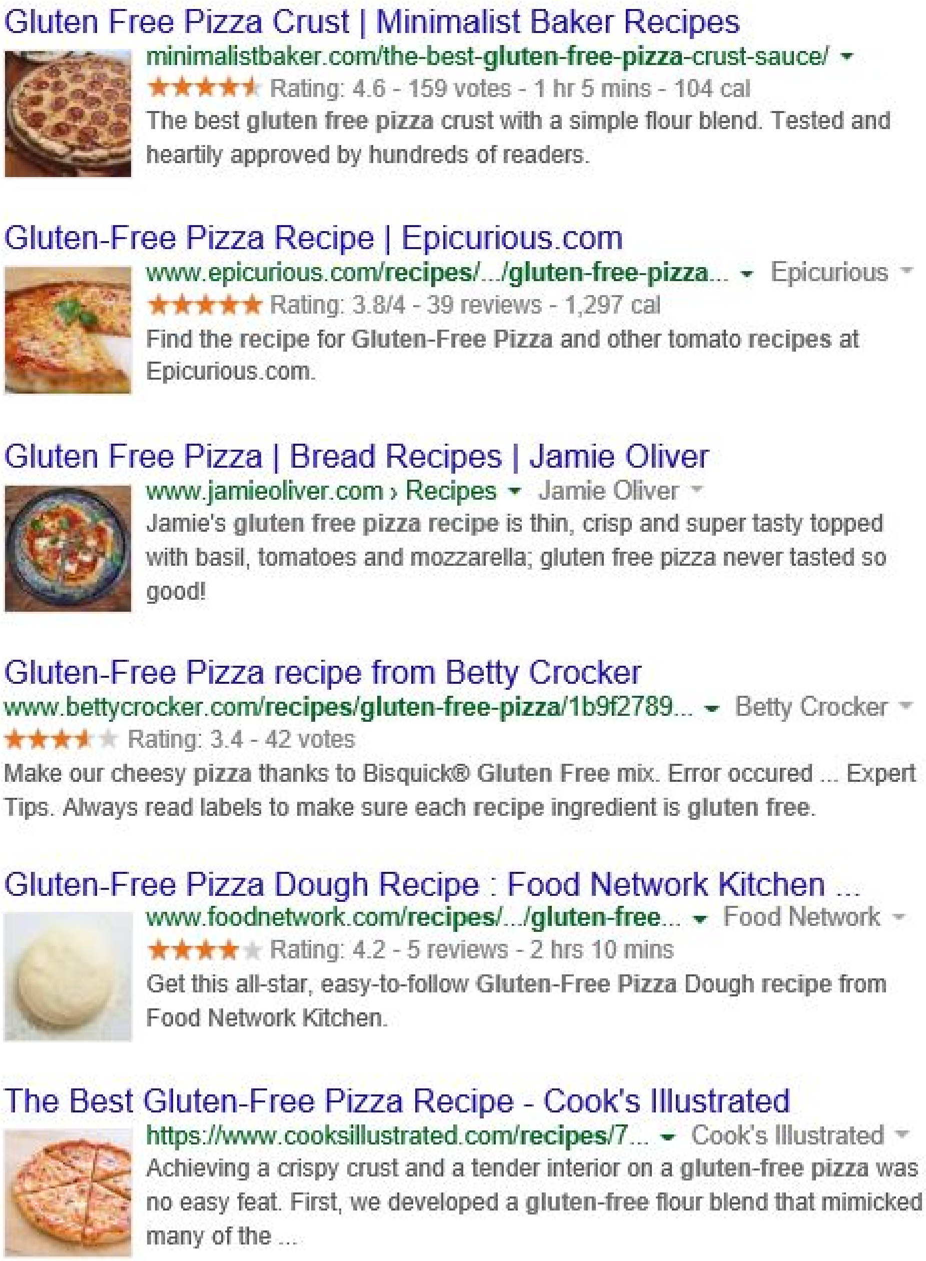
```
{<span class="nutriLabel">Calories</span><span class="nutriData"
 ↪   itemprop="calories">1,297</span>}
```

Listing 5: Example of the indication of calories

Some results in fig. 3.10 do not have enriched search results. However, they use Schema.org markup, but in order to describe other details than the ones displayed in the search result. E.g. jamieoliver.com uses Schema.org just to indicate nutritional information, with the class `NutritionInformation` and some related properties. This website focuses instead extensively on Microformats metadata (about Microformats, see section 3.8).

Another example worth discussing is cooksillustrated.com, which uses Schema.org

[29] http://www.epicurious.com/



Gluten Free Pizza Crust | Minimalist Baker Recipes
minimalistbaker.com/the-best-**gluten-free-pizza**-crust-sauce/ ▾
★★★★★ Rating: 4.6 - 159 votes - 1 hr 5 mins - 104 cal
The best **gluten free pizza** crust with a simple flour blend. Tested and heartily approved by hundreds of readers.

Gluten-Free Pizza Recipe | Epicurious.com
www.epicurious.com/**recipes**/.../**gluten-free-pizza**... ▾ Epicurious ▾
★★★★★ Rating: 3.8/4 - 39 reviews - 1,297 cal
Find the recipe for **Gluten-Free Pizza** and other tomato recipes at Epicurious.com.

Gluten Free Pizza | Bread Recipes | Jamie Oliver
www.jamieoliver.com › Recipes ▾ Jamie Oliver ▾
Jamie's **gluten free pizza recipe** is thin, crisp and super tasty topped with basil, tomatoes and mozzarella; gluten free pizza never tasted so good!

Gluten-Free Pizza recipe from Betty Crocker
www.bettycrocker.com/**recipes**/**gluten-free-pizza**/1b9f2789... ▾ Betty Crocker ▾
★★★★☆ Rating: 3.4 - 42 votes
Make our cheesy **pizza** thanks to Bisquick® **Gluten Free** mix. Error occured ... Expert Tips. Always read labels to make sure each recipe ingredient is **gluten free**.

Gluten-Free Pizza Dough Recipe : Food Network Kitchen ...
www.foodnetwork.com/**recipes**/.../**gluten-free**... ▾ Food Network ▾
★★★★☆ Rating: 4.2 - 5 reviews - 2 hrs 10 mins
Get this all-star, easy-to-follow **Gluten-Free Pizza** Dough **recipe** from Food Network Kitchen.

The Best Gluten-Free Pizza Recipe - Cook's Illustrated
https://www.cooksillustrated.com/**recipes**/7... ▾ Cook's Illustrated ▾
Achieving a crispy crust and a tender interior on a **gluten-free pizza** was no easy feat. First, we developed a **gluten-free** flour blend that mimicked many of the ...

Figure 3.10: A search for "gluten free pizza recipe", made on Google on 17 April 2015. Some results are enriched with Schema.orgś rich snippets showing pictures, ratings, cooking time and other information

markup to indicate that the object of the page is a `Recipe` class, and some general properties, such as date of creation and ingredients. However, the use of most of the markup appears to be incorrect, as it is embedded in the `<meta>` tag, and this is not considered good practice. This way of using Schema.org is suggested for use only if the "information can't be marked up because of the way it appears on the page" (e.g. a picture or a Flash object) and "should be used sparingly"(Schema.org, 2011a, § 3c).

Moreover, metadata embedded in the `<meta>` tag is considered unreliable because it can be vulnerable to spammers, and it is considered invisible metadata by search engines (Barker & Campbell, 2014, p. 8). For example Google's policy is that of not displaying metadata for content that is invisible to end users, such as markup inside the `<meta>` tag (Google, 2015b).

Although these last two websites contain Schema.org markup, it seems that right now other kind of information is given more priority to by search engines in search results, such as ratings, cooking time and calories. These examples show how structured data in HTML is important in itself to give semantic meaning to concepts in web pages. However, search engines will ultimately decide how and when to show the content and the way of displaying this content will probably change over time, when the search engines will find ways to collect more meaningful data and connect things in new ways.

As mentioned in section 3.3 some information that enriches the Knowledge Graph comes from Schema.org. E.g. the reviews information displayed in the Knowledge Graph panel is taken from Schema.org. This is clear when searching for a film title (see fig. 3.11). The websites IMDb[30], Rotten Tomatoes[31], Roger Ebert[32] and Metacritic[33] are the only ones on the result list that produce rich snippets for ratings originating from Schema.org markup. Their ratings show on the enriched search panel on the right. As for the remainder of the data showed on the panel, though the source is unclear, it seems to be a selection of images from Google Images as well as data from Wikipedia. In the picture it's also possible to see how the discovery part of the Knowledge Graph, "People also search for", is showed on the bottom. This information comes from search data (Singhal, 2012).

3.6.4 Schema.org extensions

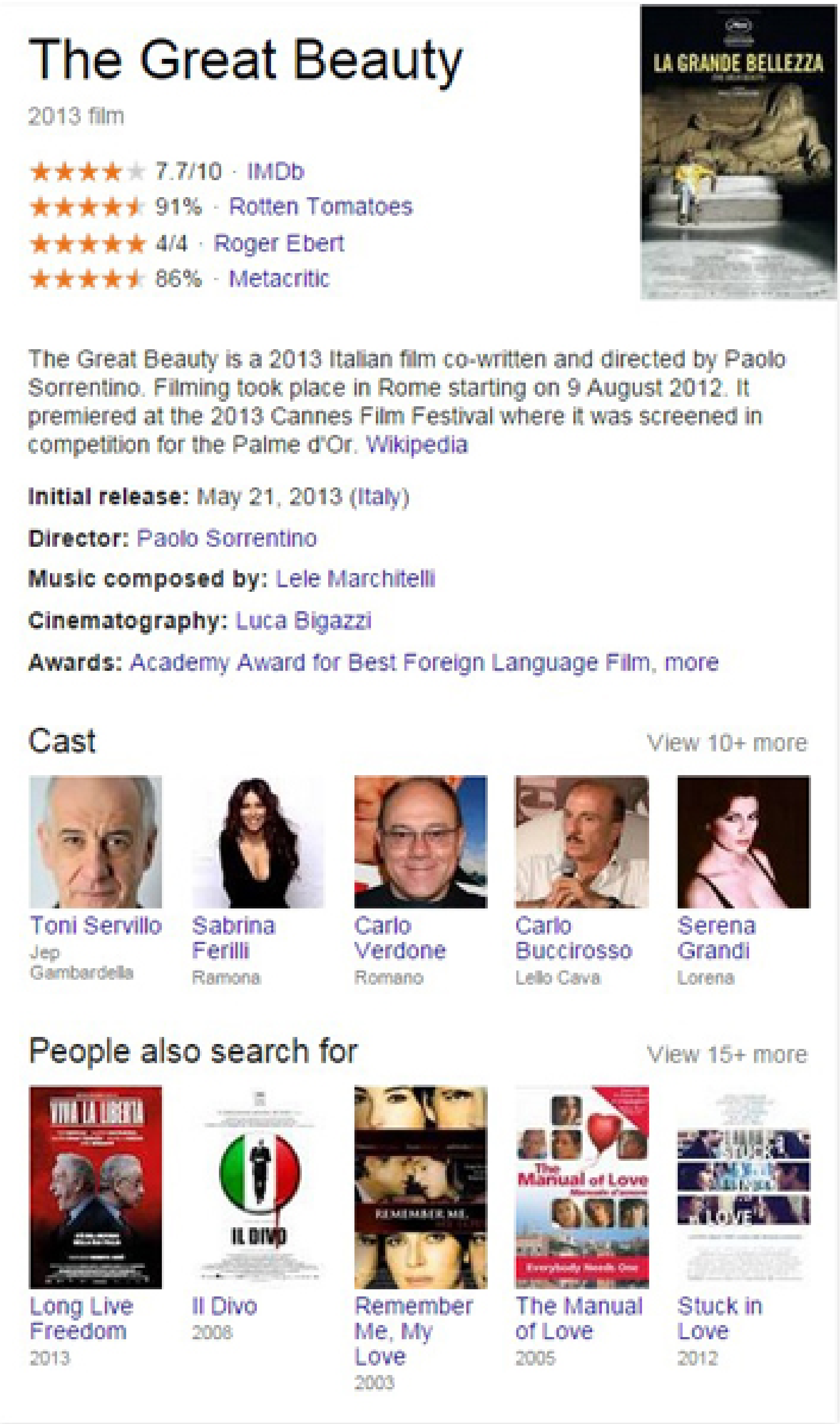
> "As schema.org adoption has grown, a number groups with more specialized vocabularies have expressed interest in extending schema.org with their terms. Examples of this include real estate, product, finance, medical and bibliographic information. [...] Groups that have a special interest in one of

[30] http://www.imdb.com

[31] http://www.rottentomatoes.com

[32] www.rogerebert.com

[33] http://www.metacritic.com



The Great Beauty

2013 film

★★★★☆ 7.7/10 · IMDb
★★★★½ 91% · Rotten Tomatoes
★★★★★ 4/4 · Roger Ebert
★★★★½ 86% · Metacritic

The Great Beauty is a 2013 Italian film co-written and directed by Paolo Sorrentino. Filming took place in Rome starting on 9 August 2012. It premiered at the 2013 Cannes Film Festival where it was screened in competition for the Palme d'Or. Wikipedia

Initial release: May 21, 2013 (Italy)

Director: Paolo Sorrentino

Music composed by: Lele Marchitelli

Cinematography: Luca Bigazzi

Awards: Academy Award for Best Foreign Language Film, more

Cast View 10+ more

Toni Servillo	Sabrina Ferilli	Carlo Verdone	Carlo Buccirosso	Serena Grandi
Jep Gambardella	Ramona	Romano	Lello Cava	Lorena

People also search for View 15+ more

Long Live Freedom	Il Divo	Remember Me, My Love	The Manual of Love	Stuck in Love
2013	2008	2003	2005	2012

Figure 3.11: An example of the Knowledge Graph panel showing information for the search "The great beauty"

> these topics often need a level of specificity in the vocabulary and operational independence. We are introducing a new extension mechanism which we hope will enable these and many other groups to extend schema.org" (Guha, 2015).

The new extension mechanism is a major change introduced with version 2.0, and it has been in use since May 2015 (Schema.org, 2015a). Extensions are based on the principle that if Schema.org does not contain the term needed, the vocabulary may be extended by adding sub-types or sub-properties through string concatenation (Schema.org, 2011b). E.g., Schema.org defines a person using the URI https://schema.org/Person. If one needs to specify that a person is a minister, one would then extend `Person` by adding a more specific type `Minister` at the end: https://schema.org/Person/Minister. The intention was to have the extensions give search engines at least some information, even when they come across unknown terms. A similar principle of "some data is better than none" is applied when one uses a text string instead of a value, as discussed above, in section 3.6.1. The extension mechanism was also intended as a way of letting the ontology grow organically, as terms which would become common might have been included in the schema (Schema.org, 2011b). However, it is also understandable when someone points out that "[y]ou can extend Schema.org but it would seem pointless doing this in isolation because Google and Bing aren't going to suddenly adopt my extensions because I've invented them – I'm guessing it would need a certain critical mass of artists, galleries, museums, and universities adopting a set of schema.org extensions to make it worthwhile for search engines to support them" (Watson, 2013). The lack of major actors doing this, and the lack of control this mechanism can generate could be some of the reasons why the developers of Schema.org have come up with a new extension mechanism, described below. Another weak side of the old mechanism was that it apparently did not conform to the Microdata specification, as Tennison (2011a) points out. The specification says that URIs should be opaque identifiers, and that one should not infer meaning through string parsing. The mechanism also seemed to encounter some difficulties in relation to disambiguation of homonyms. Using the previous example, it is impossible to infer from the URI itself if `Person/Minister` refers to a clergyman, or to the institutional role – unless of course disambiguating this through an external URI, e.g. from DBpedia. This problem would also persist when incorporating the term in the schema, and might make markup that used to be correct say something that was unintended.

In the new extension mechanism there are two kinds of extensions: reviewed/hosted, and external extensions. Both give the possibility to add classes and properties in a consistent manner, as an overlay on the core vocabulary. Added terms from two different extensions might overlap in concept, but there can't be homonymy, such as in the `Minister` example.

Reviewed extensions are external vocabularies that extend the core vocabulary, but are separate from it. These extensions are given their own namespace, e.g. a jazz music extension would look like `jazz.schema.org`. External extensions are extensions created by third parties with specific needs to their application and the specific purpose of linking with Schema.org. They should be stored on the extension developer's domain, thus having to use different namespaces than reviewed extensions, e.g. `schema.jazz.com` (Schema.org, 2015a).

Then there are proposals, i.e. vocabularies that at some point might be integrated into the core. This is the case with a vocabulary for the description of Automobiles, proposed by the Automotive Ontology community group, introduced in version 2.0 (Guha, 2015); a vocabulary for describing bibliographic resources, introduced in version 1.9, thanks to the W3C BibExtend Community Group and the Bibliographic Ontology (BIBO)[34] (Schema.org, 2015b, p. 1.9)(see also section 3.6.4); and several others[35].

It seems that extensions will become an important part of Schema.org. The potential challenge with this is that although Schema.org is trying to avoid becoming an ontology of everything, it's nonetheless become big – and will only get bigger.

An example of a recent external extension developed especially for Schema.org is BiblioGraph.net[36]. It is a OCLC sponsored initiative that adds to the core Schema.org vocabulary a small collection of classes and properties for a more specific description of bibliographic resources.

> "It is designed to be used by organizations in the bibliographic field to describe resources that they curate, aggregate, describe and reference for local use, or to serve as authoritative hubs of knowledge about these items. Such organizations will benefit from the use of the BiblioGraph.net vocabulary through greater exposure, consumption, sharing, discovery, integration,

[34] http://bibliontology.com/

[35] a list of possible future discussions for Schema.org extensions and proposals can be found here: www.w3.org/wiki/WebSchemas#Proposals_from_and_for_Schema.org and an overview of proposals can be found here: http://www.w3.org/wiki/WebSchemas/SchemaDotOrgProposals

[36] available at http://bibliograph.net/

> navigation and display of bibliographic related information across the web" (BiblioGraph, 2015).

Currently, BiblioGraph.net draws from other vocabularies, such as FOAF and Product Ontology[37] (BiblioGraph, n.d.).

As mentioned on the Schema.org extension page, a new term from this extension would get the namespace `bib.schema.org` and could be incorporated in the HTML markup, as the following example in RDFa syntax shows (Schema.org, 2015a):

```
<div vocab="http://schema.org/" prefix="bib: http://bib.schema.org/">
    <p typeof="Book" resource="http://www.freebase.com/m/0h35m">
        <em property="name">The Fellowship of the Rings</em> was written by
        <span property="author">J.R.R Tolkien</span> and was originally published
        in the <span property="publisher" typeof="Organization">
            <span property="location">United Kingdom</span> by
            <span property="name">George Allen & Unwin</span>
        </span> in <time property="datePublished">1954</time>.
        The book has been republished many times, including editions by
        <span property="workExample" typeof="Book">
            <span property="publisher" typeof="Organization">
                <span property="name">HarperCollins</span>
            </span> in <time property="datePublished">1974</time>
            (ISBN: <span property="isbn">0007149212</span>)
        </span> and by
        <span property="workExample" typeof="Book bib:Microform">
            <span property="publisher" typeof="Organization">
                <span property="name">Microfiche Press</span>
            </span> in <time property="datePublished">2016</time>
            (ISBN: <span property="isbn">12341234</span>).
        </span>
    </p>
</div>
```

Listing 6: Example of how an external extension to Schema.org could be incorporated in the HTML markup

In particular, `prefix="bib:  http://bib.schema.org/"` declares the new namespace, and `typeof="Book bib:Microform"` further specifies the Schema.org `Book` type using the new term coming from the BiblioGraph.net extension.

The already mentioned Schema Bib Extend W3C Community Group is another interest group that is in part connected to Bibliograph.net and has done some important work for the library community. This group has been active for some years, and got some additions to the Schema.org core approved along the years (Schema.org, 2015b). It is important to remark how the approach of the project has been "not to become a group emulating and duplicating metadata standards" (Wallis, 2013,

[37] http://www.productontology.org/

p. 32), showing the awareness for the problem of the proliferation of metadata standards. This is carried out through first using what existing Schema.org terms can offer to to the library community needs, then suggesting new uses for terms, examples and changes to Schema.org documentation, and lastly trying to get approval for new classes and properties, in case some major gaps are found. Thanks to the group's activities, it seems that others have been encouraged to use Schema.orgḞor example some open source library systems such as Koha or Evergreen have implemented the possibility to use Schema.org metadata in their user interfaces (Wallis, 2013).

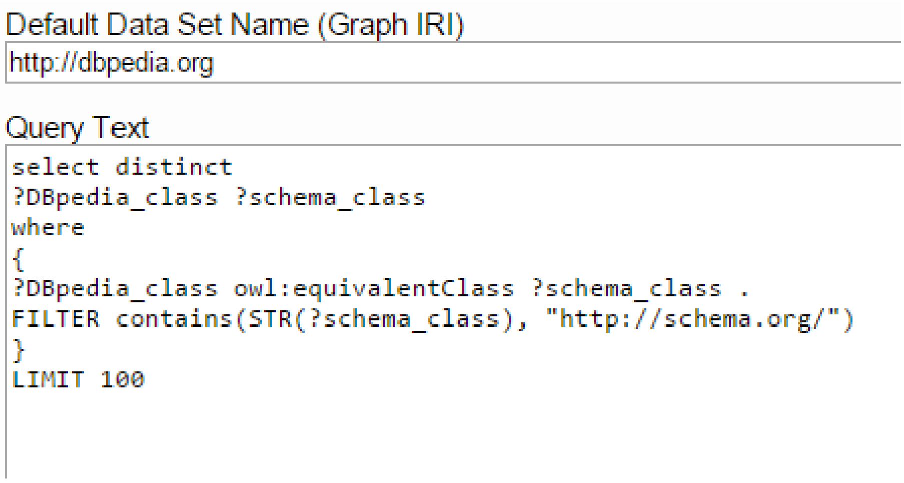
3.7 Mappings and other initiatives involving Schema.org

This section shows several mappings and work involving Schema.org. It is useful to better understand the possibilities of such a vocabulary, especially to describe and expose cultural heritage and bibliographic data. These examples could also be a starting point for further work in the TORCH project.

3.7.1 DBpedia

One data source that links to the Schema.org vocabulary in addition to its own ontology in order to increase interoperability is DBpedia. DBpedia mapped 47 of its own classes and 35 of its own properties to Schema.org classes and properties. This is done respectively using the predicates `owl:equivalentClass` and `owl:equivalentProperty` (Fleischhacker, Bryl, & Bizer, 2014, § 1). It is possible to query the public DBpedia SPARQL-endpoint[38] to obtain the mapping between the two ontologies. The queries shown in figs. 3.12 and 3.13 were executed on the DBpedia endpoint to respectively obtain URIs of subjects and objects of the RDF triples in the ontology that are linked by the two OWL predicates. The results of the two queries are shown in tables 3.1 and 3.2.

[38] available at http://dbpedia.org/sparql



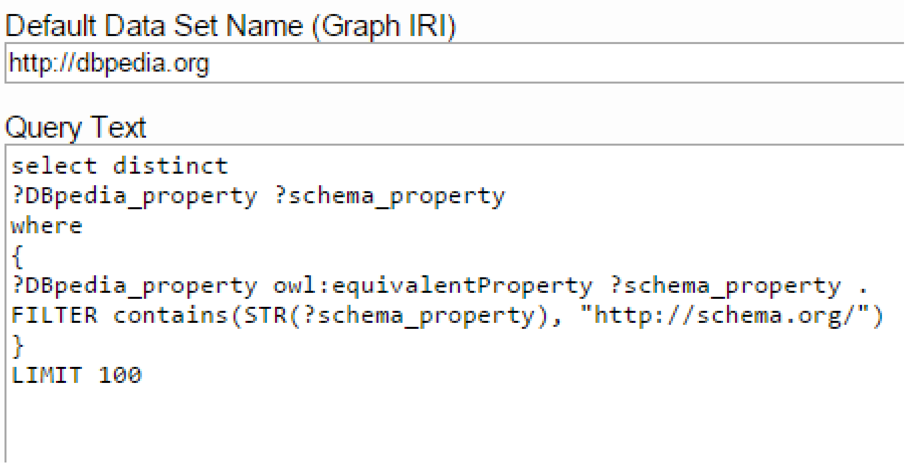
Default Data Set Name (Graph IRI)

http://dbpedia.org

Query Text

```
select distinct
?DBpedia_class ?schema_class
where
{
?DBpedia_class owl:equivalentClass ?schema_class .
FILTER contains(STR(?schema_class), "http://schema.org/")
}
LIMIT 100
```

Figure 3.12: SPARQL query executed on DBpedia in order to obtain the mapping of the classes



Default Data Set Name (Graph IRI)

http://dbpedia.org

Query Text

```
select distinct
?DBpedia_property ?schema_property
where
{
?DBpedia_property owl:equivalentProperty ?schema_property .
FILTER contains(STR(?schema_property), "http://schema.org/")
}
LIMIT 100
```

Figure 3.13: SPARQL query executed on DBpedia in order to obtain the mapping of the properties

DBpedia class	**Schema.org class**
http://dbpedia.org/ontology/Person	http://schema.org/Person
http://dbpedia.org/ontology/Hotel	http://schema.org/Hotel
http://dbpedia.org/ontology/Restaurant	http://schema.org/Restaurant
http://dbpedia.org/ontology/ShoppingMall	http://schema.org/ShoppingCenter
http://dbpedia.org/ontology/SkiArea	http://schema.org/SkiResort
http://dbpedia.org/ontology/AdministrativeRegion	http://schema.org/AdministrativeArea
http://dbpedia.org/ontology/Airport	http://schema.org/Airport
http://dbpedia.org/ontology/BodyOfWater	http://schema.org/BodyOfWater
http://dbpedia.org/ontology/Book	http://schema.org/Book
http://dbpedia.org/ontology/Canal	http://schema.org/Canal
http://dbpedia.org/ontology/City	http://schema.org/City
http://dbpedia.org/ontology/College	http://schema.org/CollegeOrUniversity
http://dbpedia.org/ontology/University	http://schema.org/CollegeOrUniversity
http://dbpedia.org/ontology/Continent	http://schema.org/Continent
http://dbpedia.org/ontology/Country	http://schema.org/Country
http://dbpedia.org/ontology/Work	http://schema.org/CreativeWork
http://dbpedia.org/ontology/EducationalInstitution	http://schema.org/EducationalOrganization
http://dbpedia.org/ontology/Event	http://schema.org/Event
http://dbpedia.org/ontology/GovernmentAgency	http://schema.org/GovernmentOrganization
http://dbpedia.org/ontology/Hospital	http://schema.org/Hospital
http://dbpedia.org/ontology/Lake	http://schema.org/LakeBodyOfWater
http://dbpedia.org/ontology/HistoricBuilding	http://schema.org/LandmarksOrHistoricalBuildings
http://dbpedia.org/ontology/HistoricPlace	http://schema.org/LandmarksOrHistoricalBuildings
http://dbpedia.org/ontology/Language	http://schema.org/Language
http://dbpedia.org/ontology/Library	http://schema.org/Library
http://dbpedia.org/ontology/Mountain	http://schema.org/Mountain
http://dbpedia.org/ontology/Film	http://schema.org/Movie
http://dbpedia.org/ontology/Museum	http://schema.org/Museum
http://dbpedia.org/ontology/Album	http://schema.org/MusicAlbum
http://dbpedia.org/ontology/Song	http://schema.org/MusicRecording
http://dbpedia.org/ontology/Organisation	http://schema.org/Organization
http://dbpedia.org/ontology/Park	http://schema.org/Park
http://dbpedia.org/ontology/Place	http://schema.org/Place
http://dbpedia.org/ontology/RadioStation	http://schema.org/RadioStation
http://dbpedia.org/ontology/River	http://schema.org/RiverBodyOfWater
http://dbpedia.org/ontology/School	http://schema.org/School
http://dbpedia.org/ontology/SportsEvent	http://schema.org/SportsEvent
http://dbpedia.org/ontology/SportsTeam	http://schema.org/SportsTeam
http://dbpedia.org/ontology/Stadium	http://schema.org/StadiumOrArena
http://dbpedia.org/ontology/Arena	http://schema.org/StadiumOrArena
http://dbpedia.org/ontology/TelevisionEpisode	http://schema.org/TVEpisode
http://dbpedia.org/ontology/TelevisionStation	http://schema.org/TelevisionStation
http://dbpedia.org/ontology/Website	http://schema.org/WebPage
http://dbpedia.org/ontology/Annotation	http://schema.org/Comment
http://dbpedia.org/ontology/Painting	http://schema.org/Painting
http://dbpedia.org/ontology/Sculpture	http://schema.org/Sculpture
http://dbpedia.org/ontology/Sea	http://schema.org/SeaBodyOfWater

Table 3.1: DBpedia classes and corresponding Schema.org types

DBpedia property	**Schema.org property**
http://dbpedia.org/ontology/starring	http://schema.org/actors
http://dbpedia.org/ontology/author	http://schema.org/author
http://dbpedia.org/ontology/award	http://schema.org/awards
http://dbpedia.org/ontology/birthDate	http://schema.org/birthDate
http://dbpedia.org/ontology/mediaType	http://schema.org/bookFormat
http://dbpedia.org/ontology/parentOrganisation	http://schema.org/branchOf
http://dbpedia.org/ontology/artist	http://schema.org/byArtist
http://dbpedia.org/ontology/child	http://schema.org/children
http://dbpedia.org/ontology/locatedInArea	http://schema.org/containedIn
http://dbpedia.org/ontology/deathDate	http://schema.org/deathDate
http://dbpedia.org/ontology/restingDate	http://schema.org/deathDate
http://dbpedia.org/ontology/director	http://schema.org/director
http://dbpedia.org/ontology/runtime	http://schema.org/duration
http://dbpedia.org/ontology/albumRuntime	http://schema.org/duration
http://dbpedia.org/ontology/duration	http://schema.org/duration
http://dbpedia.org/ontology/filmRuntime	http://schema.org/duration
http://dbpedia.org/ontology/endDate	http://schema.org/endDate
http://dbpedia.org/ontology/episodeNumber	http://schema.org/episodeNumber
http://dbpedia.org/ontology/foundedBy	http://schema.org/founders
http://dbpedia.org/ontology/genre	http://schema.org/genre
http://dbpedia.org/ontology/illustrator	http://schema.org/illustrator
http://dbpedia.org/ontology/picture	http://schema.org/image
http://dbpedia.org/ontology/language	http://schema.org/inLanguage
http://dbpedia.org/ontology/isbn	http://schema.org/isbn
http://dbpedia.org/ontology/map	http://schema.org/maps
http://dbpedia.org/ontology/musicComposer	http://schema.org/musicBy
http://dbpedia.org/ontology/nationality	http://schema.org/nationality
http://dbpedia.org/ontology/numberOfEpisodes	http://schema.org/numberOfEpisodes
http://dbpedia.org/ontology/numberOfPages	http://schema.org/numberOfPages
http://dbpedia.org/ontology/producer	http://schema.org/producer
http://dbpedia.org/ontology/publisher	http://schema.org/publisher
http://dbpedia.org/ontology/firstPublisher	http://schema.org/publisher
http://dbpedia.org/ontology/relative	http://schema.org/relatedTo
http://dbpedia.org/ontology/spouse	http://schema.org/spouse
http://dbpedia.org/ontology/startDate	http://schema.org/startDate

Table 3.2: DBpedia properties and corresponding Schema.org properties

It is noticeable that Schema.org was inspired also by DBpedia (see section 3.6.1), as many classes and properties have the same names in the two ontologies.

3.7.2 WorldCat

A major initiative that uses mappings to Schema.org was undertaken in 2012 by WorldCat, the world's largest online database for library data. In the context of a project to make WorldCat's data available as Linked Data, OCLC decided to begin marking up the HTML in their web pages with RDFa and Schema.org terms. In the initiative "the linked data models of the FAST and VIAF authority files have been redesigned with references to classes defined in Schema.org for fundamental concepts such as "Person," "Organization" "Creative Work" and "Topic". Moreover, nearly 200 million "Wor" clusters are now modeled as linked data using Schema.org and associated with persistent URIs. (Godby & Denenberg, 2015, p. 5).

To date, more than 3 million records are described with a subset of terms from Schema.org[39]. They constitute the core vocabulary of WorldCat, and were enriched with a set of terms extending Schema.org, published at http://purl.org/library/ and maintained by the Schema Bib Extend group (Coombs, n.d.).

It is also possible to paste an URL such as http://experiment.worldcat.org/entity/work/data/1151002411 in a browser to visualize an example record of how Schema.org can be connected to URIs pointing to other Linked Data sources.

3.7.3 Europeana

The Europeana project started in 2008 as an effort to create a multi-sided platform that gives users access to different kinds of cultural heritage content. To date, they have collected digitized content from hundreds of institutions in all european countries. What's digitally available on Europeana is 300 million objects, which, they argue, represent only about 10% of the european cultural heritage material that could potentially be collected by the project. Europeana aims at becoming the largest repository of cultural heritage data in Europe. The project focuses on metadata quality; data accessibility, visibility and creative re-use through open licenses (Europeana, n.d.-b; "Europeana," 2015).

[39] these can be seen at http://www.oclc.org/developer/develop/linked-data/worldcat-vocabulary.en.html

A large subset of Europeana's data was transformed into linked data in 2012 and made available in a pilot project that is still experimental and under development. The data is represented using the Europeana Data Model (EDM), Europeana's own ontology. This is a quite complex ontology that serves the purposes of a complex aggregator of contents such as Europeana. This reflects in the fact that EDM re-uses elements coming from established vocabularies, such as FOAF, Dublin Core, OAI-ORE, SKOS and CIDOC-CRM (Europeana, n.d.-a). The framework for the linked data is built with the purpose of disseminating content through different channels as well, such as projects based on data re-use in specific contexts, portals about specific topics, mobile applications, etc. Better visibility of contents through search engines obtained with the use of Schema.org is considered an important part of this dissemination strategy (Europeana, n.d.-a; Isaac et al., 2013, p. 6).

To date, Schema.org is currently not fully implemented in Europeana's records. They apparently use just one type in their HTML pages as in the following excerpt from a Europeana item page[40]:

```
<div class="row"
about="http://www.europeana.eu/resolve/record/2022362/_Royal_Museums_
Greenwich__http___collections_rmg_co_uk_collections_objects_6274"
vocab="http://schema.org/"
typeof="CreativeWork">
```

Listing 7: Example of a Europeana item

The above `<div>` is one of the main containers of all the metadata about the item in the html structure of the webpage, and it contains a `CreativeWork` type. The rest of the metadata about the item from the example uses Dublin Core, as shown in the following excerpt:

```
<h1 class="hide-on-phones" property="http://purl.org/dc/elements/1.1/title dc:title">Violin</h1>
            [...]
                <span class="bold notranslate br">Geographic coverage:</span>
                <span class="translate" property="http://purl.org/dc/terms/spatial">Oxford,
                 ↪   Oxfordshire, England</span>
            [...]
                <span class="bold notranslate br">Date:</span>
                <span class="translate" property="http://purl.org/dc/elements/1.1/date">1819
                 ↪   -1827</span>
```

Listing 8: Example of use of Dublin Core metadata on Europeana

[40] http://www.europeana.eu/portal/record/2022362/_Royal_Museums_Greenwich__http___collections_rmg_co_uk_collections_objects_6274.html

However, to date Europeana uses Facebook's Open Graph (the `og:` properties) and Twitter Cards (the `twitter:` properties) consistently in the `<meta>` tags in the html `<head>`. Here is an example of this, taken from the same record used above:

```
<meta name="twitter:card" content="summary" />
    <meta name="twitter:url" content="http://www.europeana.eu/portal/record/2022362/_Royal_
    Museums_Greenwich__http___collections_rmg_co_uk_collections_objects_
    6274.html" />
    <meta name="twitter:title" content="Violin" />
    <meta name="twitter:description" content="Violin" />
    <meta name="twitter:site" content="@EuropeanaEU" />
    <meta name="twitter:image" content="http://europeanastatic.eu/api/image?uri=http%3A%2F%2F
    collections.rmg.co.uk%2FmediaLib%2F327%2Fmedia-327630%2Flarge.jpg
    &amp;size=FULL_DOC&amp;type=IMAGE" />
    <meta property="og:title" content="Violin" />
    <meta property="og:type" content="website" />
    <meta property="og:image" content="http://europeanastatic.eu/api/image?uri=http%3A%2F%2F
    collections.rmg.co.uk%2FmediaLib%2F327%2Fmedia-327630%2Flarge.jpg
    &amp;size=FULL_DOC&amp;type=IMAGE" />
    <meta property="og:url" content="http://www.europeana.eu/portal/record/2022362/_Royal_
    Museums_Greenwich__http___collections_rmg_co_uk_collections_objects_
    6274.html" />
    <meta property="og:description" content="Violin" />
    <meta property="og:site_name" content="Europeana" />
```

Listing 9: Example of use of Open Graph protocol and Twitter cards on Europeana

However, there exists a mapping between the EDM ontology and Schema.org made by Europeana[41]. Following this mapping, Király Király (2013, p. 6), one of the developers of Europeana, gives an example of how the main elements of a record would look like with the inclusion of Schema.org:

In this example Schema.org is used in combination with Dublin Core, as Europeana makes quite consistent use of the latter, as already pointed out above.

3.7.4 The Getty Art & Architecture Thesaurus

The J. Paul Getty Trust has been developing a framework for publishing its vocabularies (thesaurus, name authorities, geographic names) as Linked Open Data. These vocabularies describe data in fields such as art, architecture and cultural heritage (2014). There was a certain interest at the project in making the data more visible to search engines. After considering some existing metadata formats for the arts, they chose to use Schema.org for marking up webpages semantically. At that time Schema.org wasn't rich enough for their need of describing

[41] It can be found at https://github.com/europeana/portal/blob/master/portal2/src/test/schema.org.mapping/schema.org.mapping.properties, retrieved on 29 May 2015

```
<div about="http://www.europeana.eu/resolve/record
/2023710/8625058B9A7750FC2B300CD0D059772D009541F9" vocab="http://schema.org/"
 ↪  typeof="CreativeWork">
  <h1 property="dc:title name">Szemely - Kodaly Zoltan</h1>
  Description:
   <span property="dc:description description">
   Szemely - Kodaly Zoltan</span>
  Creator:
   <span property="dc:creator creator">Varkonyi Laszlo</span>
  Date:
   <span property="dc:date">[1950]</span>
  Date of creation:
   <span property="dc:created dateCreated">1950-01-01</span>
  Type:
    <span property="dc:type">Photo</span>;
    <span property="dc:type">photograph</span>
  Format:
    <span property="dc:format">image/jpeg</span>
</div>
```

Listing 10: Example of how a Europeana record would look like with embedded Schema.org. Notice the simultaneous use of Dublin Core and Schema.org(see also section 3.6.2.2)

works of art quite accurately, so they sent a proposal to the W3C for some additions to the official Schema.org ontology. The resulting discussion produced a new class, `VisualArtwork`, and related properties `artEdition`, `artform`, `material`, and `surface` which were added to one of Schema.org's latest versions, 1.93 (Schema.org, 2015b).

The metadata schema used to structure and describe Getty's data is VRA Core, a standard XML schema for the description of visual works (paintings, architecture, sculptures, etc.) and the images that document them. VRA Core has two main XML elements, `<work>` and `<image>`, used respectively to describe the actual work and the digital reproduction of the work (for example a jpg image on the web). This division can be expressed with all the data about the actual artwork (e.g. title, creator, description, material, etc.) expressed under the `VisualArtwork` class and the data about the reproduction of the artwork (e.g. filename, caption, etc.) under the `ImageObject` class (Watson, 2015b).

Using Schema.org it's possible to point to URIs from a controlled vocabulary, such as the Getty's Art & Architecture Thesaurus, as in the following example:

```
{<span property="material" resource="http://vocab.getty.edu/aat/300022443">Pencil</span>}
```

Listing 11: Example of a canonical reference to an external vocabulary

This is called canonical referencing and enables to unambiguously describe some strings which are clear to a human reader, but could create confusion for machines, for example when using the synonym “graphite” instead of “pencil”. Linking to third party sites can help search engines to better understand the item you are describing (Watson, 2015a; Schema.org, 2011a, § 3b). This kind of disambiguation is also a common method used in semantic web technologies, because linking to external datasets helps the machines to better understand the data, giving them additional information about an object.

3.7.5 UMBEL

UMBEL stands for Upper Mapping and Binding Exchange Layer. It is a top-level ontology containing 35000 concepts, and it is a subset of the already mentioned OpenCyc ontology. It is designed to provide mapping points for the interoperability of different ontologies. It has about 64000 mappings to different other ontologies, including Schema.org (Bergman & Giasson, 2015c, § 3.1-3.2; “UMBEL,” 2014).

The mapping[42] consists of nearly 600 Schema.org types mapped to UMBEL classes through the use of the `rdfs:subClassOf` property. This RDF Schema property is used to state that all the instances of one class are instances of another (Brickley & Guha, 2014, § 3.4).

```
<http://schema.org/TVClip> rdfs:subClassOf <http://umbel.org/umbel/rc/VideoClip> .
<http://schema.org/TVEpisode> rdfs:subClassOf <http://umbel.org/umbel/rc/TVShow-SingleCW> .
<http://schema.org/TVSeason> rdfs:subClassOf <http://umbel.org/umbel/rc/TVSeason> .
<http://schema.org/TVSeries> rdfs:subClassOf <http://umbel.org/umbel/rc/TVSeriesProduct> .
```

Listing 12: Example of mapping between Schema.org classes to UMBEL classes

There are no classes that are mapped through the `owl:equivalentClass`, such as it has been done in DBpedia. In fact, in the UMBEL documentation we can read that they considered different kinds of relations between classes and that they have different levels of importance, or strength. `owl:equivalentClass` is an equivalence relationship, and it is therefore stronger than the sub-class-of relationship `rdfs:subClassOf` (Bergman & Giasson, 2015a, § 2.5.1-4). Besides, not all concepts in Schema.org had a mapping to UMBEL, so in the initial mapping they added 78 new reference concepts, such as `TVSeason` in the example in listing 12 (Bergman & Giasson, 2015b).

[42] available at https://github.com/structureddynamics/UMBEL/blob/master/External%20Ontologies/schema.org.n3

As indicated by (Bergman & Giasson, 2015b), there are two recommended uses for these mappings. “Existing schema.org users may wish to use UMBEL reference concepts to expand their entity relationships or relate to other third-party sources [while] UMBEL users may employ the mappings to retrieve entity and attribute information from external Schema.org sources”.

Properties haven’t been mapped yet (2015b).

3.7.6 Dublin Core

The Dublin Core Metadata Initiative formed a group called DCMI Schema.org Alignment Task Group, which was active between 2011 and 2012. Their task was to find a mapping between DC and Schema.org. Although a status document from 2014 confirms the initiative came to a halt in 2012 because of some issues related to the mappings, the group produced some mappings of which slightly different versions exist[43] (DCMI Schema.org Alignment Task Group, 2014). The work is also described as being “in progress” (Schema.org, n.d.-b).

There are no concrete implementations of this mapping, but it is possible to briefly examine it. Looking at the mapping, one can see how they used both RDF Schema’s `rdfs:subClassOf` and `rdfs:subPropertyOf` properties and OWL’s `owl:equivalentClass` and `owl:equivalentProperty` properties to map the terms. In total, 6 classes and approximately 120 properties have been mapped. Some examples of this mapping are shown in listing 13 in Turtle syntax.

Among the properties we can see that different Schema.org terms, e.g. `genre, keywords`, map to the same DC property `subject`; `branchOf` and `partOfTVSeries`, both map to `isPartOf`; the classes `schema:Organization` and `schema:Person` map to `dct:Agent`. Another example is 10 different Schema.org properties which map into `dct:contributor`. This situation creates loss of granularity, because different terms in one ontology get merged in one. This is a consequence of the fact that Schema.org is a much bigger schema than DC, that in his qualified form consists of approximately 70 properties and 30 classes.

The two OWL equivalence properties are used where there is little doubt that the two terms share the same meaning (as in Place/Location, or about/subject), or when they are synonyms.

[43] probably the most complete version can be found at http://wiki.dublincore.org/index.php/File:Mapping_schema.org-DC-sort_by_domain.rdf

```
<!-- rdfs:subClassOf (4)-->
<dct:BibliographicResource> rdfs:subClassOf     <schema:CreativeWork> .
<schema:Language>           rdfs:subClassOf     <dct:LinguisticSystem> .
<schema:Organization>       rdfs:subClassOf     <dct:Agent> .
<schema:Person>             rdfs:subClassOf     <dct:Agent> .

<!-- rdfs:subPropertyOf (109) -->
<dct:title>             rdfs:subPropertyOf  <schema:name> .
<schema:actors>         rdfs:subPropertyOf  <dct:contributor> .
<schema:additionalName> rdfs:subPropertyOf  <dct:alternative> .
<schema:branchOf>       rdfs:subPropertyOf  <dct:isPartOf> .
<schema:genre>          rdfs:subPropertyOf  <dct:subject> .
<schema:keywords>       rdfs:subPropertyOf  <dct:subject> .
<schema:mentions>       rdfs:subPropertyOf  <dct:references> .
<schema:musicBy>        rdfs:subPropertyOf  <dct:contributor> .
<schema:partOfTVSeries> rdfs:subPropertyOf  <dct:isPartOf> .
<schema:title>          rdfs:subPropertyOf  <dct:title> .

<!-- owl:equivalentClass (2)-->
<schema:Event>  owl:equivalentClass     <dctype:Event> .
<schema:Place>  owl:equivalentClass     <dct:Location> .

<!-- owl:equivalentProperty (8)-->
<schema:about>              owl:equivalentProperty  <dct:subject> .
<schema:contributor>        owl:equivalentProperty  <dct:contributor> .
<schema:copyrightHolder>    owl:equivalentProperty  <dc:rightsHolder> .
<schema:dateCreated>        owl:equivalentProperty  <dct:created> .
<schema:dateModified>       owl:equivalentProperty  <dct:modified> .
<schema:datePublished>      owl:equivalentProperty  <dct:issued> .
<schema:description>        owl:equivalentProperty  <dct:description> .
<schema:inLanguage>         owl:equivalentProperty  <dct:language> .
```

Listing 13: Example of mapping between Dublin Core and Schema.org

While the mapping goes from Schema.org terms to DC terms, one class, `dct:BibliographicResource`, and one property, `dct:title`, are mapped as respectively `subClass` and `subProperty` of Schema.org terms. This is done to indicate that all the instances in the DC class/property are also instances of the Schema.org class/property, meaning that all bibliographic resources are creative works, and that all titles are names.

One interesting thing that was also done within Europeana is that through the RDFa syntax one could describe a resource with both DC and Schema.org terms in the same markup (see section 3.7.3).

3.7.7 Others

Other mappings and initiatives are briefly described below.

3.7.7.1 BIBO

BIBO is a vocabulary for describing bibliographic resources (books, articles, patents, quotes...). A partial mapping from BIBO to Schema.org terms is available[44]. This looks as work in progress as well, and it is described as "partial", in fact only some classes, and no properties, have been mapped (Schema.org, n.d.-b).

What it is possible to see is that there are a few situations where multiple classes in schema have been mapped to one class in BIBO, e.g. different types of article mapped to `bibo:Article`, or different kinds of webpages to `bibo:WebPage`. This is due to the fact that Schema.org allows more detailed description of certain elements than BIBO.

3.7.7.2 Smithsonian American Art Museum

The Smithsonian American Art Museum's ontology is based on and extends the Europeana Data Model. The curators of the data say they couldn't "find guidance for representing the geographical information in the cultural heritage community" so they chose Schema.org as it is a widely used vocabulary to represent georaphical data as city, state and country. They incorporate Schema.org classes such as `PostalAddress, Country`, and properties such as `address, addressCountry` as subclasses and subproperties of their own classes and properties (Szekely et al., 2014, supplement, p. 154-155).

3.7.7.3 BBC Ontologies

The BBC Curiculum Ontology uses mappings to Schema.org educational vocabulary, an extension to Schema.org by the Learning Resource Metadata Initiative (LRMI)[45], to describe learning resources. Some Curriculum Ontology classes become subclasses of `http://schema.org/AlignmentObject`. This class "describes the alignment between a learning resource and a node in an educational framework" (BBC, 2013).

Another BBC ontology, the already mentioned Programmes Ontology, has established equivalence between 10 of its own and Schema.org classes. The TV and Radio metadata community has also successfully proposed extensions to Schema.org

[44] see http://schema.rdfs.org/mappings/bibo

[45] http://www.lrmi.net/

regarding the description of TV and Radio resources. These extensions can be found in RDFa form[46] and they contain the mappings to Schema.org as well, which were established using the `owl:equivalentClass` property.

3.7.7.4 SIOC

The SIOC initiative (Semantically-Interlinked Online Communities)[47] is a RDF vocabulary for describing online communities and posts (blogs, mailing lists, Wiki pages, etc.). A small mapping from SIOC to Schema.org terms is available[48]. Also in this mapping both RDF Schema's `rdfs:subClassOf` and `rdfs:subPropertyOf` properties and OWL's `owl:equivalentClass` and `owl:equivalentProperty` properties are used, with the same criteria as the previously mentioned mappings.

3.7.7.5 WordNet

WordNet® is a large lexical database of English. A comprehensive mapping from WordNet to schema.org terms is available[49]. Here it is possibe to see how approximately 30 Schema.org classes have been mapped to WordNet synsets through the property `rdfs:subClassOf`.

3.8 Other vocabularies similar to Schema.org

3.8.1 Microformats

Schema.org is not the only vocabulary used to markup HTML in web pages. The most similar initiative, started in 2005, is called Microformats. These are "a set of simple, open data formats built upon existing and widely adopted standards" ("Microformat," 2015). Microformats are "carefully designed (X)HTML class names that extend the semantics of (X)HTML and enable authors to publish higher semantic fidelity content such as people, events, reviews, etc" (Microformats, 2013). This type of metadata uses terms from a fixed set of vocabularies to annotate the HTML

[46] see https://dvcs.w3.org/hg/webschema/raw-file/c73d0e2e7013/schema.org/ext/tvradio.html

[47] http://sioc-project.org/

[48] http://schema.rdfs.org/mappings/sioc

[49] it can be found at http://schema.rdfs.org/mappings/schemaorg_wn.owl

code just like Schema.org and it has partly inspired Schema.org, as stated in section 3.6.1. It is widely used on many websites and it is supported by among others Facebook , Google and Bing for the creation of rich snippets (Schema.org, n.d.-a; Goel et al., 2009b).

The idea behind Microformats was that of bridging a gap where HTML markup wasn't being able to express semantic meaning to tags and values. The format uses mainly the `<class>` tag for class names, `<rel>` for relationships and `<rev>` for reverse relationships. The markup is highly humanly readable, which is clearly stated by their motto "for humans first and machines second" (Microformats, 2013). Similarly to Schema.org, Microformats uses a system of human readable class names with prefixes to describe different types of items ("Microformat," 2015). Wikipedia lists all types of items one can describe using Microformats. According to Wikipedia, only `hCard` and `hCalendar` have been officially ratified by the community, the others remaining as drafts ("Microformat," 2015, § Specific microformats). Despite this, Microformats looks like the most used markup format on the web, according to recent research made by Meusel, Petrovski, and Bizer (2014, § 8). However, the use of Schema.org has been rising since 2012, according to Bizer, Meusel, and Primpeli (2014, § 3.5).

3.8.2 Dublin Core

Dublin Core is a vocabulary of properties for use in resource description. It is maintained by the Dublin Core Metadata Initiative (DCMI). There are two sets of vocabularies: Simple DC that comprises 15 basic elements, and the full set of vocabularies, Qualified DC, that includes "sets of resource classes, vocabulary encoding schemes, and syntax encoding schemes" (Dublin Core Metadata Initiative, 2012).

DC was shaped in 1995, when web searching was inadequate, in order to help raise the standard of metadata on web pages, and was described as the MARC format for the web. Resource discovery is the primary function of the vocabulary, that can be embedded in the `<meta>` and `<link>` tags of the `<head>` in the HTML (Beall, 2004). It can also be used with XML and RDF and RDFa (see listing 10 in section 3.7.3).

Listing 14 shows an example of the standard when embedded in the `<meta>` tag.

```
<meta name="DC.Format" content="video/mpeg; 10 minutes">
<meta name="DC.Language" content="en" >
<meta name="DC.Publisher" content="publisher-name" >
<meta name="DC.Title" content="Title of the song" >
```

Listing 14: Example of Dublin Core

3.8.3 Open Graph protocol

Facebook has developed its own *Open Graph protocol*, which is inspired by Dublin Core, link-rel canonical, Microformats, and RDFa. The Open Graph protocol is a set of metadata properties that can be used as markup in web pages. It is similar to Schema.org regarding the description of items on web pages, but it has different purposes. On the Schema.org's frequently asked questions page, the developers write about Open Graph:

> "Facebook Open Graph serves its purpose well, but it doesn't provide the detailed information search engines need to improve the user experience. A single web page may have many components, and it may talk about more than one thing. If search engines understand the various components of a page, we can improve our presentation of the data. Even if you mark up your content using the Facebook Open Graph protocol, schema.org provides a mechanism for providing more detail about particular entities on the page" (Schema.org, n.d.-a).

The protocol allows for content creators to turn their websites into Facebook Graph objects, customizing their data so that it can be carried over from a non-Facebook website to Facebook when a page is recommended, liked, or shared. Embedding Facebook's metadata into one's page will give a representation of the page on someone's Facebook profile when that someone will like, recommend or share the webpage. Contrary to Schema.org, The Open Graph protocol is embedded in the HTML `<meta>` tag. Every tag has the prefix `og:` plus the name of the property, and a `content` attribute which contains the value of the tag (Open Graph protocol, 2014), like in the following example:

```
<meta property="og:title" content="Facebook's Open Graph protocol"/>
<meta property="og:url" content="http://www.myblog.com/ogp"/>
<meta property="og:site_name" content="Example Blog"/>
<meta property="og:type" content="blog"/>
```

Listing 15: Example of use of Open Graph protocol

3.8.4 Twitter Cards

A very similar concept has been developed by Twitter with its Cards markup. In the same way as Facebook's markup, content creators have the possibility to markup their HTML with the tags provided by Twitter. Content creators can for instance attach rich photos, videos and other media to Tweets that drive traffic to their websites. In this way, users who Tweet links to someone's content will have a Card added to the Tweet that's visible to all of their followers. Developers can choose among a few different types of cards with their own properties (Twitter, n.d.-b). Twitter Cards properties are very similar to the ones belonging to the Open Graph protocol, and actually based on the same conventions and syntax: key-value pairs, each defined in its own HTML meta tag. In fact, Twitter can use already existing Open Graph markup: if someone is already using Open Graph protocol as metadata, it's not necessary to duplicate this data. As explained on the Twitter Developers pages: "[w]hen the Twitter card processor looks for tags on your page, it first checks for the Twitter property, and if not present, falls back to the supported Open Graph property. This allows for both to be defined on the page independently, and minimizes the amount of duplicate markup required" on the page (Twitter, n.d.-a).

3.8.5 Google+

Also Google's social network Google+ uses the same mechanism as the already mentioned examples, as content creators can customize the snippet that is visualized when their own page is shared. This can be accomplished using Schema.org metadata (either in the `<meta>` tags in the `head` and in the `<body>`), which is Google's recommendation. However, the Open Graph protocol properties as well as `<title>` and `<meta name="description" ... />` tags without any other markup are understood and supported by Google (Google, 2014).

3.8.6 Rich Pins

Also the social network Pinterest uses its own metadata, called Rich Pins, for showing snippets inside Pinterest with data coming from external websites. The social network didn't implement its own vocabulary: developers can in fact choose between using Open Graph protocol metadata – in the `<meta>` tags in the `head` – and Schema.org – in the `<body>` – for the implementation of structured data on their websites (Pinterest, 2015, overview page and pages relative to each type).

CHAPTER 4

ANALYSIS AND EVALUATION

This chapter presents the proposed mapping of classes and properties of the TORCH ontology to Schema.org. The mapping is illustrated in tables 4.1 and 4.2. The discussion and the analysis is presented through examples taken from the mapping and the discussion of the challenges encountered. The evaluation will be discussed in section 4.1.

There are two main areas of concern in this analysis. The first one is the direct mapping of the terms, i.e. what classes and properties correspond to each other and how it is possible to find a correspondence when this is not so obvious. The second one is more related to the concrete use of Schema.org and to the evaluation part, i.e. how to express the relationships found in the annotation with Schema.org markup, and how to apply it to a web page and at the same time maintain the relationships established by the annotation so that also the mapping makes sense.

TORCH-Class					mapping Property	Schema.org Class
Thing					owl:equivalentClass	Thing
	Agent				inverse rdfs:subClassOf	Thing > Person + Organization
		Organization			owl:equivalentClass	Thing >Organization
			Broadcaster		rdfs:subClassOf	Thing >Organization
		Person			owl:equivalentClass	Thing >Person
			FictionalCharacter		rdfs:subClassOf	Thing >Person
				MainFictionalCharacter	rdfs:subClassOf	Thing >Person
			RealPerson		rdfs:subClassOf	Thing >Person
	BroadcastedEntity				rdfs:subClassOf	Thing >CreativeWork
		Program			inverse rdfs:subClassOf	Thing >CreativeWork >Episode >TVEpisode
		Series			owl:equivalentClass	Thing >CreativeWork >RadioSeries / TVSeries
	CreativeWork				owl:equivalentClass	Thing >CreativeWork
		Movie			owl:equivalentClass	Thing >CreativeWork >Movie
		MusicalWork			inverse rdfs:subClassOf	Thing >CreativeWork >MusicComposition
			Album		owl:equivalentClass	Thing >CreativeWork >MusicPlaylist >MusicAlbum
			Concert		rdfs:subClassOf	Thing >Event >MusicEvent / MusicComposition
			Tune		owl:equivalentClass	Thing >CreativeWork >MusicRecording
		Newspaper			rdfs:subClassOf	Thing >CreativeWork >Series >Periodical
		PerformingArt			rdfs:subClassOf	Thing >CreativeWork
			Dance		rdfs:subClassOf	Thing >CreativeWork
			Opera		rdfs:subClassOf	Thing >CreativeWork
			Play		rdfs:subClassOf	Thing >CreativeWork
			Skit		rdfs:subClassOf	Thing >CreativeWork
		Text			rdfs:subClassOf	Thing >CreativeWork
			Anthology		rdfs:subClassOf	Thing >CreativeWork
			Comic		rdfs:subClassOf	Thing >CreativeWork
			Non-FictionText		rdfs:subClassOf	Thing >CreativeWork
			Novel		rdfs:subClassOf	Thing >CreativeWork
			Poetry		rdfs:subClassOf	Thing >CreativeWork
			ShortStory		rdfs:subClassOf	Thing >CreativeWork
		Videoclip			rdfs:subClassOf	Thing >CreativeWork >Clip / VideoObject
	Event				owl:equivalentClass	Thing >Event
	Other				n/a	n/a
	Place				owl:equivalentClass	Thing >Place
	Segment				owl:equivalentClass	Thing >CreativeWork >Clip >TVClip / RadioClip
	Service				owl:equivalentClass	Thing >BroadcastService
	TimeSpan				owl:equivalentClass	Thing >Intangible >Quantity >Duration

Table 4.1: TORCH classes and corresponding Schema.org types

TORCH-Property	mapping Property	Schema.org Property (Domain)
topObjectProperty	owl:equivalentProperty	Thing >Intangible >Property
hasContributor	owl:equivalentProperty	Thing >Property >contributor (CreativeWork)
hasAnchor	rdfs:subPropertyOf	Thing >Property >contributor (CreativeWork) / roleName (Role)
hasAnnouncer	rdfs:subPropertyOf	Thing >Property >contributor (CreativeWork) / roleName (Role)
hasEditor	owl:equivalentProperty	Thing >Property >editor (CreativeWork)
hasPuppeteer	rdfs:subPropertyOf	Thing >Property >contributor (CreativeWork) / roleName (Role)
hasReporter	rdfs:subPropertyOf	Thing >Property >contributor (CreativeWork) / roleName (Role)
hasCreator	owl:equivalentProperty	Thing >Property >creator (CreativeWork)
hasAuthor	owl:equivalentProperty	Thing >Property >author (CreativeWork)
hasComposer	owl:equivalentProperty	Thing >Property >composer (CreativeWork)
hasDirector	owl:equivalentProperty	Thing >Property >director (CreativeWork)
hasScriptWriter	rdfs:subPropertyOf	Thing >Property >contributor (CreativeWork) / roleName (Role)
hasEvent	owl:equivalentProperty	Thing >Property >event (Organization / Place)
hasLocation	owl:equivalentProperty	Thing >Property >location (Organization / Event)
hasVenue	rdfs:subPropertyOf	Thing >Property >location (Organization / Event)
hasMember	owl:equivalentProperty	Thing >Property >member (Organization)
hasParticipant	rdfs:subPropertyOf	Thing >Property >contributor (CreativeWork) / roleName (Role)
hasActor	owl:equivalentProperty	Thing >Property >actor (several subtypes of CreativeWork)
hasCharacter	owl:equivalentProperty	Thing >Property >character (CreativeWork)
hasRelatedMusic	n/a	Thing >Property >description (Thing) / about (Thing)
hasRelatedLiveMusic	n/a	Thing >Property >description (Thing) / about (Thing)
hasRelatedRecordedMusic	n/a	Thing >Property >description (Thing) / about (Thing)
hasRelatedText	n/a	Thing >Property >description (Thing) / about (Thing)
hasRelatedLiveText	n/a	Thing >Property >description (Thing) / about (Thing)
hasRelatedRecordedText	n/a	Thing >Property >description (Thing) / about (Thing)
isBasedOnText	n/a	Thing >Property >description (Thing) / about (Thing)
hasSegment	owl:equivalentProperty	Thing >Property >hasPart (CreativeWork)
hasTimeSpan	owl:equivalentProperty	Thing >Property >duration (several subtypes of CreativeWork)
hasTune	owl:equivalentProperty	Thing >Property >track (MusicGroup / MusicPlaylist)
interviews	rdfs:subPropertyOf	Thing >Property >contributor (CreativeWork) / roleName (Role)
isRelatedTo	rdfs:subPropertyOf	Thing >Property >description (Thing) / about (Thing)
performs	owl:equivalentProperty	Thing >Property >performerIn (Person) / roleName (Role)
narrates	rdfs:subPropertyOf	Thing >Property >performerIn (Person) / roleName (Role)
playsIn	rdfs:subPropertyOf	Thing >Property >performerIn (Person) / roleName (Role)
playsMusic	rdfs:subPropertyOf	Thing >Property >performerIn (Person) / roleName (Role)
playsRole	rdfs:subPropertyOf	Thing >Property >performerIn (Person) / roleName (Role)

Table 4.2: TORCH properties and corresponding Schema.org properties

As also discussed in sections 3.5 and 3.6, the size and data structure of the two ontologies is different, so this is causing some challenges in trying to map the two data models.

One challenge is that the annotation process of the TORCH project is meant to define relations between classes that have been annotated, e.g. Obama is related to USA. However, this representation of information is not the purpose of Schema.org, which doesn't have so many properties to express such relations. Schema.org's purpose is that of describing web resources. However, there are some relationships in the annotation that can easily be described with Schema.org, as it will be showed below.

As mentioned, some TORCH terms have easy and obvious equivalence to Schema.org terms, i.e. they map through the property `owl:equivalentClass` and `owl:equivalentProperty`. The classes that map most easily are those that have the same name and meaning in the two schemes. Even if there is no textual descriptions for terms in the TORCH ontology, their names are expressive and human readable, so for a human it is not difficult to understand their meaning either, and to conclude that their meaning is the same in Schema.org. Here, on the pages of each element it is possible to check the description of the terms, and what they should be used for, i.e. get information about their meaning. E.g. classes such as `Organization, Person, CreativeWork, Movie, Event` and properties such as `hasContributor, hasCreator, hasMember` are easy to map to corresponding Schema.org terms.

These equivalent types and properties have been mapped from the TORCH ontology to Schema.org using OWL's property `owl:equivalentClass`[1] and `owl:equivalentProperty`[2], e.g. as it is done in several of the mentioned mappings in section 3.7.

However it is important to add that, as stated on the *OWL Web Ontology Language Reference*, " the use of owl:equivalentClass does not imply class equality. Class equality means that the classes have the same intensional meaning (denote the same concept)." In the example `USPresident equivalentClass PrincipalResidentOfWhiteHouse`, "the concept of "President of the US" is related to, but not equal to the concept of the principal resident of a certain estate. Real class equality can only be expressed with the `owl:sameAs` construct. As this requires treating classes as individuals, class equality can only be expressed in OWL Full"

[1] http://www.w3.org/TR/owl-ref/#equivalentClass-def

[2] http://www.w3.org/TR/owl-ref/#equivalentProperty-def

(Dean et al., 2004, § 3.2.2).

The equivalence of two classes made through the `owl:equivalentClass` was declared in the Protégé ontology editor software. The RDF/XML export is showed in listing 16.

```
<owl:Class rdf:about="http://torch.no#Person">
    <owl:equivalentClass rdf:resource="http://www.schema.org/Person"/>
    <rdfs:subClassOf rdf:resource="http://torch.no#Agent"/>
    </owl:Class>
```

Listing 16: Example of declaration of `equivalentClass` between the `Person` classes, in RDF/XML syntax

Some classes and properties have been mapped using RDF Schema's property `rdfs:subClassOf`[3] and `rdfs:subPropertyOf`[4], e.g. as it is done in several of the mentioned mappings in section 3.7. In the case of classes this means that all instances of a class are also instances of the other class, e.g. all instances of **FictionalCharacter** are also instances of **Person**. In the case of properties, `rdfs:subPropertyOf` means that all instances contained in the property extension of the property `hasReporter` are also members of the property extension of the property `hasContributor`, etc.

There are also some classes where it is indicated `inverse rdf:subClassOf`. This means that, as they did in the mappng with Dublin Core, the mapping goes from Schema.org to the TORCH ontology. E.g. a Schema.org `Person` is a subclass of TORCH `Agent`.

Some other challenges arise when the domain differences of the two ontologies become clear. E.g. many classes that are describing things pertaining to the domain of cultural heritage in TORCH don't have any other better mapping than `CreativeWork` in Schema.org. In this case it might be useful to use extensions such as Bibliograph.net or others, which may contain some, if not all, of these missing Schema.org classes. The same happens with the TORCH properties `hasRelatedMusic` and `hasRelatedText` properties. It is not possible to express such relationships in Schema.org. A solution, although probably poor, was to describe this properties in a textual manner using the `description` or `about` properties in Schema.org, as if it was a note. However this would probably result in misinterpretation by the search engines. A better solution might be importing a property from

[3] http://www.w3.org/TR/owl-ref/#subClassOf-def

[4] http://www.w3.org/TR/owl-ref/#subPropertyOf-def

another vocabulary or using an extension.

A class in the TORCH ontology doesn't necessarily map to a type in Schema.org. That is the case with `FictionalCharacter`. The most correct mapping that was found was just with `https://schema.org/Person`, but that results in a loss of expressivity. How is it possible to establish if an instance of person is a real person or a fictional character? However, in Schema.org there is the property `character`, which may be used to express the concept of fictional character. This property has a `CreativeWork` domain, and expects a `Person` value.

For the TORCH class `Other` a satisfying mapping with Schema.org wasn't found.

4.1 Evaluation

The original method that was planned in order to conduct the evaluation was a transformation method where a dataset of SIFT-records from NRK, whose entities populated the TORCH ontology, would have been transformed into a second dataset, utilizing the mapping to Schema.org. Then the resulting dataset would have been retransformed into TORCH. The purpose of this was to see how the data would transform and how much data would have been lost and why. However, this would require working with relatively big datasets and an automatic infrastructure built with a certain knowledge of a programming language, such as Python, PHP, Ruby, or Java. There wasn't the time nor the technical knowledge to try such a testing method, so the idea was abandoned. Later, a more detailed analysis of the two ontologies revealed that they have quite different structures and they are used in different contexts and with different purposes, so it may be that this method wouldn't have worked at all, or that too much data would have been lost in the transformation process. However, a thing to consider is that working with bigger datasets and automatic mappings could have brought some measurable results, even if poor. This remains an alternative for later tests, maybe at a later stage of the TORCH project, although the details of the process have yet to be seen.

The second method wasn't just considered, but also tried out. It consisted in preparing a few versions of files containing the same SIFT-record from NRK, one version with and one version without Schema.org markup. These files were marked up with Schema.org, following the mapping from the classes and properties contained in the source ontology. The purpose of this was to use a service by Google, called Rich Snippets Testing Tool. This enabled, just by parsing the HTML code of a web page, to display a preview of how the page would look like as an item in a Google search result[5]. However, this tool was discontinued and upgraded to another service, the Structured Data Testing Tool[6]. Using this service it is possible to parse the HTML source of a page, in order to test the markup, but it is no longer possible to see the preview of the search results.

Because of this, another method was considered. The same files were put online on a personal domain. A search page that used Google Custom Search[7], limited to just the part of the website containing the test files, was set up. However, the pages haven't been indexed by Google. Custom Search needs Google to index the pages just as it does for any pages on the internet and doesn't have any priority on its normal indexing process. So the Custom Search engine is not showing any result and it's difficult to say if and when any result will be displayed. The purpose of this was to see if Google would utilize some of the Schema.org markup produced with the mapping to populate its rich snippets, thus showing some of the data from the markup in search results, and possibly some differences with current search results, e.g. from the NRK website as it is today. However, this wouldn't probably have guaranteed to see relevant differences in the search results, because the type TVEpisode shows only rich snippets for reviews in Google search results.

Eventually, the last method considered and the one that ultimately was followed, was just showing with a concrete example how Schema.org markup can be used to mark up a web page, while taking the mapping into account. Namely, the chosen web page is the one on NRK that shows the text that is taken as example for the annotation. The text is the transcription of the 19 February 2015 episode of the TV news program Urix. The backbone of the HTML code was taken from the program page on NRK's website[8] and cleaned up just to produce a minimal working example, that would show the relevant textual information on the website. This is basically the information that shows under “Programinformasjon” and “Teksting” when clicking on them. The latter contains the SIFT-record that is the object of the annotation.

Before choosing to describe the actual NRK web page, there were also some considerations about what kind of type would suit best the description of the page.

[5] see here for some details and screenshots of how the tool looked like: http://googlewebmastercentral.blogspot.sk/2012/04/updates-to-rich-snippets.html. It was last indexed by the Internet Archive some time in february: http://tinyurl.com/qhga57d

[6] https://developers.google.com/structured-data/testing-tool/

[7] https://cse.google.com/cse/all

[8] http://tv.nrk.no/serie/urix/NNFA53021915/19-02-2015

Initially, when just the description of the annotated SITF-text was considered as object for the Schema.org mark up, a `NewsArticle` was considered the best type to describe it. However, in a real world context, the text is not appearing separately from the rest of the information on the web page. So, giving the text a context was one decisive motivation towards taking information from the actual web page. Then it became clear that the object of the description should be a `TVEpisode` type.

However it is also possible to use the property `mainEntityOfPage` to indicate a page (or other `CreativeWork`) for which this thing is the main entity being described. This is not the case however, because the `TVEpisode` is the main entity that is about some other entities (`Person, Organization, Place` etc).

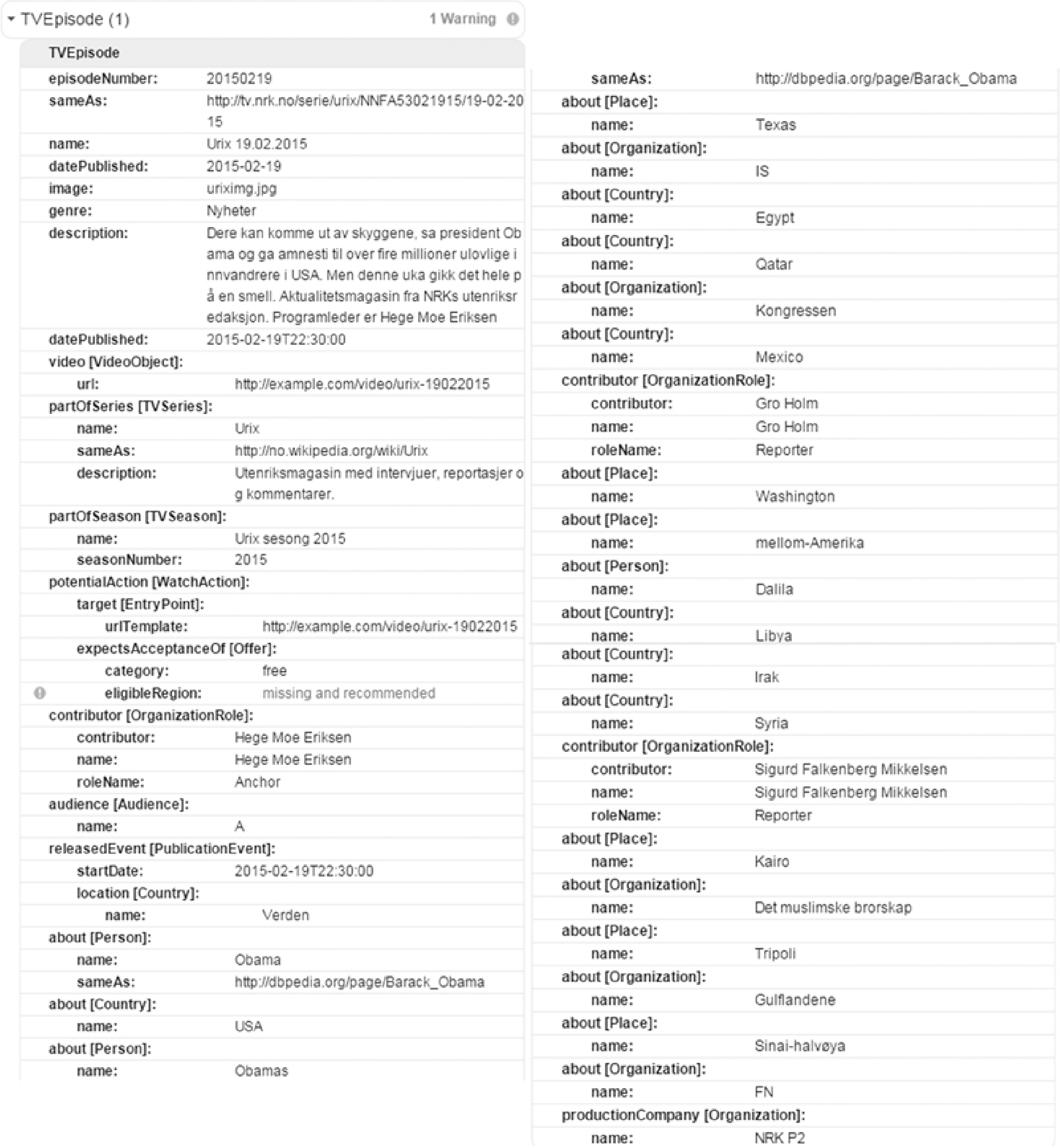
The already mentioned Structured Data Testing Tool by Google was chosen to validate the code, and at the same time to show how a search engine interprets the code. An advantage of using this is also that one can be relatively sure to follow the specification in a correct way. The syntax that was chosen to represent Schema.org was RDFa, because it is little easier to implement and well suited to show the triples that Schema.org produces.

A challenge encountered during the coding process was that it was necessary to add some information that wasn't actually displayed on the page, but that was required by Schema.org in order for the page to be validated. For example, the properties `episodeNumber`, `partOfSeason`, `partOfSeries` ans `potentialAction` were indicated as missing and required by the validator.

Another issue that manifested itself while marking up the HTML code, was that it sometimes was difficult to follow the original HTML when nesting the different Schema.org attributes. This means that it may be difficult to implement Schema.org on an existing HTML structure, and this difficulty can be greater when the code is complex and contains scripts and dynamic content. This issue is well expressed by Hepp (2014) when he states that, using RDFa, "reusing the exact same visible elements for structured data markup [. . .] was beneficial because it reduces redundancy, it also raises complexity for developers, since you violate the principle of "separation of concerns" – you have to align a given HTML tree structure with a given data structure, dictated by the vocabulary, like schema.org". A solution to this issue is the adoption of the JSON-LD syntax, which separates the data structuring elements from the visual elements of the page.

Also Coombs (2015b) expresses the fact that generating RDFa can be difficult. "If you are encoding data that doesn't already exist in a graph form, the process of creating RDFa is similar but inherently more difficult. In these cases several steps need to happen. First, you need to determine what entities you are encoding. Second you need to decide what statements you want to make about the entities and their relationships. You can use an existing vocabulary like Schema.org for your entities and statements, but you still have to figure out what you want to say".

The code is validated, as showed in fig. 4.1, except for a warning concerning the property `eligibleRegion`. It wasn't possible to find a solution to fix this warning, which may also show due to a bug in the validator.



```
▾ TVEpisode (1)                                   1 Warning ⓘ
  TVEpisode
  episodeNumber:          20150219
  sameAs:                 http://tv.nrk.no/serie/urix/NNFA53021915/19-02-20
                          15
  name:                   Urix 19.02.2015
  datePublished:          2015-02-19
  image:                  uriximg.jpg
  genre:                  Nyheter
  description:            Dere kan komme ut av skyggene, sa president Ob
                          ama og ga amnesti til over fire millioner ulovlige i
                          nnvandrere i USA. Men denne uka gikk det hele p
                          å en smell. Aktualitetsmagasin fra NRKs utenriksr
                          edaksjon. Programleder er Hege Moe Eriksen
  datePublished:          2015-02-19T22:30:00
  video [VideoObject]:
      url:                    http://example.com/video/urix-19022015
  partOfSeries [TVSeries]:
      name:                   Urix
      sameAs:                 http://no.wikipedia.org/wiki/Urix
      description:            Utenriksmagasin med intervjuer, reportasjer o
                              g kommentarer.
  partOfSeason [TVSeason]:
      name:                   Urix sesong 2015
      seasonNumber:           2015
  potentialAction [WatchAction]:
      target [EntryPoint]:
          urlTemplate:            http://example.com/video/urix-19022015
      expectsAcceptanceOf [Offer]:
          category:               free
ⓘ         eligibleRegion:         missing and recommended
  contributor [OrganizationRole]:
      contributor:            Hege Moe Eriksen
      name:                   Hege Moe Eriksen
      roleName:               Anchor
  audience [Audience]:
      name:                   A
  releasedEvent [PublicationEvent]:
      startDate:              2015-02-19T22:30:00
      location [Country]:
          name:                   Verden
  about [Person]:
      name:                   Obama
      sameAs:                 http://dbpedia.org/page/Barack_Obama
  about [Country]:
      name:                   USA
  about [Person]:
      name:                   Obamas
      sameAs:                 http://dbpedia.org/page/Barack_Obama
  about [Place]:
      name:                   Texas
  about [Organization]:
      name:                   IS
  about [Country]:
      name:                   Egypt
  about [Country]:
      name:                   Qatar
  about [Organization]:
      name:                   Kongressen
  about [Country]:
      name:                   Mexico
  contributor [OrganizationRole]:
      contributor:            Gro Holm
      name:                   Gro Holm
      roleName:               Reporter
  about [Place]:
      name:                   Washington
  about [Place]:
      name:                   mellom-Amerika
  about [Person]:
      name:                   Dalila
  about [Country]:
      name:                   Libya
  about [Country]:
      name:                   Irak
  about [Country]:
      name:                   Syria
  contributor [OrganizationRole]:
      contributor:            Sigurd Falkenberg Mikkelsen
      name:                   Sigurd Falkenberg Mikkelsen
      roleName:               Reporter
  about [Place]:
      name:                   Kairo
  about [Organization]:
      name:                   Det muslimske brorskap
  about [Place]:
      name:                   Tripoli
  about [Organization]:
      name:                   Gulflandene
  about [Place]:
      name:                   Sinai-halvøya
  about [Organization]:
      name:                   FN
  productionCompany [Organization]:
      name:                   NRK P2
```

Figure 4.1: Validation of the markup as showed on Structured Data Testing Tool

Figure 4.1 shows all the relationships that could be described through the use of

Schema.org, using the information on the NRK web page and taking the mapping into account. The page is described as a `TVEpisode`.

Confronting the results of the validator with the annotation, it is possible to see how many relationships it was possible to capture with Schema.org. It must be pointed out that the vast majority of relationships in the annotation are `isRelatedTo` and `coRefers`, that are difficult to express with Schema.org.

The annotation properties that were possible to express were those such as `hasReporter` and the properties that indicate that a certain entity belongs to a class, such as "IS" belongs to `Organization`, etc. The solution concerning the former, which was individuated in the mapping (see table 4.2), was the use of two properties. The first one indicates that the TVEpisode has a kind of `contributor`, and the second that indicates that the contributor has a `roleName`, which has been defined as "Reporter". The other kinds of properties were indicated e.g. by stating that a certain individual, such as "Obama" is a `typeOf Person`, "Washington" a `typeOf Place`, "IS" a `typeOf Organization`, and so on.

An attempt to express some kind of disambiguation was also made. This was tried with the terms "Obama" and "Obamas", by showing that both entities refer to the same DBpedia resource.

Besides the Structured Data Testing Tool, there are also other services that can parse and extract data from HTML documents marked up with RDFa, so that it may be easier to understand the relationships one can express with Schema.org. E.g. a partial graphical visualization made using the RDFa Play[9] tool is showed in fig. 4.2.

Feeding the entire page's HTML code to another tool, W3C's RDFa 1.1 Distiller and Parser[10], it is possible to realize that this small markup effort has produced 120 triples. This gives an idea of the amount of data that large institutions may produce, and how much data search engines could use to enhance their users' search experiences.

[9] http://rdfa.info/play/

[10] http://www.w3.org/2012/pyRdfa/

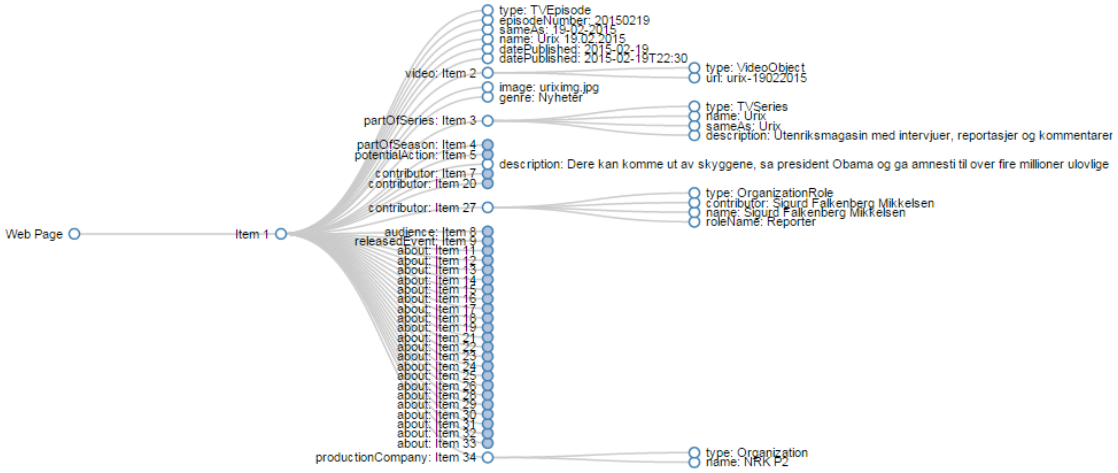


Figure 4.2: Visualization of some of the RDFa relationships using the tool RDFa Play

CHAPTER 5

CONCLUSION

Schema.org provides a simple way for cultural heritage institutions to expose their data in a structured form. This can make their resources more discoverable and useful. It seems also that things are changing and Schema.org is getting a more and more important role in the strategies of search engines as well as the Semantic Web in general. This could be a huge chance for cultural heritage organizations to expose their data, and find new ways to use the newly available structured data. The technologies are quickly getting standardized, and many things can already be done. The focus of this thesis has been the proposal of a mapping from the TORCH ontology to Schema.orgṪhe mapping is feasible and relationships between the two ontologies can be established with a series of existing properties. This process presents some challenges, in that the data structures of the ontologies are different, and one must also consider the differences in breadth and purpose. TORCH is a domain ontology, while Schema.org is a general purpose ontology.

Currently it is difficult to find ways to evaluate mappings involving Schema.org, because not all types are supported to show enriched search results, and search engines are changing their behaviour concerning rich snippets. There wasn't either a good tool to test possible results of a markup, nor the dataset tested was large enough to obtain some measurable results. This was a problem in this work, because it is not possible to test how search results with data from NRK enriched with Schema.org will be displayed. However, using the mapping and applying it to a use case situation, such as the description of a web page from NRK, it was possible to describe many relations that were defined in the mapping, and sometimes even more relations.

This may just demonstrate the potential of semantic markup if large data-centric cultural heritage organizations will start to use it in a more consistent way. Taking advantage of all the possibilities this gives, search engines may begin to enrich their search results with their data in new ways. An example of this is WorldCat, that made some big moves towards a large-scale implementation of Schema.org for structuring its own data. However, to date, search engines don't seem to have found an effective way to make use of this data. Maybe more such large-scale initiatives are required to make the focus shift towards more visibility to the cultural heritage and related communities.

5.1 Further work

Further work could be conducted in trying to find automatic mappings between the TORCH ontology and Schema.org. Also, an even deeper analysis of existing mappings and implementations involving Schema.org could give some input for further developments. Besides, keeping in touch and even getting involved in initiatives by proponents of bibliographic extensions, like Schema Bib Extend or Bibliograph.net, could bring advantages. Moreover, trying to implement a framework that would allow to exploit the data produced with semantic markup could enhance data reuse also locally.

BIBLIOGRAPHY

Aberer, K., Cudré-Mauroux, P., Ouksel, A. M., Catarci, T., Hacid, M.-S., Illarramendi, A., . . . Neuhold, E. J. (2004). Emergent semantics principles and issues. In *Database systems for advanced applications* (pp. 25–38). Springer. Retrieved May 23, 2015, from http://link.springer.com/chapter/10.1007/978-3-540-24571-1_2

Ahmet Uyar & Farouk Musa Aliyu. (2015, April 2). Evaluating search features of google knowledge graph and bing satori. *Online Information Review*, *39*(2), 197–213. doi:10.1108/OIR-10-2014-0257

Barbera, M. (2013). Linked (open) data at web scale: research, social and engineering challenges in the digital humanities. *JLIS. it*, *4*(1), 91. Retrieved March 29, 2015, from http://leo.cineca.it/index.php/jlis/article/download/6333/7868

Barker, P. (2012, August 16). Where does schema.org fit in the (semantic) web? [Sharing and learning]. Retrieved June 10, 2015, from http://blogs.pjjk.net/phil/where-does-schemaorg-fit-in-the-semantic-web/

Barker, P. & Campbell, L. M. (2014, June). *What is schema.org?* LRMI. Retrieved April 21, 2015, from http://publications.cetis.ac.uk/wp-content/uploads/2014/06/schemaBriefing.pdf

BBC. (2013, October 11). Curriculum ontology [BBC]. Retrieved March 22, 2015, from http://www.bbc.co.uk/ontologies/curriculum

Beall, J. (2004). Dublin core: an obituary. *Library Hi Tech News*, *21*(8), 40–41. Retrieved June 10, 2015, from http://www.emeraldinsight.com/doi/pdf/10.1108/07419050410567399

Bergman, M. & Giasson, F. [Frédérick]. (2015a, April 20). UMBEL annex f: mapping with UMBEL [UMBEL wiki]. Retrieved June 7, 2015, from http://techwiki.umbel.org/index.php/UMBEL_-_Annex_F#LINKING_EXTERNAL_CLASSES_.28ONTOLOGIES.29

Bergman, M. & Giasson, F. [Frédérick]. (2015b, April 20). UMBEL annex i: schema.org mapping [UMBEL wiki]. Retrieved June 7, 2015, from http://techwiki.umbel.org/index.php/UMBEL_-_Annex_I

Bergman, M. & Giasson, F. [Frédérick]. (2015c, April 20). UMBEL specification [UMBEL wiki]. Retrieved June 7, 2015, from http://techwiki.umbel.org/index.php/UMBEL_Specification#Introduction_to_UMBEL

Berners-Lee, T. (1997). Axioms of web architecture: metadata. Retrieved March 30, 2015, from http://www.w3.org/DesignIssues/Metadata.html

Berners-Lee, T. (1998). What the semantic web can represent. Retrieved March 21, 2015, from http://www.w3.org/DesignIssues/RDFnot.html

Berners-Lee, T. (2006, July 27). Linked data [Design issues]. Retrieved May 20, 2015, from http://www.w3.org/DesignIssues/LinkedData.html

Berners-Lee, T. (2007, November 21). Giant global graph. Retrieved May 20, 2015, from http://dig.csail.mit.edu/breadcrumbs/node/215

Berners-Lee, T., Hendler, J., Lassila, O., et al. (2001). The semantic web. *Scientific american*, *284*(5), 28–37. Retrieved March 20, 2015, from http://www.cs.umd.edu/~golbeck/LBSC690/SemanticWeb.html

BiblioGraph. (n.d.). BiblioGraph.net core schema [BiblioGraph.net]. Retrieved May 21, 2015, from http://bibliograph.net/docs/bibliograph_net_rdfa.html

BiblioGraph. (2015). What is BiblioGraph.net? Retrieved May 17, 2015, from http://bibliograph.net/

Bing. (n.d.). Marking up your site with structured data. Retrieved April 28, 2015, from http://www.bing.com/webmaster/help/marking-up-your-site-with-structured-data-3a93e731

Bizer, C., Eckert, K., Meusel, R., Mühleisen, H., Schuhmacher, M., & Völker, J. (2013). Deployment of rdfa, microdata, and microformats on the web–a quantitative analysis. In *The semantic web–ISWC 2013* (pp. 17–32). Springer. Retrieved June 10, 2015, from http://link.springer.com/chapter/10.1007/978-3-642-41338-4_2

Bizer, C., Meusel, R., & Primpeli, A. (2014). *Web data commons RDFa, microdata, and microformat data sets*. Retrieved May 11, 2015, from http://webdatacommons.org/structureddata/index.html

Brickley, D. (2012, May 11). Schema blog: schema.org markup for external lists [Schema blog]. Retrieved May 28, 2015, from http://blog.schema.org/2012/05/schemaorg-markup-for-external-lists.html

Brickley, D. (2013, June 3). Schema.org and JSON-LD [Schema blog]. Retrieved May 16, 2015, from http://blog.schema.org/2013/06/schemaorg-and-json-ld.html

Brickley, D. & Guha, R. (2014, February 25). RDF schema 1.1 [W3c]. Retrieved June 7, 2015, from http://www.w3.org/TR/rdf-schema/

Coombs, K. (n.d.). WorldCat linked data vocabulary [OCLC developers network]. Retrieved June 7, 2015, from http://www.oclc.org/developer/develop/linked-data/worldcat-vocabulary.en.html

Coombs, K. (2015a, March 25). Learning linked data: finding and consuming RDFa | OCLC developer network. Retrieved May 27, 2015, from http://www.oclc.org/developer/news/2015/learning-linked-data-finding-and-consuming-rdfa.en.html

Coombs, K. (2015b, March 16). Learning linked data: making your data harvestable via RDFa [OCLC developer network]. Retrieved June 7, 2015, from http://www.oclc.org/developer/news/2015/learning-linked-data-making-your-data-harvestable-via-rdfa.en.html

DCMI Schema.org Alignment Task Group. (2014, December 15). Schema.org alignment [Dublin core metadata initiative wiki]. Retrieved May 27, 2015, from http://wiki.dublincore.org/index.php/Schema.org_Alignment

Dean, M., Schreiber, G., Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., . . . Stein, L. A. (2004). *OWL web ontology language reference.* Retrieved June 10, 2015, from http://www.w3.org/TR/owl-ref/

Dublin Core Metadata Initiative. (2012, June 14). Dublin core metadata element set, version 1.1 [Dublin core metadata initiative]. Retrieved March 22, 2015, from http://dublincore.org/documents/dces/

DuCharme, B. (2011). *Learning sparql.* Sebastopol: O'Reilly.

Europeana. (n.d.-a). Data structure [Europeana labs]. Retrieved April 10, 2015, from http://labs.europeana.eu/api/linked-open-data/data-structure/

Europeana. (n.d.-b). Europeana strategy 2015-2020. Retrieved April 9, 2015, from http://pro.europeana.eu/files/Europeana_Professional/Publications/Europeana%20Strategy%202020.pdf

Fleischhacker, D., Bryl, V., & Bizer, C. (2014, September 9). DBpedia version 2014 released [DBpedia wiki]. Retrieved May 15, 2015, from http://wiki.dbpedia.org/news/dbpedia-version-2014-released

Franklin, W. & Anselimo, P. I. (2011). Information technology research in developing nations: major research methods and publication outlets. *International Journal of Information and Communication Technology Research*, *1*(6), 253–257. Retrieved June 10, 2015, from http://www.dl.is.vnu.edu.vn/dspace/handle/123456789/645

Freebase. (2015, March 26). As previously announced [1], freebase will be going read-only next tuesday... [Freebase google+ page]. Retrieved May 24, 2015, from https://plus.google.com/109936836907132434202/posts/3aYFVNf92A1

García-Silva, A., García-Castro, L. J., García, A., & Corcho, O. (2015, April 1). Building domain ontologies out of folksonomies and linked data. *International Journal on Artificial Intelligence Tools*, *24*(02), 1540014. doi:10.1142/S021821301540014X

Giasson, F. [Frederick]. (2014, April 24). Schema.org: forcing the emergence of a new web paradigm [Frederick giasson's weblog]. Retrieved June 9, 2015, from http://fgiasson.com/blog/index.php/2014/04/24/schema-org-forcing-the-emergence-of-a-new-web-paradigm/

Godby, C. J. & Denenberg, R. (2015). Common ground: exploring compatibilities between the linked data models of the library of congress and OCLC. Retrieved June 7, 2015, from http://www.wip.oclc.org/content/dam/research/publications/2015/oclcresearch-loc-linked-data-2015.pdf

Goel, K., Gupta, P., & Hansson, O. (2009a, October 26). Help us make the web better: an update on rich snippets [Official google webmaster central blog]. Retrieved March 22, 2015, from http://googlewebmastercentral.blogspot.com/2009/10/help-us-make-web-better-update-on-rich.html

Goel, K., Gupta, P., & Hansson, O. (2009b, May 12). Introducing rich snippets [Official google webmaster central blog]. Retrieved March 22, 2015, from http://googlewebmastercentral.blogspot.com/2009/05/introducing-rich-snippets.html

Google. (n.d.). Crawling & indexing [Google inside search]. Retrieved April 23, 2015, from http://www.google.com/search/about/insidesearch/howsearchworks/crawling-indexing.html

Google. (2014, June 11). Snippet [Google developers]. Retrieved May 3, 2015, from https://developers.google.com/+/web/snippet/

Google. (2015a, February 12). Rich snippets [Google developers]. Retrieved April 23, 2015, from https://developers.google.com/structured-data/rich-snippets/

Google. (2015b, March 13). Structured data policies [Google developers]. Retrieved April 21, 2015, from `https://developers.google.com/structured-data/policies?rd=1`

(2012, October 10). *Does the use of schema.org markup create a ranking benefit?* Retrieved June 11, 2015, from `https://www.youtube.com/watch?v=OolDzztYwtQ`

Gruber, T. R. (1993, June). A translation approach to portable ontology specifications. *Knowledge Acquisition*, *5*(2), 199–220. doi:`10.1006/knac.1993.1008`

Guha, R. (2011). Introducing schema.org: search engines come together for a richer web [Official google blog]. Retrieved March 20, 2015, from `http://googleblog.blogspot.com/2011/06/introducing-schemaorg-search-engines.html`

Guha, R. (2015, May 13). Schema.org 2.0 [Schema blog]. Retrieved May 19, 2015, from `http://blog.schema.org/2015/05/schema.html`

Hepp, M. (2014, March 24). JSON-LD: finally, google honors invisible data for SEO [The hepp research blog on marketing with data]. Retrieved June 10, 2015, from `http://blog.heppresearch.com/2014/03/24/json-ld-finally-google-honors-invisible-data-for-seo/`

Isaac, A. & Baker, T. (2015). Linked data practice at different levels of semantic precision. *Bulletin of the Association for Information Science and Technology*, *41*(4), 34–39. Retrieved May 21, 2015, from `https://www.asis.org/Bulletin/Apr-15/Bulletin_AprMay2015.pdf`

Isaac, A., Király, P., Kats, P., Mamakis, Y., Markakis, G., & Charles, V. (2013, November 27). *Semantic web technology in europeana.* SWIB13, Hamburg. Retrieved April 10, 2015, from `https://prezi.com/0zv1sd8xxcdp/semantic-web-technology-in-europeana/`

Jeffrey Beall. (2014, October 28). Dublin core is still dead. *Library Hi Tech News*, *31*(9), 11–13. doi:`10.1108/LHTN-07-2014-0058`

Kenning Arlitsch & Patrick S. O'Brien. (2012, March 2). Invisible institutional repositories. *Library Hi Tech*, *30*(1), 60–81. doi:`10.1108/07378831211213210`

Khalili, A. & Auer, S. (2013). WYSIWYM authoring of structured content based on schema.org. In X. Lin, Y. Manolopoulos, D. Srivastava, & G. Huang (Eds.), *Web information systems engineering – WISE 2013* (Vol. 8181, pp. 425–438). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved April 30, 2015, from `http://link.springer.com/10.1007/978-3-642-41154-0_32`

Király, P. (2013). *Schema.org in libraries.* Retrieved April 10, 2015, from `https://prezi.com/x9y_60qtmdo-/schemaorg-in-libraries/`

Laniado, D., Eynard, D., & Colombetti, M. (2007). Using WordNet to turn a folksonomy into a hierarchy of concepts. In *Semantic web application and perspectives-fourth italian semantic web workshop* (pp. 192–201). Citeseer. Retrieved June 1, 2014, from `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.94.4393%5C&rep=rep1%5C&type=pdf#page=200`

Legg, C. (2007, January 1). Ontologies on the semantic web. *Annual Review of Information Science and Technology*, *41*(1), 407–451. doi:`10.1002/aris.2007.1440410116`

Legg, C. (2013). Peirce, meaning, and the semantic web. *Semiotica*, *2013*(193), 119–143.

Lin, H. & Davis, J. (2010). Computational and crowdsourcing methods for extracting ontological structure from folksonomy. In *The semantic web: research and applications* (pp. 472–477). Springer. Retrieved May 31, 2014, from `http://link.springer.com/chapter/10.1007/978-3-642-13489-0_46`

Manola, F., Miller, E., & McBride, B. (2014, June 24). RDF 1.1 primer [W3c]. Retrieved May 20, 2015, from `http://www.w3.org/TR/2014/NOTE-rdf11-primer-20140624/`

Mao, M. (2007). *Ontology mapping: an information retrieval and interactive activation network based approach.* Springer. Retrieved June 10, 2015, from `http://link.springer.com/chapter/10.1007/978-3-540-76298-0_72`

Meusel, R., Petrovski, P., & Bizer, C. (2014). The WebDataCommons microdata, RDFa and microformat dataset series. In *The semantic web–ISWC 2014* (pp. 277–292). Springer. Retrieved May 11, 2015, from `http://link.springer.com/chapter/10.1007/978-3-319-11964-9_18`

Microformats. (2013, September 22). What are microformats? [Microformats wiki]. Retrieved May 4, 2015, from `http://microformats.org/wiki/what-are-microformats`

Mika, P. (2005). Ontologies are us: a unified model of social networks and semantics. In *The semantic web–ISWC 2005* (pp. 522–536). Springer. Retrieved May 23, 2015, from `http://link.springer.com/chapter/10.1007/11574620_38`

Net Market Share. (2015). Mobile/tablet search engine market share [Net market share]. Retrieved May 3, 2015, from `http://marketshare.hitslink.com/search-engine-market-share.aspx?qprid=4%5C&qpcustom=%5C&qpcustomb=%5C&qptimeframe=Y%5C&qpcustomd=1`

NISO. (2004). *Understanding metadata.* Bethesda: NISO Press.

Ontology Matching. (2015, May 29). Welcome to ontology matching. Retrieved March 30, 2015, from `http://www.ontologymatching.org/index.html`

Open Graph protocol. (2014, October 20). Open graph protocol. Retrieved May 1, 2015, from http://ogp.me/

UMBEL. (2014, December 9), In *Wikipedia, the free encyclopedia.* Page Version ID: 637278794. Retrieved June 7, 2015, from http://en.wikipedia.org/w/index.php?title=UMBEL%5C&oldid=637278794

Europeana. (2015, March 31), In *Wikipedia, the free encyclopedia.* Page Version ID: 654315805. Retrieved April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Europeana%5C&oldid=654315805

Knowledge Graph. (2015, April 15), In *Wikipedia, the free encyclopedia.* Page Version ID: 656551762. Retrieved May 24, 2015, from http://en.wikipedia.org/w/index.php?title=Knowledge_Graph%5C&oldid=656551762

Microformat. (2015, February 21), In *Wikipedia, the free encyclopedia.* Page Version ID: 648222359. Retrieved May 2, 2015, from http://en.wikipedia.org/w/index.php?title=Microformat%5C&oldid=648222359

Semantic Web. (2015, May 2), In *Wikipedia, the free encyclopedia.* Page Version ID: 660491845. Retrieved May 21, 2015, from http://en.wikipedia.org/w/index.php?title=Semantic_Web%5C&oldid=660491845

SPARQL. (2015, May 12), In *Wikipedia, the free encyclopedia.* Page Version ID: 661970711. Retrieved May 22, 2015, from http://en.wikipedia.org/w/index.php?title=SPARQL%5C&oldid=661970711

Web search engine. (2015, May 22), In *Wikipedia, the free encyclopedia.* Page Version ID: 663535276. Retrieved May 23, 2015, from http://en.wikipedia.org/w/index.php?title=Web_search_engine%5C&oldid=663535276

Pinterest. (2015). Rich pins overview [Pinterest]. Retrieved May 3, 2015, from https://developers.pinterest.com/rich_pins_overview/

Pretorius, A. J. (2004). Ontologies-introduction and overview. *adapted from: Pretorius, AJ, Lexon Visualisation: Visualising Binary Fact Types in Ontology Bases, MSc Thesis, Vrije Universiteit Brussel.* Retrieved March 22, 2015, from https://www.starlab.vub.ac.be/teaching/Ontologies_Intr_Overv.pdf

Purohit, S. & Harrison, M. (2014, September 15). RDF AND JSON-LD UseCases [W3c wiki]. Retrieved March 30, 2015, from https://www.w3.org/2013/dwbp/wiki/RDF_AND_JSON-LD_UseCases#Inline_markup_.28e.g._using_RDFa_or_Microdata.29:

Rahm, E. (2011). Towards large-scale schema and ontology matching. In *Schema matching and mapping* (pp. 3–27). Springer. Retrieved June 10, 2015, from http://link.springer.com/chapter/10.1007/978-3-642-16518-4_1

Ronallo, J. (2012, February 3). HTML5 microdata and schema.org. *The Code4Lib Journal*, (16). Retrieved April 30, 2015, from http://journal.code4lib.org/articles/6400

Schema.org. (n.d.-a). Frequently asked questions [Schema.org]. Retrieved April 23, 2015, from https://schema.org/docs/faq.html

Schema.org. (n.d.-b). Mappings of web data vocabularies to schema.org terms ... [Schema.rdfs.org]. Retrieved March 22, 2015, from http://schema.rdfs.org/mappings.html

Schema.org. (2011a). Getting started with schema.org using microdata. Retrieved March 22, 2015, from https://schema.org/docs/gs.html

Schema.org. (2011b). ("slash-based") extension mechanism. Retrieved March 22, 2015, from https://schema.org/docs/old_extension.html

Schema.org. (2013, June 3). Schema blog: schema.org and JSON-LD [Schema blog]. Retrieved May 27, 2015, from http://blog.schema.org/2013/06/schemaorg-and-json-ld.html

Schema.org. (2014, May 14). Data model. Retrieved April 29, 2015, from https://schema.org/docs/datamodel.html

Schema.org. (2015a). Extension mechanism. Retrieved May 16, 2015, from https://schema.org/docs/extension.html

Schema.org. (2015b, February 4). Releases. Retrieved April 7, 2015, from http://schema.org/docs/releases.html

Schlenoff, C. (2009). Ontology formalisms: what is appropriate for different applications? In *Proceedings of the 9th workshop on performance metrics for intelligent systems* (pp. 180–187). PerMIS '09. New York, NY, USA: ACM. doi:10.1145/1865909.1865947

Schmachtenberg, M., Bizer, C., & Paulheim, H. (2014). Adoption of the linked data best practices in different topical domains. In *The semantic web–ISWC 2014* (pp. 245–260). Springer. Retrieved June 10, 2015, from http://link.springer.com/chapter/10.1007/978-3-319-11964-9_16

Searchmetrics. (2014, April). *Schema.org in google search results.* Searchmetrics Inc. New York.

Semy, S. K., Pulvermacher, M. K., & Obrst, L. J. (2004). *Toward the use of an upper ontology for US government and US military domains: an evaluation.* Retrieved May 21, 2015, from http://oai.dtic.mil/oai/oai?verb=getRecord%5C&metadataPrefix=html%5C&identifier=ADA459575

Shadbolt, N., Hall, W., & Berners-Lee, T. (2006). The semantic web revisited. *Intelligent Systems, IEEE, 21*(3), 96–101. Retrieved March 20, 2015, from `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1637364`

Shirky, C. (2005). Ontology is overrated – categories, links, and tags [Clay shirky's internet writings]. Retrieved from `http://www.shirky.com/writings/ontology_overrated.html`

Shvaiko, P. & Euzenat, J. (2013). Ontology matching: state of the art and future challenges. *Knowledge and Data Engineering, IEEE Transactions on, 25*(1), 158–176. Retrieved June 10, 2015, from `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6104044`

Singhal, A. (2012, May 16). Introducing the knowledge graph: things, not strings [Official google blog]. Retrieved May 24, 2015, from `http://googleblog.blogspot.com/2012/05/introducing-knowledge-graph-things-not.html`

Smarty, A. (2013, November 18). Schema-org guide for beginners (everyone can understand!) + schema cheatsheet [Internet marketing ninjas blog]. Retrieved April 23, 2015, from `http://www.internetmarketingninjas.com/blog/search-engine-optimization/schema-org-guide-beginners-cheatsheet/`

Sporny, M., Bazaar, D., & Inc. (2015, March 17). RDFa lite 1.1 - second edition. Retrieved May 27, 2015, from `http://www.w3.org/TR/rdfa-lite/`

Suchanek, F., Abiteboul, S., & Senellart, P. (2011). Ontology alignment at the instance and schema level. *arXiv preprint arXiv:1105.5516*. Retrieved June 10, 2015, from `http://arxiv.org/abs/1105.5516`

Szekely, P., Knoblock, C. A., Yang, F., Fink, E. E., Gupta, S., Allen, R., & Goodlander, G. (2014). Publishing the data of the smithsonian american art museum to the linked data cloud. *International Journal of Humanities and Arts Computing, 8*, 152–166. Retrieved June 7, 2015, from `http://www.euppublishing.com/doi/abs/10.3366/ijhac.2014.0104`

Tallerås, K., Massey, D., Husevåg, A.-S. R., Preminger, M., & Pharo, N. (2014, November 27). Evaluating (linked) metadata transformations across cultural heritage domains. In S. Closs, R. Studer, E. Garoufallou, & M.-A. Sicilia (Eds.), *Metadata and semantics research* (478, pp. 250–261). Communications in Computer and Information Science. Springer International Publishing. Retrieved March 19, 2015, from `http://link.springer.com/chapter/10.1007/978-3-319-13674-5_24`

Tennison, J. (2011a, June 10). Lessons for microdata from schema.org [Jeni's musings]. Retrieved March 22, 2015, from `http://www.jenitennison.com/2011/06/10/lessons-for-microdata-from-schema-org.html`

Tennison, J. (2011b, June 12). Schema.org and the responsibility of monopoly [Jeni's musings]. Retrieved March 22, 2015, from `http://www.jenitennison.com/2011/06/12/schema-org-and-the-responsibility-of-monopoly.html`

The J. Paul Getty Trust. (2014, February). The getty vocabularies: project to publish as linked open data. Retrieved April 7, 2015, from `http://www.getty.edu/research/tools/vocabularies/vocab_lod_flier.pdf`

Twitter. (n.d.-a). Getting started guide [Twitter developers]. Retrieved May 3, 2015, from `https://dev.twitter.com/cards/getting-started`

Twitter. (n.d.-b). Twitter cards [Twitter developers]. Retrieved May 3, 2015, from `https://dev.twitter.com/cards/overview`

W3C OWL Working Group. (2012, December 11). OWL 2 web ontology language document overview. Retrieved May 21, 2015, from `http://www.w3.org/TR/owl2-overview/`

Wallis, R. (2013). Schema bib extend. *Information Standards Quarterly, Winter*, *25*(4), 30–32. Retrieved May 19, 2015, from `http://www.niso.org/apps/group_public/download.php/11936/IP_Wallis_Schema_Bib_Extend_isqv25no4.pdf`

Watson, P. (2013, April 6). Semantic HTML for artwork – can you help? [The lazarus corporation]. Retrieved May 10, 2015, from `http://www.lazaruscorporation.co.uk/blogs/arts-tech/posts/semantic-html-for-artwork`

Watson, P. (2015a, February 4). Getty AAT linked open data in schema.org VisualArtwork [The lazarus corporation]. Retrieved April 7, 2015, from `http://www.lazaruscorporation.co.uk/blogs/arts-tech/posts/getty-aat-linked-open-data-in-schemadotorg-visualartwork`

Watson, P. (2015b, February 4). Mapping VRA core 4 to schema.org VisualArtwork [The lazarus corporation]. Retrieved April 7, 2015, from `http://www.lazaruscorporation.co.uk/blogs/arts-tech/posts/mapping-vra-core-4-to-schemdotorg-visualartwork`

Wu, H., Zubair, M., & Maly, K. (2006). Harvesting social knowledge from folksonomies. In *Proceedings of the seventeenth conference on hypertext and hypermedia* (pp. 111–114). ACM. Retrieved May 31, 2014, from `http://dl.acm.org/citation.cfm?id=1149962`

Zaino, J. (2013, November 13). Where schema.org is at: a chat with google's r.v. guha [DATAVERSITY]. Retrieved June 11, 2015, from http://www.dataversity.net/schema-org-chat-googles-r-v-guha/