TECHNISCHE UNIVERSITÄT DRESDEN

Department of Computer Science Institute for Systems Architecture, Chair of Computer Networks

Diploma Thesis

SEMANTIC FEDERATION OF MUSICAL AND MUSIC-RELATED INFORMATION FOR ESTABLISHING A PERSONAL MUSIC KNOWLEDGE BASE

Submitted by:

Thomas Gängler

Student ID: 3174504
E-Mail: Thomas.Gaengler@mailbox.tu-dresden.de

Supervisor:

Prof. Dr. rer. nat. habil. Dr. h. c. Alexander Schill

Advisor:

Dr.-Ing. Daniel Schuster

External advisor:

Dr. Markus Schedl, Department of Computational Perception, Johannes Kepler University Linz

Submitted on April 26th 2011

SOLEMN DECLARATION

With my signature I truthfully and solemnly declare that I have prepared the Diploma Thesis with the topic:

> Semantic Federation of Musical and Music-Related Information for Establishing a Personal Music Knowledge Base

entirely myself and without the help of any third party and that all literal quotations and other authors have completely been accounted for.
Thomas Gängler

Dresden, April 26th 2011

ABSTRACT

Music is perceived and described very subjectively by every individual. Nowadays, people often get lost in their steadily growing, multi-placed, digital music collection. Existing music player and management applications get in trouble when dealing with poor metadata that is predominant in personal music collections. There are several music information services available that assist users by providing tools for precisely organising their music collection, or for presenting them new insights into their own music library and listening habits. However, it is still not the case that music consumers can seamlessly interact with all these auxiliary services directly from the place where they access their music individually.

To profit from the manifold music and music-related knowledge that is or can be available via various information services, this information has to be gathered up, semantically federated, and integrated into a uniform knowledge base that can personalised represent this data in an appropriate visualisation to the users. This personalised semantic aggregation of music metadata from several sources is the gist of this thesis. The outlined solution particularly concentrates on users' needs regarding music collection management which can strongly alternate between single human beings.

The author's proposal, the *personal music knowledge base* (PMKB), consists of a client-server architecture with uniform communication endpoints and an ontological knowledge representation model format that is able to represent the versatile information of its use cases. The PMKB concept is appropriate to cover the complete information flow life cycle, including the processes of user account initialisation, information service choice, individual information extraction, and proactive update notification.

The PMKB implementation makes use of Semantic Web technologies. Particularly the knowledge representation part of the PMKB vision is explained in this work. Several new Semantic Web ontologies are defined or existing ones are massively modified to meet the requirements of a personalised semantic federation of music and music-related data for managing personal music collections. The outcome is, amongst others,

- a new vocabulary for describing the *play back* domain,
- another one for representing information service categorisations and quality ratings, and
- one that unites the beneficial parts of the existing advanced user modelling ontologies.

The introduced vocabularies can be perfectly utilised in conjunction with the existing Music Ontology framework. Some *RDFizers* that also make use of the outlined ontologies in their mapping definitions, illustrate the fitness in practise of these specifications.

A social evaluation method is applied to carry out an examination dealing with the reutilisation, application and feedback of the vocabularies that are explained in this work. This analysis shows that it is a good practise to properly publish Semantic Web ontologies with the help of some Linked Data principles and further basic SEO techniques to easily reach the searching audience, to avoid duplicates of such KR specifications, and, last but not least, to directly establish a "shared understanding". Due to their project-independence, the proposed vocabularies can be deployed in every knowledge representation model that needs their knowledge representation capacities. This thesis added its value to make the vision of a *personal music knowledge base* come true.

ACKNOWLEDGEMENTS

There are so many people out there which I have to be thankful for supporting me the whole time of my studies and especially at that time when I wrote this thesis. At first, I would especially like to thank my advisers, Daniel Schuster, from the TU Dresden (Germany), and Markus Schedl, from the JKU Linz (Austria), for being patient with me all the time, and guiding me in reducing my pretty ambitious project vision to an appropriate level that fits into the scope of this work. Furthermore, thanks a lot Markus for three inspirational months at the Institute for Computational Perception (JKU Linz) last summer.
In addition, I would to massively thank all the people of the different Web and research communities. These are in broader fields: the Semantic Web community, the Linked Data community, and the Music Information Retrieval community; and in a narrow sense: the Music Ontology community, the FOAF community, and the *SemanticOverflow* community. I believe that the Web is the best place for doing modern research and exchanging ideas in a simple manner. Since, I cannot start name dropping now to mention all people that influenced my thoughts and creativity through the time of writing this thesis, I just want to name a few in no particular order: Yves Raimond, Kurt Jacobson, Toby Inkster, Mats Skillingstad, Jirí Procházka, Alexandre Passant, and Stephan Baumann.
Finally, special thanks to my whole family and friends who particularly supported me in realising to walk my way. Thank you so much, Astrid, for all the countless things you had to manage for me during the last time and the power you gave me to finish this project. I love you with all my heart.

Believe in music to survive!

Copyright TU Dresden, Thomas Gängler

CONTENTS

Copyright TU Dresden, Thomas Gängler

LIST OF FIGURES

1 INTRODUCTION AND BACKGROUND

1.1 INTRODUCTION

The phenomenon of music is omnipresent. It exists beyond the barriers of societies, classes or cultures. Despite that it seems to be rather difficult to exactly define what music really is, every one can talk about it in a very personal way. That is why music still bears certain mystery in its realisations. It seems to be that this is exactly the aspect which makes music especially interesting for its providers and consumers (cf. [Hol05]).
A personal music collection can say a lot about a human being who owns this aggregation of musical items. One can establish a strong emotional relationship to a music collection's single entities or the whole library, e.g., by associating individual experience and remembrances to the music (cf. [CJJ04]).
Currently, the shift from physical music collections to digital ones is in full progress. Thereby, the sensation of haptics, that is caused by analogue carrier media, such as vinyl records, usually get lost during this digitalisation process and people lose their individual mnemonics with this digital music evolution. For that matter, new methods and paradigms are needed to put forward the music consumption - the listening experience including its discovery and selection part - again.

Problem Description Music is perceived and described very subjectively by every individual. Nowadays, people often get lost in their steadily growing, multi-placed, digital music collection. Existing music player and management applications get in trouble when dealing with poor metadata that is predominant in personal music collections. There are several music information services available that assist users by providing tools for precisely organising their music collection, e.g., *Discogs*[1] or *MusicBrainz*[2], or for presenting them new insights into their own music library and listening habits, e.g., *Last.fm*[3] or *Echo Nest*[4]. However, it is still not the case that music consumers can seamlessly interact with all these auxiliary services directly from the place where they access their music individually.
To profit from the manifold music and music-related knowledge that is or can be[5] available via

[1] `http://discogs.com`
[2] `http://musicbrainz.org`
[3] `http://last.fm`
[4] `http://echonest.com`
[5] Some music metadata has to be derived from complex analysis tasks before it is available on a high level.

various information services, this information has to be gathered up, semantically federated, and integrated into a uniform knowledge base that can personalised represent this data in an appropriate visualisation to the users. This personalised semantic aggregation of music metadata from several sources is the gist of this thesis. It is a continuum of the work that was carried out by Yves Raimond in his dissertation about a distributed music information system (see [Rai08a]). The solution that is outlined in my work particularly concentrates on users' needs regarding music collection management which can strongly alternate between single human beings.

Related Research Areas The superior research area that concerns this thesis is called Music Information Retrieval (MIR). It is an interdisciplinary field of research which primarily deals with an "(semi-)automatic extraction and processing of descriptions and features of music on the basis of music documents (incl. their audio signals and metadata) to represent retrieved or generated knowledge in a manner that is appropriate for its consumers, and make this data accessible" [Gän09a].

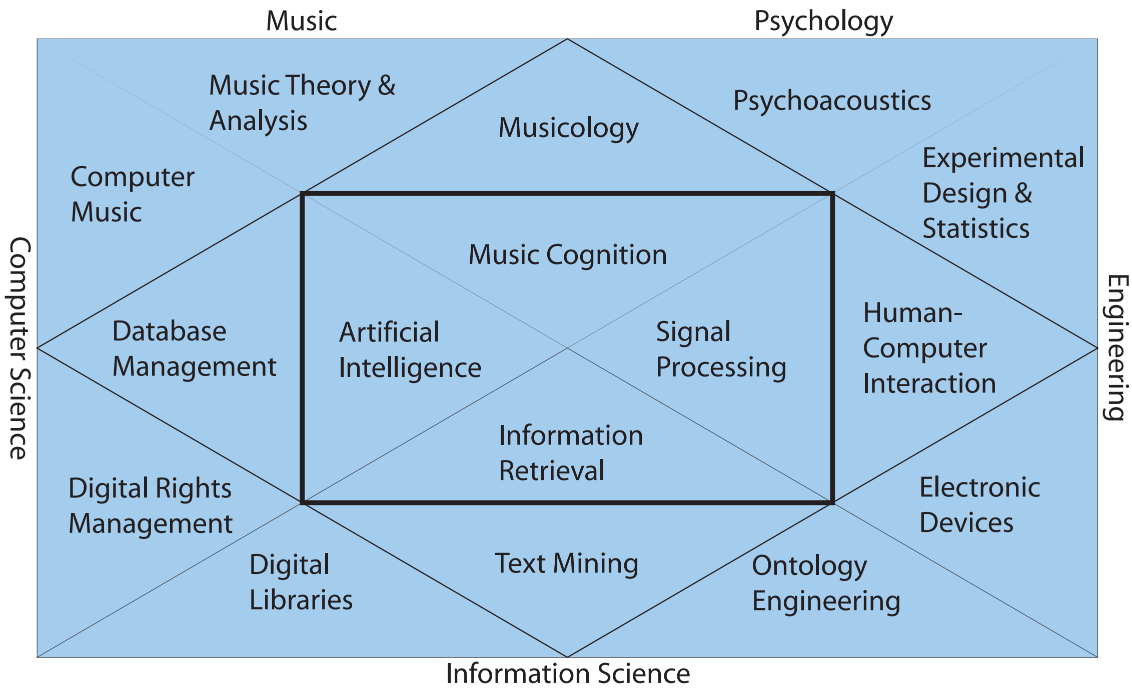

Figure 1.1: Disciplines of MIR (according to [Ser08])

An overview of involved sciences is illustrated in Figure 1.1 and includes, amongst others, the areas Artificial Intelligence and Ontology Engineering. These disciplines contribute their findings, which stemming from researches of the highly complex process of music perception by human beings, to developing systems that can provide versatile interfaces for huge, digital music collections to users (cf. [MKC06]).
Figure 1.2 shows a largely simplified map of music content and context data concepts that are involved in the broad variety of MIR tasks which are exhibited and analysed by the different MIR disciplines. This graphic nicely showcases the complexity music descriptions can reach to satisfying cater usually fuzzy user requests that make use of high level categories, such as mood or music genre (see Section 2.3.1).
Most recently, the results of the manifold MIR researches find their way into music player and management applications that are used for handling personal music collections, e.g., *Songbird*[6] or mufin player[7] (see Section 1.2), as well as music metadata services, such as *Last.fm* or *MusicBrainz* (see Section 2.3.3).

[6] http://getsongbird.com
[7] http://mufin.com

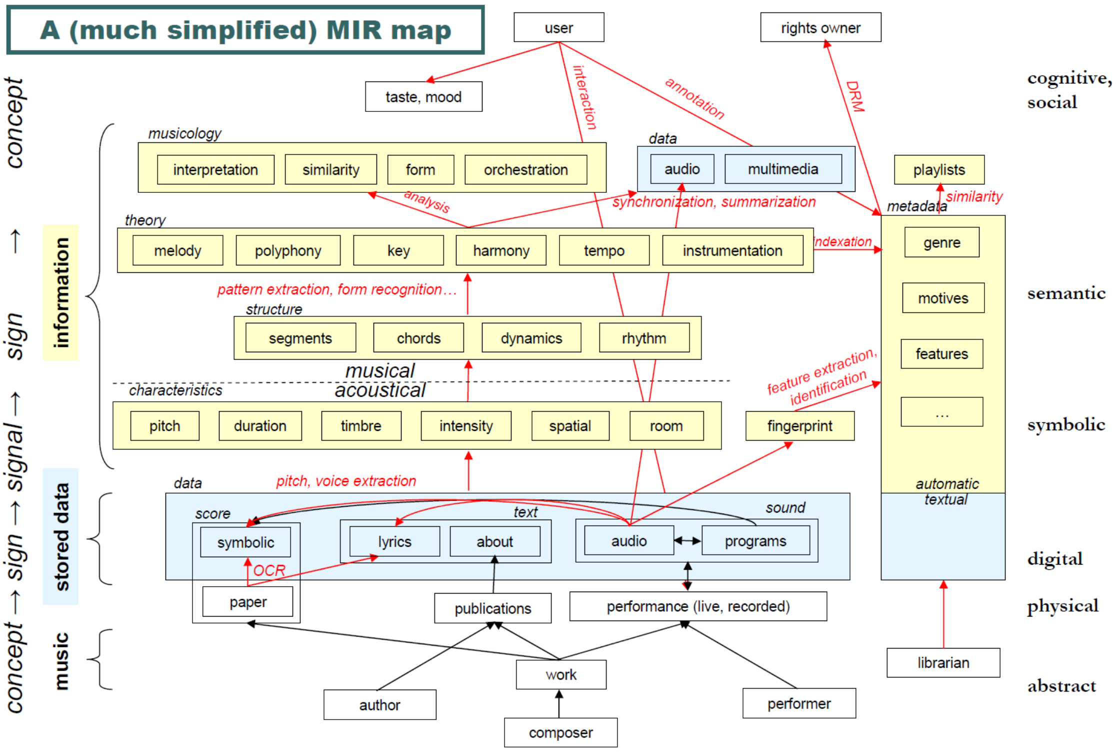

Figure 1.2: Musical characteristics - the subject of MIR (see [Fin04])

Thesis Outline This work, whose aim it is to illustrate and showcase appropriate technologies that can be utilised for a semantic federation of music and music-related information to establish a personal music knowledge base, is divided into six chapters. Thereby, the first part contains this introduction as well as a compact categorisation of the principal application scenarios that are relevant for handling personal (digital) music collections (see Section 1.2).
The prelude is followed by a chapter that introduces the basic areas and their technologies that are relevant for personal music information management as well as being necessary to understand the subsequent chapters (see Chapter 2). Starting with a general introduction to the extensive topic of knowledge management (KM; see Section 2.1), this chapter furthermore includes an overview section about Semantic Web technologies and their application (see Section 2.2), an explanation of music content and context data including its formats and services (see Section 2.3), and a reflection of personalisation aspects (see Section 2.4).
This technology-centred chapter provides the foundations for the Chapters 3 and 4, which represent the author's proposal for handling the personalised semantic federation of music metadata. Whereby, the former one explains the rationale of the abstract concept *personal music knowledge base* (PMKB), the latter one describes a concrete implementation of the PMKB vision on a rather technology-independent level with a special focus on knowledge representations (KR).
Chapter 5 exemplifies the utilisation of the designed PMKB vocabularies and an evaluation of their fitness in practise. This thesis is closed with a small conclusion and future work part (see Chapter 6).

Limitations Due to the lack of time, unforeseen problems, and a probably too ambitious sketched-out vision that cannot be resolved within the scope of a diploma thesis[8], this work had to undergo some further limitations. Despite having tried to pre-check every part of my concept regarding its suitability of implementation with the help of *state-of-the-art* technologies, I had to realise that this did not apply to every component. That is why, I had to design and implement parts of the

[8]A diploma thesis is similar to a master's thesis.

current solution, whereas I previously thought that they already existed in usable statuses.
Unfortunately, the time that I needed to resolve these unforeseen problems was no longer available to to put the knowledge management system (KMS) part of the PMKB concept into practise. So, this work particularly concentrates on a proposal for the KR part of the PMKB vision. Fortunately, as one can see in the evaluation part of this work, the current results have found their way into the Web ecosystem.
Besides, the technology chapter of this thesis took up more time for writing up than expected. This was caused by the lack of available, appropriate definitions of some technologies and concepts that are outlined in Chapter 2. In addition, it was a special concern of mine that I would like to briefly illustrate the whole background story of the Semantic Web vision, which seems to be a quite natural step in the evolution of the Web, and thereby has its roots in ancient philosophy.

1.2 PERSONAL MUSIC COLLECTION USE CASES

There are manifold application scenarios that are imaginable regarding a personal music collection. One can differentiate between those that can be applied physical music collections, and use cases that are possible on digital music collections. Albeit, the latter ones more or less only reflect a transformation of the former forms of usage. That is why, it is important to primarily implement the use cases that can be applied to physical music collections also for digital music libraries (cf. [CJJ04, Gän09a]).
Music players in all their variety represent the primary type of user agents for dealing with digital music collections. Many actions that are executed with the help of the software applications lead to a play back activity of a piece of music at a certain point in time. Interesting, notable *state-of-the-art* music player implementations are:

- *Songbird*, which is an open-source music player and collection management application that is based on an easily extensible component architecture where manifold add-ons already exist, e.g, the information aggregation plug-in *mashTape*[9],
- *Tomahawk* [Tea11], which is an open-source software that primarily concentrates on music consumption, i.e., playing back pieces of music, and, therefore, implements a multi-source music song resolution concept[10] with social network capacities, or
- *mufin player*, which is a music player and collection management programme that makes especially use of advanced music content analysis techniques.

I already carried out a comprehensive analysis of application scenarios on personal music collections as a part of my Belegarbeit[11] [Gän09a]. For this reason, brief summaries of the separate use cases: organisation, exploration, search, recommendation, playlist and mix generation, modification, and sharing are given in the following paragraphs. An overall outcome of this examination was that the different forms of usage require a wide variety of knowledge in total. This necessity vehemently justifies the semantic federation of multiple music information services.

Organisation There are countless variants for organising a music collection. For digital media, a certain arrangement does not always need to conform to the underlying storage hierarchy. Each organisation strategy can be represented with the help of separate views of a music collection. A traditional structure consists of a layering of music artist's name, album title, and music track. Contextual arrangement strategies, such as 'current favourites', 'popularity', 'storage place' or 'play back counter', are often in use as well. Thereby, *idiosyncratic genre* definitions [CJJ04] provide the most flexibility to meet a user's individual needs. One has to be able to create such *music contexts* [CGD06] by utilising every available feature of music content and context data (see Section 2.3). For example, "music for programming" that has to meet a certain rhythm and has to be taken from my *alltime-favourite-songs*.

[9] `http://addons.songbirdnest.com/addon/73`
[10] This music song resolver component is based on the concept of the *Playdar* service [Jon10] (see [Tea11]).
[11] A 'Belegarbeit' is an extensive essay that can be compared to a bachelor thesis.

Exploration Exploration deals with the issues of how a user browse through a music collection and occasionally discovers interesting aspects of it she/he did not knew before. It is an explorative search where user experience is achieved by the visualisation that is utilised for browsing ("the journey is its own reward"). On the one side, these can be classical ones, such as cover art displaying, e.g., *iTunes Cover Flow* (see [App11]), or *faceted browsing* [Rai08a]. On the other, these can be modern context- or content-based views, for example,

- a map with geographical information and related attributes, e.g., release country or country of origin of an music artist,
- a timeline with temporal information and related attributes, e.g., release year or period of time of an artist cooperation,
- a semantic net that illustrates various relations between different entities, e.g., music artist and band associations, or
- a 3D visualisation that arranges resources regarding various similarity methods in a virtual 3D space, e.g., music songs regarding their dancability, mood and play back count.

Of course, these variants can be combined with each other, e.g., a timeline with a semantic net. The exploration use case particularly requires a huge amount of information of various kinds.

Search Music search should principally deliver a result as exact as possible regarding a given query. *Music queries* are verbalised expressions and/or music content representations, where users express requests regarding one or more music object(s), or they give some information about those. (see [Gän09a]). They can be divided into three different types (according to [CLP+07]). These are *Query-by-Text* (or *verbalised queries* to include spoken requests too; see [Gän09a]), *Query-by-Content* (e.g. *Query-by-Melody*) and *hybrid music queries*. Due to their different nature, *music queries* also require comprehensive music content and context information to be able to process usually fuzzy search requests. Nevertheless, only the first type is commonly provided by music player software for personal music collection management (see [Gän09a]).

Recommendation The use case of recommendation can be viewed from two perspectives. Firstly, that instance which gives suggestions, e.g., a music recommendation service á la *Last.fm*. Secondly, that participant which consumes recommendations - usually a user of such a service. An agent can hold both positions, e.g., a person who provides recommendations to another one and simultaneously consumes recommendations of that other human being.
Recommendation tasks can be initialised by *music queries* as well. However, they deliver, in contrast to a search, fuzzy results with music objects that are similar to a given input, e.g., a music song. Similarity analysis methods are commonly divided into four main categories (cf. [Her08]). These are *collaborative filtering*, *content-based filtering*, *context-based filtering* and *user profile analysis* (incl. a music taste analysis). They have to be used in in conjunction with one another in order to be effective, e.g., to be able to also recommend music from the *long tail*, and to overcome certain drawbacks of the single approaches, e.g., a *popularity bias* of a community that is the foundation for *collaborative filtering*. Thereby, user should always be able to customise the similarity analysis methods in an easy way.
Besides, it is important to inform the user about the backgrounds of a recommendation. This type is usually called *explorative recommendation*. For example, Alexandre Passant demonstrated such kind of recommendations in the Web application *dbrec*[12] [Pas10a] which utilises Semantic Web technologies in its implementation (see Section 2.2).

[12] The idea of this system is adopted in the forthcoming music recommendation and discovery service *seevl*, see `http://seevl.net`.

Playlist and Mix Generation There are two kinds of music object compositions (according to [CBF06]). The first one is a (personal) playlist, which is usually composed in a loosely manner and has a varying length. The second one is a formal mix, which commonly got a well thought-out subject and fixed sequence and length. Both types can be manually construed or (semi-) automatically generated. A user always is to be intuitively assisted by a composition process. Such a system has to be able to process exact as well as fuzzy information of different varieties.
Different (semi-)automatic playlist generation methods can be applied. These are *shuffle mode*, and *personalised* or *smart playlists*. Even the utilisation of the *shuffle mode* can lead to serendipity effects (see [LVH05]). *Personalised* or *smart playlist* generation tasks usually require some *seed* or *music context* definitions. Such values can also be selected, for example, by drawing a path on a 3D visualisation (see [Lil08]). The playlist generation process has to be adaptable at all times, e.g., a user can give feedback, such as liking or skipping a piece of music, which influences the computation algorithm.

Modification A personal music collection or single parts of it can be modified in a versatile manner. This can be, for instance, its size, structure, storage location (incl. carrier media), or format. Hence, this use case additionally requires the handling of administrative metadata. Besides, information that is related to a music object can be modified as well, e.g., new relations between entities can be defined or existing ones be changed and enriched. Usually personal music collections include many documents with poor music content and context data. For that matter, (semi-) automatic completion or correction techniques can be applied, such as data validation tasks via semantic federation of information that is provided by several information services, e.g., *MusicBrainz*.
In addition, analyses of usage statistics, e.g., play back or skip counter, can be done. On the one hand, their results can influence other application scenarios, for example, music recommendation. On the other, this information can be used for clean-up tasks that are executed on personal collections, e.g., in order to downsize them. Information services, such as *Last.fm*, support the tracking of user's music consumption activities. Tools that (semi-) automatically support music collection clean-up tasks are, for instance, the *Songbird* extension *Exorcist*[13].

Sharing Users usually do not use a personal music collection only for themselves. They tend to share parts of their collections, particularly favourite music songs, with other interested parties. This application scenario requires advanced access control[14] and sharing capacities. A user model has to be a able to represent knowledge about personal views and usage statistics on shared music collections. For example, the music player software *Tomahawk* implements a basic music collection sharing approach that makes use of social network functionalities, i.e., one can connect to a collection of a friend with the help of certain social network services (see [Tea11]).
Besides, sharing methods, such as *digital lending*, which maps the physical borrowing process one-to-one onto the digital world, can bring back some social aspects that often get lost in the consumption overflow that is predominant nowadays (see [Gän09a]). The application of such techniques can help to trim the size of a personal digital music collection to a manageable amount.

1.3 SUMMARY

The versatile use cases that can be applied on personal music collections (see Section 1.2) require manifold information which can be fuzzy or exact, and be interlinkable in various ways. In addition, user profiles are often needed to guide the processes so that they can deliver more satisfaction for the users. Currently, there is no music player and management application that can satisfy all application scenarios to some degree at once. A customisable semantic federation that is supplied by multiple information sources seems to be a promising approach to deliver a more satisfying user experience with one's personal music collection. This idea is further investigated in the following chapters of this work.

[13] `http://addons.songbirdnest.com/addon/216`
[14] These are capabilities go beyond "party modus" functionalities.

2 MUSIC INFORMATION MANAGEMENT

There are countless definitions of the concept 'information' and its related terms 'data' and 'knowledge'. An unambiguous characterisation does not or cannot exist. An attempt of an interpretation is:

> Knowledge is all meaning of data, and information is a subset of knowledge to a concrete question or domain (cf. [sev11k]).

It defines knowledge as interpreted data - pieces of information that bear a meaning or a sense (cf. [Hal09]). In addition to that, this definition states that information is related to a specific concern or issue, e.g., the domain of music. Nevertheless, the terms 'information, 'data' and 'knowledge' are fuzzy and often used interchangeably.
Music information management is a specialisation of the broad field of information management. It particularly addresses issues that are related to the management of music metadata[1]. Albeit, music information management shares many features with its general superior subject.
For that reason, this chapter starts with a brief introduction to the topic 'knowledge management' (see Section 2.1). This part is followed by a field that exemplifies concepts of KM (see Section 2.2). These technologies are aggregated under one umbrella that is called Semantic Web. Afterwards, a compact overview of music content and context data is given (see Section 2.3). This section includes examples of the application of Semantic Web technologies in the domain of music information management. Since a PMKB has to be adaptive, this chapter closes with a small section about personalisation and environmental context (see Section 2.4).

2.1 KNOWLEDGE MANAGEMENT

Knowledge management can be described as "the process of turning information into useful knowledge that is accessible when needed" [SKR06]. It has the aim to enable information consumers to gain instantaneous, uniform and seamless access to most relevant information, wherever, whenever and however needed. Furthermore, KR has to support knowledge creation, staging and interpretation (incl. inferencing, e.g., categorising or indexing), and transfer. To manage knowledge explicitly, one has to represent or describe knowledge in some way. This is the topic of knowledge representation (see Subsection 2.1.1). The service that manages all the data, information and knowledge is called knowledge management system (see Subsection 2.1.2).

[1] To especially emphasise the aspect that it is "knowledge about music" [Pac05]

2.1.1 Knowledge Representation

The foundation of every KMS is the way in which knowledge is expressed in order to be processed by such a system. It is this kind of knowledge representation that is supposed to enable all participants, which interact with a KMS, to try to understand the same meaning or to grasp the same content including its sense of the information they are accessing and using (cf. [Hal09]).

2.1.1.1 Knowledge Representation Models

A (formal) knowledge representation model (KRM) is to realise the content of at least a single piece of information. It is encoded in a given knowledge representation language (KRL). A KRM can be interpreted by an information consumer to ideally get the intended meaning (the purpose) of the information. Besides, this interpretation can cause a meaningful behaviour for the information consumer as it was intended by the information provider (cf. [Hal09]). Knowledge representation languages power knowledge representation models. The applied terms of the KRLs are part of these models to describe or represent (parts of) the world. A KRM consists of interpretable sentences of at least one KRL. These are statements (*assertions*, see [BL09e]). Moreover, dialects or languages that are embedded as subsets of (other) languages are called vocabularies. KRLs can be layered upon another. Higher layered languages use the functionalities of lower layer languages.
The field of KR has a long history. Ancient Greek philosophers, e.g., Socrates, Plato and Aristotle, with their thoughts established the basis for the field of Epistemology and the "science of describing being and the world" that later was called Metaphysics. The latter term was later redefined as *metaphysica generalis* which is also called *ontologia*[2] (ontology) [SKR06]. The most accepted idea that knowledge is made up of *concepts* is essential for a representation of meaning of a thing and to investigate the most general concepts of being.
Thereby, the division of abstract and concrete objects, *entities*, or resources is led by Plato's definition of shared *entities* as *universals*. These are resources that can simultaneously be exemplified, or instantiated, or exhibited by several different objects [Lou02]. For that matter, the category of concrete objects is called *particulars* or *individuals* [Col07], which non-philosophers typically think of as *things* like human beings, animals or plants. Additionally, *particulars* can be treated as *universals* for other *particulars* and recursively.
Universals are divided into two main concepts. The first one is called *one-place* or *monadic universal*. It is sometimes divided into *property* and *kind*. *Properties* are used for characterisation, e.g., the adjective "courageous" can be referred to the *universal* "courage". *Kinds* are utilised to "mark out their members as what they are", e.g., human beings [Lou02]. Especially *kinds* can be seen as categories or types. The second specialisation of *universal* is known as *relation*. *Relations* are exemplified by several *entities* in relation to each other, e.g., "being next to" or "father of". In general they are called *polyadic* or *many-place universals* and are also known as *n-ary relations*. A common specialisation are *two-place* or *dyadic universals* which are known as *binary relations* as well.

2.1.1.2 Semantic Graphs

An important conclusion that the history of KR draws is that it is good to have a kind of simple formal KRL. Usually, such a formalisation is able to represent knowledge in a form that is similar or based on the concepts described above. Machines, by their nature, are currently not able to a priori process natural language statements (cf. [Hal09, Sow91]). A quite natural-language-like structure for representing knowledge is given by *semantic networks* as "a pattern of interconnected *nodes* and *arcs*" (see [Hal09, Sow91]). Thereby, *arcs* are used for describing the *relations* (incl. relation types) between *nodes* that hold the *particulars* or the *monadic universals*. Natural

[2] from the Greek *onto* for "being" and *logia* for "written or spoken discourse"

language terms are used as labels for identification and reference, which can cause ambiguity in an interpretation task.
From now on, we will use the term 'semantic graph' as a general, abstract term for instantiations of a formal knowledge representation structure (KRS) such as *semantic networks*. They can be used to model cognitive structures and have various features of other graph types, e.g., being directed or attributed (see [Sow91, RN10]). Hence, a concrete semantic graph is a knowledge representation structure or part of a KRM. On the one side, a vocabulary or an ontology can be represented by a semantic graph, for instance, a Semantic Web ontology. On the other, instantiations of *universals*, *particulars*, can be described by a semantic graph, e.g., a Semantic Web KRM.

2.1.1.3 Ontologies

Ontologies play quite a central role when dealing with KRMs. We will use the term 'ontology', more in the sense of vocabulary, for naming KRMs that more or less only include definitions of *universals* (shared *entities*). This point of view is in line with the popular definition of ontology in the context of computer and information science that was stated by Thomas R. Gruber : "an ontology is an explicit specification of a conceptualization" [Gru93]. This characterisation was later redefined by himself to "an ontology defines a set of representational primitives with which to model a domain of knowledge or discourse" [Gru09]. He additionally explained that ontologies are "intended for modelling knowledge about individuals, their attributes, and their relationships to other individuals" [Gru09]. So, ontologies provide a basis for "common understanding" or "shared meaning" and hence enable information integration (cf. [SKR06, Hal09]). Albeit, in general, one can consider every KRM as an ontology (cf., e.g., [WAO08]). This is more in line with the definition of ontology as it is used in philosophy (see Subsection 2.1.1.1).

2.1.1.4 Summary

The previous introduction to knowledge representation includes main terms that we need for understanding knowledge representation models, including their languages, and their application in KMSs. For a deeper reading on this topic, [Lou02, Sow91, SKR06, Hal09, Gru93] are interesting sources.

2.1.2 Knowledge Management Systems

A knowledge management system is to support KM tasks as mentioned in the introduction of this section. For that matter, it is a specialised information (management) system that is "dedicated to manage knowledge processes and represent a key element of knowledge-oriented information services" [SKR06].

2.1.2.1 Information Services

Information services are in general defined as the following:

> An **information service** is that part of an *information system* which provides information, serves it to customers and collects it from contributors in order to manage and store it (by optionally using administrators[3]).

[3] In this context, administrators can be moderators, too. They reflect a controlling and coordinating role.

This definition implies that an information service can be both, an information provider and an information consumer. It is also important to note that this definition is technology-independent, i.e., it includes non-computer (sometimes referred as *offline*[4]) related *things*, e.g., a human being, a book, or a library, and computer (sometimes referred as *online*) related *things*, e.g., a website (cf. [Gän10l]).
The term 'information service' can cover a broad range of things. In this work, its usage primarily concentrates on the Internet in our context, especially on the Web. The main focus is set on information services which either have or deliver a huge amount of information[5] (of a specific domain or domain independent). One could say: "let us define the term *Web Information Service* as a subset and a specific kind of information service". However, since the Web only delivers another kind of carrier medium or transmission form, it is enough to say that an information service simply uses this carrier medium or transmission form next to other carrier mediums or transmission forms.
The fundamental knowledge management processes of the mentioned KM tasks are communication (knowledge exchange), integration and reasoning (cf. [SKR06]). They can be organised in a centralised or distributed way. The decentralised KMS type provides advantages for easier extensibility, more efficient reasoning and context-awareness, and knowledge sharing across autonomous entities - hence scalability of the overall system. Thereby, the reasoning engine can make use of a partitionally complete, limited knowledge, e.g., by applying a kind of Access-Limited Logic, see [CK91].

2.1.2.2 Ontology-based Distributed Knowledge Management Systems

Ideally a distributed KMS has a strong focus on information integration and personalisation. Consequently, the information integration component has to deal with data of every kind while it should be able to utilise every available information service. The personalisation component manages customisation requests as well. Such an archetypal KMS can overcome the drawbacks of the existing information silos, e.g., information overload, that are often the result of centralised approaches (cf. [SKR06]).
Therefore, Vasudeva Vama and Liana Razmerita in their chapters of [SKR06] propose, as well as R.M. Colomb suggest in [Col07] that ontologies and a uniform interface to them play a key role in building effective knowledge representation models. As a result a KRM of a KMS is supposed to be powered by multiple ontologies. They can help in all three fundamental KM processes (see Subsection 2.1.2.1) and especially to make tacit knowledge explicit.
The KRMs of the ontology-based knowledge management systems (OKMS) are typically stored in knowledge bases (KB). These are specific databases that usually include not only raw data, i.e., furthermore ontologies and rules for reasoning. They have to fulfil the requirements that are defined by the specific KM organisation types (see Subsection 2.1.2.1). Generally, KBs are a kind of advanced, open databases. They can be seen as their descendants that also profit from experiences of their developments and deployments.
The book "Ontologies" by Sharman et al. [SKR06] can be a good starting point for further reading on ontology-based KMS.

---

[4]The terms *offline* and *online* are generally used for computer related *things*. They express whether information is available via Internet (*online*) or not (*offline*)

[5]However, the society tends more and more into the *long tail* [sev11d]. That is why, it might be interesting to describe and rate all these very subjective information services, e.g., a personal blog or website, too. It depends all on the subjective mutual trust each information service consumer has to specific information services. It is at least a mutual trust between consumers and providers of an information service.

2.1.2.3 Knowledge Management System Design Guideline

Ontology-based distributed KMSs require an appropriate overall architecture that is optimised for establishing efficient communication between independent decentralised components of the system that interact with each other. For that matter, the architectural design of such systems can be guided by an architectural style that propagates to reflect an ideal to emphasise the outlined behaviour. It has the name Representational State Transfer (REST) [FT02].
This hybrid architectural style is derived from a couple of basic network-based architectural styles. It was initially introduced to capture all aspects of a distributed hypermedia system that are important regarding information management requirements (see [Fie00]). These demands are adopted by ontology-based distributed KMSs due to the fact that these systems are specialised distributed information systems. The coordinated set of architectural constraints that forms the REST architectural style can be applied as needed to mostly reach an approximation of the ideal of architectural properties (see [Fie00]). They can be induced or improved via a complete implementation of the defined constraints.
Design principles, such as, separations of concerns or generality of interfaces, are realised by an architecture with client, server and intermediary components that make use of a uniform interface for component interaction processing. Thereby, the layered system components are encapsulated by their abstract interfaces (typed connectors). They allow hiding of implementation details (allowing substitutability) and enable mechanisms for (shared) caching, security policies and integration of legacy systems. The individual view of a connector is restricted to the immediate layer of its interactions. For that reason, in order to be understandable independently by all participating components, all component-interactions are stateless. This allows parallel processing of transferred messages and enforces a loose coupling between components.
Consequently,

1. (uniform) identification of resources,
2. universal action semantics[6],
3. manipulations of data through representation,
4. self-descriptive messages, and
5. hypermedia as the engine of application state

are defined as component interface constraints of the REST architectural style to emphasise the feature of a uniform interface (see [FT02]). Since the application of caching can significantly enforce user-perceived performance and network efficiency, it is required to explicitly label the cacheability state of response data. To induce a client-side simplicity, it is possible to individually extend the functionality of a client on demand ("code on demand") as an option.
All in all, implementations of different constraints of the REST style influence the effects on architectural properties in a different manner. A careful selection of constraints is recommended and necessary to achieve an acceptable overall balance of these properties regarding the given functional requirements of a system. The whole constraint package of REST is designed for the feature of evolving requirements. Hence, it can establish a high degree of durability of a system, dependent on the application of possible and appropriate constraints.
The dissertation of Roy T. Fielding [Fie00] is an essential reading that helps to understand the quite impressive architectural style REST.

[6]This constraint was later explicitly phrased, see [Fie08].

2.1.3 Summary

The whole section should provide a good overview about important aspects of distributed and decentralised knowledge management. On the one side, a coarse introduction on the broad field of knowledge representation explains the concepts: knowledge representation model (see Subsection 2.1.1.1), semantic graph (see Subsection 2.1.1.2) and ontology (see Subsection 2.1.1.3). On the other side, necessary basic facts about knowledge management systems are given, in order to be able to grasp the fundamental architecture and design decisions that are made in this work. These subjects are information services (see Subsection 2.1.2.1, ontology-based KMS (see Subsection 2.1.2.2), and Representational State Transfer (see Subsection 2.1.2.3). All this knowledge provides the basis to explain the Semantic Web and its technologies as an exemplification of KM and KR. This is the topic of the following section.

2.2 SEMANTIC WEB TECHNOLOGIES

The Semantic Web is a step in the evolution of Web, which seems to take precedence over the media for communication and knowledge-sharing in our daily life. For that matter, the first subsection shows its natural integration in the currently biggest, existing distributed information (management) system (see Subsection 2.2.1). Afterwards, a common set of basic knowledge representation languages in the Semantic Web is introduced (see Subsection 2.2.2). This section is followed by an explanation of different resource description levels and their relationships, which are important for knowledge representation and management in the Semantic Web (see 2.2.3). Finally, the information flow life cycle of Semantic Web knowledge representation models is outlined in the last part of this section (see Subsection 2.2.4).

2.2.1 The Evolution of the World Wide Web

At a first glance, the World Wide Web (WWW) can be seen as an "ahistorical and unprincipled 'hack' that came unto the world unforeseen and with dubious academic credentials" [Hal09]. The WWW, or short and simply 'the Web', was initially a project with the goal of developing a distributed hypertext (hypermedia) system for information management by Tim Berners-Lee et al. (see [BL89, BLC90]). However, at a second view, one will see that it is well aligned to existing solutions and foundings, which are not only from the research field of computer and information science.

On the one side, one can recognise parallels and inspiration of the Web by the vision and projects of Vannevar Bush ('Memex', 1945 [Bus45]), J. C. R. Licklider ('Man-Computer Symbiosis', 1960 [Lic60]) and Douglas Engelbart ('Human Augmentation Framework', 1962 [Eng62]). On the other side, it was a crucial decision to build the Web on an existing, standardised computer networking architecture, known as the Internet, which is by itself a descendant of Licklider's vision of an 'intergalactic computer network' [Lic63] that resulted into the ARPANet project. As a result of this design decision, the architecture of the Web follows the design principles of its underlying technology as well. These are, for instance, simplicity, modularity, decentralisation and tolerance (see [BL10d, Car96]).

2.2.1.1 The Hypertext Web

As the history of the Web teaches us, it was a wise decision to firstly deploy the vision of a shared 'universal space of information' [BL97] as a *hypertext Web*. This is primarily built on the (now) well-known specifications of Uniform Resource Identifier (URI) [BLFM05], HyperText Markup Language (HTML) [Hic10] and HyperText Transfer Protocol (HTTP) [FGM+99]. Furthermore, the *hypertext Web* uses untyped *binary relations* for linking resources that are referenced by URIs. The success of the *Web of Documents* [BL10a] is still leading. Albeit, this is "just" one step on the road towards a 'universal [inter-]linked [decentralised] [shared] information system' [BL89]. This evolution is always guided by the *normative Principles of Web architecture* (see the following subsection; cf. [Hal09]).

2.2.1.2 The Normative Principles of Web Architecture

The *normative principles of Web Architecture* were formed during the design and evolution of the Web and are investigated profoundly by Harry Halpin in his PhD dissertation [Hal09]. The first one is the *Principle of Universality* [BL09f]. It says that any resource (*entity*) can be identified by a URI. For that matter, URIs are used as names to denote or refer to resources and as locators to access information[7] in the 'universal information space'.
The second foundation is the *Principle of Linking*. It specifies that any resource can be linked to other resources that are identified by URIs. This is important for establishing a real web of interlinked resources. Thereby, the deployment of decentralised link indices (Web servers) was a crucial decision for the global scalability of the Web.
The third guideline is the *Principle of Self-Description* (cf. [Men09]). It states that the information that can be accessed by resolving a URI on the Web is supposed to help an agent to discover an interpretation of this name. For that matter, a "resource is successfully described if an interpretation of a sense is possible" for an information consumer [Hal09]. This process can be continued recursively with every linked name ("follow your nose" [sev07b]), if it is not possible to reach an adequate interpretation of a name by using the given information and the local knowledge of the agent. A general 'follow-your-nose' algorithm cannot really exist, due to different capabilities of information consumers and information providers (cf. the *Principle of Nose-following* [RBH10]). Yet, "The ability to rely on sophisticated common-sense interpretative capacities" is crucial for the sharing of information [Hal09].
The forth *normative Principle of Web architecture*, the *Open World Principle*, follows a relatively natural condition because on the Web, the number of resources can always increase, too. Since there is no central repository of the entire Web's state, only partitional completeness and limited knowledge can be achieved by using it (cf. [CK91]). Due to this condition, the *Open World Assumption*[8] prevails on the Web because the world is not fixed.
The last principle is the *Principle of Least Power* [BL10d]. It expresses that an information resource that describes a resource should be realised in the least powerful but adequate language(s). Following this principle, languages have to be chosen well-thought-out, so that they are capable of conveying a sense of transmitted information. In other words, information consumers have to be able to understand the languages that information providers use for encoding the delivered information resources. This principle intends to enforce tolerance on knowledge transfer - "be strict when sending and tolerant when receiving" [Car96]. Besides, it propagates simplicity for establishing a low entry barrier to enable sufficient adoption of the Web (see [FT02]). Finally, stacks of languages that are built upon another and applied as needed, are a consequence of the application of the *Principle of Least Power*.

[7] A message as a result of a URI dereferenciation process can be used to emphasise discovering a sense of a name.
[8] Statements that cannot be proven to be true cannot be assumed to be false.

2.2.1.3 The Semantic Web

Information that is available on the *hypertext Web* is primarily intended for human consumption. However, the initial vision of Tim Berners-Lee is a 'universal information space', i.e., the Web must be usable by *everyone* - human beings and machines (cf. [BL97]). Since the amount of information on the Web is constantly growing, human information consumers increasingly rely on the external assistance of machines (cf. Licklider's 'Man-Computer Symbiosis'). Hence, the step to make the Web 'machine-understandable' and tacit knowledge explicit, is simply consequential as well as for web-scale knowledge management (see [SKR06]). This enhancement is often called the Semantic Web (*Web of Data* [BL09f]), which is led by Tim Berners-Lee's slogan: "It is is not the documents which are actually interesting, it is the things they are about" [BL10a]. Therefore, a common, layered stack of formal KRLs is necessary to enable the description of resources as fully as possible and to share this knowledge (cf. Section 2.1.1). It allows machines to interpret a sense from such a KR in the same manner as human beings would[9]. The common, layered stack of formal KRLs for the Semantic Web (see Subsection 2.2.2) is part of the Semantic Web technology stack (see Figure 2.1) and follows the introduced *normative Principles of Web architecture* (as explained more in detail in [Hal09]). Furthermore, this technology stack includes aspects that are important for the information flow life cycle of Semantic KRMs (see Subsection 2.2.4). Some of these concerns are covered quite well, e.g., SPARQL as an abstract query interface. However, others are still at an earlier state of development, e.g., proof and trust.

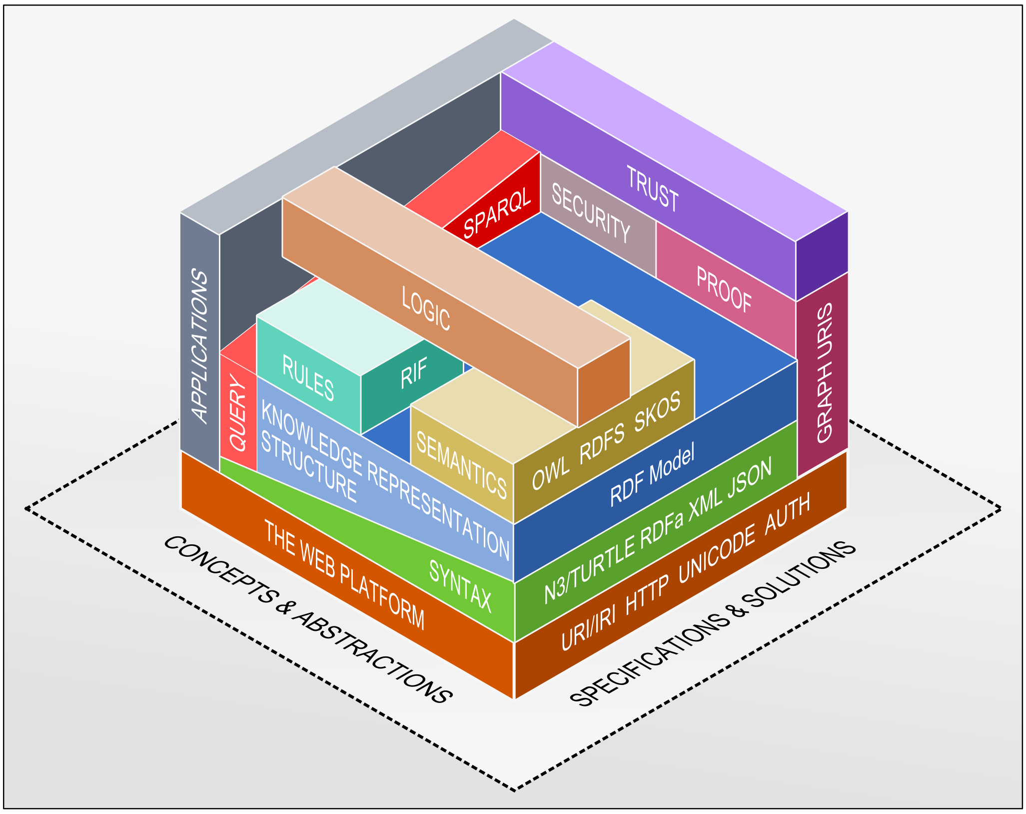

Figure 2.1: The common, layered Semantic Web technology stack (a modification of [Now09], see also [Gän11g])

[9]This does not mean that humans and machines always need to deduce the same meaning from a given piece of information, because neither all knowledge can be formalised.

2.2.2 Common Semantic Web Knowledge Representation Languages

The common knowledge representation languages for the Semantic Web are outlined in the following paragraphs. Different serialisation formats are utilised and designed to embody descriptions of the abstract specifications of them. The most important syntaxes are

- Notation 3 (N3) [BL06] and its subset Terse RDF Triple Language (Turtle) [BBLP10],
- Resource Description Framework in attributes (RDFa) [ABMH10] (as it can be embedded into (X)HTML [McC10, ABP10]),
- Extensible Markup Language (XML) [BPSM+08] (e.g. RDF/XML [Bec04a]) and
- recently JavaScript Object Notation (JSON) [Cro06] (RDF/JSON is currently standardised, see [sev11i]).

Resource Description Framework Model The Resource Description Framework (RDF) Model [KC04] is the core language of this framework. It is a KRS to connect resources by links (*binary relations*) that is inspired by *semantic networks* (see Section 2.1.1.2). *Arcs* are, at present, always *binary relations* (properties) in this model. The key difference between both KRSs is that with RDF Model all *arcs* and *nodes* can be labelled with URIs (cf. the *Principle of Universality* and the *Principle of Linking* in Subsection 2.2.1.2).
This name reference mechanism evokes unambiguousness. RDF Model allows to use HTTP URIs for references that can be resolved to access information on the Web. Nevertheless, URIs can be as ambiguous as natural languages labels. Their usage and their related information resources can be forced to enable interpretations in a sense of identifying a referent. There is no *Unique Name Assumption* due to the *Open World Principle* (see Subsection 2.2.1.2; cf. [Hal09]).
A graph modelled with the help of RDF Model, an RDF graph, is a directed, labelled, attributed, semantic multi-graph (property graph [RN10]). The *assertions* or statements in such a graph are specialised triples of the form 'subject-predicate-object' (RDF statement). This structure is similar to the form of simple natural language sentences. RDF Model provides the basis for supporting the crucial requirement of 'machine-understandable' *self-descriptions* in decentralised information spaces of which the Web is a *particular*.
Even languages themselves can be denoted by URIs to retrieve descriptions of them on the Web. A URI of a language is, in this context, the *namespace* of that language, for instance, `http://www.w3.org/1999/02/22-rdf-syntax-ns#` is the *namespace* of RDF Model. References of *namespaces* are often written in an abbreviated form by using prefixes, e.g., `rdf:` for the RDF Model *namespace*. Identifiers in this format are called CURIEs (Compact URIs) [BM10] and can easily be mapped to URIs and vice versa.

Resource Description Framework Schema The second KRL of the stack of common KRLs for the Semantic Web, RDF Schema (RDFS) [Bri04], is built on top of RDF Model. RDFS enhances the support for definition and classification. Therefore, it provides concepts for, e.g., resource (*entity*) and class (*kind*), inheritance, and domain and range definitions for properties. The formal semantics of RDF Model and RDF Schema [Hay04] considers the *Open World Principle* and obey the *Open World Assumption* (see Subsection 2.2.1.2). For that matter, inferences based on this semantics can only enrich the descriptions of existing triples. Generally, the term 'Resource Description Framework' (RDF) [MM04], without further descriptions, is often used for naming both RDF Model and RDF Schema.

Web Ontology Language Much richer logical inferences are possible with definitions that are based on terms of the Web Ontology Language (OWL) [W3C09]. With OWL one can define various axioms, for instance, inverse properties, symmetric properties, cardinalities, conjunctions or disjunctions. This language can be layered on top of the previously introduced languages, on the logical layer. Thereby, different OWL variants define the level of expressiveness by allowing or restricting specific terms of OWL, e.g., OWL DL aligns with the expressiveness of Description Logics [BCM+07]. In other words, the semantics of OWL DL is strongly related to the semantics of Descriptions Logics. In general, the OWL Semantics [Mot09, Sch09] is defined in a way that semantic graphs remain monotonic when extensions are added to them (cf. *Open World Assumption*; Subsection 2.2.1.2). This feature is inherited from the RDF semantics (see [Hay04]).

Rule Interchange Format One has to make use of a rule language to define and exchange more advanced inference rules (incl. usage descriptions) for domain or application depended reasoning. The Rule Interchange Format (RIF) [KB10] is particularly defined to enable the exchange (sharing) of rules that are commonly predefined, often by using a specific rule language, e.g., Prolog or N3Logic [BLCK+08], and processed in specific reasoning systems. Thus, RIF enables, for instance, the description of the semantics of a specific OWL profile as a set of RIF rules (see, e.g., [Rey10] for OWL RL). An RDF graph can be utilised in RIF rules [dB10], which themselves can be realised as an RDF graph [Haw10]. RIF itself, or the predefined RIF dialects, are constructed on the basis of an extendible framework to enable the creation of further dialects for exchanging rules with an expressiveness that is currently not covered by the given RIF dialects (see [BK10]). All in all, RIF should be nicely suited for describing and sharing KRM mappings, especially vocabulary mappings, e.g., of KRLs that are originally not Semantic Web ontologies (see [PHG+08]).

2.2.3 Resource Description Levels and their Relations

It is crucial to understand the circumstances of how the Web tries to *describe* and *represent*, or model, the world of which it is part of. This is especially crucial, since the most recent step in the evolution of the Web as Semantic Web is being realised. It is supposed to enable machines to try to "understand" knowledge representations. Therefore, the terms 'resource', 'information resource' and 'document' are defined and their relationships are described in this subsection (see Figure 2.2 for an overview). They are especially important to grasp KR as it is also propagated in the Web (see Section 2.1.1).

Resources In terms of the Semantic Web, a resource can be typed as `rdfs:Resource`[10], i.e., it can be characterised by a *description* that is related to this CURIE. One can use a URI to *denote* a resource, e.g., to *name* it in the Web. It is utilised for identification purposes in this context. So, a URI can be applied to *identify* a resource in some way. Even though, other resources can contribute to this identification process, e.g., a *description* that has as (a) topic the intended resource. One can call such a URI *resource URI*.

[10] `http://www.w3.org/2000/01/rdf-schema#Resource`

Copyright TU Dresden, Thomas Gängler

Information Resources An information resource is a piece of information that *describes* (or *represents*) a resource. It includes the amount of information that is necessary to *describe* a resource in a certain kind. Hence, the *subject*[11] of an information resource is this characterised resource. In general, an (abstract) information resource is usually *realised* by a semantic graph, which consists of a (concrete) *description* of a resource. This is necessary to make an information resource easily machine-processable. Of course, plain text is (more or less) easily processable by a human being, but not by a machine (cf. Section 2.1.1).
An information resource *represents* a resource in some way. Nevertheless, it cannot really *represent* a resource completely. This is simply grounded in the nature of things that we cannot really define what a complete *description* is to be about. Basically, there are always subjective *descriptions* (*assertions*), which are unforeseeable.
To be more precise, an information resource can be

- a Minimum Spanning Graph [ZX10], or
- an RDF Molecule [DPdSM05] (cf. [BL10b]), or
- a URI Declaration [Boo10], or
- a Concise Bounded Description [Sti05], or
- a *resource-description* [Dav07], or
- an *associated description* [BCH08].

It depends on the definition one specifies for "the amount of information that is necessary to *describe* a resource in a certain kind" (see above). An example of such a definition is given by the following rule set:

- statements that have the *resource URI* as object + dereferencing the subject + predicate URI of that statement for human-readable titles or names
- + statements that have the *resource URI* as subject + dereferencing the object + predicate URI of that statement for human-readable titles or names
- + (optionally) include all information of the statements of these object resources recursively, if they are part of this semantic graph, i.e., include at least all *blank node*[12] objects[13]

A resource can have multiple information resources. Each of them is *embodied* as a *representation*[14] that is *delivered* by dereferencing a *resource URI*. That is why, a resource can have multiple *resource URIs*. Every *resource URI* belongs to one information resource[15]. Due to that reason, a *resource URI* can be an alias for another *resource URI*, because there can be different *descriptions* (information resources) of one and the same resource.

[11] Here *subject* should not be considered in terms of the subject position of a triple in an RDF statement.
[12] RDF terms that have no URI.
[13] *Resource URIs* that are part of other semantic graphs are dereferenced only once for human-readable label retrieval. Otherwise, the algorithm would be a 'follow-your-nose' algorithm (see Subsection 2.2.1.2).
[14] *Representation* can be defined as "data that encodes information about resource state" [JW04], whereby "information about resource state" can be seen as *description*.
[15] Equivalence of information resources and follow-up inferences that two *resource URIs denote* the same resource is another big issue, which can not be clarified in this scope. For example, a *resource URI* can be related to an information resource that do not *describe* the *essence* of a resource. Hence, two such information resources are ambiguous (cf. definition of *substances* in [Lou02], and [Boo10]).

Documents A document can be used to *deliver* an information resource to an information consumer (as a specific information service, cf. Section 2.1.2.1). Therefore, it should include a semantic graph, which consists of at least a semantic graph of that information resource. In other words, a document *contains* at least one information resource. However, a semantic graph of a document can include further information - in Semantic Web terms, further RDF statements. Thus, a document is the envelope that is sometimes needed to *represent* or *deliver* an information resource to an information consumer.
It is a concrete *thing*, a *particular*. A document is a specific *representation*, e.g., a computer document. Such a sequence of bits has at least a content type, for instance `application/xhtml+rdfa` (see Subsection 2.2.2), and also a name, such as a *document URI*, for example, `http://example.com/test.html`.
Some information consumers do not need this document envelope yet. They can process, e.g., a semantic graph of an information resource without further information. These information consumers will simply get a serialised version of the information resource of a requested *resource URI* in an appropriate *representation* format, e.g., Notation 3 (see Subsection 2.2.2).

Resource Description Level Relations An overall expectation of an information consumer is that an information provider (as a specific information service, cf. Section 2.1.2.1) should *deliver* an information resource as response to a *requested* resource (*denoted* via a *resource URI*). To transfer this statement to the Semantic Web one can say:

> The Web of Things is built on top of a Web of (realisations of) **information resources** (that are delivered by information services; cf. [BL10a]).

Following this definition, an information resource can have multiple *representations* that at least *embody* a semantic graph of that information resource. A *resource URI* and a *document URI* can be the same, if a resource is a document. However, the relation of the resource description levels can be different in this context. Dereferencing that *resource URI* can deliver the information resource that *describes* this document

1. in a computer document that has the *document URI* as a name (a *self description*).
2. in another computer document that has another *document URI* as a name.
3. in another *representation* format.

All in all, at any time we should not directly infer knowledge about a resource from *resource URIs*, but always from semantic graphs (cf. [JW04] and "... the separation of layers ... is fundamental" [BL10c]). For this interpretation task we ought to especially utilise those concrete *descriptions* of the related information resources. These abstract *descriptions* are addressed by the *resource URIs* of a resource. In that context, *resource URIs* are simply present for dereferencing their information resources. Good URI design may sometimes help an information consumer to find a *requested* information resource. However, these interpretations are always expectations and are not fundamentally definitive.
Figure 2.2 illustrates the relationships of the described terms. Finally, one can say that everything can have an information resource (at least one). Additionally, one can claim, by insisting on Representationalism, that everything (every resource) is an information resource (cf. [Hal06, Gän10e]). Generally, 'information resource' is a massively debated term[16], especially in the Web science community (see, e.g., [BL09a, BL09e, BL09d, Wan07, Hay08, sev09b, RBH10]).
An attempt of an abbreviated definition of the term 'information resource' is given in the following (see [Gän10e]):

[16] This discussion includes the alignment of the term 'information resource' to the terms 'representation' and 'document' as well.

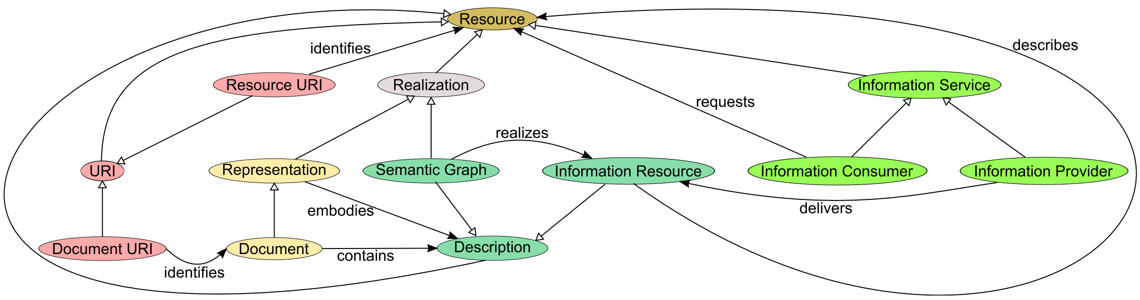

Figure 2.2: The relationships of the concepts resource, information resource and document and related ones

> An **information resource** is a resource which can convey or describe (essential) characteristics of a resource in a particular way, e.g., as a semantic graph. This description can, for example, be realised (or embodied) as a concrete message, e.g., a serialisation (representation) of a semantic graph in Notation 3 syntax. The resource can also be the information resource itself, in which case it is referred to as a self description.

It is aligned to the definition of 'information resource' that is given by the Technical Architecture Group (TAG) [JW04] and Harry Halpin's reflection of this TAG definition (see [Hal09]). So, an information resource itself is more like a piece of *abstract information* [Hal09], rather than a concrete document. Albeit, for simplification purposes we will use the term 'information resource' to refer to both *abstract information* and particular *realisations* (cf. [Hal09]).
One has to keep in mind that ambiguousness always exist. We simply only have a *partial understanding* [BL09b] of *things* we perceive[17] at a specific temporal and spatial point. In that context, we cannot further reason to get the *essence* of a *thing*. It is a natural sensation, which is not bad at all. *partial understanding* is even very helpful most times. However, it does not prevent us to reach a level of unambiguousness, which is not always necessary (cf. [HH08]).

2.2.4 Semantic Web Knowledge Representation Models

We can dynamically create and share all kinds of knowledge representations of different complexity to properly describe parts of the world. In this KM and KR exemplification, the modelling is realised on the basis of the common KRLs for the Semantic Web (see Subsection 2.2.2). Besides, the provision and consumption of these knowledge representation models require an understanding of the relationships between resources, information resources and documents (see Subsection 2.2.3). The following subsections explain different aspects of the information flow life cycle of Semantic Web KRMs, i.e., construction, mapping, context modelling, storing, providing and consuming (see [HB11] for a quite good reading on that topic). These tasks are common in OKMSs (see Subsection 2.1.2).
Semantic Web programming environments that are specialised in dealing with such KRMs are available in, more or less, every programming language (see [sev09c] for a good overview). Two Semantic Web frameworks with more comprehensive capabilities are *Jena* [Sea10] and *Sesame* [Adu09].

[17] Perception is/is a kind of/can be seen as dereferenciation or vice versa.

2.2.4.1 Construction

The construction of KRMs is the foundation of the information flow life cycle in the Semantic Web. On the one side, we can decentralised build and publish extensible vocabularies for specific domains or purposes, e.g., a bibliographic ontology (e.g. the Bibliographic Ontology [GD10]) or a music ontology (see Section 2.3.2). These specifications are Semantic Web ontologies that contain descriptions of *universals*. Thereby, domain-specific vocabularies can be constructed with the help of so-called upper level ontologies, e.g., the Upper Mapping and Binding Layer (UMBEL) vocabulary [BG11a]. The mentioned example contains a broad range of more general concepts and *relations* that are intended for specialisations. UMBEL itself particularly makes use of the Simple Knowledge Organisation System (SKOS) vocabulary [MB09], which is a KR vocabulary whose specific purpose is to model knowledge organisation systems, such as thesauri, taxonomies or classification schemata. On the other side, we can flexibly apply Semantic Web ontologies to collectively create and distributively provide instance data - *particulars* - to establish connectivity and share our knowledge.
It is important to initially utilise as much appropriate, existing Semantic Web ontologies as possible for describing *particulars* and, if needed, new *univerals* as well. This step helps to easily establish "shared meaning", of knowledge that should be conveyed, directly (cf. Section 2.1.1.3), rather than having to perform an expensive mapping of instance data whose descriptions are based on proprietary vocabularies. Thus, only if reutilisation cannot be applied to fit the requirements for a specific KR, one has to create a new ontology for that purpose. Hopefully, these new vocabularies are aligned to existing Semantic Web ontologies. This can be achieved, e.g., by following the top-down approach.
Noteworthy tools that particularly support the development of Semantic Web vocabularies and ontologies are TopBraid Composer [Top11], Protégé [Sta11]. Both are full-fledged ontology modelling tools. A lightweight, web-based one is Neologism [DER11].

2.2.4.2 Mapping

A program that creates a mapping from knowledge representations which are not powered by Semantic Web KRLs into a Semantic Web KRM is often called *RDFizer*. Such source KRs can be, e.g., stored in a database or described in another language. Most of the retained and published data today is not available as a Semantic Web KR. That is why, these tools are essential for information integration tasks in OKMSs (see Section 2.1.2.2). They are available in all variants and forms.
This can be, for example, *RDFizer* for existing

- metadata formats, e.g., ID3 [ADG10] or EXIF [Maz06], or
- data description formats of existing information services, e.g., *Last.fm* (see Section 2.3.3.2) or *Echo Nest* (see Section 2.3.3.3), or
- database schemata, e.g., of relational databases (*D2R* [BC09]), or
- content analysis', e.g., audio signal analysis' (*Sonic Annotator* [KC08]).

They are available in different complexities. Thereby, the applied mapping technologies are varying

- from transformation standards, such as, Extensible Stylesheet Language Transformation (XSLT) [Kay07] whose descriptions can be
  - linked by using Gleaning Resource Descriptions from Dialects of Languages (GRDDL) [Con07] or

  - processed in generic mapping components, e.g., *Virtuoso Sponger* [sev10k], of bigger software environments

- to whole, extensible frameworks in different programming languages, for instance, *Aperture* [AD10] (written in Java), that can itself be processed in generic mapping components á la *Virtuoso Sponger*, too.

All mentioned *RDFizers* for specific data sources are often mere examples. Usually there are different mapping implementations. Everybody can write an own *wrapper* whose mapping differs from the existing ones, e.g., by utilising different Semantic Web ontologies. A good and maintained general overview of available *RDFizers* is given on the W3C community wiki[18] (see [sev10b]). Besides, a *Virtuoso-Sponger*-specific one can be found on the *Virtuoso* wiki[19] (see [sev10h]).
Besides the proposed approaches, a vocabulary or data source mapping can be described by utilising a rule language, e.g., RIF (see Subsection 2.2.2). This approach is especially recommended for mappings between KRMs that are already powered by Semantic Web KRLs. In addition, Link discovery frameworks, such as *Silk* [IJBV11], can be utilised on the *particular* level of Semantic Web KRMs to deploy interlinking strategies for resources of different datasets.
Finally, the design of mappings from relational database schemata to (existing) Semantic Web ontologies and recommendations to the mapping process itself (incl. a utilisation of reusable identifiers) can be described by uniform specifications, e.g., the RDB[20] to RDF Mapping Language (R2RML) [DSC10]. Currently, these developments are in the process of being standardised (see [AFM+10]).

2.2.4.3 Context Modelling

The modelling of context and the separation of different context types for efficient processing causes difficulties for the triple structure of RDF Model (see [Tol04, GMF04, MK03]). A quite general distinction between *internal* and *external context* was stated by Karsten Tolle in [Tol04]. This differentiation was simplified in [MK06] as

- *internal context* is semantically related information, e.g., grouped by a named semantic graph (cf. information resource, Subsection 2.2.3)

- *external context* is semantically unrelated information, e.g., provenance, trust, privacy and security statements (see [GCG+10] for a deeper view of this topic).

Both context types have a specific importance of their own regarding KM tasks, e.g., information service selection (mainly *external context*) and information selection in information federation tasks. It is usually a difficult task to find an appropriate set of statements, in a given situation, that represent a satisfiable amount of *internal context* (cf. Subsection 2.2.3 and [Tol04]).
The inbuilt RDF reification vocabulary has no formally defined semantics (see [Bri04, MM04]) and is disreputable. Often, it unnecessarily increases the number triples, e.g., when describing provenance information in data aggregation tasks (as outlined in [CS04]). Non-standardised extensions of the triple structure of RDF Model usually result in quadruple formats. Thereby, the forth element commonly is of different usage and semantics. It is used, for example, as

- statement identifier for reified statements [Kly00],

- graph identifier á la Named Graphs [Car08] (that have RDF based semantics, see [CBHS05]) for grouping triples and naming semantic graphs, or

[18] `http:/w3.org/wiki/`
[19] `http://virtuoso.openlinksw.com/dataspace/dav/wiki/Main`
[20] relational database

- more general context identifier as implemented in N-Quads [CHH09].

Generally, approaches in which context separation is reflected more explicitly in the KRS end up in polyadic structures with tuples that have more than four elements. This can be, for example, quintuples, as applied in, e.g., the Ontology Representation and Data Integration (ORDI) model [MK06]. Finally, Karsten Tolle concluded in [Tol04] that there cannot really be a universal approach to a KRS that equally satisfies all needs of context modelling in its different types.
We might use the forth element of quadruples explicitly for statement identifiers whose descriptions consist of *external contexts* (see [Gän11e]). Consequently, we have to consider the abstract concept of reification as an approach for refining semantically related information (*internal context*). We call this kind of reification *property-oriented context reification*. Both description variants - the short and the detailed (reified) one - can be used interchangeably. One can, for example, expand *binary relations* to *n-ary relations* (cf. [Noy06] and its various renamed relatives with quite similar intentions, e.g., *Extended Triples* [Hec05], *Record Description* [sev05], *On-the-Fly Properties* [sev07c], *Curried Function* [sev07a]). Besides, statement identifiers are often implemented by RDF storage systems internally (see next subsection) and are supported by Semantic Web frameworks (e.g. *Redland* RDF libraries [Bec11]; see [Bec04b]) already.

2.2.4.4 Storing

As explained in Section 2.1.2, knowledge representation models of KMSs are stored in KBs. Those knowledge bases that have enhanced capacities for storing Semantic Web KRMs are commonly called *triple stores*, due to the specific KRS of RDF Model that has the form of triples (see Subsection 2.2.2). They benefit from the standardisation of the Semantic Web technology stack (see Subsection 2.2.1.3). Usually, *Triple stores* are natively implemented in specialised graph database systems. This is a category which is part of the recently aggregated, popular NoSQL ("Not only SQL") solutions (see [Edl11] for a good overview). These database systems, generally, have a special focus on *horizontal scaling*. Semantic Web KRMs can be stored in database system of other categories as well. These are, for example,

- relational database systems (by deploying a specific database schema, e.g. that one from *Jena* [Hew09]),
- document database systems (e.g. *MongoDB*) [Imb10]), or
- one stores the information simply in file(s) on a (distributed) file system (e.g. *SHARD* [Roh10]).

Basically, it is a crucial issue that distributed database system architectures for handling large amounts of data ("web-scale") must scale vertically (machine-wide performance) and horizontally (cluster-wide performance). Besides supporting the inherent requirement of high-performance query processing of simple and complex requests (query engine), KBs ideally have to implement complex inferencing capabilities too. The latter feature realisation is often called inference or reasoning engine. Both engine types (query and inference engine) should be able to work on highly interlinked large semantic graphs.
Currently, large scale (distributed) *triples stores* often will only perform well on a certain satisfiable level by supporting basic inbuilt (optimised) inference capacities. Usually, these features rely on RDF and OWL DL semantics (see Section 2.2.2). However, inference engines mostly have interfaces for adding further inference rules for reasoning. An information service can also perform inferencing tasks by utilising a separate reasoner, e.g., Pellet [Cla11], since reasoning engines are only an optional part of a *triple store* environment.
Commonly, *triples stores* are implemented at least as *quad stores* to be able to store, for instance, the identifiers of named semantic graphs that are normally URIs, too (see Subsection 2.2.4.3). Thereby, semantic graphs are used as natural partition mechanisms, e.g., for data source separation. Descriptions of semantic graphs are intended to specify provenance information, e.g., the party which stated the *assertions* in the named semantic graph. Such quadruples can be serialised as *Named Graphs* or the more general *N-Quads*.
Scalable, distributable, native *triple stores* are, e.g.,

- the light-weight (no inference engine), highly scalable and very fast (commercial) *5store* [Gar10] as the descendant of the open-source version *4store* [Gar09],
- *AllegroGraph* [Fra10], which has a Prolog implementation for enhanced reasoning tasks on top of the basic inference capabilities and extended queries,
- *BigOWLIM* [Ont11], which performs materialisation of inferences on loading tasks and implements the ORDI model, and
- *Bigdata* [SYS10], which is especially designed for *horizontal scaling* and implements, therefore, a dynamic key-range partitioning of indices to allow an incremental capacity-increase that does not require a full reload of all data.

A quite comprehensive piece of software is the multi-purpose and multi-protocol data server *Virtuoso* [Ope10b, sev10j]. The hybrid database system of this server architecture can handle relational, graph, hierarchical and document data models. Additionally, the *Virtuoso* universal server has a built-in Web server component for serving information (incl. Web Service support). The mentioned generic mapping engine *Virtuoso Sponger* is a part of *Virtuoso* (see Subsection 2.2.4.2). Enhanced support for generating RDF views of existing relational databases is given by a HTML-based wizard (see [sev09a]). The powerful *Virtuoso* has even more capabilities to act as or support a modern OKMS. Its most recent feature extensions are listed on the *Virtuoso* homepage[21] (see [Ope10a]).
A good, however not complete overview of the landscape of existing *triple stores* or tools with *triple store* components is given on the W3C community wiki (see [sev09d]). All in all, due to the reason that *triple stores* currently are the only kind of a standardised NoSQL solution, it should be much easier to change a backend database system, when a new one is needed for, e.g., scalability issues or new functional requirements (cf. [Ben10]).

2.2.4.5 Providing

An information service that publishes information resources should, thereby, follow the principles of Linked Data. These present a general data publishing guideline that can satisfy the needs of the resource description levels, which are explained in Section 2.2.3. For that matter, it might be beneficial to guide the architectural design of a KMS of an information service by the set of architectural constraints of the REST architectural style to enforce reliability and a uniform interoperability (cf. Section 2.1.2.3).

The Linked Data Publishing Guideline Initially (in summer 2006), Tim Berners-Lee composed the Linked Data publishing guideline [BL10b] to address shortcomings of predominant data providing practices in the Web and to promote the benefits of utilising Semantic Web KRMs for that issue. Later, people started realising the independence of these principles from concrete applied technologies, i.e., the Linked Data publishing guideline is generally independent from Semantic Web technologies. This movement is especially propagated by Kingsley Idehen (see [Ide11d]). The Linked Data publishing guideline addresses the following five main principles (see also [Gän11b]). Whereby, the fifth one is a recently made, optional addition that was suggested by Matthew Rowe (see [Row10]).

1. Use a **name reference mechanism** for resource identification, e.g., URIs (cf. the first interface constraint of the REST architectural style, see Section 2.1.2.3).

[21] `http://virtuoso.openlinksw.com`

2. Use an **information resource delivering mechanism** on top of that name reference mechanism, e.g., a URI scheme, for example, HTTP URIs, to enable the transfer of a document that contains at least the information resource to a requested resource identifier (cf. the second interface constraint of the REST architectural style, see Section 2.1.2.3).

3. Use a **resource description mechanism** to facilitate the expression of knowledge (*assertions*) regarding the denoted resource, e.g., RDF (see Subsection 2.2.2; cf. the forth interface constraint of the REST architectural style, see Section 2.1.2.3).

4. Provide a **resource connection mechanism** to permit the embedding of links. These refer to other resources to establish meaningful (as unambiguously as possible) interpretations of a received information resource. This enforces knowledge exploration, e.g., via the HTTP Link header field [Not10]. The mentioned example resource connection mechanism has to be chosen, if, for instance, the specification of the media type does not enable link embedding in the representation (cf. "follow your nose", see Section 2.2.1.2, and the fifth interface constraint of the REST architectural style, see Section 2.1.2.3).

5. Provide an **information resource modification mechanism** by (optionally) allowing the transfer of (a) modification(s) of received content by using the resource identifier of the request. This can be, for example, changes that are entered in a (HTML) form and sent to the server by applying the HTTP PATCH method [DS10] (cf. the third interface constraint of the REST architectural style, see Section 2.1.2.3).

In particular, the last principle directs into a read and write enabled Linked Data information space scenario (see [BL10e]). A good overview of relevant specifications for Linked Data in the context of the Semantic Web can be found on the *linkeddata-specs.info* website [Hau10b]. Overall, this guideline, applied on Semantic Web technologies, conforms to the *normative Principles of Web architecture* (see Subsection 2.2.1.2; cf. [Hal09]). The *Virtuoso* universal server (see Section 2.2.4.4), for example, provides an enhanced Linked Data deployment support (see [sev10c]).
Since Semantic Web vocabulary specifications are specialised information resources for describing *universals*, they have to be published by following the principles of Linked Data as well. Providing a documentation with good explanations, visualisations, and examples is a necessity to enforce a "shared understanding" as well as applicability and reutilisation. Fortunately, some helpful tools already exist for assisting the specification documentation generation, e.g., *SpecGen* [sev10i, Gän10d].
The same guideline has to be applicable for the publishing of rules for inferencing processes. This is an application of the "code on demand" paradigm (see Subsection 2.1.2.3; cf. [Rai08a]). Publishing of rules is, at the moment of writing this thesis, rarely applied. A reason for this might be that rule usage descriptions are currently only supported inappropriately in the context of rule provision and application scenarios (cf. [sev10e, sev11g]).

Content Negotiation Overall, it is vitally important that information providers deploy a content negotiation mechanism for handling requests and providing appropriate representation formats for the responses, e.g., that one of the HTTP protocol. For that matter, it is beneficial to provide as much appropriate serialisations as possible to broaden the range of reachable information consumers and to be able to deliver a response in the best fitting representation format (cf. Subsection 2.2.1.2). Content negotiation is one mechanism to enforce durability of resource identifiers. Another one is the application of so called ’cool URIs’ [BL98]. This guideline propagates a careful treatment of URI changes as a consequence of content changes. In addition, it suggests publishers to keep resource identifiers that deliver content for a specific purpose alive, e.g., URIs that are used to permanently deliver the latest version of a resource description. This type of URIs are usually called permanent URLs (PURL).

Datasets Description Finally, a publication of a Linked Data dataset[22] is to contain a description of the content and its quality, and further relevant information, e.g., references to the information provider, contributors, or information services whose information resources are consumed in that dataset. A vocabulary that is intended for the description of Semantic Web KRMs is the voiD Vocabulary [ACHZ10]. Moreover, a good starting point for information quality concerns of Linked Data is the *qualitywebdata.org* website [Hau10c].

2.2.4.6 Consuming

Ideally, an information consumer should be able to apply a small set of actions, which have universal semantics, to an abstract component interface to deal with pieces of information that are served by an information provider (cf. the second interface constraint of the REST architectural style, see Section 2.1.2.3). Albeit, not every information resource is initially published in that form. For that reason, an information consumer should be able to perform KM tasks directly via an interface to the query engine of the KB.

SPARQL Protocol and RDF Query Language The standardised SPARQL query language [HS10] is primarily designed to fulfil the above mentioned requirement for *triple stores* (see Subsection 2.2.4.4). That is why, nearly every *triple store* query engine implements the specification of that interface (at least in a certain level). SPARQL facilitate the opportunity to perform arbitrary queries on so-called *SPARQL endpoints*. Not only are information requests possible, but also resource description constructions, modifications and deletions (see [SGP10, Ogb10]).
To propagate these functionalities, information providers can serve a description of their SPARQL service(s), e.g., by utilising the SPARQL Service Description Vocabulary [Wil10]. This gives an overview of the capabilities of their endpoints of underlying KBs. The SPARQL query language provides an extension that especially addresses the requirements of information federations tasks [Pru10]. This addition supports query executions over the boundaries of *SPARQL endpoints*.
Finally, one can also describe queries with the help of the Semantic Web knowledge representation language SPARQL Inferencing Notation (SPIN) [Knu09]. This KR vocabulary includes a mapping of the native SPARQL syntax onto an RDF-Model-based one. A publication of queries by following the Linked Data publishing guideline is easily realisable by utilising SPIN (see Subsection 2.2.4.5). Besides, the capability of describing SPARQL queries and functions, SPIN can be used to express rules or constraints that are processable by a reasoning engine. Therefore, it has features of a rule language (cf. RIF, see Subsection 2.2.2).

Authentication and Authorisation Generally, it is a crucial prerequisite for an information provider to implement control mechanisms that information consumers can perform various KM tasks, e.g., a modification task, in a secure and privacy-aware distributed KMS. Authentication and authorisation components that can satisfy that need must be inbuilt in such systems and usable by their information consumers. Two protocols that are specialised to handle the authentication and identification process in a decentralised manner are the quite well established OpenID protocol [Ope11] and the recently designed WebID protocol[23] [SIS+11]. The latter one utilises well-known Internet technologies for security (e.g. the Transport Layer Security (TLS) protocol [DR08] and the X.509 standard [ITU11]) and Semantic Web vocabularies (e.g. the *Friend of a Friend Vocabulary*; see Section 2.4.1). Thereby, it follows the REST architectural design principles (see [SHJJ09] and Subsection 2.1.2.3).
On the other side, the authorisation process can be managed with the help of the quite popular OAuth protocol [sev10g] or the experimental Web Access Control (WAC) system [sev10l]. The second one is a decentralised implementation of access control lists. WAC can be used elegantly in conjunction with the WebID mechanism. A combination of these techniques in a system is in line with the principles of Linked Data (see Subsection 2.2.4.5) and the REST architectural style guideline (see Subsection 2.1.2.3). The *Virtuoso* universal server (see Section 2.2.4.4), for example, provides an enhanced WebID deployment support (see [sev10m]).

[22] Semantic Web KRM that is published by following the Linked Data publishing guideline
[23] This protocol is still in an early development state, see [HS11].

Change Propagation Finally, the propagation of changes is essential for information consumers of all kinds in a distributed KM environment. On the one hand, this issue requires a change detection mechanism, such as a change notification implementation, e.g., the PubSubhubbub protocol (PSHB) [FSA10] or the Simple Update Protocol (SUP) [BBCT10]. The former one is optimised for efficient server-to-server communication, because it is event-driven. Information consumers register on a hub to receive updates of relevant information resources on time. On the other, SUP aggregates digests of changes and clients can poll them to receive a complete update of subscriptions as needed.
Both protocols can especially utilise the well-known standard for syndication of Web content, the Atom Syndication Format (Atom) [NS05]. This can be done in combination with the Atom Publishing Protocol (AtomPub) [Gdh07]. The introduced change notification techniques can be deployed side by side in a distributed KMS, because PSHB and SUP are complementary regarding their features. A PSHB application in the context of Semantic Web technologies is sparqlPuSH [PM10]. It implements the PSHB change propagation mechanism for *SPARQL endpoints*. The *Virtuoso* universal server (see Section 2.2.4.4), for example, provides an enhanced PSHB deployment support (see [sev10d]).
On the other hand, changes themselves and their semantics have to be described for Semantic Web KRMs. For example, SPIN or the Changeset Vocabulary [TD09] can be utilised for describing changes on the triple level. An ontology such as the DSNotify Eventset Vocabulary [PH10], can be used for handling this issue on the information resource level. Thereby, both views can be connected with the help of the this ontology. Furthermore, the DSNotify Eventset Vocabulary can be applied when performing integrity maintenance tasks in a distributed KM environment (see [PHMR10]). Such description formats can be aligned with the expression capacity of Atom.
Moreover, the Dataset Dynamics Vocabulary [Hau10a] can be utilised to describe coarse-grained change characteristics of datasets. Besides, this ontology provides *relations* to refer to change notification mechanisms on that level. This makes them easily discoverable.
Overall, the issue of dataset dynamics handling is still not completely resolved. That is why, it is a topic of active research (see [sev11b] for a good overview of this topic). At the end, it is especially important to consider information services that do not handle or support Semantic Web KRMs (see [sev11h]). Currently, they represent the majority on the Web.

2.2.5 Summary

This section shows the way in which Semantic Web technologies can be applied for implementing a system that deals with knowledge management (as described in Section 2.1). Therefore, it is illustrated that the Semantic Web is one step of the evolution of the Web. It was outlined right from the beginning of the WWW vision and still needs its time to evolve properly. Since KRLs build the foundation of every KRM, the common Semantic Web KRLs are introduced after the prelude of this section. The core of the *Web of Data* is resource description. That is why, it is important to know the different levels of abstraction of resource description and their relations. This issue is represented in Section 2.2.3.
Finally, the Semantic Web technologies section ends with a summary of aspects of the information flow life cycle of Semantic Web KRMs. It illustrates that is especially important to close the information flow cycle, i.e., information services ideally act as both - information provider and information consumer. For that reason, information consumers have to enable the creation, modification and removal of information resources and push back changes to its original information providers.
The next section provides an overview of music content and context data, its metadata formats and relevant information services of that domain. Thereby, the application of Semantic Web technology is explained by appropriate examples.

2.3 MUSIC CONTENT AND CONTEXT DATA

Music description is divided into two main categories - music content data (musical information) and music context data (music-related information). The former addresses music itself, e.g., expressed via an audio signal. The latter one specifies knowledge that is related to music, e.g., a music artist or a music genre that is formed by community.
At the beginning, the main categories of musical characteristics are introduced and an overview about their analysis process is given (see Subsection 2.3.1). This part is followed by a short summary of available music metadata formats (MMF) and a representation of an appropriate Semantic Web ontology framework for the music domain (see Subsection 2.3.2). Finally, this section closes with a categorisation of music metadata services (MMS) that includes noticeable ones of each type (see Subsection 2.3.3).

2.3.1 Categories of Musical Characteristics

Overall, the complex phenomenon of music in its full range can be described by manifold concepts and subjectively perceived features. In a nutshell, these are musical characteristics. Their origins stem from a variety of sciences, for example, physics, psychoacoustics, music theory, and musicology. They can also be established by a community or culture.
Pachet [Pac05] introduced a music metadata classification with three categories for being able to better differentiate the aspects of music context data. Hence, music content data bears the type *acoustic metadata* and music context data can be of the type *editorial metadata* or *cultural metadata*. Another orthogonal and more universal division is that one of subjective, e.g., a dedicated mood, and objective descriptions, e.g., a song title (see [DHSW02]).
Generally, musical characteristics are highly versatile. They can be arranged on different levels and sub categories. Figure 2.3 illustrates an example of such a classification, which is taking its clue from the Musical Audio Mining (MAMI) conceptual framework [Les06]. This categorisation was already introduced and fully explained in my Belegarbeit [Gän09a]. It consists of three main types: basic audio signal features, derived audio signal features and taxonomies. Classifications of the latter ones are derived from features of the former ones. Besides, entities of taxonomies, e.g., music genre taxonomies, can be evolved from music context data or a combination of both main types of music description.
Even the process of feature extraction and derivation from audio signals is a rather complex one. Figure 2.4 illustrates an abstraction of the main tasks that are necessary to gain compact descriptions that are known as *musical fingerprints* from given audio signals. They should ideally act as a robust and correct song identification mechanism and be applicable for a quick song recognition. That is why, they have to be efficiently computable and as compact as possible (see [Can06]). Fortunately, the whole audio signal analysis process can be outsourced and various music information services provide different functionalities to support this task (see Subsection 2.3.3).
Besides an audio signal analysis task to retrieve music content data, a full-fledged work flow of the analysis of musical characteristics includes a music context data extraction process. Thereby, a metadata enhancement task makes use of further information services, instead of only relying on information that can be gained from a music document (see Figure 2.5). It is important to note that the processes of classification, categorisation and similarity calculations have to be driven by user profiles (see Section 2.4) in order to be adaptable [Gän09a]. Finally, fuzzy and abstract feature descriptions are necessary to enable an intuitive handling of the music KB.
Please have a look at my Belegarbeit [Gän09a] to get a deeper insight into the background of music content and context data, and the non-trivial extraction and derivation process of musical characteristics. Even this whole analysis process can be outsourced (as needed) to external music information services (see Subsection 2.3.3).

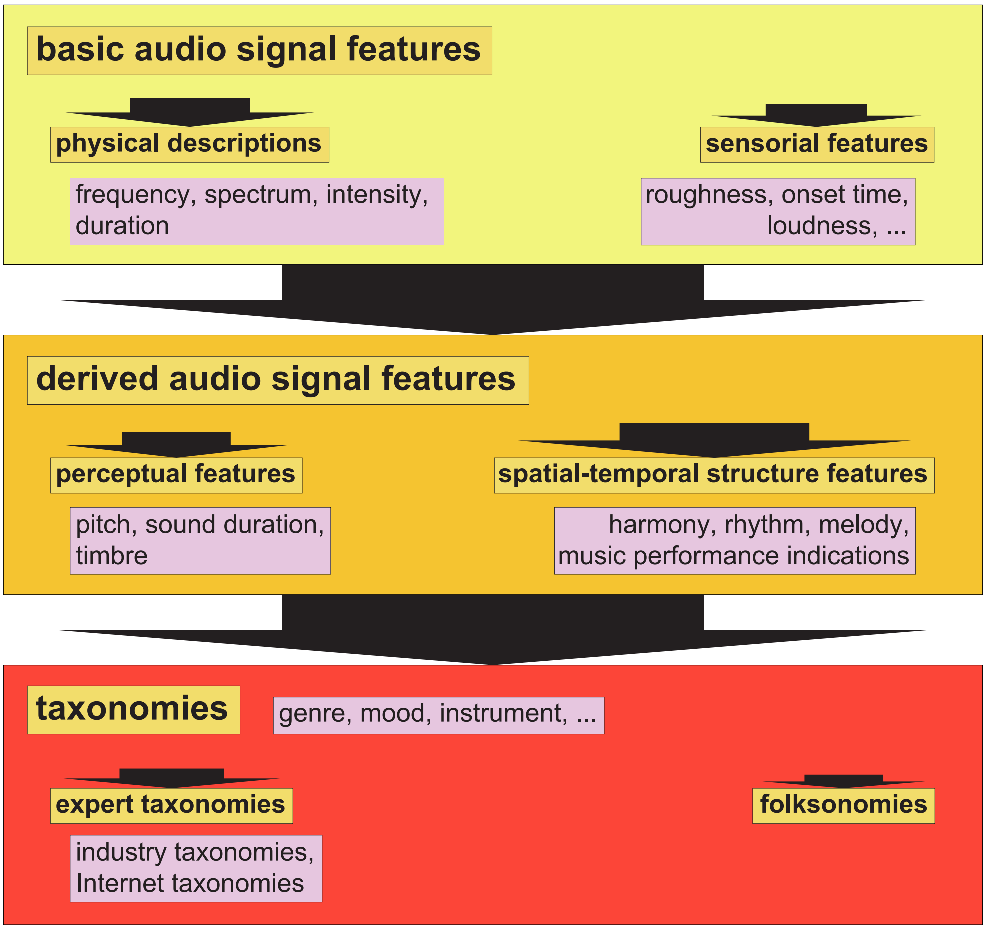

Figure 2.3: Categories of musical characteristics with subcategories and examples (lilac coloured) [Gän09a]

2.3.2 Music Metadata Formats

Due to the evolution of description formats for music content and context data, there are presently manifold specifications available for representing various musical characteristics to a different complexity. A main division can be made regarding their bondage to audio signal file formats [Gän09a]. On the one side, there are *format-bound specifications*, e.g., the well-known ID3 tags [O'N10] or Vorbis Comments [sev11j]. These are predefined data containers that are part of the audio or multimedia files of the respective file formats, e.g., MP3 [Nil00] or OGG [GPM08]. On the other side, there are *format-independent specifications*, e.g., the Music Ontology framework (see below) or MPEG-7 Audio [MSS03]. They usually make use of different data models, e.g., semantic graphs, and description formats, e.g., RDF or XML, for the purpose of defining specifications and for representing instantiations. Music metadata formats of MMSs also belong to this category (see Subsection 2.3.3). Nevertheless, they are often based on proprietary database schemata, e.g., the *Next Generation Schema* (NGS) of *MusicBrainz* (see Section 2.3.3.1). For that matter, explicit mappings are always required in a information federation KMS to process information resources that are originally represented with the help of such varying KR vocabularies. Corthaut et al. did an analysis of various common MMFs [CGVD08] by comparing their applicability to a set of important music application domains, and their coverage regarding a set of *universal* clusters (semantically related concepts and relations). They concluded that there was not a single music metadata standard that fulfils all requirements of the different application domains at once. However, *format-independent specifications* in general attempt to compensate

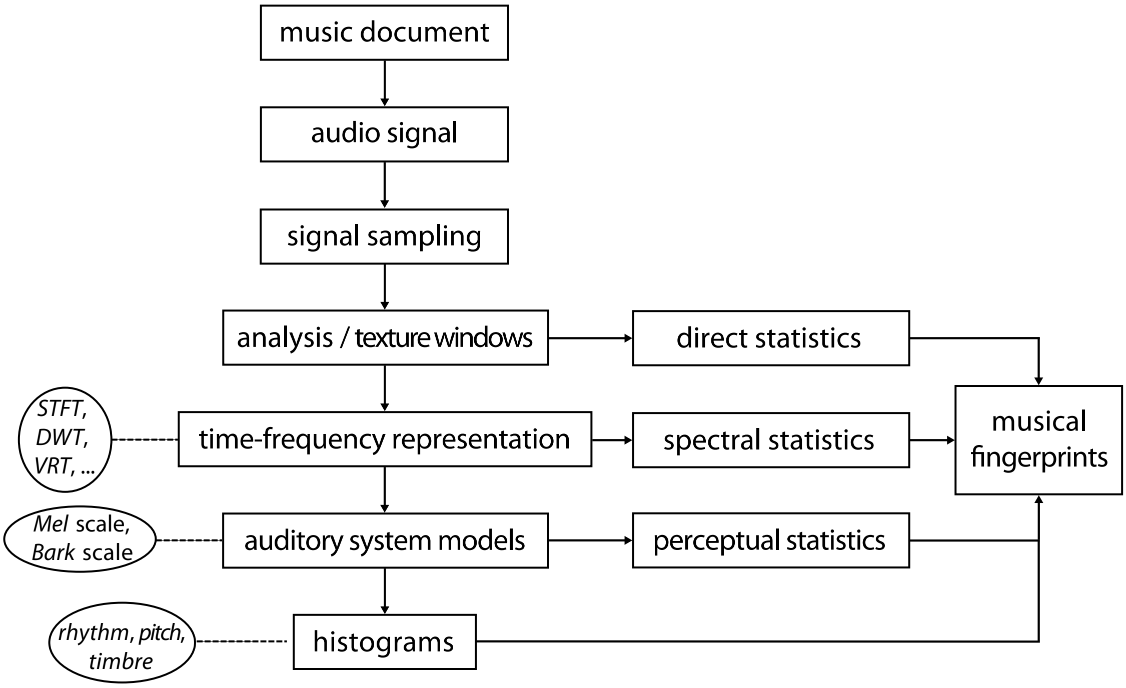

Figure 2.4: The audio signal feature extraction and derivation process [Gän09b]

weaknesses of *format-bound specifications* to enhance interoperability [Gän09a]. One outcome of the MMF analysis that I undertook as a part of my Belegarbeit [Gän09a] and which can be seen as a succession of the research of Corthaut et al. [CGVD08] was, that it is important to utilise an open, variable, and extensible KR framework, such as Semantic Web ontologies. Such a set of KRLs is to be able to satisfy evolving requirements of different music application domains.

The Music Ontology Framework A framework of Semantic Web ontologies that establishes a good foundation for describing musical characteristics is the Music Ontology framework as it is introduced by Raimond et al. in [RASG07]. It is explained, compared and evaluated in detail in the dissertation of Yves Raimond [Rai08a]. Besides, the analysis of Corthaut et al. [CGVD08] and that one carried out in my Belegarbeit [Gän09a], both demonstrated the expressiveness and applicability of the Music Ontology framework, especially regarding the requirements of describing personal music collections.

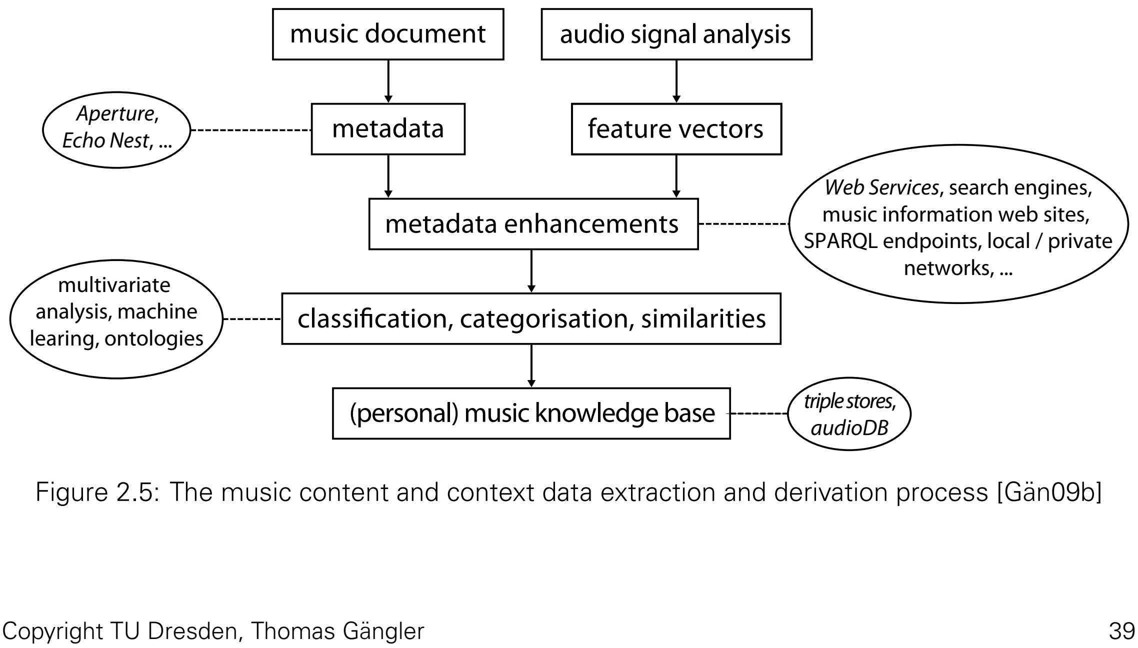

Figure 2.5: The music content and context data extraction and derivation process [Gän09b]

This vocabulary framework consists of several Semantic Web ontologies that attempt to heavily involve existing KR definitions, e.g., the *Friend of a Friend Vocabulary* (see Section 2.4.1) or the *Functional Requirements for Bibliographic Records Vocabulary* (FRBR) [DN09]. Furthermore, it has a strong aspect on modularisation and reutilisation. For this reason, several *simple specialised ontologies*[24], which represent a concrete aspect, are an outcome of the development process of the Music Ontology framework. These are, for example, the *Timeline Ontology* (TL) [RA07b], which is designed on top of the *OWL Time Ontology* (TIME) [Hob06] to express a timeline concept, or the *Event Ontology* (EVENT) [RA07a] to be able to model events as first-class entities.
The core ontology of this framework, the *Music Ontology* (MO) [RGJ+10], amongst others, makes use of these *simple specialised ontologies*. This vocabulary is divided into three main levels, which are related to a different expressiveness. Starting from basic *editorial metadata* (e.g. music artist, track or label), over terms that can be used to describe a creation workflow (e.g. composition, performance, recording), to event decomposition (e.g. to describe the structure of a song). Especially, the last level is mainly modelled in separate ontologies that are built on top of the core ontologies of the Music Ontology framework, e.g. the *Audio Features Ontology* (AF) [Rai08b].
Besides the MO specification documentation [RGJ+10], a community[25] wiki [sev11e] provides further explanations, examples, etc. For instance, comprehensive illustrations of different conceptual parts of this vocabulary (see [RGG10]) or an exhaustive example, which contains descriptions starting from an abstract idea of a music album and its songs to concrete items of a release of this album that are owned by someone (see [Gän11c]), are available in the wiki. An overview of extension modules and related ontologies, e.g. the *Similarity Ontology* (SIM) [JRG10], can be found as well on this website (see [sev11c]).
The *Music Ontology* is one of the most popular Semantic Web ontologies (see, e.g., its usage on the Linked Open Data cloud [BJC11]). It is utilised in, e.g., *BBC Music*[26], *Libre.fm*[27] and *MusicBrainz* (via the *LinkedBrainz* project [DJ10]). Moreover, several *SPARQL endpoints* of proof-of-concept datasets are accessible on *DBTune*[28] that demonstrate the manifold applicability of MO.

2.3.3 Music Metadata Services

Music metadata services are intended for analysing, categorising, identifying, searching and recommending songs [Gän09a]. As already mentioned in the prior section about categories of musical characteristics (see Subsection 2.3.1), there are various web information services available that primarily deal with music content and context analysis (see Subsection 2.3.3.3). In addition to that type of MMSs this kind of information services can be classified with the help of two more categories (cf. [Gän09a]).
The first one is called 'audio signal carrier indexing services' (ASCIS, see Subsection 2.3.3.1) and the second one is named 'music recommendation and discovery services' (MRDS, see Subsection 2.3.3.2). Nowadays, MMSs often supports manifold functionalities. That is why they usually cover multiple types of the introduced music information service classification. Since, an analysis of such services was also a part of my Belegarbeit, the following subsections only include a summary of this research (see Subsection 4.3 of [Gän09a]). The set of presented MMS is by far complete, although it tries to cover the most representative ones.

[24] *Simple specialised ontologies* are an ontology category whose representatives concentrate on representing "(a) basic and commonly-used concept(s)" [SKR06].
[25] The *Music Ontology* has a quite active developer community. Since autumn 2008, I am an active contributor to it and are deeply involved in the recent revisions of MO.
[26] `http://www.bbc.co.uk/music`
[27] `http://libre.fm`
[28] `http://dbtune.org`

2.3.3.1 Audio Signal Carrier Indexing Services

Audio signal carrier indexing services are music cataloguing services. They have the strongest historical background of the three introduced MMS types. They arose right at the beginning of the WWW. At this time, ASCISs were primarily intended for delivering related information during the digitalisation or cataloguing process of common physical audio signal carrier media, such as CDs or vinyl records. Nevertheless, their features naturally evolved during that period of time until today. ASCISs are usually used for the management of personal music collections.

All Music Guide One of the oldest ASCIS is the *All Music Guide*[29] (AMG). It is driven by professional editors and for that reason delivers high quality musical characteristics and background information. Generally, this setting restricts the coverage of music in the *long tail* in the AMG music database. Still, AMG approximately doubled the amount of album releases and songs in their database during a period of roughly two years[30].

Compact Disc Data Base and freeDB Another quite mature ASCIS is the *Compact Disc Data Base*[31] (CDDB). This service has its origin in identifying audio CDs to deliver music metadata of theirs content. Since its buyout by *Gracenote*, CDDB is heavily revised (incl. musical fingerprinting support) and incompatible to former versions of this MMS. Its a part of the product portfolio of this company, which primarily delivers solutions for music consumption, e.g., playlist generation for embedded platforms, such as car music systems (see [Gra10]).
On the other side, *freeDB*[32] is a freely available, community-driven MMS that is based on CDDB in its initial version. Hence, it only supports an outdated MMF, which is not suitable for music metadata needs of customers today.

MusicBrainz *MusicBrainz* is, in its origin, another follow-up project of the initial CDDB version. During the last few years, it evolved very much. Currently, This community-driven, open-source MMS is introducing a new music metadata format called *Next Generation Schema* [sev11f]. It suits better for evolving music information requirements of the *MusicBrainz* information consumers. NGS is quite similar to MO in its expressiveness (see also its extensible set of *relationship types*[33] [sev10a]). *MusicBrainz* utilises a third party musical fingerprint service called *AmpliFIND* [Amp11].
This ASCIS is very popular for its universal unique identifiers called *MusicBrainz Identifiers* (MBID) [sev10f]. They are intended to unambiguously address music artists, releases, song etc. MBIDs are consumed by several information services, e.g., *Last.fm*, *BBC Music* or *Echo Nest*. For that matter, *MusicBrainz* is largely used as an information provider in Linked Data environments too. At present, this needs to be done via third party services, e.g., the *MusicBrainz* dataset on *DBTune*[34]. These services often make use of MO as a part of mappings to terms of Semantic Web ontologies. Fortunately, the ongoing *LinkedBrainz* project [DJ10, Jac10c] features a support of directly integrating Linked Data (see Section 2.2.4.5) into the *MusicBrainz* information service. This project heavily utilises MO for a mapping of NGS to appropriate concepts and *relations* of Semantic Web vocabularies.
In addition to its website, this MMS provides a Web Service[35] for accessing and submitting various music metadata and user generated data, such as reviews or ratings, via remote applications.

[29] `http://allmusic.com`
[30] See [Gän09a] for statistics of the AMG music database from 2009 and [Rov11] for statistics from 2011.
[31] `http://gracenote.com`
[32] `http://freedb.org`
[33] *Relationship types* are similar to properties in RDF Model, see Section 2.2.2.
[34] `http://dbtune.org/musicbrainz/`
[35] `http://wiki.musicbrainz.org/XML_Web_Service/Version_2`

Discogs *Discogs* is another community-driven ASCIS that especially serves *editorial metadata*. Thereby, it is specialised on describing artist-, song-, release- and label-related information. *Discogs*' MMF is similar to *MusicBrainz* NGS in its expressiveness. In addition to community features that are also supported by *MusicBrainz*, *Discogs* provides an online market place where users can offer and buy musical items.
This ASCIS provides a Web Service[36] as an alternative way of consumption as well. There are attempts to serve parts of the *Discogs* MMS as Linked Data, e.g., that one of the *Data Incubator* project[37]. Generally, the mappings from the *Discogs* MMF to appropriate terms of Semantic Web ontologies often include many concepts and *relations* of MO, too.

2.3.3.2 Music Recommendation and Discovery Services

Music recommendation and discovery services make us of all three kinds of music metadata (*editorial*, *contextual* and *acoustic metadata*) to provide music recommendations and/or guide a music discovery process. They often include information taken from user profiles to personalise recommendations and discovery journeys. MRDS are usually used as a foundation for playlist generation tasks. Generally, their main differences are the application and combination of various recommendation techniques, e.g., *collaborative* or *content-based filtering*. Due to its importance, only *Last.fm* is explained in the following as a representative MRDS. Further MMS of this category are, for example, *Pandora*[38] or *mufin*[39].

Last.fm One of the most popular Internet music streaming services is *Last.fm*. This MRDS mainly generates customised radio stations that make use of *collaborative filtering* algorithms. *Last.fm* enables its users to track their listening habits by *scrobbling* their listening events, i.e., every time a user is listen to a piece of music via a *Last.fm*-supported music player, this information is sent to the *Last.fm* server. On the basis of tracking data, a music listening history can be created. Interesting analyses can be carried out on top of music listener activity streams. This can be, for example, fancy visualisations, e.g. *LastGraph* [God09] that generates a graph of a users' listening history, or tag clouds, e.g. the user tag cloud generator by Anthony Liekens [Lie07] that calculates personalised tag clouds of the most frequent tags in one's listening history.
All mentioned music data analysis examples make use of the comprehensive *Last.fm* Web Service[40]. One advantage of its *scrobbling* method is that metadata corrections can be served if misspellings are recognised and corrections are available (see [Las11]). This feature can be used to enhance the data quality of a users' personal music collection. The data cleaning task heavily benefits from the application of *Last.fm*'s own musical fingerprinting engine [Cas10b] (see [Cas10a]).
There are some *RDFizers* that convert (and enrich) data taken from this MMS into Semantic Web KRs. Two of them are part of *DBTune*. The first one is the *AudioScrobbler* RDF Service [Rai06, RG10a]. It transforms the last 10 *scrobbled* tracks, the list of friends, and events (e.g. one has planned to attend) into KRs that make use of several Semantic Web ontologies, e.g., FOAF or MO. The second one is the *Last.fm* Artist RDF Service [Jac10a]. It converts a list of similar artist into Semantic Web KRs that make use of MO and SIM. Furthermore, both *RDFizers* enrich their transformation results by looking for identifiers that represent the same resource in other information services, e.g., *BBC Music*.

[36] http://www.discogs.com/help/api
[37] http://discogs.dataincubator.org
[38] http://pandora.com
[39] Besides, its MMS capacities, *mufin* also offers a music player application that makes use of this capabilities (see Section 1.2).
[40] http://last.fm/api

2.3.3.3 Music Content and Context Analysis Services

Music content and context analysis services (MCCAS) concentrate on computation and management of MIR tasks and their results. They are slightly different from the other two MMS categories. MCCASs do not consist of a website or another application that can be used directly by users. These MMS integrate their frameworks into a backend of a user client software and/or provide a Web Service as an interface for their musical characteristics analysis platforms. They power third party applications of services of the media domain, e.g., music recommendation components or media monitoring components. During the last few years several MCCASs were founded by researchers that already had a strong background in academic MIR research, e.g., *Barcelona Music & Audio Technologies*[41] (BMAT[42]) or the *Echo Nest*[43]. The latter one is characterised in the following paragraph representative.

Echo Nest The *Echo Nest* is a MCCAS that currently provides probably the most comprehensive Web Service[44] of this type of MMS. It can serve its information consumers musical characteristics of every music metadata type and on the music artist and song level (see [The11a, The11e]). Besides, the track analysis methods of *Echo Nest*'s Web Service (see [The11f]), this MCCAS also offers a client-side musical fingerprinting framework (see [Whi10]). In addition, to its musical characteristics analysis methods, further components are available, e.g., for playlist generation or personal catalogue management (see [The11d, The11b]).
Echo Nest forces a good interlinking of other MMSs. On the one side, an initiative called *Project Rosetta Stone* concentrates on matching identifiers of different identifier spaces[45], e.g., *MusicBrainz* (see [The11c]). On the other side, the music artist analysis component offers a method that serves websites of a music artist from several MMSs, e.g., *Last.fm* (see [The11a]). For all these reasons, *Echo Nest* can power a personal music collection management application with potentially useful features and capabilities.
DBTune also provides two *RDFizers* that make use of the Web Service of this MCCAS. The first one is an XSLT template[46]. It transforms the result of an audio signal analysis processed by the *Echo Nest* platform into a Semantic Web KR that utilises, amongst others, MO and AF (see [Rai08c]). The second one is the *Echo Nest* Artist Similarity RDF Service [Jac10b]. It has the same functionality as the *Last.fm* Artist RDF Service, but this *RDFizer* is utilising the results of the *Echo Nest* artist similarity method instead.

2.3.4 Summary

This section starts with illustrating a categorisation of musical characteristics and outlining an abstraction of the related complex task of music analysis. These topics lay the foundations for music metadata formats which are introduced after the prelude of this part. Former studies already concluded that ontological KRMs are qualified for representing the music domain properly. Due to that finding, the most comprehensive Semantic Web ontology framework that covers this field, the Music Ontology framework, is shortly explained. A categorisation of music metadata services with prototypical examples outlines the application of music content and context data. The MMS example descriptions contain the utilisation of Semantic Web technologies via third party applications. With the help of these information services researchers already demonstrated the benefits of this KM exemplification.
In general, music perception will rather be more subjective than objective. Representing and managing this knowledge still bears a challenge and is only partly implemented by existing MMS solutions. For this reason, the following section provides an overview about the important subject: How to handle personalisation and environmental context.

[41] A spin-off the Music Technology Group of the Universitat Pompeu Fabra in Barcelona, Spain.
[42] `http://bmat.com`
[43] Co-founded by two former PhD students of the MIT Media Lab of the Massachusetts Institute of Technology, Cambridge , USA.
[44] `http://developer.echonest.com/docs`
[45] *Echo Nest* maintains also an own identifier space.
[46] Please note that this XSLT template was created on the basis of prior version of the *Echo Nest* Web Service.

2.4 PERSONALISATION AND ENVIRONMENTAL CONTEXT

Personalisation is a crucial issue that is usually still a big challenge for many application developers today. However, it is not only the facet of user modelling that matters (see Subsection 2.4.1). It is, furthermore, the knowledge of the environmental context that concerns a user (see Subsection 2.4.2). Both aspects can be a part of stereotype modelling to approximate and predict further characteristics that can be used for personalisation tasks (see Subsection 2.4.3).

2.4.1 User Modelling

The field of user modelling is relatively mature. It has a history of 40 years of research by now (cf. [Sos08]). Countless modelling approaches and implementations were developed during that time. User models often include static and dynamic parts. For example, demographics usually are static and the physical state of a person varies a lot. Furthermore, every behaviour that can be profiled (user profiling) is a dynamic part of a user model. *AcitvityStreams* [sev11a] is a relatively popular format for sharing activity KRs.
User profiles can either be designed for a specific domain or they are generic. For example, the *User Modeling for Information Retrieval Language* (UMIRL) [CV00] was created to meet the requirements of information retrieval processes and can be applied for the personalisation of MIR tasks. Generic user models represent a unified way of general user modelling. Their type of users is initially undefined. However, refinements and extensions are typically supported to enable an application that requires more precise knowledge representations.
Generally, user profile approaches differ in their supporting of various user modelling dimensions, e.g., competences (skills, expertises, beliefs etc.), interests, or emotional states. Even the definitions of these clusters of characteristics often vary between the implementations. Dominikus Heckmann in his dissertation [Hec05] gives an exhaustive overview about the range of user modelling dimensions, as well as Sergeyan Sosnovsky is doing so in [Sos08]. In addition, there are different user model implementation techniques, e.g., overlay, keyword-based or stereotype user modelling (see [Sos08]). A basic requirement of user profiles usually is that they should be adaptive and, thereby, grant users an appropriate level of control. This personal view may be selected by an analysis of user's knowledge and experience regarding the handling of the related system. Different user models already demonstrated the application of ontological KRs. They show the benefits of this modelling approach by utilising axioms, e.g., functional properties of OWL (see Section 2.2.2), *relations*, e.g., `is-a` or `part-of`, and associated values, e.g., weights. Moreover, user models that make use of Semantic Web KRLs can easily be shared between different applications. This helps to also establish a "shared understanding" in the domain of user modelling. Three different user profile implementations that utilise Semantic Web KRLs are shortly introduced in the following paragraphs.

Friend of a Friend Vocabulary The *Friend of a Friend Vocabulary* (FOAF) [Mil10] is probably the most prominent Semantic Web ontology. This might be caused by the fact that it is primarily intended for modelling user profiles of social networks. In addition, FOAF is an early adopter of RDF and demonstrates a good showcase for this Semantic Web KRL framework (see [Bri07]).
It enforces a decentralised social network approach, where everyone is able to host and maintain one's own profile on a preferred web space, e.g., a user's homepage (see [BM08]). This information can be reutilised and, for example, aggregated to illustrate one's global social web. Therefore, Web Services, such as the *Social Graph API* [Goo11], are already available. A user can primarily express basic demographics, interests, and relations to specific web pages, with FOAF. For that reason, one can attribute, for example, a personal homepage or a user profile page in an online social network, for example, *Facebook*[47].
FOAF represents a basic user modelling vocabulary. Several extensions are available via separate

[47] `http://facebook.com`

namespaces to follow the design principle of modularity. They address specific related (sub-)domains or simply enhance the expressiveness of existing KRs. Representatives of the former type are the *BIO schema* [Gal10] and *ResumeRDF* [Boj07]. Examples of the latter type are the *Weighted Interest Vocabulary* [MBRI09] or *e-foaf:interest Vocabulary* [ZWDH10]. Besides theses extensions, the WebID protocol makes us of FOAF in its KR vocabulary part (see Section 2.2.4.6).

General User Model Ontology A comprehensive implementation of a generic user model is the *General User Model Ontology* (GUMO) [HSB+05, Hec05]. This vocabulary follows the ontology design principle *property-oriented context reification* to describe user-specific aspects, e.g., interests or knowledge, in detail (see Section 2.2.4.3).
GUMO supports a huge variety of user modelling dimensions. Beyond basic ones, such as demographics or contact information, this Semantic Web ontology provides terms for describing, e.g., emotional states, personalities or abilities. Albeit, GUMO has an exhaustive expressiveness, it does not contain terms that enable simple KRs, i.e., *binary relations*, for modelling user characteristics.

Unified User Context Model The *Unified User Context Model* (UUCM) [NSMH04] is another ontology-based, generic user modelling approach. It follows an extensible two-level design that includes a meta-model. With the help of UUCM one cannot only describe user profiles, it is rather intended to represent contextual information as well.
This vocabulary implements four user modelling dimensions. These are 'cognitive pattern', 'task', 'relationship' and 'environment'. Thereby, only the first one qualifies a user him-/herself. Cognitive patterns characterise user specific-aspects, such as interests, knowledge, or preferences (see [NSMH04]). UUCM is also guided by the design principle of modularity. Each dimension is represented by a separate ontology.

2.4.2 Context Modelling

Personalisation requires not only models of users, but also KRs of context. In this case context is defined by a user's situation and its environmental resources and involved agents. This information is evaluated regarding the task or activity a user or user agent is performing. Basic context models at least consist of the dimensions time and space. More advanced context spaces contain attributes, such as language, device, application, or role of the user (cf. [NSMH04]). For example, UUCM considers environmental context information regarding a given working context. Hence, it is particularly suitable for context modelling as well.
In addition, the more general field of association or annotation modelling can be utilised to the description of context. There are several approaches expressed in form of Semantic Web ontologies, which address this issue[48], e.g., *Annotea* [Koi05], *Open Annotation* [San10] or the *NEPOMUK Annotation Ontology* [SSvEH07]. Their annotation concepts hardly differ from each other, because they are often simply a reification of the semantic relation "annotates". Sometimes, *Named Graphs* are applied in addition to enclose the associations (cf. Section 2.2.4.3).
Primarily, these approaches often have the same drawbacks, e.g.,

- their association statements are badly reusable, i.e., reusing the same association statement for a different information resource is not always possible,
- the sources and targets are pre-typed,
- their semantic relations are too general, or

[48] If one considers *tagging* as a special case of annotation statements, then even more implementations are available, e.g., the *Nice Tag Ontology* [MLL05] or the *Tag Ontology* [New05].

- one cannot rate, comment, or express in another form feedback, e.g., *like*.

However, KRs that enable semantically rich associations, e.g., a mood, a genre or an occasion, have to be enabled. Their meaning can simply be inferred from the relation itself. This issue has to be investigated further.

2.4.3 Stereotype Modelling

There are certain sets of similarities that are shared by groups of users or contexts. These characterisations are usually called stereotype and consist of generalisable characteristics. Such stereotypes can be clustered regarding a subject or concept, e.g., user type, music listening type or music knowledge type. A classification of music listening types, for example, can consist of: savants, enthusiasts, casuals, indifferents (cf. [LC07] or [Gän09a]). Occasionally, stereotype definitions contain attributes that overlay between different stereotype descriptions, e.g., the ones of different stereotype clusters. Furthermore, they can be part of a hierarchy, where features of higher-layered stereotype definitions are inherited by lower-layered ones.
Multiple stereotypes can be assigned to a user or context model, e.g., one stereotype per stereotype category. This can be beneficial to overcome the *cold start problem* of user profile initialisation (see [GKP05, Her08]). In the music domain such templates are called *music context* or *idiosyncratic genre* (see Section 1.2) and are used for automatic playlist generation or individual definitions of own music clusters.
Ontological KRs can be applied to stereotype modelling. For example, Gawinecki et al. [GKP05] demonstrated the applicability of Semantic Web KRLs for the modelling of stereotypes in travel support systems. The utilisation of ontologies can be three-fold (according to [Sos08]):

1. to populate stereotype profiles into user or context profiles
2. to implement a stereotype definition as an ontology
3. to organise the structure of stereotype descriptions

Stereotype profiles can easily be made shareable and reusable by following the principles of the Linked Data publishing guideline (see Section 2.2.4.5). Rule languages, such as SPIN (see Section 2.2.4.6), can be applied to integrate knowledge of stereotype definitions into user or context models. Moreover, Semantic Web KRLs, such as OWL or SKOS (see Sections 2.2.2 and 2.2.4.1), can be utilised to defined structures of stereotype descriptions.
It is crucial that these templates are adaptive, i.e., initially associated stereotypes[49] can evolve over time. For example, it is not natural to implement a strict, universal division of mandatory and obligatory musical characteristics and descriptions. These settings can be guided by stereotype profiles. If a user has the need to change this preference, she or he can simply select another stereotype that, for example, supports more advanced features.

2.5 SUMMARY

This chapter explores necessary concepts and technologies that one basically has to be familiar with, to be able to understand the PMKB concept. This vision is introduced in the following chapter and is refined afterwards.

[49]These stereotypes can exemplary be assigned via a user type analysis.

The prelude gives a general overview about knowledge management (see Section 2.1). It illustrates knowledge representation concepts, such as ontology (see Subsection 2.1.1.3), and foundations of knowledge management systems, such as the REST architectural style (see Subsection 2.1.2.3).
Semantic Web technologies implement KM concepts (see Section 2.2). RDF Model (see Subsection 2.2.2), for example, makes use of the concept of semantic graphs (see Subsection 2.1.1.2). Moreover, the Linked Data publishing guideline (see Subsection 2.2.4.5) shares ideas of the REST architectural style.*Triple stores* (see Subsection 2.2.4.4) are knowledge bases (see Subsection 2.1.2.2).
Since this chapter is about music information management, Section 2.3 outlines basic facts about music metadata, its formats and services. Ontological KRs are also appropriate for describing musical characteristics. The *Music Ontology* is a prototypical KRL for the music domain (see Subsection 2.3.2). Furthermore, the application of Semantic Web technologies in music metadata services is expanding, e.g., the *LinkedBrainz* project is a noteworthy initiative in the context of *MusicBrainz* (see Subsection 2.3.3.1).
Personalisation is an issue of the PMKB concept. For that reason, the main concerns regarding user, context and stereotype modelling are explained in Section 2.4. User models, such as FOAF or GUMO, are summarized and classified (see Subsection 2.4.1). The introduction of the second main aspect of personalisation, context profiling, considers the more general domain of association or annotation modelling as well (see Subsection 2.4.2). This includes a list of common drawbacks of existing approaches that make use of Semantic Web KRLs. Finally, the field of stereotype modelling, that can be applied for user and context modelling, closes the last section of this chapter (see Subsection 2.4.3). It illustrated the benefits of utilising ontological KR approaches for stereotype models too.

Copyright TU Dresden, Thomas Gängler

3 THE PERSONAL MUSIC KNOWLEDGE BASE

To explain and illustrate the needs of the mainly computational and fully platform independent concept *personal music knowledge base* this chapter is divided into three main parts. Firstly, foundations for a short definition of requirements for knowledge representation and management are identified (see Subsection 3.1). Secondly, the architecture to meet these requirements is established (see Subsection 3.2). Thirdly, the (main) workflows to describe how this information service works are introduced (see Subsection 3.3).

3.1 FOUNDATIONS

3.1.1 Knowledge Representation

A PMKB can contain manifold information of a wide variety that is part of a real world model. Section 2.3 shows the complexity and diversity of music content and context data and the possibilities of multiple relationships between single pieces of information. Furthermore, Section 2.4 outlines that user profiles and the modelling of users' behaviour, which includes the continuously changing environmental context, can be a deeply interlinked net of a big amount of data. Due to these reasons, we need a KRM that is able to represent all addressed issues of the music, personalisation and context domain in a uniform manner. This is particularly required, if one wishes to reason over data of a KB that include information of all these domains (see Section 2.1.1. Therefore, this KRM has to be able to

- represent and describe semantic relations between different pieces of information,
- name resources in a uniform way (at least be able to map equal resources to each other),
- enable the opportunity to describe a specific *thing* on different levels of detail, which includes that it is easily extendible and follows the *Open World Assumption* (see Section 2.2.1.2), and
- allow to draw conclusions and infer new knowledge from the results (that can be represented as part of this KRM, too).

Semantic Relations On the basis of these requirements, one can see that an ontological KRM might be the best choice in this case (see Section 2.1.1.3). Such a model cannot only represent hierarchies of *universals* of a domain like one can do with a taxonomy based KRM, e.g., a music artist is a person is an agent is a *thing*. It can also use semantic relationships, which are different from such a hierarchy, e.g., a music artist is a friend of another music artist, who plays in a band called "The Table". A hierarchy defined by a taxonomy can be a part of an ontological KRM, too. Albeit, it consists of many semantic relationship definitions between *universals* and/or *particulars* that categorise or describe *something*.

Uniform Naming To establish a uniform naming, one can introduce identifiers. They enable a KMS to address and reference every *entity* with the help of at least one of them (cf. Sections 2.1.2.3 and 2.2.3). If a human being or machine is able to reason that two identifiers represent the same *thing*, a semantic relation can be applied which represents that information, e.g., a *same-as* relation. In this case, an instantiation of this *same-as* relation, that is related to the two identifiers, was inferred from a reasoning task.

Expressive Power A KRM has to be able to somehow describe the real world in a uniform manner (cf. Section 2.1.1.1). However, this does not mean that every *particular* of a specific concept has to be described by the same level of detail. On the one side, there can exist different definitions of a concept as *universals*. They can be instantiated to describe one and the same real world thing on different levels of detail in different *particulars* of different *universals*. On the other side, a single *universal* definition can be instantiated to describe one and same real world thing from different points of view. Both types can be combined, i.e., every time there can exist many *particulars* that describe one and the same real world thing.
Parts of an ontological KRM can be applied as needed. A KMS only reasons over those resources which are available at this moment on this location (cf. Section 2.1.2.1). Generally, it is quite impossible to define all concepts and properties (*relations*), their variants, and instantiations in one *ontology*. Following the principle of *partial understanding*, every *entity* that is not part of the model is *unknown* at this moment, but might be included some day (cf. Section 2.2.1.2). For example, if I know that a music track description can have an artist relation, but this attribute is for whatever reason currently not instantiated, I will only know that this track description has an artist relation, but I will not know the artist description itself.

Reasoning The previous example illustrates a reasoning capability, which is possible for our KRM. It is especially important that someone can infer new knowledge from the existing model, e.g., the example with the *same-as* relation above. This can be of different complexity. A simple reasoning task is, for example, a *part-whole* relation as one can layer over a taxonomy. Taken the taxonomy example from above, one can infer that a music artist is also a *thing*.
A complex reasoning task is, e.g., the inference of a stereotype profile of a specific stereotype category for a concrete user (see Section 2.4). Firstly, an agent provides a user profile of a user, who uses a music player, and which records the usage behaviour of his/her music collection in context of this music player. Secondly, the reasoning processor needs stereotype definitions to be able to infer that a user can be associated with a specific stereotype. Finally, if this reasoning task delivers a positive result that is a user can somehow be related to stereotype definition(s), one can exactly model this relationship with the help of parts of the (same) KRM.

Summary Such a KRM can provide the basis of a *personal music knowledge base*, because of its different functionalities. Due to its generality, it can also be used for other use cases of various domains. This is an important design criterion, since it has to be possible to integrate knowledge of a specific domain into a KB that consists of a KRM of another specific domain at ease.

3.1.2 Knowledge Management

The PMKB needs a knowledge management system (see Section 2.1.2) that must be able to handle a variety of features that are defined by its own knowledge representation model (see Subsection 3.1.1) and KRMs of its information providers and consumers. For that reason, pieces of information (information resources) have to be

- creatable (write),
- accessible (read),
- modifiable (update) and
- removable (delete).

These four primary knowledge management tasks are explained separately in the following paragraphs.

The Write Task It depends on the provenance of the data, how the write task is structured. On the one hand, if the KMS consumes pieces of information from another information service, e.g., *MusicBrainz* or an ID3 tag of an MP3 file, it must be able to include these data into the KRM of the PMKB. For that matter, the KMS adds at least some provenance and trust information to the retrieved data. However, it also has to provide an alignment between *universal* definitions, if the source information service, the information provider, cannot deliver the information by using existing *universal* definitions (of the PMKB KRM) or in the supported KR format, e.g., the language framework of the Semantic Web (see Section 2.2.2). On the other hand, if the information is inferred from knowledge of a reasoning task, it has to be easily attachable to the knowledge representation model of the KMS.

The Read Task All open knowledge[1] of the PMKB KRM is accessible by its KMS and by all information consumers of the *personal music knowledge base*. Knowledge from KRMs of other information services has to be provided as information that can be consumed by the PMKB KMS. Hence, the pieces of information are accessible to its information consumers too. The KMS is able to deliver information resources in different serialisation formats. An information consumer has the ability to choose a preferred serialisation format or to provide a wrapper, which can consume at least one of the available serialisation formats. This requires the ability to process knowledge (management) requests from information consumers. For example, when following the Linked Data philosophy (see Section 2.2.4.5), the implementation has to provide dereferenceable URIs and/or (a) *SPARQL endpoint*(s). These can process queries that are defined with the help of the SPARQL query language (see Section 2.2.4.6).

The Modification Task The KMS of the PMKB is able to modify existing data its KRM. On the one side, it must be able to process changes, which are caused by the KMS itself. This can be , e.g., new inferences of a reasoning task that affect existing pieces of information, or the synchronisation between different working nodes of the PMKB (e.g. a client update to the server; see Section 2.2.4.6). On the other side, it must be able to handle updates from external information services. Thereby, both main information service types can be involved. Information consumers can propagate updates, e.g., when a user changes the title of a music track on his/her music player. Information providers can also spread modifications, e.g., a concert of a music artist is cancelled and a concert information service is notified about this change to propagate this update to its information consumers, too.

[1] In other words, information resources without any access restrictions.

The Delete Task To complete the explanation of information processing features, the KMS of the PMKB is able to remove pieces of information from its KRM. Thereby, the synchronisation aspect must also be treated as it is important for every mentioned KM task of the KMS (see Section 2.2.4.6). For example, if the user wishes to remove a music track from his/her music collection, the application, in which the user executes this task, processes this KM request to the PMKB KMS. This server unit handles the request and the internal synchronisation of the working nodes as well.

Security and Privacy All information processing capabilities are secured by an access control mechanism. This is an important requirement to treat personal information that is part of the KRM secure ly(see Section 2.2.4.6). The access control mechanism also includes an anonymisation (de-personalisation) method to enable data evaluation tasks across user accounts. Results of such information interpretations tasks can influence personalisation tasks too.

Summary To sum up, a knowledge management system of a PMKB is able to process all knowledge and KM requests from itself and its related information services (consumers and providers). For that matter, it enables knowledge exchange and reasoning.

3.2 ARCHITECTURE

The concept of a *personal music knowledge base* can be seen as a part of the concept of an interlinked, distributed KB that powers the *Web*. As explained in Section 2.1.2.3 the REST architectural style is appropriate for guiding the design and development of an architecture of an OKMS. Therefore, it is chosen to lead the architectural design of the PMKB.
The application is divided into two main parts, regarding the need for customisation (see Section 2.4), offline application, and performance issues (application of client-server style). Every device, which has at least one application that utilises the PMKB, also has a *local music knowledge base* (LMKB) that interacts with the *global music knowledge base* (GMKB).

Knowledge Bases Figure 3.1 illustrates the main architecture of the *personal music knowledge base* concept. On the one side, a LMKB is responsible for interacting with client applications (e.g. a music player), a (part of a) digital music collection (which is stored on this device), and to provide an intermediated layer for these information services to the GMKB. This component always plays the role of an origin server for all local user agents that interact with it.
On the other side, the GMKB provides and serves information to LMKBs ,to satisfy needs (requests) of the client applications. Furthermore, it is responsible for information aggregation and federation of information from external information services. This component always plays the role of an origin server for all LMKBs.
Viewing the overall system, both components act as "intelligent" intermediaries with their own functionalities, including cache, proxy and gateway features. However, the roles of the components in a communication chain are clear to each interacting part in this layered system.

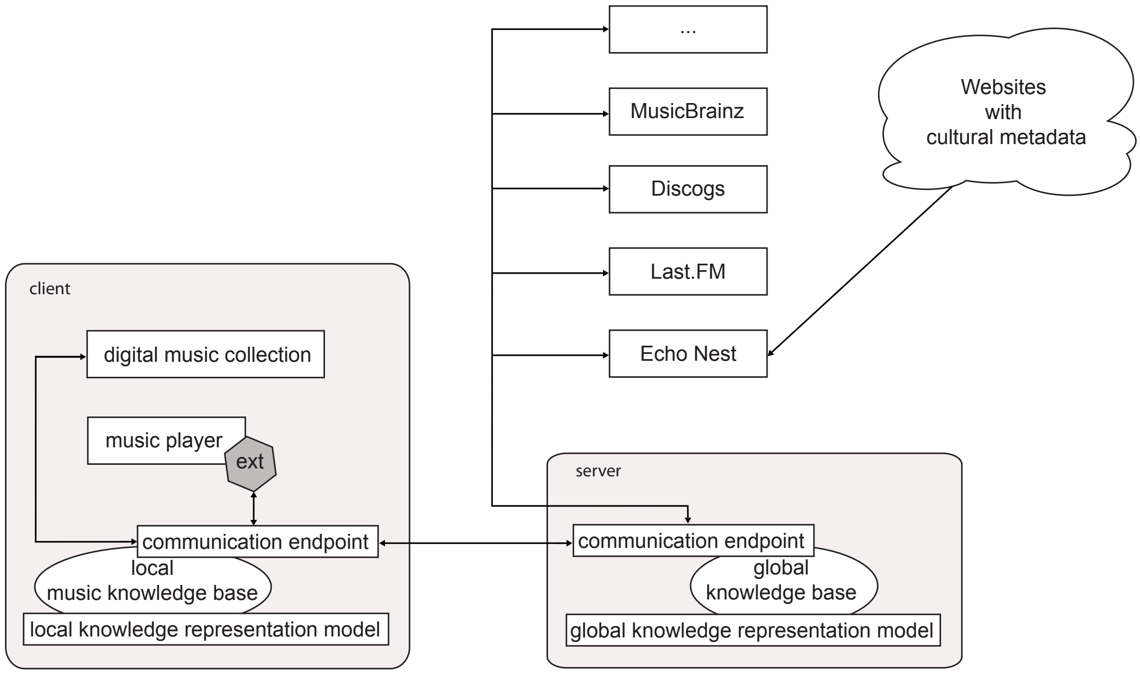

Figure 3.1: Main architecture of the *personal music knowledge base* concept

Knowledge Representation Models One can see from the architecture graphic (Figure 3.1) that there is a local knowledge representation model[2] (LKRM) and a global one (GKRM). This setting has several reasons. Firstly, the GMKB can contain data of multiple personal music collections. Common knowledge, such as musical fingerprints, does not need to be recalculated from scratch repeatedly as it can be shared between different users of the service. Thus, every KRM of a user account shares a subset of the global one. In addition, it includes KRs of personal associations and context. Secondly, a LKRM can differ from the global one of a user, regarding synchronisation issues and additional client specific information, which does not need to be shared. Thirdly, LKRMs of different client devices of a user can also be different.

Communication Endpoints The communication endpoints (CE) in the architecture graphic (Figure 3.1) illustrate a part of the PMKB KMS that is responsible to interact with other information services. They are connectors. Communication can take place on different working nodes. That is why, each working node has its own CE. On the one hand, the CE of a client can, for example, establish a connection to the digital music collection of this client. It uses this channel to extract metadata and to process audio signal data of digital music documents, or to talk with an extension of a music player. On the other hand, the CE of a server can, for instance,

- retrieve information of an external information service (e.g. *Echo Nest*, see Section 2.3.3.3),
- send a proactive update notification to a LMKB (which is then retrieved by its CE) or
- serve information to an information consumer, which exists somewhere on the *Web* (e.g. *MusicBrainz*, see Section 2.3.3.1).

The main architecture of a CE is constant, because they represent the interface of the PMKB KMS.
The (main) workflow of the information service PMKB, that is powered by the illustrated architecture, is explained in the next section.

[2]For simplification, the graphic includes only one client. However, even one and the same user can have several client devices that use the PMKB.

3.3 WORKFLOW

3.3.1 User Account Initialisation

At the beginning, a user, who wishes to use the PMKB, has to provide some information to "feed" this information service. First of all, this is information about the user her- or himself, which is stored into a user profile. Moreover, this is information about his or her personal music collection. To make the initialisation step of a user profile as easy as possible, two main auxiliary opportunities should boost the user's comfort. Firstly, it is possible to provide information by (own) user profiles from other information services, e.g., from social network services (e.g. *Facebook*), or (music) recommendation services (e.g. *Last.fm*). Secondly, a wizard can be created which includes a simple questionnaire to provide basic personal information. This is supposed to help to infer further knowledge and make basic settings. Consequently, the user can, for example,

- set specific stereotype instances of specific stereotype categories, e.g., a stereotype category of music listening types that include savants, enthusiasts, causals and indifferents as instances/sub categories (see Section 2.4.3), and stereotype instance definitions include specific further presettings, or
- select some information services[3] she/he likes, and which act as information providers for information retrieval tasks.

To integrate information about the personal music collection, the user also has two more auxiliary opportunities. Firstly, the import of an existing KRM of this collection can help to establish a good foundation. This can be from, e.g., *Songbird* or *Last.fm*, if it is a completely digital music collection, or another service which supports the representation of analogue music collections too. Secondly, a device's operation system grants the CE of the PMKB KMS access rights to the folders, where the personal digital music collection is located on the device. This information extraction task is described in the following subsection.

3.3.2 Individual Information Extraction

The individual information extraction task is massively executed during the user account initialisation task, and later, when a user updates a collection or requests further information about a specific *entity*. Thereby, it directly utilises music documents of one's personal digital music collection. This information extraction task can be described by the following steps:

1. The KMS receives (consumes) the metadata of a music document or the knowledge request on the client device via its LMKB CE (as described in subsection 3.1.2).
2. The KMS triggers a synchronisation task via the local CE to the GMKB, if this data is not already available in the LKRM. This initialise the verification task regarding the extracted metadata of the music document as well.
3. The GMKB CE consumes this information and the KMS continues the verification task by checking the availability of the data in the GKRM.

[3]This can be supported by information service quality ratings, which are provided by information service quality rating agencies (see Section 3.3.3)

(a) A probably cleaned and/or completed piece of information is the result of a successful verification task. This is the case, if it can be matched with an information resource that includes basic music context and content data[4].

(b) Requests to a preselected and optionally pre-ordered list of external information providers are triggered by the global CE. Received information is evaluated by the KMS, if the data cannot be verified at this stage. This is the case, if it cannot be matched to an information resource in the GKRM.

(c) An audio signal extraction request will be assigned to the LMKB CE, if the data still cannot be verified. In other words, the inclusion of the external information services was non-satisfying to match or complete the given amount of data.

   i. The follow up audio signal analysis task[5] results in a high-level musical feature vector and a unique audio signal-based fingerprint, where the grounded evidence vectors can be stored separately for revaluation tasks etc. (see Section 2.3.1).

   ii. The verification task is repeated with the help of the results from the previous step.

(d) A new unverified information resource is created, if the verification task has finally failed.

4. In both cases an information resource is written to the GKRM and the GMKB CE returns an information resource to the LMKB.

5. The KMS includes the information resource, that was retrieved via the local CE, in the LKRM.

6. The KMS registers update feeds on utilised pieces of information from external information services for the GMKB and for synchronisation purpose of the LMKBs. This is necessary to enable a proactive update notification (see, e.g., Section 2.2.4.6).

3.3.3 Information Service Choice

Information services (see Section 2.1.2.1) present the foundation of the information federation task. Since there is a great variety of web information services available, it will be useful to describe, categorise and rate them according to their quality. At least, it will be beneficial if there are general objective characteristics about common web information services (e.g. *Wikipedia*) and music web information services (e.g. *MusicBrainz*, see Section 2.3.3.1). These characteristics can be complemented by information service quality ratings provided by information service quality rating agencies. It should not be an issue, whether these are official ones, with a broader consumer acceptance and a higher general trust value, or inofficial ones, made by a group of domain experts or friends, with a higher subjective trust value.
However, the user has the possibility to preselect information services of his/her choice on the basis of these information service quality ratings. There are two auxiliary opportunities, which can alleviate this task. As a result of selected stereotype instances of different stereotype categories or a complete analysis of the personal music collection of a user, the KMS can suggest (semi-automatic) or set (automatic) specific appropriate information services as a foundation for the information federation task of knowledge requests. Furthermore, the KMS can consider, whether some information services are more qualified for a specific knowledge request, or whether other information services can be a priori excluded on the basis of the preselected information services of the user and the characteristics of these information services.
This enables an reduction of the overall information space, which can in general be considered, in the first place. Besides, information service choices can facilitate an optimisation regarding specific knowledge requests, for instance, by processing a query specific ranked list of source information services. The user pulls appropriate information, rather than all information is pushed to him/her.

[4] The amount of that basic music context and content data can vary regarding user settings. This information is fetched a priori, for example, if a user preconfigure an interest in background information of music artists. Another one might get background information about music artists later, because he/she requests such kind of information quite infrequently.

[5] It depends on the processor power of the device and/or its implementation, whether this task will be executed by the local machine, the server machine, or an external information service.

Copyright TU Dresden, Thomas Gängler

3.3.4 Proactive Update Notification

Synchronisation that is based on proactive update notifications is an important aspect of the PMKB. On the one hand, the KMS must be able to keep LMKBs up-to-date with the GMKB. On the other hand, it must also be able to keep the GMKB up-to-date with utilised information services and hopefully vice versa. Such a setup is supposed to close the communication cycle of information (see Section 2.2.4.6). Due to proactive update notifications, synchronisation descriptions are always pulled from the information consumer or information provider.
Therefore, an implementation of a KMS has the freedom to choose between a lazy and/or an ad-hoc update strategy. The first one follows an *as-needed principle*. This principle works as follows:

1. A client application or LMKB requests an information resource.
2. An update task will be triggered on this information resource, if a change delta (or several change deltas) is available to it.
3. This update task maintains (a) change delta(s) before an information resource will be served to its consumer.

By applying the ad-hoc update strategy, every change delta will be directly maintained in the affected information resource, when a proactive update notification is consumed by a communication endpoint.
A KMS can mix both update strategies, too. For instance, it uses an ad-hoc update strategy for consumed update notifications from utilised information services at the GMKB and a lazy update strategy for synchronisation on LMKBs. It is crucial that a PMKB must always be able to deliver up-to-date information for knowledge requests from client applications. Consequently, for instance, a LMKB will be synchronised before one switches to the offline mode[6].

3.3.5 Information Exploration

As described in Subsection 3.3.1, after having initialised his/her account, the user can access information resources in the LMKB. Each of them describes basic music context and content data (as defined by the user) of a song that is contained in the (part of the) personal music collection. However, this knowledge must be requested by (a) further knowledge request(s), if a client application (for several reasons) needs information that goes beyond this basic data (cf. Subsection 3.3.2).
Such knowledge requests are needed, for example, when a user demands further background information about a music artist, which is not presented in the LMKB yet. This can be the case, if he/she is barely interested into this kind of information. Consequently, every (locally) new piece of information that is requested by a LMKB, will be included into the LKRM and be maintained like every other information resource (see Subsection 3.3.4). This is the same case for the GMKB, when the information resource is not available in the GKRM (cf. Subsection 3.1.2).
The processing of knowledge requests may also include an information federation task on external information services by the KMS if the requested information is not available or up-to-date in the GMKB. To reduce the overall information space, the KMS uses user's preselected information services and/or evaluates their characteristics, whether they are appropriate for satisfying the knowledge request or not (see Subsection 3.3.3).
Retrieved information from external information providers can be

[6]This excludes unforeseen connectivity errors.

- validated against each other, e.g., regarding spelling or correctness by using NLP techniques and/or applying reasoning tasks[7], and/or
- ranked, e.g., by a predefined order of preferred information services.

Finally, it depends on the client application in which way the result of a knowledge request will be represented. A client application can provide specific appropriate views for specific knowledge domains, e.g., a timeline for temporal information, or format the result of a validation task for information from external information providers, e.g., to enable further user feedback.

3.3.6 Personal Associations and Context

It is important that users can express their subjective perceptions about music at every level, e.g., music artist, song, genre or compilation. This information is processable by the PMKB KMS. For instance, when a user associates a specific mood or occasion with a self created playlist. All this subjective knowledge can be used for further tasks, e.g., a music recommendation based on a specific mood.
On the other side, this knowledge can influence evaluation tasks to satisfy specific knowledge requests, e.g., when a user re-defines a description of a specific music genre or she/he provides specific preferences regarding specific environmental contexts. In this case a user creates *idiosyncratic genres* or *music contexts* (see Section 2.4.2). Knowledge request processing is strongly dependent on the user and the context and probably delivers quite different results for the same query (see Section 2.4).
To enable such a knowledge processing in a multiple device environment, which is common nowadays, these KRs of personal associations and contexts have to be synchronised with the GKRM (see Subsection 3.2).

3.4 SUMMARY

The important features of the concept *personal music knowledge base* are given in the following short list:

- The knowledge representation model has to be able to represent music content and context, user profile and profiling data (especially personal associations and context, see Subsection 3.3.6) in the same manner (see Subsection 3.1.1)
- The knowledge management system has to be able to handle the four important information processing capabilities (creatable, accessible, modifiable, removable), data synchronisation, access control, and knowledge exchange and reasoning (see Subsection 3.1.2)
- The PMKB has a *global music knowledge base* and *local music knowledge bases* (see Subsection 3.2)
- The PMKB has a strong concern on information federation, personalisation and timeliness/-data synchronisation (see Subsection 3.3)

[7] Thereby, different representations of the result of this validation can be made.

4 A PERSONAL MUSIC KNOWLEDGE BASE

This chapter describes an exemplary, platform specific realisation of the concept *personal music knowledge base*. The PMKB is described as mainly computational and fully platform independent concept in the previous chapter. This chapter is aligned to the structure of the "abstract concept" chapter, to map the features and requirements of the abstract PMKB description directly onto the concrete concept. This instantiation is on a very high level platform specific, i.e., only main technologies, e.g, KRLs (vocabularies) and protocols are defined. A detailed explanation of novel designed or massively modified KR vocabularies is given in Section 4.1. This part is followed by a small summary of design decisions, that are made on this implementation level, regarding the PMKB KMS (see Section 4.2).

4.1 KNOWLEDGE REPRESENTATION

A conclusion of Section 3.1.1 is that an ontological KRM is a good choice to satisfy the needs of a KRM for the PMKB. As introduced in Section 2.2.4, Semantic Web KRLs and ontologies are an appropriate basis to satisfy the needs of such a KRM. One can

- model *universals* (classes and properties) and *particulars* (individuals) with them in the same manner,
- separate domain specific descriptions into separate ontologies (which can also have a different level of detail),
- apply inferencing rules on them for reasoning and
- name resource in a uniform way by utilising URIs.

To model the main domains that can be part of a KRM for the PMKB, one should try reusing existing Semantic Web ontologies as much as possible. Firstly, the Music Ontology framework is an appropriate entry point for music content and context data (see Section 2.3.2). Secondly, FOAF Vocabulary is a good foundation for describing user profiles (see 2.4.1).
However, since this knowledge can be federated from several, selectable information services, we also need an ontological KR to describe them (see Subsection 4.1.1). We require further

ontological KRs to be able to record user behaviour, describe recommendations, personal associations and environmental context (see Subsections 4.1.2.4, 4.1.3, and 4.1.2.3). Moreover, some additions for modelling users are useful and necessary (see Subsection 4.1.4.2). Finally, a taxonomy for media types closes a further gap that was discovered during the implementation of the PMKB concept (see Subsection 4.1.5).
All these ontological KRs are expressed in several new Semantic Web ontologies, which can be included into the PMKB KRM. They are presented in the following subsections[1].

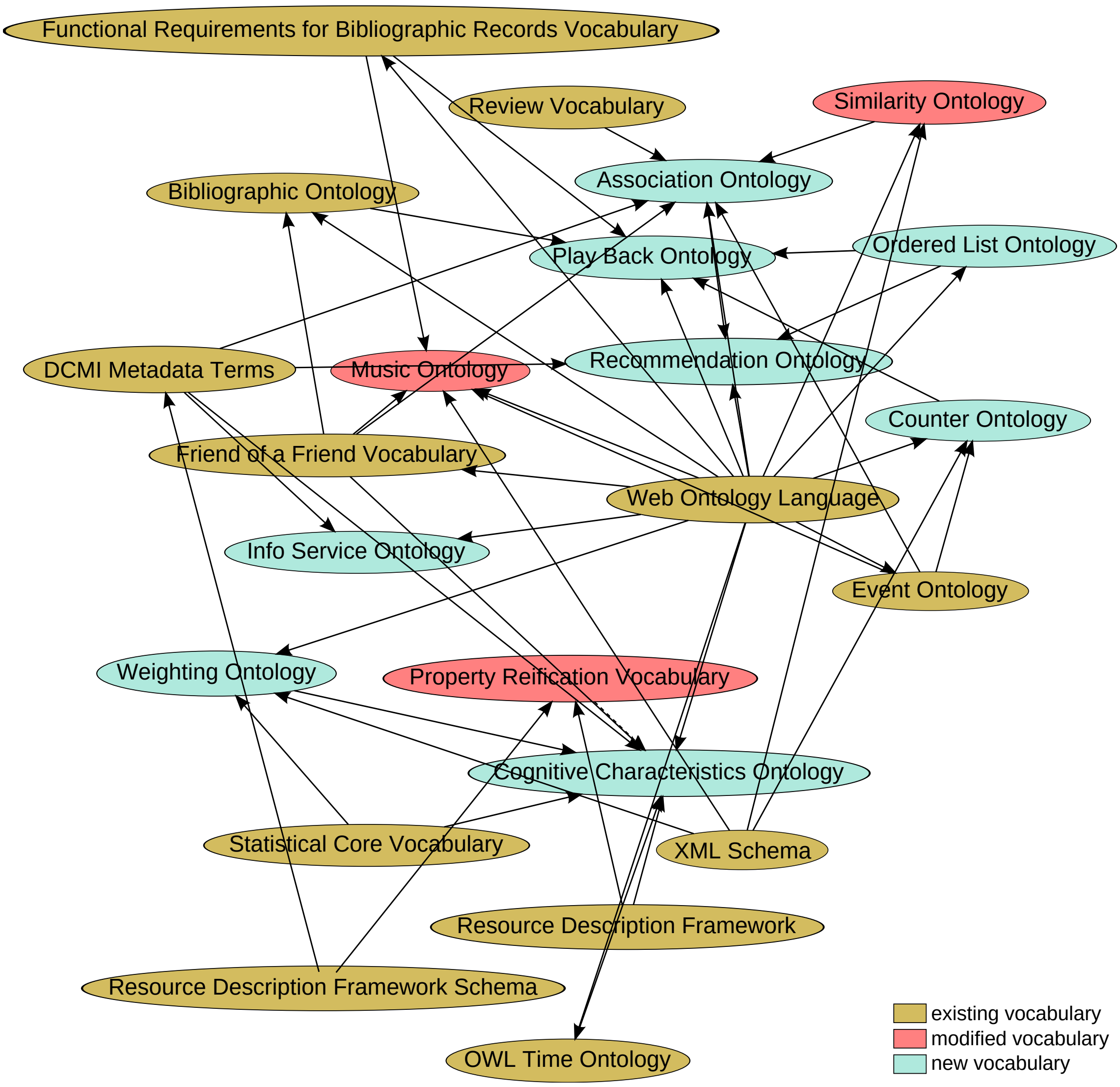

Figure 4.1: An overview of applied ontologies and vocabularies (see [Gän10f])

Figure 4.1 shows an overview of applied ontologies and vocabularies in the PMKB KRM. On the one side, one can see the reutilisation of existing Semantic Web ontologies in new ones. On the other side, this graphic illustrates the application of the new vocabularies in other new ones, too. The majority[2] of the represented (re-)utilised existing KRLs and ontologies are already introduced in Chapter 2. A link from one vocabulary to another one was created, when at least one sub class, sub property, domain or range relation to another vocabulary exists. Some existing Semantic Web ontologies were also modified during the development process for the PMKB KRM.

[1] KRM examples are always serialised in N3 (see Section 2.2.2). Graphical illustrations in this section are mainly composed with the help of TopBraid Composer (see Section 2.2.4.1).

[2] Please have a look at [Gän10f] for references to those KRLs and vocabularies which are not separately introduced in this thesis.

The *Property Reification Vocabulary* is explicitly outlined in Subsection 4.1.4.3, due to its noteworthy modifications. An overview of new created or initially published Semantic Web ontologies, vocabularies and taxonomies in this context is given in [Gän10g].

4.1.1 The Info Service Ontology

The *Info Service Ontology* (IS) [Gän10l, Gän10k] is a new Semantic Web ontology which was created as part of this implementation. This vocabulary provides basic concepts and properties for describing different information services, e.g., web information services, for instance, *Wikipedia*, *MusicBrainz*, *Last.fm* or *Echo Nest*. It is intended to be used for information service characterisation (see Subsection 3.3.3). The main concept of this ontology is information service, which is already defined in Section 2.1.2.1.

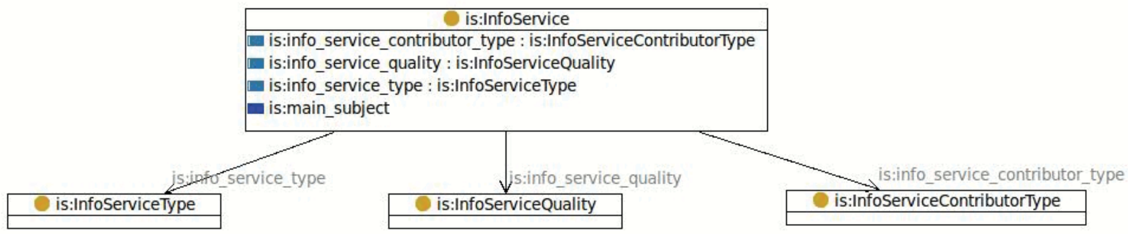

Figure 4.2: The information service concept as graph with relations

The information service concept is represented as class `is:InfoService` in the *Info Service Ontology* and illustrated in Figure 4.2. In this illustration one can see that information service type descriptions, information service quality descriptions and information service contributor type descriptions can easily be related to information service *particulars*, because their *universal* acts at least as domain for *relations* with specific ranges for *universals* of the specific mentioned descriptions. The forth *relation* in the graphic is a property, which enables the opportunity to address a topic that is the main subject of an information service.
In Listing 4.1 one can see that an information resource (in this example a HTML webpage that describes a music artist), can easily be related to an information service description by using the property `is:info_service`. One can infer that this HTML webpage is provided by the information service *MusicBrainz* and a description of this service can be retrieved by requesting the information resource `isi:musicbrainz`. The statement that references to an information service description can be useful for implementing an efficient 'follow-your-nose' algorithm (cf. 2.2.1.2).
The information resource `isi:musicbrainz` is an information service, has a title and a small literally description. Furthermore, besides a topic, which is the main subject, this information service is associated with further topics that are related by the property `dcterms:subject`. All topics are information resources of categories of the information service *DBPedia*[3]. It is important for a description of a web information service that the domain (hostname) of this service is associated with its characterisation. This can be done by using the property `foaf:homepage`.
To continue the categorisation of this information service,

- the information service contributor type is set to `isct:mixed`, which means that a user community and experts contribute to this information service,
- the information service type is set to `ist:encyclopedia` and `ist:knowledge_base`, which are dereferencable resources of the information service types encyclopedia and knowledge base, and

[3] `http://dbpedia.org`

```
@prefix dc:      <http://purl.org/dc/elements/1.1/> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix foaf:    <http://xmlns.com/foaf/0.1/> .
@prefix is:      <http://purl.org/ontology/is/core#> .
@prefix ist:     <http://purl.org/ontology/is/types/> .
@prefix isct:    <http://purl.org/ontology/is/ctypes/> .
@prefix isq:     <http://purl.org/ontology/is/quality/> .
@prefix isi:     <http://purl.org/ontology/is/inst/> .
@prefix rdf:     <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs:    <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd:     <http://www.w3.org/2001/XMLSchema#> .

<http://musicbrainz.org/artist/8a1fe33d-6029-462e-bcb7-08e0ebaba6dd.html>
   a foaf:Document ;
   is:info_service isi:musicbrainz .

isi:musicbrainz
   rdf:type is:InfoService ;
   rdfs:isDefinedBy isi: ;
   dc:description "An open content music database."@en ;
   dc:title "MusicBrainz"^^xsd:string ;
   dcterms:subject <http://dbpedia.org/resource/Category:Metadata_registry> ,
      <http://dbpedia.org/resource/Category:Acoustic_fingerprinting> ,
      <http://dbpedia.org/resource/Category:MusicBrainz> ,
      <http://dbpedia.org/resource/Category:Library_2.0> ,
      <http://dbpedia.org/resource/Category:Online_encyclopedias> ,
      <http://dbpedia.org/resource/Category:Online_music_and_lyrics_databases>
          ,
      <http://dbpedia.org/resource/Category:Free_websites> ;
   is:info_service_contributor_type
      isct:mixed ;
   is:info_service_quality
      isq:good ;
   is:info_service_type
      ist:encyclopedia , ist:knowledge_base ;
   is:main_subject <http://dbpedia.org/resource/Category:Music> ;
   foaf:homepage <http://musicbrainz.org/> .
```

Listing 4.1: A description of the information service *MusicBrainz*

- the information service quality is set to `isq:good`, which is currently simply the second highest rating on a scale of six levels.

Especially, descriptions of information quality can be made on different levels of detail and complexity. Still, it is important that very simple and high level descriptions can be represented to the user. Accordingly, a KMS must be able to infer high level descriptions from complex and more precise information service quality ratings.
An implementation of a reference information service quality ontology to enable detailed information service quality description can be done in future work. Such an ontology has to be based on a specific classification for that purpose, for example, this one from Wang et al. [WS96]. Thereby, the property `is:info_service_quality` always acts as *relation* to such information service quality descriptions.
Information service descriptions and/or information service quality descriptions can be provided by different information service rating agencies. This condition can assists the user to select information services on the basis of information service descriptions provided by information service rating agencies he/she trusts.

4.1.2 The Play Back Ontology and related Ontologies

We have to introduce several new Semantic Web ontologies that are later reutilised and applied in the domain-dependent *Play Back Ontology* (PBO; see Subsection 4.1.2.4) and that assist to describe *particulars* in this context. They all have a broader range and can be applied in other domains and use cases as well. The first one is the *Ordered List Ontology* (OLO; see Subsection 4.1.2.1). This vocabulary is followed by the *Counter Ontology* (CO; see Subsection 4.1.2.2). Finally, the *Association Ontology* (AO; see Subsection 4.1.2.3) close this explanation of required *simple specialised ontologies*.

4.1.2.1 The Ordered List Ontology

There is a need to describe typed ordered lists or sequences of *entities*. The existing approach in the RDF vocabulary (see Section 2.2.2), `rdf:Seq`, has some drawbacks. One shortcoming is that an explicit definition of a range for items of a sequence is not possible. Another one is related to an efficient query opportunity with SPARQL for such ordered lists. This issue is due to the reason that every slot of a sequence is related by a separate property of the form `rdf:_N`, where `N` is a positive integer number, e.g., `rdf:_1` (cf. [Bri04, Hay04]). That is why I co-designed the *Ordered List Ontology* (OLO) [AG10, Gän10m] according to a proposal made by Samer A. Abdallah. This *simple specialised ontology* is supposed to overcome the drawbacks of the existing ordered list modelling approach by enabling more semantics to describe sequences of resources. It aligns with the basic ontology design pattern for representing ordered lists (see [Blo10, BG11b]).

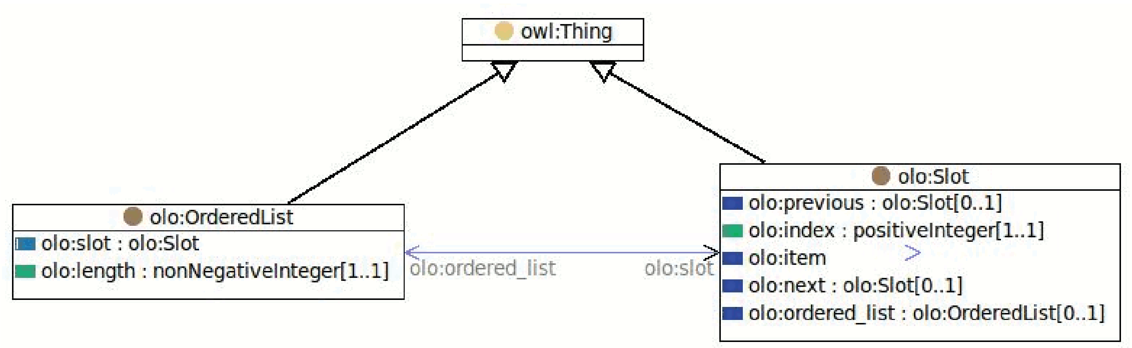

Figure 4.3: The ordered list concept as graph with relations

As one can see in Figure 4.3, OLO consists of two concepts - "ordered list" and "slot". In this context, an ordered list is a composite of all slots, i.e., they are part of this sequence. The ontology provides "backward" compatibility to the RDFS world, although its definitions are OWL-based (cf. Section 2.2.2).
The initial and primary access method to single slots in an ordered list is their index, which is fixed for a slot in a sequence[4]. Thereby, the length of a sequence is the number of included slots. The secondary access method is its optional iterator `olo:next` as shortcut to a next slot in a list. Items which are arranged in an ordered list, are associated with the item *relation* to a slot.
All properties also have an inverse property, since OLO will not restrict its users in how they have to apply this ontology. These are explicitly defined for properties, which introduce a new meaning (`olo:ordered_list`, `olo:previous`), and anonymously defined for the rest of the inverse properties.
Listing 4.2 shows a simple sequence that is modelled with the help of OLO. It consists of an ordered list of the length two and has some further editorial metadata attached. These are a title, a description and a reference to its creator. Each slot has its index and an item relation. The items

[4] This is in contrast to the sequence modelling approach of the *SWAN Collections Ontology* [Cic09]. See [Gän11a] for a detailed comparison of the ordered list part of this ontology with OLO.

```
@prefix xsd:     <http://www.w3.org/2001/XMLSchema#> .
@prefix dc:      <http://purl.org/dc/elements/1.1/> .
@prefix olo:     <http://purl.org/ontology/olo/core#> .
@prefix owl:     <http://www.w3.org/2002/07/owl#> .
@prefix ex:      <http://example.org/> .

ex:AThing a owl:Thing .

ex:AnotherThing a owl:Thing .

ex:ASequence a olo:OrderedList ;
   dc:title "A sequence example"^^xsd:string ;
   dc:description "A sequence modelled with the help of OLO"^^xsd:string ;
   dc:creator <http://foaf.me/zazi#me> ;
   olo:length 2 ;
   olo:slot [
      a olo:Slot ;
      olo:index 1 ;
      olo:item ex:AThing
   ] ;
   olo:slot [
      a olo:Slot ;
      olo:index 2 ;
      olo:item ex:AnotherThing
   ] .
```

Listing 4.2: A sequence modelled with the help of OLO

are simple `owl:Thing` instances.
One can define more specific ordered list definitions, e.g., for media playlists (see Subsection 4.1.2.4) or ranked recommendations (see Subsection 4.1.3), on the basis of the ordered list concept defined in OLO.

4.1.2.2 The Counter Ontology

Usage statistics get more and more important, especially in the context of personalisation. It might be useful to represent them with the help of a multiple purpose counter concept as part of a KRM that is based on Semantic Web ontologies. That is why I co-developed the *Counter Ontology* (CO) [RSG10, Gän10j]. It is a generalisation and extension of the *Playcount Ontology*[Rai08d] and the *Scrobble Ontology* that both were proposed by Yves Raimond (see [Rai08a]).

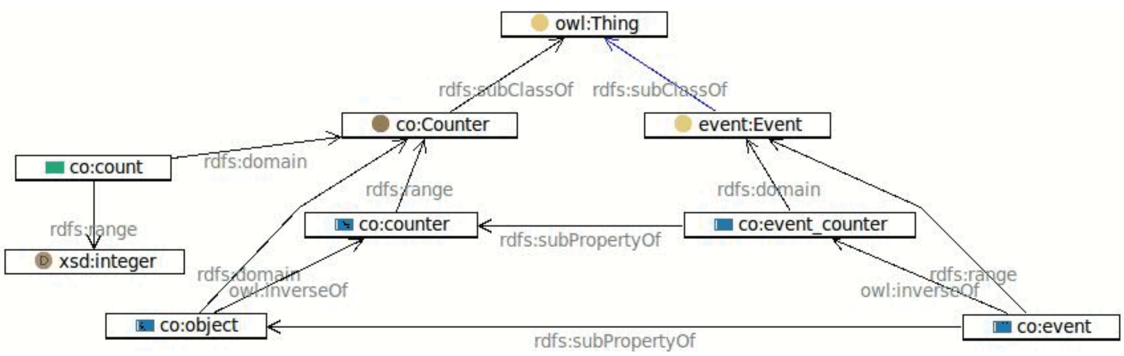

Figure 4.4: The counter concept as graph with relations

Figure 4.4 illustrates the design of the intended counter concept that is represented by the *universal* `co:Counter`. One can associate countable information resources with a counter by using

the property `co:counter` or its inverse one, `co:object`. The count *relation* assigns the numeric value of a counter. It is a simple integer-based datatype property.
The *Counter Ontology* already includes predefined properties to associate event specific counter to its related events. These are *particulars* of the event concept of the *Event Ontology* (see Figure 4.4). A refinement of this *universal* is the *scrobble event*[5]. This is a general multiple purpose activity record event concept. Due to its importance, *scrobble event* is a part of CO (see Figure 4.5).

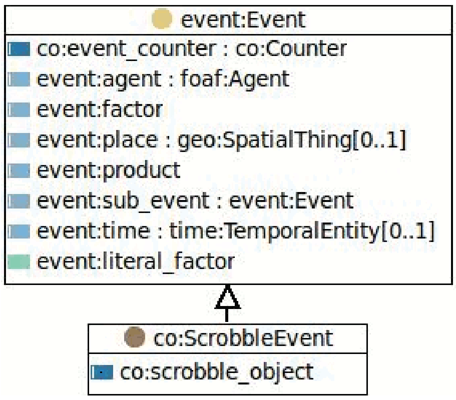

Figure 4.5: The *scrobble event* concept as graph with relations

This enables the opportunity to trace back all related events which are responsible for a specific count. Of course, this is also possible with all other concepts that are utilised by a counter. Objects that are a factor of a *scrobble event* can be related separately by using the *scrobble* object property. This might be useful, if, for instance, the exemplar (e.g. a specific music document) of a music track varies between different *scrobble events* that are related to one and the same counter for that music track.

```
@prefix bibo:    <http://purl.org/ontology/bibo/> .
@prefix xsd:     <http://www.w3.org/2001/XMLSchema#> .
@prefix dc:      <http://purl.org/dc/elements/1.1/> .
@prefix co:      <http://purl.org/ontology/co/core#> .
@prefix ex:      <http://example.org/> .

ex:WebpageCounter a co:Counter ;
   dc:title "Webpage Counter"^^xsd:string ;
   dc:creator <http://foaf.me/zazi#me> ;
   dc:description "A counter of a specific web page."^^xsd:string ;
   co:count 10 .

<http://smiy.org/> a bibo:Document ;
   co:counter ex:WebpageCounter .
```

Listing 4.3: A web page counter modelled with the help of CO

For this purpose one might create further, more specific *universals* on the basis of the general counter type, e.g., a play back or skip counter (see Subsection 4.1.2.4). Such specialisations enable counting different activities separately and accessing those counters in an easy way.
On the basis of the counter concept one can already easily model a web page counter[6] as it is shown in Listing 4.3. This example describes a web page that is associated with a counter. A document has a count of ten and some further editorial metadata (a title, a description and a reference to the creator) are attached.

[5]Inspired by the term 'scrobble' as introduced by *Last.fm*, which means, to record a listening activity regarding a specific music track

[6]Toby Inkster implemented an RDFa serialisation of a web page counter based on CO in PHP (see [Ink10])

4.1.2.3 The Association Ontology

Modelling association or annotation statements has a long history in the Semantic Web community. The basics of this domain and the principal disadvantages of its existing approaches are already outlined in Section 2.4.2. To overcome the named drawbacks of the majority of existing KR vocabularies in that field, I co-designed the *Association Ontology* (AO) [GI10, Gän10h].
AO combines features of SIM, the *Review Vocabulary* (REV) [AH07] and *DCMI*[7] *Metadata Terms* (DC/DC Terms) [DCM08]. The intend behind this *simple specialised ontology* is to reutilise a mechanism of SIM to append (personal) association statements (`sim:Association`) to *something*[8] by applying the relation `sim:association`. This step of indirection is necessary to enable

- reusable association statements (RAS) and
- voting, rating and reviewing of association statements in a specific context

Therefore, the likeable association statement concept was created in AO, which combines the association statement concept of SIM and review concept of the REV. Simple voting (the "like button") can be realised by using the property `ao:likeminded`, which creates a relation between an association statement and an individual (based on the agent concept of FOAF). Ratings and reviews can be described by using the features of the *Review Vocabulary*, e.g., the rating *relation* or the feedback concept.
To address (general) associations of a specific domain, e.g., genre, mood or occasion, new sub properties based on the subject *relation* of *DC Terms* or the context *relation* of AO (for environmental context associations) were created and further ones may be created in the future. These are (currently):

- `ao:genre`, for genre descriptions of all kind, e.g., a music or film genre
- `ao:mood`, e.g., happy or sad
- `ao:occasion`, e.g., a birthday or Christmas
- `ao:activity`, e.g., dancing, sleeping, driving
- `ao:application`, e.g., a music player (e.g. that is related to its currently playing track)
- `ao:device`, e.g., a CD player (e.g. that is related to its currently playing CD)
- `ao:location`, e.g., my house, my country, my current whereabouts
- `ao:time`, e.g., morning, afternoon, evening

These attributes are intended to be an abstract and general hook into their specific domains. Furthermore, new, more specific sub properties based on these *relations* should be created to provide a hook in more specific domains, e.g., a music genre *relation* to associate music genres/styles[9].
The property `ao:included_association` was created to enable voting, rating and reviewing of a RAS in a specific context. By using this attribute one can include a RAS into another annotation statement (preferably based on the likeable association concept).
As mentioned at the beginning of this section, the modelling approaches for annotation statements are occasionally applying the concept of *Named Graphs*. However, the *Association Ontology* at its current state does not really include this modelling issue. There is an experimental branch in the source code repository of AO that tries to handle this feature (see [Gän10a], especially the remodelled examples). *Named Graphs* support will be added to AO officially maybe some day in the future.
As one might notice, the terms of the *Association Ontology* can be perfectly combined with terms of other ontologies, e.g., the following *Play Back Ontology*. They enable the attachment of (personal) association statements to existing KRs of any kind, e.g., a mood and an occasion to a music playlist description (see also the example in the next subsection).

[7] DCMI is an abbreviation for *Dublin Core Metadata Initiative* (see [Dub11a]).
[8] In this context *owl:Thing* based particulars.
[9] This sub property relation exist since MO version 2.1, see [RGJ+10, RG10d].

4.1.2.4 The Play Back Ontology

This subsection demonstrates the reutilisation, specialisation and application of the previous introduced, quite domain independent Semantic Web ontologies by presenting the *Play Back Ontology* (PBO) [GJ10b, Gän10n]. Of course, further, other Semantic Web ontologies, e.g., the *Bibliographic Ontology* (BIBO) [GD10] or FRBR, are utilised to support a helpful ontology alignment and to realise the modelling of the intended use cases.
The *Play Back Ontology* describes basic concepts and properties for modelling ontological KRs that are related to the *play back domain*, e.g., playlist, play back and skip counter. One can define the *play back domain* as following:

> The **play back domain** is a subset of the *media domain*. It deals with playing back some media, e.g., videos, music tracks or slideshows, and how used media can be structured or organised.

Objects that are included into this domain are *media objects*. All concepts of PBO are somehow related to at least one *media object*. The range of these *media objects* is currently restricted to the document concept of BIBO and the endeavour concept of FRBR. This includes all their sub classes, e.g., the track concept of MO.

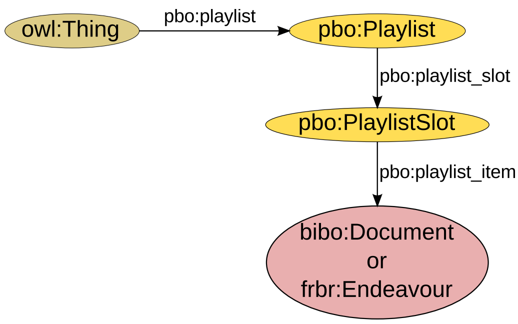

Figure 4.6: The playlist concept as graph with relations

Figure 4.6 illustrates the playlist concept of PBO. This *universal* is derived from the document concept of BIBO and the ordered list concept of OLO. Hence, it is a specialised ordered list of the *media domain*. A playlist consists of particular playlist slots. They are related to at least one *media object*. The playlist concept has a further specialisation. It is a fixed playlist concept, to describe playlists or sections of playlists which have a strict order. Such an arrangement can be, for example, a section of music tracks that should always be played one after another, or that are related to each other in some way. This feature is especially useful for automatic playlist generation tasks. Playlists can be attached to *something*[10] by using the property `pbo:playlist`. The latest addition to the PBO playlist concept is the a transition *relation*. It can be used to associate a description of a transition between two neighbouring playlist slots, e.g., two successive music tracks in a dj mix.
Furthermore, PBO consists of a couple of *media action counters* (MAC). They are related to concrete activities that are usually done when dealing with *media objects*. For example, a music track can be, amongst others, played back or skipped. The MAC concept is derived from the counter concept of CO. One can see this *universal* and its current specialisations, the play back and skip counter concept, in Figure 4.7. The *media object relation* only associates *media objects* to MAC *particulars*. The same behaviour is realised for *media object* relations to *scrobble events* by using the property 'media scrobble object'. Finally, the refined *scrobble event* concept 'skip event', enables the representation of a skip time of skip activity regarding a *media object*.
Now, one should have been given a good overview about the main concepts of the *play back domain*. Some illustrating examples of use cases demonstrate the applicability of PBO and related

[10] Every `owl:Thing` based concept.

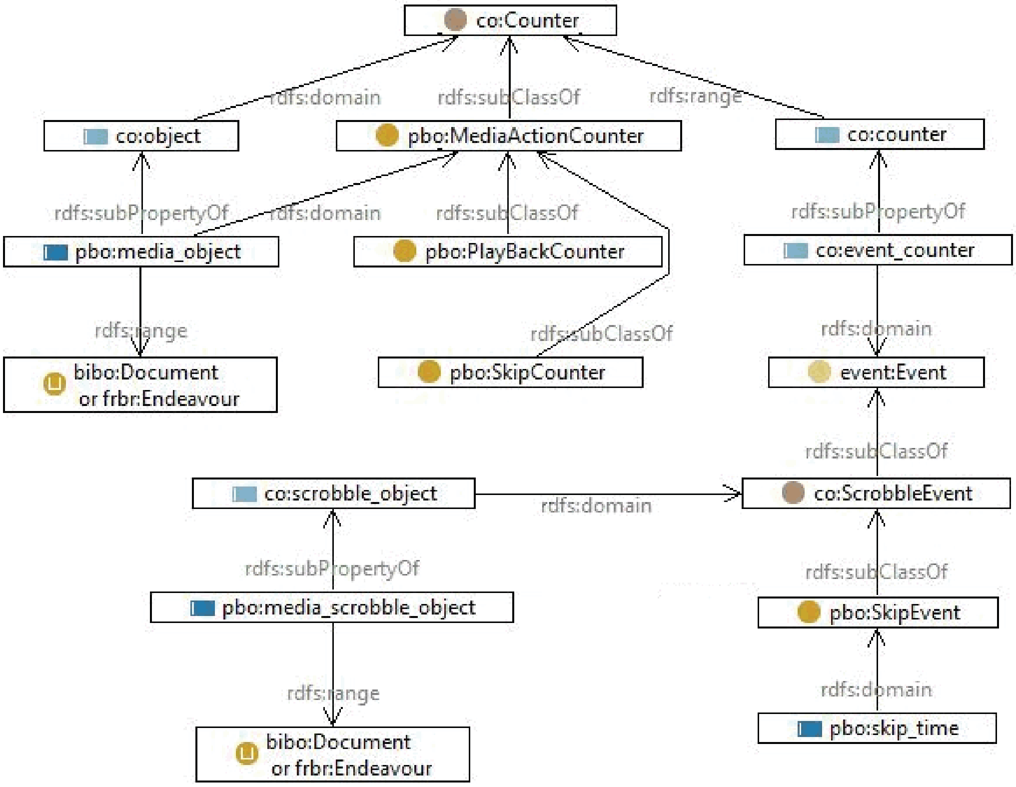

Figure 4.7: The play back and skip counter concepts as graph with relations

Semantic Web ontologies. The first example describes a music playlist and the second one the usage of play back and skip counter.
Figure 4.8 illustrates a music playlist example created with concepts of PBO. This playlist is annotated with further editorial metadata - a title and a creator. It has a fixed length of two. Each slot is modelled as a playlist slot to ensure that the items are *media objects*. In the example these are music tracks.
Moreover, he music playlist is annotated with two association statements. These are Zazi's association and `ex:BobsAssociationInUse`. The first one is modelled as an association to a specific occasion. Besides, it describes (music) genre and mood categorisations as simple string-typed literals. The second association statement is a likeable association statement that includes a RAS (Bob's association). This RAS associates[11] a specific music genre - *Funk*. Other people can express that they like the resource `ex:BobsAssociationInUse` in this context (the described music playlist). In the example the person Yves Raimond[12] likes the related association statement.
Listings 4.4 and 4.5 illustrate a *media action counter* counter example. It shows a play back counter and as a skip counter for a specific music track. Each time this specific music track was played, a *scrobble event* was created and linked to its play back counter. The *scrobble* time[13] and a related agent are part of the description of these an activity trackings. Further context information is addressed to model a *scrobble event* more in detail. In the given example this are a used application (*iTunes*) and a used device (my PC).
When this specific music track was skipped, a specialised skip event was created (see Listing 4.5). Such an event is related to its skip counter instance. To represent the occurred skip time, the object of this relation is modelled as a time instant on a timeline[14]. It describes the moment where the user skipped the music track. That is 30 seconds after the start of the music

[11] This music genre can maybe precisely related by using the music genre *relation* of MO (cf. Subsection 4.1.2.3).
[12] Represented by the information resource `http://moustaki.org/foaf.rdf#moustaki`.
[13] This is a *particular* of the time instant concept of the *OWL Time Ontology* (TIME) [Hob06].
[14] This is a *particular* of the timeline concept of TL.

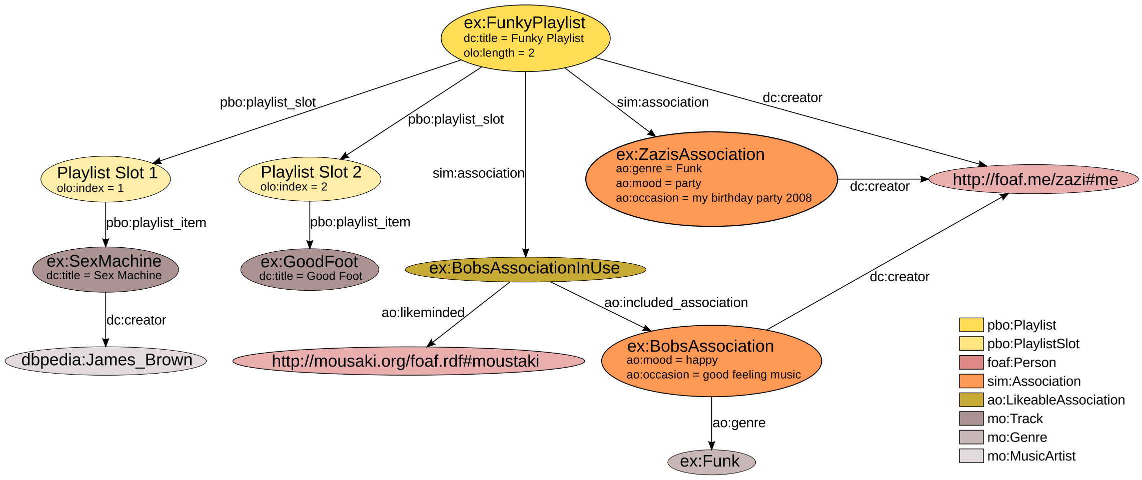

Figure 4.8: A music playlist created with the help of PBO

track. The specific timeline is related to the original audio signal of the published music track (`ex:SexMachine`, see Listing 4.4).
As one could see, the *Play Back Ontology* and the illustrated examples demonstrate the reutilisation, specialisation and application of concepts of existing ontologies for modelling domain specific KRs. Especially the use cases of the example can be the foundation of further (personalised) usage behaviour analysis tasks.

```
@prefix xsd:     <http://www.w3.org/2001/XMLSchema#> .
@prefix dc:      <http://purl.org/dc/elements/1.1/> .
@prefix co:      <http://purl.org/ontology/co/core#> .
@prefix pbo:     <http://purl.org/ontology/pbo/core#> .
@prefix ex:      <http://example.org/> .

ex:PlayBackCounter a pbo:PlayBackCounter ;
   dc:title "Play Back Counter"^^xsd:string ;
   dc:creator <http://foaf.me/zazi#me> ;
   dc:description "A play back counter of a specific music track"^^xsd:string ;
   co:count 1 ;
   pbo:media_object ex:SexMachine .

ex:SkipCounter a pbo:SkipCounter ;
   dc:title "Skip Counter"^^xsd:string ;
   dc:creator <http://foaf.me/zazi#me> ;
   dc:description "A skip counter of a specific music track"^^xsd:string ;
   co:count 1 ;
   pbo:media_object ex:SexMachine .

ex:SexMachine a mo:Track ;
   dc:title "Sex Machine"^^xsd:string ;
   dc:creator <http://dbpedia.org/resource/James_Brown> .
```
Listing 4.4: A PBO play back and skip counter example (to be continued in Listing 4.5)

```
@prefix xsd:    <http://www.w3.org/2001/XMLSchema#> .
@prefix dc:     <http://purl.org/dc/elements/1.1/> .
@prefix co:     <http://purl.org/ontology/co/core#> .
@prefix pbo:    <http://purl.org/ontology/pbo/core#> .
@prefix mo:     <http://purl.org/ontology/mo/> .
@prefix ex:     <http://example.org/> .
@prefix event:  <http://purl.org/NET/c4dm/event.owl#> .
@prefix time:   <http://www.w3.org/2006/time#> .
@prefix tl:     <http://purl.org/NET/c4dm/timeline.owl#> .
@prefix am:     <http://vocab.deri.ie/am#> .
@prefix owl:    <http://www.w3.org/2002/07/owl#> .
@prefix ao:     <http://purl.org/ontology/ao/core#> .

ex:SexMachineTL a tl:TimeLine .

ex:SexMachineSignal a mo:Signal ;
   mo:time [
      a time:Interval ;
      tl:duration "PT5M18S"^^xsd:duration ;
      tl:timeline ex:SexMachineTL
      ] ;
   mo:published_as ex:SexMachine .

ex:iTunes a am:Application .

ex:MyPC a owl:Thing .

ex:SexMachineSE1 a co:ScrobbleEvent ;
   event:time [
      a time:Instant ;
      time:inXSDDateTime "2010-07-15T11:21:52+01:00"^^xsd:dateTime
      ] ;
   event:agent <http://foaf.me/zazi#me> ;
   ao:used_application ex:iTunes ;
   ao:used_device ex:MyPC ;
   co:event_counter ex:PlayBackCounter .

ex:SexMachineSE3 a pbo:SkipEvent ;
   pbo:skip_time [
      a time:Instant ;
      tl:timeline ex:SexMachineTL ;
      tl:at "PT30S"^^xsd:duration
      ] ;
   event:time [
      a time:Instant ;
      time:inXSDDateTime "2010-07-15T11:35:52+01:00"^^xsd:dateTime
      ] ;
   event:agent <http://foaf.me/zazi#me> ;
   ao:used_application ex:iTunes ;
   ao:used_device ex:MyPC ;
   co:event_counter ex:SkipCounter .
```

Listing 4.5: A PBO skip event example (as sequel of Listing 4.4)

Copyright TU Dresden, Thomas Gängler

4.1.3 The Recommendation Ontology

Music recommendation is the show case to demonstrate the added value of the concept PMKB. For that reason, it is important to be able to represent high-level recommendations in our KRM. Such a KR especially makes sense, if recommendations are provided by several information services that are consumed and processed by our KMS (see Section 3.3.3).
I co-designed the *Recommendation Ontology* (REC) [GJ10c, Gän10o], since a Semantic Web ontology for representing high-level recommendations as semantic graph did not exist. This *simple specialised ontology* hooks up parts of SIM, DC Terms, the *Association Ontology* (see Subsection 4.1.2.3) and the *Ordered List Ontology* (see Subsection 4.1.2.1).

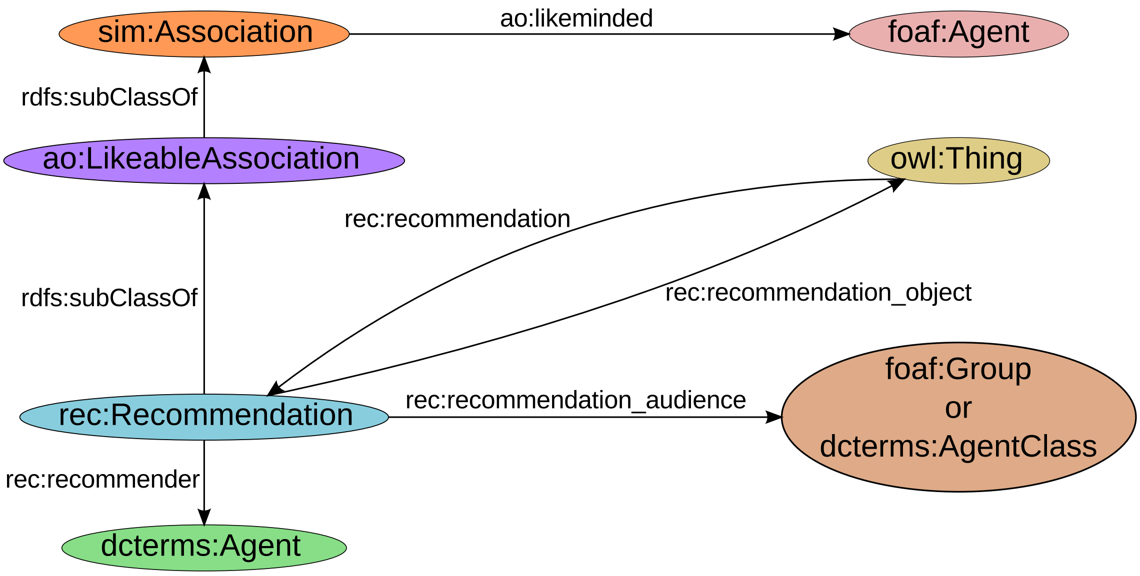

Figure 4.9: The recommendation concept as graph with relations

Figure 4.9 illustrates the core of REC, the recommendation concept. A recommendation can (/is supposed to) consist of

- an item or agent relation (`rec:recommendation` or its inverse property `rec:for`) to associate the resource for which the recommendation was made
- a recommender relation to associate the *particular*, which provided or calculated the recommendation, e.g., an information service (see Subsection 4.1.1)
- recommendation object relations to associate appropriate objects to the recommendation subject
- recommendation audience relations to associate groups or stereotypes, which are probably appropriate as target group for this recommendation (at least).

It is possible to like, rate or give feedback to a recommendation, since the recommendation concept is based on the likeable association statement concept (cf. Subsection 4.1.2.3). Furthermore, one can include detailed association or similarity statements (`sim:Similarity` based) in a recommendation description (cf. Subsection 4.1.2.1).
As an extension of the recommendation concept, I created the ranked recommendation concept to enable ordered (ranked) recommendations at a high level. Figure 4.10 illustrates this concept as graph with relations. A ranked recommendation is not only based on the recommendation concept. Besides, it is based on the ordered list concept of OLO. This enables all features of sequences to ranked recommendations. Following this design, the ranked recommendation object

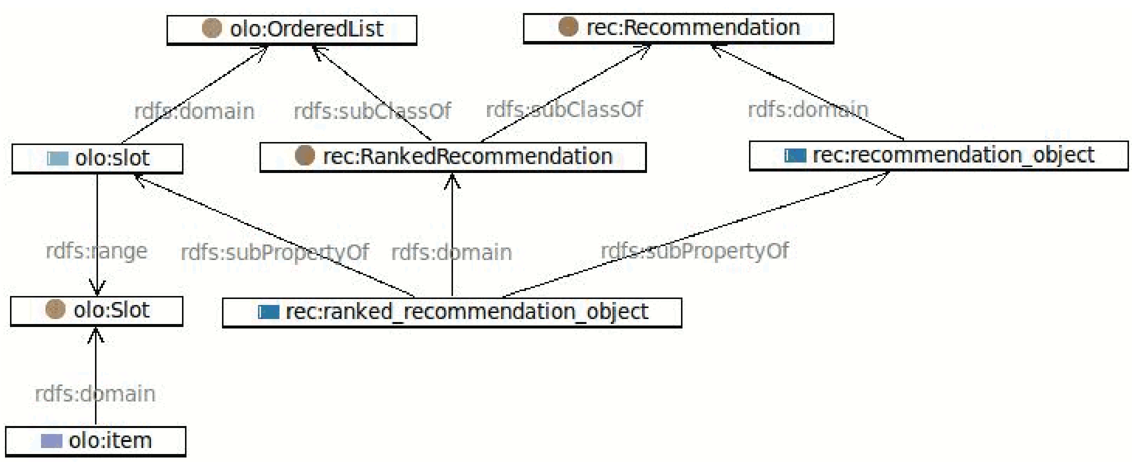

Figure 4.10: The ranked recommendation concept as graph with relations

attribute is not only a sub property of the recommendation object *relation*. In addition, it is a sub property of the slot *relation* of OLO. In this context, recommendation objects are associated by using the item *relation* of OLO.
Below are examples of use cases of REC. The first one describes a simple music track recommendation and the second one an extended music track recommendation.

```
@prefix rec: <http://purl.org/ontology/rec/core#> .
@prefix ex: <http://example.org/> .
@prefix isi: <http://purl.org/ontology/is/inst/> .
@prefix sim: <http://purl.org/ontology/similarity/> .
@prefix is: <http://purl.org/ontology/is/core#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix dbtune: <http://dbtune.org/musicbrainz/resource/track/> .

ex:AMusicRecommendation a rec:Recommendation ;
   rec:for dbtune:008e72df-6469-4557-8b5b-c54c3285fbd3 ;
   rec:recommender isi:lastfm ;
   rec:recommendation_audience ex:FunkyPeople ;
   sim:subject dbtune:008e72df-6469-4557-8b5b-c54c3285fbd3 ;
   rec:recommendation_object dbtune:097c362d-72b7-4a53-96e2-d9ff02f8be1f ;
...
   rec:recommendation_object dbtune:0348eea1-8178-4dc1-8a37-d09b5897ace2 ;
   sim:method ex:lfmTrackSimilarity .

ex:lfmTrackSimilarity a sim:AssociationMethod ;
   is:info_service isi:lastfm .

ex:FunkyPeople a foaf:Group .
```

Listing 4.6: A simple music recommendation example modelled with the help of REC (see [GJ10c] for the complete example)

The music recommendation example in Listing 4.6 is simply stripped down to its high-level recommendation. It is based on the music track recommendations of James Brown's "Get Up (I Feel Like Being a) Sex Machine" from *Last.fm*[15]. Hence, the subject of the `sim:subject` relation is an information resource of this music track at *MusicBrainz*[16], modelled and represented as semantic graph and provided by *DBTune*.
The recommendation target is related by the property `rec:for` (which is equal to its subject in

[15]See http://bit.ly/bcjFcR
[16]See http://musicbrainz.org/artist/20ff3303-4fe2-4a47-a1b6-291e26aa3438.html

this example). This recommendation includes a fictional audience group (`ex:FunkyPeople`). The recommendation objects are simply addressed via the recommendation object attribute. Thereby, each recommendation object is also a music track description at *MusicBrainz*, modelled and represented as semantic graph and provided by *DBTune*. This example does not contain a ranking. Currently, it is an unordered list. However, this recommendation involves a relation to its similarity method. This association method cannot be further specified in this context, because it is based on a proprietary algorithm from *Last.fm*. Therefore, the complete recommendation and also the association method itself has as an attribute value the information service *Last.fm*.
Listings 4.7 and 4.8 show an extended music track recommendation modelled as a *particular* of the ranked recommendation concept. It is quite similar to the simple music track recommendation example as described in Listing 4.6. The subject of the `rec:recommendation` attribute is the same information resource as in the `rec:for` relation of the simple music recommendation example. The ranked recommendation concept makes use of ordered list features.
Those imply the opportunity to provide an ordered (/ranked) list of recommendation objects by hiding the details of their ranking features on this level. These details are supplied by further similarity statements. Each slot item represents a music track description equal to those which are directly associated by the recommendation object relation in the simple music recommendation example (see Listing 4.6).
Each of the attached similarity statements includes

- the seed music track (related by `sim:subject`), on which this similarity calculation is based,
- the other (recommended) music track (related by `sim:object`),
- the weight of the similarity, and
- the association method of the similarity calculation.

The *Recommendation Ontology* and the illustrated examples demonstrated the reutilisation, specialisation and application of concepts of existing ontologies again. One can think about more specialised concepts and properties which are based on the introduced ontologies. Moreover, further examples that make use of REC are imaginable, e.g., suggested items from *Amazon*, which might be modelled with the help of the *GoodRelations Vocabulary* [Hep10], too.

```
@prefix rec: <http://purl.org/ontology/rec/core#> .
@prefix ex: <http://example.org/> .
@prefix olo: <http://purl.org/ontology/olo/core#> .
@prefix isi: <http://purl.org/ontology/is/inst/> .
@prefix sim: <http://purl.org/ontology/similarity/> .
@prefix dbtune: <http://dbtune.org/musicbrainz/resource/track/> .

dbtune:008e72df-6469-4557-8b5b-c54c3285fbd3
   rec:recommendation ex:AMusicRecommendation .

ex:AMusicRecommendation a rec:RankedRecommendation ;
   rec:recommender isi:lastfm ;
   sim:subject dbtune:008e72df-6469-4557-8b5b-c54c3285fbd3 ;

   rec:ranked_recommendation_object [
      a olo:Slot ;
      olo:item dbtune:097c362d-72b7-4a53-96e2-d9ff02f8be1f ;
      olo:index 1
      ] ;
```

Listing 4.7: An extended music recommendation example modelled with the help of REC (to be continued in Listing 4.8; see [GJ10c] for the complete example)

```
@prefix rec: <http://purl.org/ontology/rec/core#> .
@prefix ex: <http://example.org/> .
@prefix olo: <http://purl.org/ontology/olo/core#> .
@prefix isi: <http://purl.org/ontology/is/inst/> .
@prefix ao: <http://purl.org/ontology/ao/core#> .
@prefix sim: <http://purl.org/ontology/similarity/> .
@prefix is: <http://purl.org/ontology/is/core#> .
@prefix dbtune: <http://dbtune.org/musicbrainz/resource/track/> .

...
   rec:ranked_recommendation_object [
      a olo:Slot ;
      olo:item dbtune:0348eea1-8178-4dc1-8a37-d09b5897ace2 ;
      olo:index 10
      ] ;
   ao:included_association ex:SimAssociation01 ;
...
   ao:included_association ex:SimAssociation10 .

ex:lfmTrackSimilarity a sim:AssociationMethod ;
   is:info_service isi:lastfm .

ex:Association01 a sim:Similarity ;
   sim:subject dbtune:008e72df-6469-4557-8b5b-c54c3285fbd3 ;
   sim:object dbtune:097c362d-72b7-4a53-96e2-d9ff02f8be1f ;
   sim:weight 1.0 ;
   sim:method ex:lfmTrackSimilarity .
...
ex:Association10 a sim:Similarity ;
   sim:subject dbtune:008e72df-6469-4557-8b5b-c54c3285fbd3 ;
   sim:object dbtune:0348eea1-8178-4dc1-8a37-d09b5897ace2 ;
   sim:weight 0.336997 ;
   sim:method ex:lfmTrackSimilarity .
```

Listing 4.8: An extended music recommendation example modelled with the help of REC (as sequel of Listing 4.7; see [GJ10c] for the complete example)

4.1.4 The Cognitive Characteristics Ontology and related Vocabularies

The introduction of the *Cognitive Characteristics Ontology* (CCO; see Subsection 4.1.4.2) involves two further KR vocabularies. They are utilised, reutilised and applied in the domain dependent CCO and assist to describe *particulars* in this context. However, the auxiliary vocabularies have a broader range and can be applied in other domains and use cases as well. The first one is a Semantic Web ontology called *Weighting Ontology* (WO; see Subsection 4.1.4.1). The second one, is a vocabulary called *Property Reification Vocabulary* (PRV; see Subsection 4.1.4.3). It is explained after the introduction of CCO itself, because the utilisation of PRV is demonstrated by applying terms of CCO.

4.1.4.1 The Weighting Ontology

The *Weighting Ontology* [Gän10p] is guided by the design principle of modularisation and abstraction as usual. It is a Semantic Web ontology for describing weightings and their referenced scales as semantic graphs. For that matter, this *simple specialised ontology* includes a general multiple purpose weight concept on top of *Statistical Core Vocabulary* (SCOVO) [AFH+10] terms. Weightings are especially important for statistical analysis tasks. The design of this ontology is an application of the *interpretation property* pattern as described by Tim Berners-Lee in [BL09c] that is intended for describing the interpretation of datatype values, e.g., numbers or strings, as *object-oriented context* [MK03].

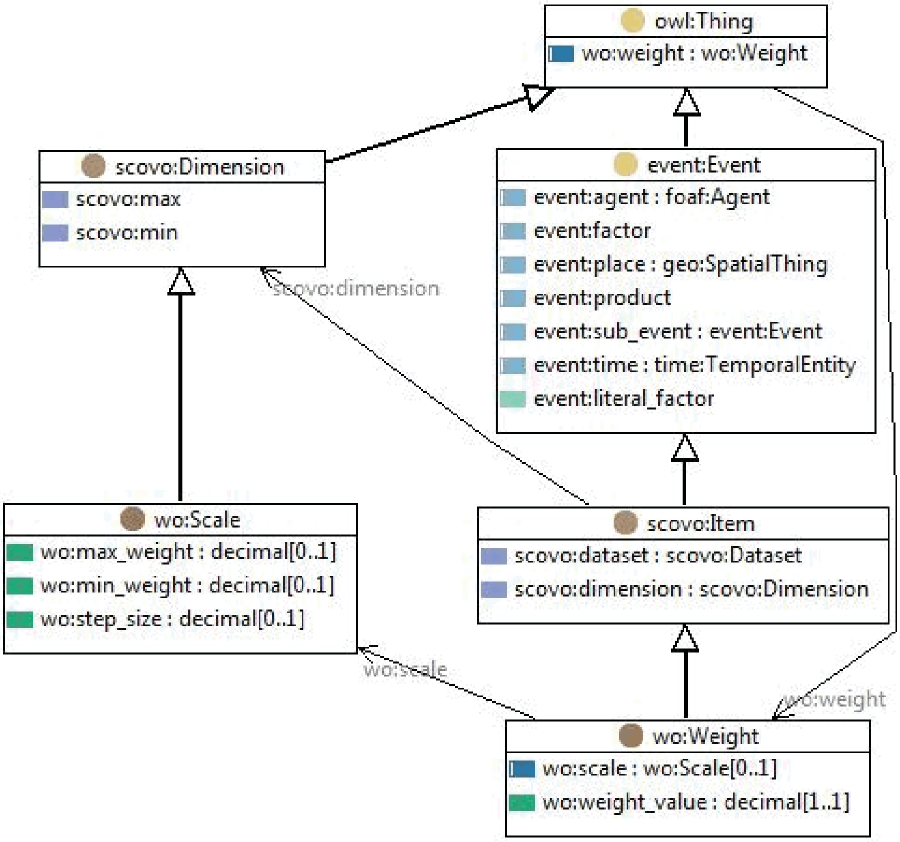

Figure 4.11: The weight concept as graph with relations

In Figure 4.11 one can see that the weight concept, a specialisation of the item concept of SCOVO, can be used to associate any *owl:Thing* based information resource to (a) weight *particular*(s). The numeric value of a weighting is related by the weight value *relation*, which is a simple decimal typed datatype property.
Furthermore, this ontology includes a scale concept, which is based on the dimension concept of SCOVO. It is enables the relation of a specific scale to a weighting and prepares such weight KRs for further automatic processing. One can associate a minimum and a maximum weight to this dimension to define the range of a scale. These *relations* are based on the minimum and maximum relations of SCOVO and REV. Finally one can define a step size for weight scales.
Due to its generalisation, this ontology can be applied side-by-side with other Semantic Web ontologies to model a huge variety of use cases. One of them is the subjective weighting of cognitive patterns of a human being as it is explained in the following subsection.

4.1.4.2 The Cognitive Characteristics Ontology

It is important that a PMKB KMS can work with a proper KR for user profiles. The description of cognitive patterns of a user, e.g., interests, skills or expertise, is a part of user modelling (see Section 2.4.1). As a result, it is a main concern that they are represented in an elegant form. Currently, several approaches regarding the representation of interests exist in form of separate Semantic Web ontologies, e.g., the *Weighted Interest Vocabulary*, the *e-foaf:interest Vocabulary*, or the *Interest Mining Ontology* [PKB10]. Each of the mentioned KR vocabularies is aligned to

the interests modelling concept of FOAF in some way.
Due to that reason, I decided to roll all these interest related ontologies into one hood[17], properly model some concepts and add useful extensions. The result of this task was the *Weighted Interests Vocabulary* version 0.5 (WI) [BMI+10]. However, the range of that ontology is still a bit limited, because it "only" provides basic concepts and properties for describing preferences (interests) within contexts, their temporal dynamics and their origin.
That is why, I designed the *Cognitive Characteristics Ontology* [BRM+10, Gän10i] on top of the latest revision of WI to broaden the scope and being able to model cognitive patterns. For that matter, all concepts and properties are imported from this ontology (WI) into CCO initially. Some of them are redefined and renamed to broaden their meaning. Moreover, CCO is inspired by UUCM, GUMO, UMIRL, all their fundamental sources[18], and, finally, the discussions on the FOAF developers mailing list[19].

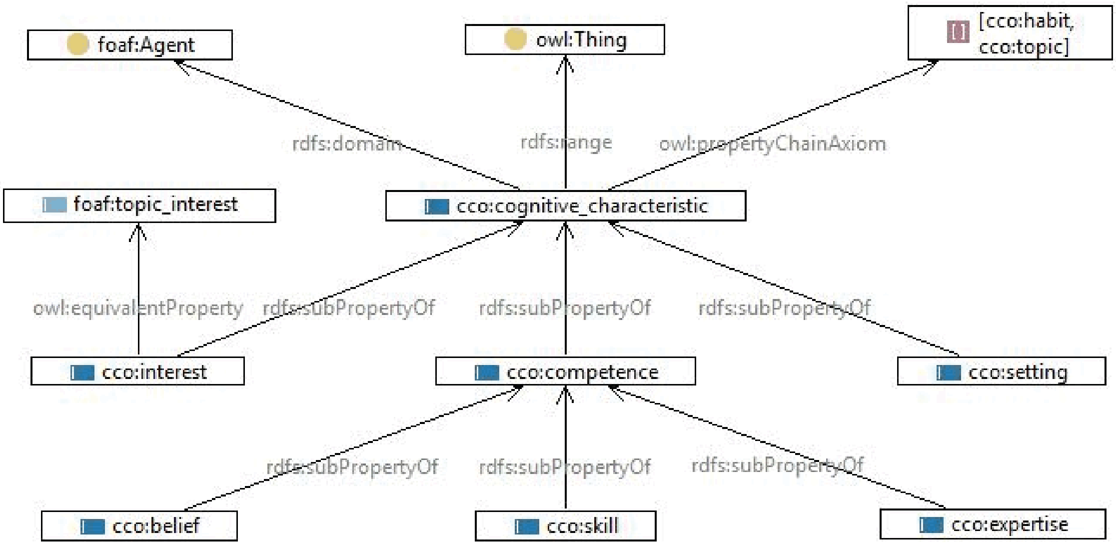

Figure 4.12: The cognitive characteristic *relation* as graph with further relations

CCO includes two opportunities to model cognitive patterns. The first one is a representation of cognitive characteristics by using a simple semantic relation, the cognitive characteristic attribute, to associate topics of cognitive patterns to users. Even better is the application of its more specialised sub properties (see Figure 4.12). The second opportunity is the *property-oriented context reification* of the cognitive characteristic *relation*, the cognitive characteristic concept (cf. Section 2.2.4.3). It is a general multiple purpose cognitive characteristic concept to describe cognitive patterns in more detail for a specific user or a user group.
As one can see in Figure 4.12, the specialised sub properties of the cognitive characteristic *relation*, the cognitive patterns, are currently

- an interest *relation* (equivalent to the interest relation of the FOAF), to associate a certain area of interest or a preference,
- a competence *relation*, to associate a competence to (be able to) do or know something or
- a setting *relation*, to associate a setting, often regarding a specific environment, e.g., an application.

One can refine the semantic relation of a competence association by using a sub property of the competence *relation*. Currently, these are:

[17]This ontology alignment process is strongly influenced by the outcome of the User (weighted) Interests Ontology working group from Hypios VoCamp Paris 2010, see [PS10].
[18]The ontologies whose terms are revised and united in WI.
[19]`http://lists.foaf-project.org/pipermail/foaf-dev/`

- a skill *relation*, to associate the ability or skill to (being able to) do something, e.g., to walk, to play the piano, or to work in a team
- an expertise attribute, to associate knowledge or expertise in a certain domain or a specific topic, e.g., football, programming languages, or music
- a belief property, which is an uncertain relation for a competence representation, i.e., persuasions or opinions which can also be misconceptions

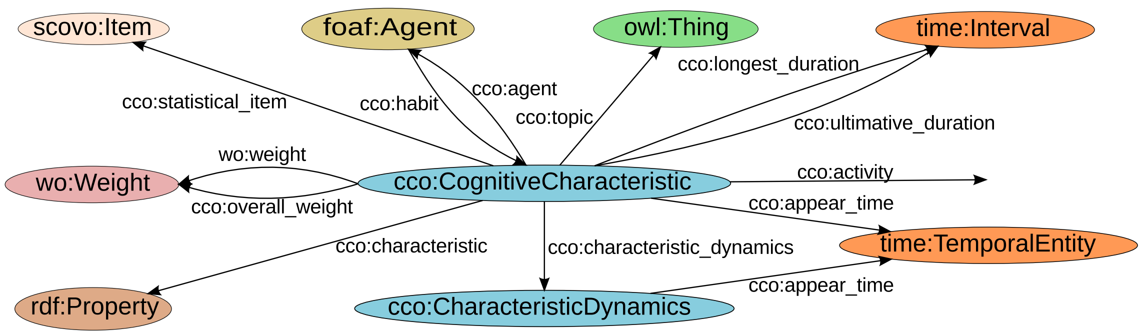

Figure 4.13: The cognitive characteristic concept as graph with relations

In Figure 4.13 ne can see the second opportunity to model cognitive patterns, the cognitive characteristic concept. It is a specialisation of the item concept of SCOVO. This concept can be used to associate any `foaf:Agent` instance with (a) cognitive characteristic instance(s). Therefore, one has to utilise the habit or agent *relation*. The topics of a cognitive characteristic are address by using the topic attribute. A property chain of the habit and topic *relation* directs to topics of a cognitive pattern of a user or a user group, too.
Because of that, a statement which is modelled with the simple semantic relation approach based on the cognitive characteristic property can also be represented by a cognitive characteristic *particular*. Such an instance has an agent or habit, a topic, and a characteristic attribute relation at least. The last property in this list is used to place the applied cognitive pattern relation (specialised cognitive characteristic).
Different statistics can be made on cognitive characteristics. These are currently:

- an overall weight *relation*, to reflect the overall interest in a topic and which should differ from the actual weight (associated by the weight *relation* of WO; see Section 4.1.4.1) of a cognitive characteristic
- a longest duration property, to associate the longest continuous interval of attention for a cognitive pattern
- an ultimate duration attribute, to relate the overall duration of attention for a cognitive pattern

Besides these statistics, one can also place

- a concrete activity, to differentiate topics of cognitive patterns, e.g., between football playing (topic = football; activity = playing) and football watching (topic = football; activity = watching), and
- further statistical items to a cognitive pattern description.

It is important, to be able to describe dynamics of a cognitive characteristic. In CCO they can be modelled with the help of the characteristic dynamics concept. It is based on the weight concept of WO. Furthermore, instances of this concept can be related to a cognitive characteristic *particular* by using the characteristics dynamics *relation*. Thereby, one can associate

- concrete times of appearance (time instants or intervals), i.e., when a cognitive pattern gets attention by someone, or
- a description of evidence, i.e., where this characteristic or dynamics was derived from,

to a cognitive characteristic or characteristic dynamics *particular*.
The example in Listings 4.9, 4.10 and 4.11 shows a part of a user profile as it can be taken from a user account of the PMKB. It is inspired by an example that was originally modelled with the help of UMIRL (see [CV00]). It is remodelled with more expressiveness by utilising CCO.
The user profile comes from the person John White, who can play the piano and sing despite having no formal education in music. Furthermore, he is interested in the music genre Blues, the music artist Wolfgang Amadeus Mozart, the music group the Beatles, the music song "Yesterday" from the Beatles and some self-defined *idiosyncratic genres* (see Section 2.4.3).
At the beginning, all cognitive patterns of this user are described by simple, specialised cognitive characteristic relations. Thereby, most of the topics of the cognitive characteristic attributes are information resources from the information service *DBPedia* and one topic is an information resource from *DBTune*. Afterwards, some of these *shortcut relations* (see Subsection 4.1.4.3) are described in more detail as cognitive characteristic *particulars*.
Each of these cognitive characteristic descriptions is related to a weighting with a different weight value and the same scale, which is represented in Listing 4.11. Moreover, the *particulars* are associated with a topic of a *shortcut relation* and a property of such a relation. Besides this information, some of the cognitive characteristic *particulars* are also related to an activity.

```
@prefix foaf:     <http://xmlns.com/foaf/0.1/> .
@prefix cco:      <http://purl.org/ontology/cco/core#> .
@prefix wo:       <http://purl.org/ontology/wo/core#> .
@prefix ex:       <http://example.org/> .
@prefix dbpedia:  <http://dbpedia.org/resource/> .
@prefix dbtune:   <http://dbtune.org/musicbrainz/resource/signal/> .
@prefix opencyc:  <http://sw.opencyc.org/concept/> .

ex:APerson
   a foaf:Person ;
   foaf:name "John White" ;
   foaf:gender "male" ;
   cco:skill dbpedia:Piano ;
   cco:skill dbpedia:Vocal ;
   cco:expertise dbpedia:Music ;
   cco:interest <http://dbpedia.org/resource/Category:Blues> ;
   cco:interest dbpedia:Wolfgang_Amadeus_Mozart ;
   cco:interest dbpedia:The_Beatles ;
   cco:interest dbtune:8aefa373-2858-4643-b691-cad4ac7c971a ;
   cco:interest ex:IdiosyncraticGenre1 ;
   cco:interest ex:IdiosyncraticGenre2 ;
   cco:habit [
      a cco:CognitiveCharacteristic ;
      cco:topic dbpedia:Piano ;
      cco:characteristic cco:skill ;
      wo:weight [
         a wo:Weight ;
         wo:weight_value 6.0 ;
         wo:scale ex:AScale
         ] ;
      cco:activity opencyc:Mx4rvVjUJ5wpEbGdrcN5Y29ycA
      ] ;
```

Listing 4.9: A part of a user profile modelled with the help of CCO (to be continued in Listings 4.10 and 4.11; see [BRM+10] for the complete example)

```
@prefix cco:      <http://purl.org/ontology/cco/core#> .
@prefix wo:       <http://purl.org/ontology/wo/core#> .
@prefix days:     <http://ontologi.es/days#> .
@prefix tl:       <http://perl.org/NET/c4dm/timeline.owl#> .
@prefix xsd:      <http://www.w3.org/2001/XMLSchema#> .
@prefix ex:       <http://example.org/> .
@prefix dc:       <http://purl.org/dc/elements/1.1/> .
@prefix dbpedia:  <http://dbpedia.org/resource/> .

   cco:habit [
      a cco:CognitiveCharacteristic ;
      cco:topic dbpedia:Vocal ;
      cco:characteristic cco:skill ;
      wo:weight [
         a wo:Weight ;
         wo:weight_value 7.0 ;
         wo:scale ex:AScale
         ] ;
      cco:activity <http://dbpedia.org/resource/Category:Singing>
      ] ;
   cco:habit [
      a cco:CognitiveCharacteristic ;
      cco:topic dbpedia:Music ;
      cco:characteristic cco:expertise ;
      wo:weight [
         a wo:Weight ;
         wo:weight_value 0.0 ;
         wo:scale ex:AScale
         ] ;
      ] ;
   cco:habit [
      a cco:CognitiveCharacteristic ;
      cco:topic <http://dbpedia.org/resource/Category:Blues> ;
      cco:characteristic cco:interest ;
      wo:weight [
         a wo:Weight ;
         wo:weight_value 9.0 ;
         wo:scale ex:AScale
         ] ;
      ] ;
...
   cco:habit [
      a cco:CognitiveCharacteristic ;
      cco:topic ex:IdiosyncraticGenre2 ;
      cco:characteristic cco:interest ;
      wo:weight [
         a wo:Weight ;
         wo:weight_value 7.0 ;
         wo:scale ex:AScale
         ] ;
      cco:appear_time [
         a days:WeekdayInterval ;
         dc:title "bedtime" ;
         tl:at "23:00:00"^^xsd:time
         ] ;
      ] .
```

Listing 4.10: A part of a user profile modelled with the help of CCO (as sequel of Listing 4.9 and to be continued in Listing 4.11; see [BRM+10] for the complete example)

Copyright TU Dresden, Thomas Gängler

```
@prefix wo:         <http://purl.org/ontology/wo/core#> .
@prefix xsd:        <http://www.w3.org/2001/XMLSchema#> .
@prefix ex:         <http://example.org/> .
@prefix rdf:        <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs:       <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl:        <http://www.w3.org/2002/07/owl#> .
@prefix dc:         <http://purl.org/dc/elements/1.1/> .
@prefix sim:        <http://purl.org/ontology/similarity/> .
@prefix ao:         <http://purl.org/ontology/ao/core#> .

ex:tempo
   a rdf:Property , owl:DatatypeProperty ;
   rdfs:range xsd:string ;
   rdfs:subPropertyOf ao:context .

ex:IdiosyncraticGenre2
   a sim:Association ;
   dc:title "romantic music" ;
   ex:tempo "largo"^^xsd:string .

ex:AScale a wo:Scale ;
   wo:min_weight 0.0 ;
   wo:max_weight 9.0 ;
   wo:step_size 1.0 .
```

Listing 4.11: A part of a user profile modelled with the help of CCO (as sequel of Listings 4.9 and 4.10; see [BRM+10] for the complete example)

Each action is described by an information resources from *OpenCyc*[20] ("playing a musical instrument"[21]) or *DBPedia* ("singing"[22]). Due to the weightings and activity descriptions, we can conclude that John White can sing and play the piano pretty good. In addition, he has a strong interest in the music genre *Blues* and some sympathies for romantic music. However, the described person has no knowledge of musical foundations.
Especially Listing 4.10 includes detailed descriptions of John White's interest in specific music genres. Thereby, 'romantic music' is an *idiosyncratic genre* (see Section 2.3). He prefers to listen to it weekdays at "bedtime" (ca. 11 pm). This *music context* itself is defined as an association statement in Listing 4.11 (see Section 4.1.2.3). It describes that musical pieces should be very slow (*largo*) to be considered by this template.
Due to the two modelling opportunities of cognitive patterns in CCO, there is a need in formal semantics to associate statements with a *shortcut relation* (*binary relation*) and instances of a *reification class* (*n-ary relation*) that semantically belonging together or to infer such knowledge with a reasoning engine. The *Property Reification Vocabulary* that is introduced in the next subsection is going to fulfil these requirements.

4.1.4.3 The Property Reification Vocabulary

As explained in Section 2.2.4.3, there is still a lack in modelling contextual information for semantic graph triples and deploying this KR in order to reuse it in a distributed Linked Data environment (see Section 2.2.4.5). Probably applicable solutions for representing *external context*, e.g., *Named Graphs* or *N-Quads*, are available. However, these methods are not really appropriate for representing refinements of semantically related information (*internal context*) by keeping clear semantics regarding their described semantic graph triples (information resource). Even the existing method for RDF statements, RDF reification (*statement reification*), has not defined clear

[20] `http://opencyc.org`
[21] See `http://sw.opencyc.org/concept/Mx4rvVjUJ5wpEbGdrcN5Y29ycA`
[22] See `http://dbpedia.org/resource/Category:Singing`

semantics between statement triples and their reification information resource(s). This method is intended to be applied at the instance level (cf. [Noy06]).
It might be beneficial to be able to describe a reification on the vocabulary level. In other words, an explanation of a reification from a semantic relation that is established by a property. This kind of reification is called *property reification*. It is quite interesting for ontology mapping and alignment processes on existing Semantic Web ontologies. A simple (concrete) semantic relation, represented by subject, predicate and object, is defined as *shortcut relation* in this context. It is important to note that the predicate is always the same property for a single *shortcut relation* definition. In addition, a class that enables a detailed description of such an *n-ary relation* is called *reification class*.

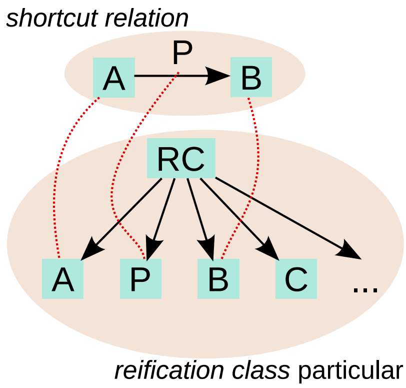

Figure 4.14: A *property reification* example

Figure 4.14 shows an example of a *property reification*. On the one side, there is a *binary relation* that simply links a *particular* `A` to another *particular* `B` by using a property `P`. This semantic relation is a *shortcut relation*. On the other side, there is an *n-ary relation* which is a *particular* of a *reification class* that not only links to the parts of the *shortcut relation*. However, it describes a semantic relation with further attributes. Hence, it is important that a reasoning engine can infer that these ontological KRs semantically belonging together.
I co-designed the *Property Reification Vocabulary* [PCIG10], due to the lack of a published vocabulary that addresses a mapping and its semantics between *shortcut relations* and *reification class particulars* and vice versa. This vocabulary is designed after a proposal made by Jirí Procházka, Richard Cyganiak and Toby Inkster [PCI10]. It was also extended by myself to broaden its functionality. This KR is supposed to reflect the important use case and ontology design pattern of *property reification*. PRV gives ontology designers the freedom to separate *property reification* definitions from a core ontology definition. For that matter, it enables the opportunity to make them optional.
Figure 4.15 illustrates the *property reification* concept and its related properties as extension of RDFS (see Section 2.2.2). It associates

- the *reification class*,
- the property of the *shortcut relation* (by `prv:shortcut`),
- the *shortcut property*, i.e., the property that relates to the property of a *shortcut relation* in a *reification class particular*,
- the *subject property*, i.e., the property that relates to a subject of a *shortcut relation* in a *reification class particular*, and
- the *object property*, i.e., the property that relates to an object of a *shortcut relation* in a *reification class particular*,

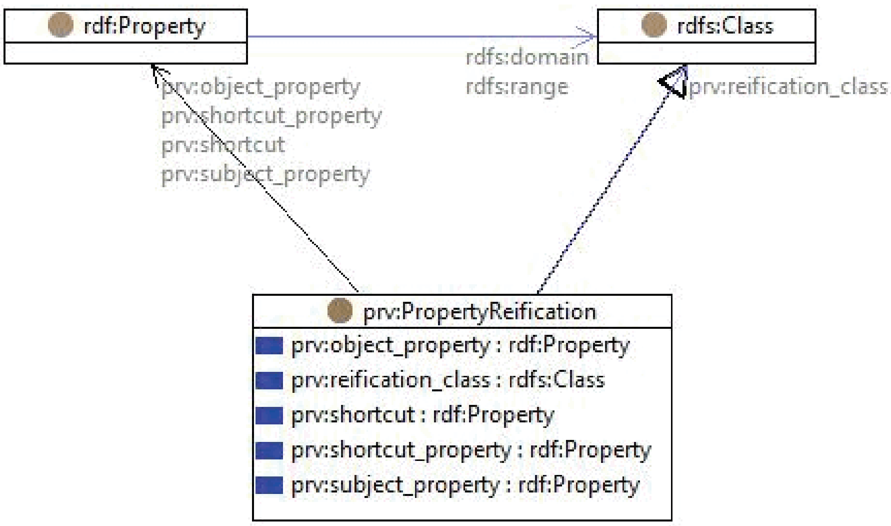

Figure 4.15: The *property reification* concept as graph with relations

of a *property reification* definition. This design provides the opportunity to utilise a *reification class* for several *property reification* definitions. It enables the application of a class for several *shortcut relation* definitions (properties) as well. PRV enables a reasoning engine to apply the implications as described in Listing 4.12 and 4.13 (represented as N3 rules; see Section 2.2.2). These rules are also published as SPIN rules [Gän11d] by following some Linked Data principles (see Section 2.2.4.6 and 2.2.4.5).

```
@prefix prv: <http://purl.org/ontology/prv/core#> .

{
   ?r a ?rc ;
      ?scp ?sc ;
      ?sp ?s ;
      ?op ?o .

   ?pr a prv:PropertyReification ;
      prv:shortcut ?sc ;
      prv:reification_class ?rc ;
      prv:shortcut_property ?scp ;
      prv:subject_property ?sp ;
      prv:object_property ?op .
}
=>
{ ?s ?sc ?o } .
```

Listing 4.12: The *shortcut relation* rule of PRV

The *shortcut relation* rule in Listing 4.12 means that if a *property reification* particular `?pr` exists which includes

- the *shortcut relation* property `?sc` (associated by `prv:shortcut`), and
- a *particular* of a *reification class* `?rc` (associated by the *reification class* relation of `?pr`) exists, where
  - the *shortcut relation* predicate `?sc` is associated by the *shortcut property* `?scp` (assigned by the *shortcut property* relation of `?pr`),

  - the *shortcut relation* subject `?s` is associated by the *subject property* `?sp` (assigned by the *subject property* relation of `?pr`) and
  - the *shortcut relation* object `?o` is associated by the *object property* `?op` (assigned by the *object property* relation of `?pr`),

then a *shortcut relation* of a *reification class particular* can be constructed, which has

- as its subject the *particular* `?s`,
- as its predicate the property `?sc` and
- as its object the *particular* `?o`.

```
@prefix prv: <http://purl.org/ontology/prv/core#> .

{
   ?s ?sc ?o .

   ?pr a prv:PropertyReification ;
      prv:shortcut ?sc ;
      prv:reification_class ?rc ;
      prv:shortcut_property ?scp ;
      prv:subject_property ?sp ;
      prv:object_property ?op .
}
=>
{
   ?r a ?rc ;
      ?scp ?sc ;
      ?sp ?s ;
      ?op ?o .
} .
```

Listing 4.13: The *property reification* rule of PRV

The *property reification* rule in Listing 4.13 means that if a *shortcut relation* exists which has

- as its subject the *particular* `?s`,
- as its predicate the property `?sc` and
- as its object the *particular* `?o`,

and a *property reification particular* `?pr` exists which relates

- the *shortcut relation* property `?sc` (associated by `prv:shortcut`),
- the *reification class* `?rc` (associated by the *reification class* relation)
- the *shortcut property* `?scp` (associated by the *shortcut property* relation),
- the *subject property* `?sp` (associated by the *subject property* relation) and
- the *object property* `?op` (associated by the *object property* relation),

then a *particular* `?r` of a *reification class* `?rc` can be constructed, which relates

- the *shortcut relation* prediate `?sc` (associated by the *shortcut property* `?scp`),
- the *shortcut relation* subject `?s` (associated by the *subject property* `?sp`) and
- the *shortcut relation* object `?o` (associated by the *object property* `?op`).

With the help of the *property reification* and *shortcut relation* rule, a reasoning engine is supposed to not only be able to infer (construct) related ontological KRs, but to also reason that existing *shortcut relations* and *particulars* of *reification classes* semantically belonging together.

```
@prefix prv: <http://purl.org/ontology/prv/core#> .
@prefix cco: <http://purl.org/ontology/cco/core#> .
@prefix ex:  <http://example.org/> .

ex:SkillReification a prv:PropertyReification ;
   prv:shortcut cco:skill ;
   prv:shortcut_property cco:characteristic ;
   prv:reification_class cco:CognitiveCharacteristic ;
   prv:subject_property cco:agent ;
   prv:object_property cco:topic .

ex:ExpertiseReification a prv:PropertyReification ;
   prv:shortcut cco:expertise ;
   prv:shortcut_property cco:characteristic ;
   prv:reification_class cco:CognitiveCharacteristic ;
   prv:subject_property cco:agent ;
   prv:object_property cco:topic .

ex:InterestReification a prv:PropertyReification ;
   prv:shortcut cco:interest ;
   prv:shortcut_property cco:characteristic ;
   prv:reification_class cco:CognitiveCharacteristic ;
   prv:subject_property cco:agent ;
   prv:object_property cco:topic .
```

Listing 4.14: A part of a *property reification* example of CCO modelled with the help of PRV (see [PCIG10] for the complete example)

One can see *property reification* definitions that utilise terms of CCO (see Subsection 4.1.4.2) in Listing 4.14. The example includes specifications for the *shortcut relation* properties: `cco:skill`, `cco:expertise`, and `cco:interest`. Thereby, every *property reification* definition utilises the same *reification class*, the cognitive characteristics concept. Furthermore, the *subject* and *object property* are in every definition the same - the agent relation (as *subject property*) and the topic relation (as *object property*).

Due to the *property reification* definitions from Listing 4.14 and the *shortcut relation* and *property reification* rules (see Listings 4.12 and 4.13), one can clearly reason that *shortcut relations* and *reification class particulars* in Listing 4.15 are semantically belonging together (here pairwise). This user profile example with different cognitive characteristics of the same topic is created with the help of CCO. It represents the person John Wayne, who has tree different cognitive patterns - a skill, an expertise and an interest - with the same topic, soccer. At the beginning, each semantic relation is modelled as *shortcut relation*. Afterwards, follow more detailed descriptions of these cognitive characteristics. They are associated to John Wayne by using the habit relation. Besides the basic information of the *shortcut relations*, each of these cognitive characteristic *particulars* includes a weight description (see Subsection 4.1.4.1). In addition, two of them are related to an activity.

As a result, a reasoning engine is not only able to automatically infer knowledge from the *shortcut relations*, but rather from *reification class particulars* as well. Because of that, one can conclude that John Wayne

- can play[23] soccer quite moderate,
- has a quite good knowledge about the topic soccer and
- and is moderately interested in watching[24] soccer.

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix cco:  <http://purl.org/ontology/cco/core#> .
@prefix wo:   <http://purl.org/ontology/wo/core#> .
@prefix ex:   <http://example.org/> .

ex:APerson
   a foaf:Person ;
   foaf:name "John Wayne" ;
   cco:skill <http://dbpedia.org/resource/Football_(soccer)> ;
   cco:expertise <http://dbpedia.org/resource/Football_(soccer)> ;
   cco:interest <http://dbpedia.org/resource/Football_(soccer)> ;
   cco:habit [
      a cco:CognitiveCharacteristic ;
      cco:topic <http://dbpedia.org/resource/Football_(soccer)> ;
      cco:characteristic cco:skill ;
      wo:weight [
         a wo:Weight ;
         wo:weight_value 6.0 ;
         wo:scale ex:AScale
         ] ;
      cco:activity <http://sw.opencyc.org/concept/Mx4rwJRiEpwpEbGdrcN5Y29ycA>
      ] ;
   cco:habit [
      a cco:CognitiveCharacteristic ;
      cco:topic <http://dbpedia.org/resource/Football_(soccer)> ;
      cco:characteristic cco:expertise ;
      wo:weight [
         a wo:Weight ;
         wo:weight_value 7.0 ;
         wo:scale ex:AScale
         ] ;
      ] ;
   cco:habit [
      a cco:CognitiveCharacteristic ;
      cco:topic <http://dbpedia.org/resource/Football_(soccer)> ;
      cco:characteristic cco:interest ;
      wo:weight [
         a wo:Weight ;
         wo:weight_value 5.0 ;
         wo:scale ex:AScale
         ] ;
      cco:activity <http://sw.opencyc.org/concept/Mx4rwOOJ55wpEbGdrcN5Y29ycA>
      ] .
```

Listing 4.15: A part of a user profile example with different cognitive characteristics of the same topic created with the help of CCO (see [BRM+10] for the complete example)

Such a modelling has the advantage that one can talk about one and the same topic, regardless of the activities that are related to cognitive pattern descriptions of a person. This is possible because 'activity', 'cognitive characteristic' and 'topic' have their own separate dimension. Moreover, with the help of PRV one can semantically relate multiple detailed *reification class particulars* to one *shortcut relation*, e.g., cognitive patterns from one and the same user which have the same cognitive characteristic and topic, but different activities.

[23] http://sw.opencyc.org/concept/Mx4rwJRiEpwpEbGdrcN5Y29ycA
[24] http://sw.opencyc.org/concept/Mx4rwOOJ55wpEbGdrcN5Y29ycA

Copyright TU Dresden, Thomas Gängler

4.1.5 The Media Types Taxonomy

Since, the *Music Ontology* (see Section 2.3.2) follows the four levels of abstraction of FRBR[25], it enables the description of musical or music-related *entities* on each of them. For that matter, one can represent a single copy of, e.g., a CD as well as a characterisation that bears all similarities of all existing exemplars of this realisation. The former description is done on the item level. It can make use of existing concepts of MO and apply terms of its separated *Media Formats Ontology* [Jac10d]. The latter characterisation is done on the manifestation level. This requires a different modelling to relate, for example, a media type to a *particular* of this level of abstraction. Such KRs were previously not possible.
That is why, I co-designed the *Media Types* taxonomy (MT) [GJ10a]. It can be used to assign[26] appropriate media types to a music manifestation, e.g., a *mo:Record* or `mo:Track` *particular*. MT is, at least, well aligned to similar media type taxonomies of the music metadata format *ID3* (see Section 2.3.2) and those of the MMSs *MusicBrainz* and *Discogs* (see Section 2.3.3).
This taxonomy utilises SKOS (see Section 2.2.4.1) for its specification, i.e., every media type is modelled as a SKOS concept and related media types are associated by applying SKOS' broader or narrower *relation* as needed. Besides, preferred and alternative labels are assigned to single media type definitions by using appropriate SKOS *relations*. The media types are classified as generic media type, physical medium or file format with the help of concepts of DC Terms (see [DCM08]). If possible, a media type definition is related to an appropriate information resource of *DBPedia* to provide a detailed description of a format.

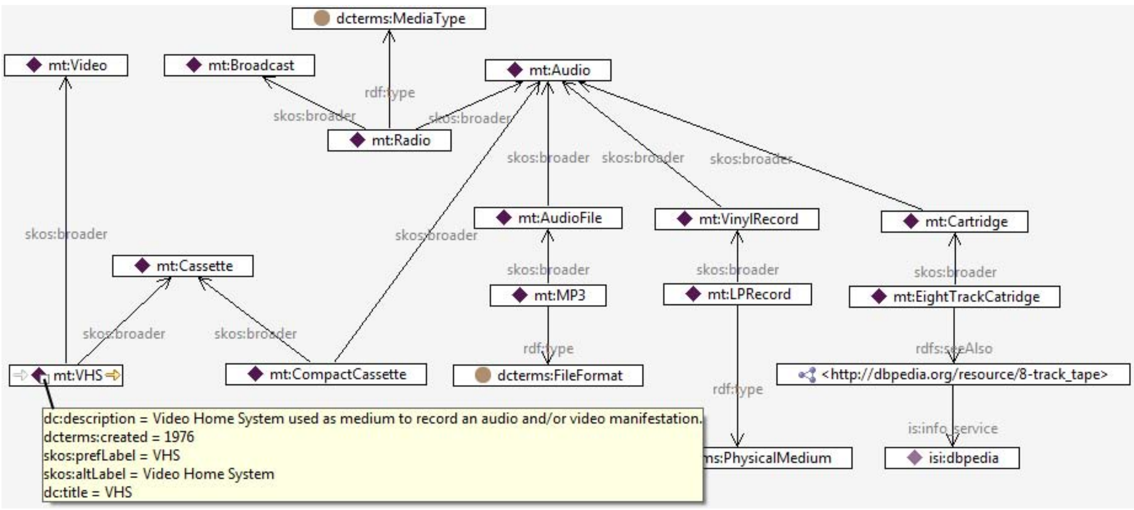

Figure 4.16: An excerpt of the *Media Types* taxonomy

Figure 4.16 illustrates an excerpt of MT. It includes broad categories, such as audio or video, and narrow ones, e.g., compact cassette or vinyl record. A media type can have multiple broader *relations*, for example, if there are orthogonal aspects that can be classified. For instance, the media type 'radio' has two superior concepts - 'broadcast' and 'audio'. Furthermore, one can see exemplarily assigned media type categories, e.g., MP3 is a file format. The 8-track cartridge definition showcases the relation to an appropriate *DBPedia* information resource[27]. An example of a minimal media type description is given for the video cassette format VHS. It contains preferred (`skos:prefLabel`) and and alternative labels (`skos:altLabel`).
The *Media Types* taxonomy can also be utilised by KRMs of other domains, e.g., the movie domain. Its media type coverage is far from complete. However, a relatively useful set of format descriptions is defined and related to each other. Currently, MT consist of 54 media type specifications.

[25]The four levels of abstraction of FRBR are: work, expression, manifestation, and item (see [DN09]).
[26]The media type *relation* is part of the most recent version of MO (see [RG10d]).
[27]This resource (`http://dbpedia.org/resource/8-track_tape`) is marked as *DBPedia* resource via the information service *relation* (see Section 4.1.1) that references a description of *DBPedia* (see [Gän10b]).

4.1.6 Summary

This section introduces various Semantic Web vocabularies that were implemented as part of the PMKB KRM. These ontologies close the gap of missing or inappropriate designed concepts and relations. They are necessary to properly model different aspects of the PMKB concept.
The first KR vocabulary of this set, the *Info Service Ontology* (see Subsection 4.1.1), is to be used in information service selection (see Section 3.3.3) and information exploration tasks (see Section 3.3.5). It enable the characterisation of any information service.
Further *simple specialised ontologies* are presented in the context of the *Play Back Ontology* (see Section 4.1.2). This modularisation boosts the reutilisation of the single small vocabularies. The *Ordered List Ontology* consists of an implementation of a basic ontology design pattern for describing sequences (see Subsection 4.1.2.1). Besides, the *Counter Ontology* specifies a universal counter concept and a specialised event *universal* to enable the description of single count actions (see Subsection 4.1.2.2). The third Semantic Web vocabulary of this series, the *Association Ontology*, is intended for association and context modelling (see Subsection 4.1.2.3). All these *simple specialised ontologies* are reutilised or applied in the modelling of PBO and its *particulars* (see Subsection 4.1.2.4). This vocabulary defines concepts for playlist and *media action counters* that are related to the *play back domain*. PBO and related *simple specialised ontologies* are part of the context modelling for implementing personalisation (3.3.6).
The *Recommendation Ontology* is another new Semantic Web ontology that delivers a high-level concept for describing recommendations (see Subsection 4.1.3). For representing ranked recommendations it makes use of the ordered list concept of OLO.
Morever, the *Cognitive Characteristics Ontology* (see Subsection 4.1.4.2) is the last main KR vocabulary that was designed to meet the requirements of a proper user modelling as a part of the personalisation implementation. Thereby, a further *simple specialised ontology*, the *Weighting Ontology* (see Subsection 4.1.4.1), and a basic Semantic Web KRL, the *Property Reification Vocabulary* (see Subsection 4.1.4.3), are introduced. The former ontology simply consists of a weight concept whose *particulars* can be related to a scale that represents the range of a certain weighting model. The latter vocabulary establishes semantic relations between *binary relations* and related *n-ary relations* (see section 2.2.4.3). Both are reutilised in term definitions or applied in modellings related with CCO. This ontology can represent cognitive pattern (see Section 2.4) on different levels of expressiveness. On the one side, as simple *binary relation*. On the other, as detailed *n-ary relation*.
Finally, the *Media Types* taxonomy (see Subsection 4.1.5) defines a set of useful media type specifications that are related to each other by using properties of SKOS. This classification can be used to assign media types to KRs on the manifestation level.
All these Semantic Web ontologies are part of the PMKB KRM. It is managed by its KMS whose applied protocols and technologies are drafted in the following section.

4.2 KNOWLEDGE MANAGEMENT SYSTEM

During the period of my research, the focus of this work strongly shifted to knowledge representation. For that reason, this section is dealt with in brief. Main ideas and design decision are only sketched. The KMS part of the PMKB vision has to be described more detailed in further works.
As is explained in the previous section, Semantic Web KRLs and ontologies are chosen to lay the foundation of the PMKB KRM. For this reason, *triples stores* are an ideal knowledge base category to handle RDF graphs (see Section 2.2.4.4). They often provide inbuilt reasoning capacities and implement the SPARQL specification (see Section 2.2.4.6). The latter feature can be utilized for processing the four primary KR tasks (see Section 3.1.2).
In addition to SPARQL, information resources of the PMKB KRM have to be published by following the principles of Linked Data (see Section 2.2.4.5). They align quite well to the REST architectural style, which is a requirement of the architecture of the PMKB KMS (see Section 3.2). In other words, the design of this KMS is to be guided by REST principles. Thereby, the

HTTP protocol (see Section 2.2.1.1) implements the 'universal action semantics' constraint of REST. HTTP's URI scheme allows a uniform identification of resources. Furthermore, via its headers, HTTP partly supports the 'self-descriptive messages' principle of REST. This requirement is met by additionally utilising Semantic Web KRLs and ontologies for resource description. Therefore, the GMKB can serve its information resources in as many different serialisation formats as possible to reach a broad audience of information consumers.
The PMKB KRM is powered by the WebID protocol and WAC (see Section 2.2.4.6) to support an appropriate authentication and authorisation mechanism. This secures the individual GKRM of PMKB users and allows flexible access control settings.
Moreover, the mapping of non-RDF KR formats into Semantic Web KRs is conducted with the help of various *RDFizers* (see Section 2.2.4.2). An alignment of Semantic Web vocabularies that are not part of the PMKB KRM to ontologies that are included, has to be realised via a rule language. N3Logic (see Section 2.2.2) and SPIN (see Section 2.2.4.6) are the preferred ones in this implementation.
Information resources that are retrieved from external information providers during information federations tasks is to be assigned with provenance data. At a minimum, the information service *relation* of IS (see Section 4.1.1) has to be used for this *external context* information. Nevertheless, detailed provenance models can be applied in conjunction with this basic evidence information (see Section 2.2.4.3).
Change propagation and synchronisation (see Section 3.3.4) is implemented with the help of multiple protocols and formats that are introduced in Section 2.2.4.6.

1. Atom and the AtomPub protocol provide the basis for communicating changes of parts of the PMKB KRM to interested information consumers.
2. The PSHB protocol is used to enable an efficient server-to-server communication for update notification.
3. The SUP protocol is applied on the "last mile", i.e., for synchronisation between the GMKB and LMKBs.
4. The DSNotify Eventset Vocabulary is utilised for the orchestration of these KM tasks, i.e., it supports the integrity maintenance proccess.

Stereotype profiles and categorisations (see Section 3.3.1) are modelled by means of of OWL (see Section 2.2.2), SKOS (see Section 2.2.4.1) and SPIN. The latter one is also used to populate attributes of stereotypes in user profiles, e.g., during a user account initialisation process.
All in all, these are the main design decisions that are made regarding protocol and technology usage in the PMKB KMS. They are intended to be applied in a concrete implementation which makes use of as many existing software solutions and frameworks as possible.

4.3 SUMMARY

This chapter illustrates on a very high, platform-specific level a refinement of the PMKB concept. Its strong focus on knowledge representation is reflected in the rather detailed knowledge representation section (see Section 4.1). This part is shortly summarised in Subsection 4.1.6. Since, this work concentrates on the KR part of the PMKB, design decisions regarding PMKB KMS are only sketched in Section 4.2.
Nevertheless, this chapter should have been given an good overview about a realisation of the PMKB vision on this implementation level. The next section outlines the application and evaluation of the ontologies of the PMKB information space and explains some concrete implementations.

5 PERSONAL MUSIC KNOWLEDGE BASE IN PRACTICE

At the beginning of my research, it was intended to develop a comprehensive prototype that implements principal aspects of the PMKB vision. This application should make use of ontologies and vocabularies of the PMKB information space (see Section 4.1) and further, already existing and applicable ones that are needed for the modellings of the PMKB KRM. Thereby, the most important representative of the latter category is the *Music Ontology* (see Section 2.3.2).
The PMKB prototype, at least, ought to be able to communicate with the MMSs: *MusicBrainz* (see Section 2.3.3.1), *Last.fm* (see Section 2.3.3.2) and the *Echo Nest* (see Section 2.3.3.3). Moreover, the foundation of the PMKB KMS should be provided by utilising the *Virtuoso* universal server (see Section 2.2.4.4). This solution consists of many useful built-in capabilities, such as the *RDFizer* framework *Virtuoso Sponger* (see Section 2.2.4.2). It probably implements the most state-of-the-art KM technologies, e.g., Linked Data, WebID, or PSHB deployment support (see Section 2.2.4.5 and 2.2.4.6).
Finally, it was planned to develop a *Songbird* extension that can communicate with a LMKB. This add-on ought to showcase the capacities of the PMKB prototype in a user application for music consumption.
Unfortunately, a complete implementation of the sketched proof-of-concept PMKB could not be realised within the scope of this work. This has to be developed as part of further activities. The following sections explain the application of the PMKB ontologies in several components of the PMKB prototype (see Section 5.1) and the evaluation of those vocabularies via the Semantic Web community (see Section 5.2).

5.1 APPLICATION

So far, two *RDFizer* were already realised as part of the *PMKB* prototype. They utilise rather different technologies and are shortly introduced in the following paragraphs.

5.1.1 AudioScrobbler RDF Service

The first *RDFizer* is an enhanced version of the *AudioScrobbler* RDF Service [Rai06, RG10a]. It executes transformations of information resources that are delivered by the *Last.fm* Web Service into Semantic Web KRs (see Section 2.3.3.2). These semantic graphs can contain the following descriptions of a *Last.fm*-user's profile:

- the last ten *scrobble events* (see Section 4.1.2.2)
- one's friends
- events a user has planned to attend (or personal event recommendations[1])

This service is implemented in SWI Prolog [2]. Thereby, the following existing Semantic Web ontologies are mainly used. The *Music Ontology* is utilised for music artist, album and track representations. User modelling is done with the help of FOAF (see Section 2.4.1). Furthermore, event descriptions are represented by applying terms of EVENT. The *AudioScrobbler* RDF Service is already deployed in its initial version on *DBTune* (see [Rai08a]).
I completely redesigned the *AudioScrobbler* RDF Service in version 2.0 [RG10b]. This was necessary to make it compatible with the latest version of the *Last.fm* Web Service. Since version 2.5 of this *RDFizer* [RG10c], a mapping to appropriate ontologies of the PMKB information space is integrated.
Scrobble events are represented with the help of CO (*scrobble event* concept; see Section 4.1.2.2) and PBO (media *scrobble* object *relation*; see Section 4.1.2.4). Original information resources from *Last.fm*, e.g. user profiles or music artist pages, are assigned with minimal provenance information by utilising the information service *relation* of IS (see Section 4.1.1). Furthermore, REC provides terms for modelling personal event recommendations, e.g., the recommender or recommendation *relation* (see Section 4.1.3).

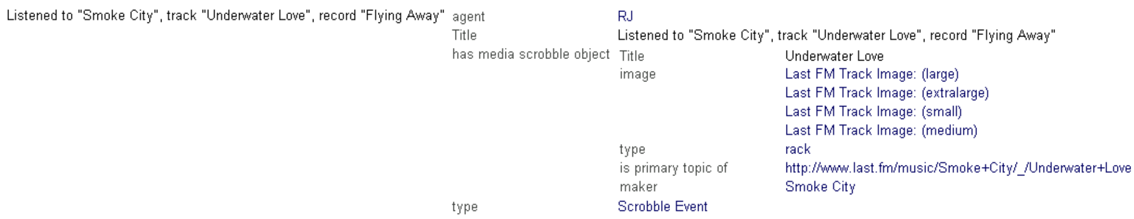

Figure 5.1: An *RDFized Last.fm scrobble event*

Figure 5.1 shows an excerpt of an *RDFization* of parts of RJ's user model on *Last.fm*[3]. It is illustrated in the Semantic Web browser *Tabulator* [sev08]. This example contains a *scrobble event* of the song "Underwater Love" from Smoke City that occurs on the album "Flying Away".

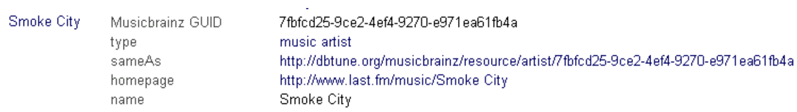

Figure 5.2: An enriched, *RDFized* music artist description from *Last.fm*

Smoke City is shortly described in Figure 5.2. This is another excerpt of the *Tabulator* visualisation of the transformed *Last.fm* information resources. Besides a reference to the *Last.fm* artist page of Smoke City, this KR is related to an equal *entity* of the *MusicBrainz D2R* Server on *DBTune*[4] (see 2.3.3.1). The referred Linked Data information resource of Smoke City provides additional knowledge that is originally served by *MusicBrainz*. Such an alignment easily enables information integration and knowledge discovery.
It was intended to integrate the *AudioScrobbler* RDF Service independently as an external information provider that communicates with the GMKB CE via HTTP, or as a cartridge of *Virtuoso Sponger* (see Section 2.2.4.2). However, the latter option would require an implementation of a Prolog binding to the Virtuoso Service Extension Interface that is written in C (see [Ope09]).

[1] Therefore, the user has to authorise the *AudioScrobbler RDF Service*.
[2] `http://www.swi-prolog.org`
[3] `http://last.fm/user/RJ`
[4] `http://dbtune.org/musicbrainz/resource/artist/7fbfcd25-9ce2-4ef4-9270-e971ea61fb4a`

5.1.2 PMKB ID3 Tag Extractor

The second *RDFizer* of the PMKB prototype is the *PMKB ID3 Tag Extractor* (PITE) [ADG10], which is programmed in Java. It is a massively rewritten adaptation of the *MP3 File Extractor* component [AD08] of the *Aperture* framework (see Section 2.2.4.2). The existing extractor makes use of various vocabularies of the *NEPOMUK* ontology framework [CO09]. It especially utilises the *NEPOMUK ID3 Ontology* [MSSvE07] for the mapping of ID3 tags (see Section 2.3.2) into Semantic Web KRs. The design of this ontology is tightly aligned to the MMF of the MP3 sources that can be processed by the *MP3 File Extractor*. It barely makes use of enhanced modelling capacities that go beyond the expressiveness of ID3 tags. Besides, the existing ID3 tag *RDFizer* only implements mappings of a few frames (description units) of the latest version of this MMF. For that reason, I redesigned the *MP3 File Extractor* as PITE to apply advanced Semantic Web KR capabilities. Thus, I integrated several ontologies of the Music Ontology framework (see Section 2.3.2) and the PMKB information space (see Section 4.1). The mapping mainly utilises terms of the *Music Ontology*. In addition, it applies, amongst others, PBO and CO for play back counter modelling (see Section 4.1.2.4), or AO for expressing personal associations regarding a song (see Section 4.1.2.3). The *Media Types* taxonomy (see Section 4.1.5) delivers appropriate media type URIs as a replacement of the ID3-specific strings for media format classification. Finally, basic provenance modelling is realised with the help of the information service *relation* of IS again (see Section 4.1.1).

```java
public static void createPlayCounter(Model model, int playCount,
    String playCounterTitle, Resource mediaObject)
{
    Resource playCounter = ModelUtil.generateRandomResource(model);
    model.addStatement(playCounter, RDF.type, PBO.PlayBackCounter);
    addIntegerLiteral(model, playCounter, CO.count, playCount);
    addStringLiteral(model, playCounter, DC.title, playCounterTitle);
    model.addStatement(playCounter, PBO.media_object, mediaObject);
}
```

Listing 5.1: A code fragment of PITE that creates a play back counter description

Listing 5.1 illustrates a code fragment of PITE that contains the creations of a play back counter description. Firstly, it associates a type to the play back counter resource. Secondly, the numerical value is added. A human-readable title is assigned as well. Finally, the play back counter *particular* is related to its *media object*, which is a music track.

```java
MOOD(FrameBodyTXXX.MOOD, true)
{
    public void process(AbstractTagFrameBody body, RDFContainer result,
    HashMap<String, Resource> resourceMap)
    {
        FrameBodyTXXX txxx = (FrameBodyTXXX) body;
        Model model = result.getModel();

        // these are personal mood associations and added to the own exemplar
        ID3Util.addStringLiteral(model, result.getDescribedUri(), AO.mood,
        txxx.getFirstTextValue());
    }
},
```

Listing 5.2: A code fragment of PITE that creates personal mood association

A personal mood association is extracted from a specific ID3 frame in the code example of Listing 5.2. This mood description is related to one's own copy of a music track and not to a more abstract KR, such as a master signal of a song. Such a general mood association can be assigned separately and will be attached to the master signal description instead of a music item

Copyright TU Dresden, Thomas Gängler

description. Besides, the object of this mood relation is currently modelled as a simple string. However, one can create a (personal) mood taxonomy and provide more descriptive information resources which can be dereferenced by defining a HTTP URI for each of them[5]. This enables a mapping of mood strings of an ID3 tag to URIs of a mood taxonomy.
It was intended to integrate PITE with the help of *Virtuoso Sponger* (see Section 2.2.4.2) in the LMKB KMS. Therefore, *OpenLink* provides an *Aperture* cartridge which integrates all extractors that are part of the associated *Aperture* deployment (see [sev09e]).

5.2 EVALUATION

In general, the evaluation of KR vocabularies is a rather difficult task. One might tend to say that this is not really feasible, because something like a "golden standard" does not and cannot exist at all. Yves Raimond analysed several ontology evaluation methods as part of his dissertation [Rai08a]. The main outcome of this examination was that a combination of a task-based and a data-driven methodology is a good choice for driving an evaluation that approximately meets certain "real world" requirements.
In his work, Raimond outlined a query-driven ontology evaluation approach which is similar to the query-driven methodology that was described by Baumann et al. in [BKN02]. Raimond's method utilises a set of verbalised user requirements of the application domain that is being assessed. Several queries that reflect needs of different user groups have to be derived from the given information. A given KR framework has to provide features to satisfy these queries. The level of expressiveness is measured on the results of this ontology fitting task. Since an application of this method requires a comprehensive dataset of verbalised needs of users, an evaluation that follows the query-driven ontology methodology cannot be realised within the scope of this work. However, a comparatively easier evaluation approach that reflects "real world" requirements, is an examination of applications or peer reviews of the developed ontologies. One can count this approach to the category of social evaluation methods. This methodology is carried out on the PMKB ontology information space. Since ontologies live on the Web, I tried to track as many reutilisations, applications, interests in utilisation, reviews and mentions, and indexations of the outlined ontologies in Section 4.1 as possible outside of the PMKB project to illustrate their usefulness. All information is primarily gathered from the results of several search requests on *Google*[6]. Every time, the full name of an ontology, enveloped in quotation marks, was utilised as search term for a query regarding that vocabulary. This examination is by far incomplete, since sometimes references are made quite undisclosed.

5.2.1 Reutilisation

Reutilisation in this context means that terms of an ontology are utilised in another vocabulary definition. This helps to create a "shared understanding" more easily, i.e., a reasoning engine does not require separate ontology mappings to be able to conclude that a term description is semantically related to another one. It can directly infer relations via a specification of an ontology (see Section 2.2.4.1).
The first reutilisation of OLO occurs in the *Stories Ontology* [JRH10]. This vocabulary refines the ordered list concept of OLO to model "the order of 'telling' the story" [JRH10], which can be different from the temporal sequence of the single events that make this story. Thereby, the ordered list concept of OLO is overlayed with the event concept of EVENT, i.e., lists of events can be defined. The *Stories Ontology* is successfully applied in the *BBC Stories* project [Har10].
In addition, the developers of the *Sport Ontology* [RWO11] showed some interest in reutilising the ordered list concept of OLO to represent sequences of competitions that can consist of further sub competitions, sessions or rounds (see [Oli11]).

[5] Of course, the information resource has to be deployed by following some Linked Data principles, that a resource request will deliver some processable and interpretable KRs as response (see Section 2.2.4.5 and 2.2.3).
[6] `http://google.com`

5.2.2 Application

Application, in this context, means that terms of an ontology are included in a KRM of a software solution. The benefits are identical to those of the reutilisation use case, i.e., the deployment of common vocabularies provides a good foundation for information federation tasks (see Section 2.1.2.2).
On the basis of the query results evaluation that I carried out, the *Recommendation Ontology* is applied in the *OpenRecommender* project[7]. This framework, at least, powers the *BCMoney MobileTV*[8] recommender (see [Bcm11b]). However, it is designed to be an open-source, universal recommender engine (see [Bcm11a]). Both projects are still at a very early development state.
Besides, the developers of *DBrec* / *Seevl* intend to integrate REC into the vocabulary part of the KRMs of these systems (see [Pas10a, Pas10b]). The coordinators of *Libre.fm* also showed some interest[9] in integrating appropriate ontologies of those vocabularies that are illustrated in Section 4.1. I proposed the application of terms of CO, PBO and REC in the *Libre.fm* KRM (see [Gän10c]).

5.2.3 Reviews and Mentions

Several people have already peer-reviewed or have mentioned the ontologies that are outlined in the knowledge representation section of chapter 4. At first, Ben Fields and Paul Lamere recommend in their music playlist tutorial [FL10] the playlist concept of PBO for modelling music playlists. Furthermore, Fields emphasises the need for a common model for playlist representation in his dissertation [Fie11]. He suggests the playlist concept of PBO as an appropriate proposal for such a model.
The ongoing developments of WI, whose present successor is CCO, were endorsed in a presentation [ZWDH10] that was given by an *e-foaf:interest Vocabulary* developer. In addtion, Nima Dokoohaki, a developer of the *SMARTMUSEUM* project[10] that utilised GUMO for user modelling, wrote an insightful review [DG10] regarding the design of WI and honoured its development. At the point of time of this feedback, this ontology specification was at a development state shortly before the shift of WI to CCO.
The evaluated vocabularies got some mentions and approval by several reputable people and institutions from the Semantic Web community. For instance, Kingsley Idehen[11] recommended these ontologies several times (see [Ide11a] for a general, [Ide11b] for a CCO, and [Ide11c] for a REC recommendation). In addition, DCMI continuously label different vocabularies that are also part of the PMKB information space as useful, e.g., REC (see [Dub10a]), IS and PRV (see [Met10]), AO (see [Dub10b]), and PBO (see [Dub11b]). Danny Ayers (see [Aye11]) and Bernard Vatant (see [Vat11]) positively mentioned the whole vocabulary set, that was evaluated, as well. Finally, Joshua Shinavier suggested the application of REC in KRMs of graph-based recommender systems (see [Shi11]).

5.2.4 Indexing

The designed ontologies are catalogued in several ontology registries and submitted to various Semantic Web search engines. I contributed some of these enrolments by myself. However, others were conducted by the maintainers of these aggregation tools or automatically by their algorithms.
First of all, I added the ontologies co-developed to *Schemapedia*[12]. This is a very practical Semantic Web vocabulary repository, because it supports the embedding of examples and annotations

[7] `http://openrecommender.org`
[8] `http://bcmoney-mobiletv.com`
[9] Unfortunately, this reaction was part of an *offline* e-mail conversation.
[10] `http://smartmuseum.eu`
[11] Kingsley Idehen is the CEO of *OpenLink*.
[12] `http://schemapedia.com`

via tags. Secondly, I submitted these specifications to the *Ontology Design Patterns*[13] repository. It is designed for describing ontology design patterns and referring prototypical implementations to them. Moreover, I registered the vocabulary URIs on *Sindice*[14], which is a comprehensive Semantic Web indexing service with search and exploration functionalities.
Besides these contributions of my own, the creators of the *Linked Open Vocabularies*[15] (LOV) information service added the ontologies of this work to their catalogue. Interesting analyses regarding the different relationships between single vocabularies are illustrated for each entry in LOV. These can be, for example, specialisations, generalisations, or equivalence relation types. Finally, Kingsley Idehen submitted all described specifications to the *Linked Open Data* cloud cache[16].
In contrast to these Semantic Web registry and search engine contributions, I tried to apply some search engine optimisation (SEO) methods, so that people can find the proposed ontologies with the help of a traditional search engine, such as *Google*, more easily. Firstly, I generated *sitemap* from the web space where the ontology specifications (incl. their documentations) are hosted. This is a quite common SEO task. Therefore, I utilised the *Remote Sitemap Generator* tool [LG11] that was initially developed by Martin Lévesque. I had to contribute some minor refinements to this open-source application to enable the processing of PURLs (see [Gän11f]), which is an important deployment method for Linked Data (see 2.2.4.5).
Additionally, every specification documentation is serialised in the *HTML+RDFa* format. In other words, each documentation contains the RDF graph of its ontology specification as well. The RDFa serialisation format is also acknowledged by an increasing number of traditional search engines. Websites that embed this KR format usually receive higher page rankings. I generated most parts of these documentation files automatically with the help of an *SpecGen* version that I massively extended for a comprehensive RDFa support (see [Gän10d]).

5.3 SUMMARY

This chapter outlines the application (see Subsection 5.1) and evaluation (see Subsection 5.2) of the Semantic Web ontologies that were created or massively modified to be integrated into the PMKB information space. So far, two *RDFizers* were created as part of the PMKB prototype. The first one is the *AudioScrobbler RDF Service*, which transforms results from the *Last.fm* Web Service to Semantic Web KRs by utilising terms of PBO, CO, REC, and IS amongst other *universals* from further Semantic Web vocabularies (see Subsection 5.1.1). The second *RDFizer*, PITE, can handle the mapping of ID3 tags into Semantic Web KRs by applying concepts and *relations* of PBO, CO, AO, MT, and IS amongst other definitions from separate namespaces (see Subsection 5.1.2).
The evaluation of the ontologies demonstrates various reutilisations and applications of them in other projects (see Subsections 5.2.1 and 5.2.2). In addition, some quite ambitious MMSs, *Seevl* and *Libre.fm*, showed interest in an utilisation of parts of the developed set of vocabularies. Besides, these specifications already received multiple peer reviews and mentions on the WWW (see Subsection 5.2.3). Finally, some hints regarding an optimisation of Semantic Web ontology deployment are outlined in Subsection 5.2.4. These steps should particularly help to reach a broader audience and probably encourage some developers to also make use of the proposed vocabularies. This contributes to an easy establishment of a "shared understanding" of the domains and purposes that are modelled in the different domains.

[13] `http://ontologydesignpatterns.org`
[14] `http://sindice.com`
[15] `http://labs.mondeca.com/dataset/lov/`
[16] `http://lod.openlinksw.com`

Copyright TU Dresden, Thomas Gängler

6 CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

This thesis describes the concept *personal music knowledge base* and outlines a refinement of the PMKB vision with a strong focus on the knowledge representation part. The PMKB concepts manifests the aspects of a music KMS that are necessary to perform a personalised semantic federation of music and music-related information for further KM tasks.
At the beginning, several basic use cases, that are related to personal music collection management, are classified, and their requirements regarding (music) metadata are worked out (see Section 1.2). Thereby, the main outcome is that a wide variety of information is needed to satisfy fuzzy and exact user requests when consuming music of personal libraries.
All basic knowledge that is necessary to understand the different parts of the proposed solution PMKB, is introduced in Chapter 2. This technology background part illustrates the whole way down from abstract basics about knowledge management (see Section 2.1) to a pretty powerful instantiation that is in line with the outlined knowledge representation and knowledge management system concepts.
This exemplification is called the Semantic Web and its essential technologies are explained in Section 2.2. The suitable integration of the Semantic Web into the existing Web ecosystem and a verification of the design of the basic Semantic Web KRLs regarding their foundations and predecessors that contributed to this quite natural KR format, is particularly described in that part. Furthermore, it is revealed that Semantic Web KRMs meet the requirements of the versatile information that has to be expressed to be able to properly manage personal music collections.
Since Chapter 2 deals with music information management, a section about music metadata has to be a part of it as well (see Section 2.3). Thereby, the *Music Ontology* is suggested as a very expressive and extensible music metadata format which is embedded in a whole framework of further Semantic Web vocabularies that can be used for describing musical characteristics. Besides, various music metadata services that can act as information providers for a PMKB are presented in this section including proof-of-concept examples that showcase their integratability into a uniform Semantic Web KRM.
Finally, the *state-of-the-art* regarding user and context modelling that is used for personalisation tasks is briefly outlined in Section 2.4 to close the chapter on technology background. Albeit, this part again demonstrates that Semantic Web ontologies are a good choice for representing such models, it shows that proper user profile and context descriptions are not possible with the existing KR vocabularies in an easy way.
The comprehensive technology basics chapter provides the foundation for the abstract PMKB

concept that is introduced in Chapter 3. It explains the client-server architecture with its uniform communication endpoints and ontological knowledge representation model format that powers the *personal music knowledge base*. Moreover, the complete information flow life cycle is illustrated, including the processes of user account initialisation, information service choice, individual information extraction, and proactive update notification.
The PMKB vision is refined in Chapter 4 with a special focus on its knowledge representation part (see Section 4.1). Several new Semantic Web ontologies are defined or existing ones are massively modified. The outcome is, amongst others,

- a new vocabulary for describing the *play back domain*, the *Play Back Ontology*,
- another one, the *Info Service Ontology*, for representing information service categorisations and quality ratings, and
- one, the *Cognitive Characteristics Ontology*, that unites the beneficial parts of the existing advanced user modelling ontologies.

In addition, a Semantic Web KRL, the *Property Reification Vocabulary*, that can be used to represent that certain exemplified *binary relations* belong together semantically to certain instantiated *n-ary relations*, is presented.
The refinement of the PMKB knowledge management system is only briefly sketched-out in Section 4.2. Albeit, even the enhancements that are proposed for the KR part of the PMKB concept are of much value for guiding the realisation of this vision a bit more into its full, but non-trivial implementation. The introduced Semantic Web ontologies can be perfectly utilised in conjunction with the existing Music Ontology framework to represent the majority of the *universals* of the PMKB KRM. Due to their project-independence, these vocabularies can be deployed in every KRM that needs their KR capacities. These can be, for example, KRMs of music metadata services or *format-bound* music metadata formats, as well as, KRMs of information services of other domains, e.g., the movie domain.
The applications and evaluation of the proposed vocabularies of the PMKB information space in practise are outlined in Chapter 5. Firstly, some *RDFizers* are described that also make use of the new ontologies in their mapping definitions. Secondly, a social evaluation method is applied to carry out an examination dealing with the reutilisation, application and feedback of the vocabularies that are explained in Section 4.1 of this work. This analysis shows that it is a good practise to properly publish Semantic Web ontologies with the help of some Linked Data principles and further basic SEO techniques to easily reach the searching audience, to avoid duplicates of such KR specifications, and, last but not least, to directly establish a "shared understanding".
This thesis added its value to make the vision of a *personal music knowledge base* come true.

6.2 FUTURE WORK

The realisation of the PMKB vision probably requires several further master thesis projects or other basic conditions, such as an open-source project with several contributors. Since, enhancements in the KR part of the PMKB concept are particularly the outcome within the scope of this work, especially the implementation of the sketched-out PMKB KMS has a high importance for future activities on this proposal (see Section 4.2 and the introduction of Chapter 5).
Besides, the stereotype and context modelling requires some further refinements as well as the user modelling in general to be able to represent user aspects such as personality or emotional state. Stereotype profiles can probably be modelled and deployed in user profiles with the help of a combination of the Semantic Web KRLs OWL, SKOS and SPIN (see Section 4.2). A music context ontology as a specialisation of AO is imaginable. Further cognitive patterns can be defined within the scope of CCO.
Moreover, the *Info Service Ontology* only represents the beginning of a comprehensive description of information services. A detailed information service quality ontology and an implementation of a full information service rating system might be the next steps for this aspect of a semantic information integration system.
All given extension ideas are only examples of future work that can be done to make the *personal music knowledge base* vision alive.

BIBLIOGRAPHY

[ABMH10] Ben Adida, Mark Birbeck, Shane McCarron, and Ivan Herman. RDFa Core 1.1 [online]. October 2010. Available from: http://www.w3.org/TR/rdfa-core/ [cited 10-01-2011].

[ABP10] Ben Adida, Mark Birbeck, and Steven Pemberton. HTML+RDFa 1.1 [online]. October 2010. Available from: http://www.w3.org/TR/rdfa-in-html/ [cited 10-01-2011].

[ACHZ10] Keith Alexander, Richard Cyganiak, Michael Hausenblas, and Jun Zhao. Describing Linked Datasets with the voiD Vocabulary [online]. December 2010. Available from: http://www.w3.org/2001/sw/interest/void/ [cited 16-02-2011].

[AD08] Aduna and DFKI. aperture/extractore/mp3 [online]. 2008. Available from: http://aperture.svn.sourceforge.net/viewvc/aperture/aperture/trunk/core/src/main/java/org/semanticdesktop/aperture/extractor/mp3/ [cited 21-04-2011].

[AD10] Aduna and DFKI. Aperture Framework [online]. July 2010. Available from: http://aperture.sourceforge.net/ [cited 11-01-2011].

[ADG10] Aduna, DFKI, and Thomas Gängler. pmkb-aperture-extractor [online]. November 2010. Available from: http://smiy.svn.sourceforge.net/viewvc/smiy/pmkb-aperture-extractor/ [cited 11-01-2011].

[Adu09] Aduna. openRDF.org [online]. 2009. Available from: http://www.openrdf.org/ [cited 22-01-2011].

[AFH+10] Danny Ayers, Lee Feigenbaum, Michael Hausenblas, Tom Heath, Wolfgang Halb, and Yves Raimond. The Statistical Core Vocabulary [online]. September 2010. Available from: http://vocab.deri.ie/scovo [cited 10-12-2010].

[AFM+10] Use Cases and Requirements for Mapping Relational Databases to RDF [online]. June 2010. Available from: http://www.w3.org/TR/rdb2rdf-ucr/ [cited 10-01-2011].

[AG10] Samer A. Abdallah and Thomas Gängler. The Ordered List Ontology Specification [online]. July 2010. Available from: http://purl.org/ontology/olo/orderedlistontology.html [cited 10-07-2010].

[AH07] Danny Ayers and Tom Heath. RDF Review Vocabulary [online]. November 2007. Available from: http://purl.org/stuff/rev# [cited 10-09-2010].

[Amp11] AmpliFIND Music Services. AmpliFIND (formerly MusicDNS) [online]. 2011. Available from: http://www.amplifindmusicservices.com/what/amplifind.php [cited 10-04-2011].

[App11] Apple Inc. Apple - iTunes [online]. 2011. Available from: http://www.apple.com/itunes/features/ [cited 23-04-2011].

[Aye11] Danny Ayers. Twitter / @danja [online]. February 2011. Available from: http://twitter.com/danja/status/40109749286477824 [cited 22-04-2011].

[BBCT10] Paul Buchheit, Gary Burd, DeWitt Clinton, and B. Taylor. simpleupdateprotocol [online]. 2010. Available from: http://code.google.com/p/simpleupdateprotocol/ [cited 22-02-2011].

[BBLP10] David Beckett, Tim Berners-Lee, and Eric Prud'hommeaux. Turtle - Terse RDF Triple Language [online]. January 2010. Available from: http://www.w3.org/2010/01/Turtle/ [cited 28-02-2011].

[BC09] Chris Bizer and Richard Cyganiak. D2R Server [online]. August 2009. Available from: http://www4.wiwiss.fu-berlin.de/bizer/d2r-server/ [cited 11-01-2011].

[BCH08] Chris Bizer, Richard Cyganiak, and Tom Heath. How to Publish Linked Data on the Web [online]. July 2008. Available from: http://www4.wiwiss.fu-berlin.de/bizer/pub/LinkedDataTutorial/ [cited 08-01-2011].

[BCM+07] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook*. Cambridge University Press, 2nd edition edition, August 2007.

[Bcm11a] Bcmoney. OpenRecommender [online]. January 2011. Available from: http://wiki.bcmoney-mobiletv.com/index.php?title=OpenRecommender [cited 22-04-2011].

[Bcm11b] Bcmoney. Recommender System [online]. April 2011. Available from: http://wiki.bcmoney-mobiletv.com/index.php?title=Recommender [cited 22-04-2011].

[Bec04a] Dave Beckett. RDF/XML Syntax Specification (Revised) [online]. February 2004. Available from: http://www.w3.org/TR/rdf-syntax-grammar/ [cited 09-03-2011].

[Bec04b] Dave Beckett. Redland librdf RDF API Library - Contexts [online]. January 2004. Available from: http://librdf.org/notes/contexts.html [cited 16-04-2011].

[Bec11] Dave Beckett. Redland RDF Libraries [online]. 2011. Available from: http://librdf.org/ [cited 16-04-2011].

[Ben10] Arto Bendiken. How RDF Databases Differ from Other NoSQL Solutions [online]. April 2010. Available from: http://blog.datagraph.org/2010/04/rdf-nosql-diff [cited 22-01-2011].

[BG11a] Michael Bergman and Frédérick Giasson. UMBEL - Reference Concept Ontology and Vocabulary [online]. February 2011. Available from: http://umbel.org/ [cited 15-02-2011].

[BG11b] Eva Blomqvist and Thomas Gängler. Ordered Lists [online]. February 2011. Available from: http://ontologydesignpatterns.org/wiki/Community:Ordered_Lists [cited 20-04-2011].

[BJC11] Chris Bizer, Anja Jentzsch, and Richard Cyganiak. State of the LOD Cloud [online]. March 2011. Available from: http://www4.wiwiss.fu-berlin.de/lodcloud/state/ [cited 05-04-2011].

[BK10] Harold Boley and Michael Kifer. RIF Framework for Logic Dialects [online]. June 2010. Available from: http://www.w3.org/TR/rif-fld/ [cited 08-01-2011].

[BKN02] Stephan Baumann, Andres Klüter, and Marie Norlien. Using natural language input and audio analysis for a human-oriented MIR system. In *Proceedings of Web Delivery of Music (WEDELMUSIC) 2002*, 2002.

[BL89] Tim Berners-Lee. Information Management: A Proposal [online]. March 1989. Available from: http://www.w3.org/History/1989/proposal.html [cited 07-01-2011].

[BL97] Tim Berners-Lee. Realising the Full Potential of the Web [online]. December 1997. Available from: http://www.w3.org/1998/02/Potential.html [cited 07-01-2011].

[BL98] Tim Berners-Lee. Cool URIs don't change [online]. 1998. Available from: http://www.w3.org/Provider/Style/URI [cited 16-02-2011].

[BL06] Tim Berners-Lee. Notation 3 [online]. March 2006. Available from: http://www.w3.org/DesignIssues/Notation3 [cited 08-01-2011].

[BL09a] Tim Berners-Lee. A Short History of "Resource" in web architecture [online]. August 2009. Available from: http://www.w3.org/DesignIssues/TermResource.html [cited 10-01-2011].

[BL09b] Tim Berners-Lee. Evolvability [online]. August 2009. Available from: http://www.w3.org/DesignIssues/Evolution.html [cited 10-01-2011].

[BL09c] Tim Berners-Lee. Interpretation Properties [online]. August 2009. Available from: http://www.w3.org/DesignIssues/InterpretationProperties.html [cited 14-01-2011].

[BL09d] Tim Berners-Lee. Meaning [online]. August 2009. Available from: http://www.w3.org/DesignIssues/Meaning.html [cited 10-01-2011].

[BL09e] Tim Berners-Lee. Metadata Architecture [online]. August 2009. Available from: http://www.w3.org/DesignIssues/Metadata.html [cited 06-01-2011].

[BL09f] Tim Berners-Lee. Web Architecture from 50,000 feet [online]. August 2009. Available from: http://www.w3.org/DesignIssues/Architecture.html [cited 07-01-2011].

[BL10a] Tim Berners-Lee. Levels of Abstraction: Net, Web, Graph [online]. June 2010. Available from: http://www.w3.org/DesignIssues/Abstractions.html [cited 07-01-2011].

[BL10b] Tim Berners-Lee. Linked Data [online]. 2010. Available from: http://www.w3.org/DesignIssues/LinkedData.html [cited 08-01-2011].

[BL10c] Tim Berners-Lee. Long Live the Web: A Call for Continued Open Standards and Neutrality. *Scientific American*, 2010-12:n/a, December 2010. Available from: http://www.scientificamerican.com/article.cfm?id=long-live-the-web.

[BL10d] Tim Berners-Lee. Principles of Design [online]. March 2010. Available from: http://www.w3.org/DesignIssues/Principles.html [cited 07-01-2011].

[BL10e] Tim Berners-Lee. Read-Write Linked Data [online]. December 2010. Available from: http://www.w3.org/DesignIssues/ReadWriteLinkedData.html [cited 16-02-2011].

[BLC90] Tim Berners-Lee and R. Cailliau. WorldWideWeb: Proposal for a HyperText Project [online]. November 1990. Available from: http://www.w3.org/Proposal.html [cited 07-01-2011].

[BLCK+08] Tim Berners-Lee, Dan Connolly, Lalana Kagal, Yosi Scharf, and Jim Hendler. N3Logic: A logical framework for the World Wide Web. *Theory and Practice of Logic Programming*, 8(03):N3Logic: A logical framework for the World Wide Web, April 2008. Available from: http://www.journals.cambridge.org/abstract_S1471068407003213, doi:10.1017/S1471068407003213.

[BLFM05] T. Berners-Lee, R. Fielding, and L. Masinter. Uniform Resource Identifier (URI): Generic Syntax [online]. January 2005. Available from: http://tools.ietf.org/html/rfc3986 [cited 07-01-2011].

[Blo10] Eva Blomqvist. List [online]. December 2010. Available from: http://ontologydesignpatterns.org/wiki/Submissions:List [cited 20-04-2011].

[BM08] Dan Brickley and Libby Miller. Introducing FOAF [online]. 2008. Available from: http://www.foaf-project.org/original-intro [cited 17-04-2011].

[BM10] Mark Birbeck and Shane McCarron. CURIE Syntax 1.0 [online]. December 2010. Available from: http://www.w3.org/TR/curie [cited 23-01-2011].

[BMI+10] Dan Brickley, Libby Miller, Toby Inkster, Yi Zeng, Yan Wang, Danica Damljanovic, Zhisheng Huang, Sheila Kinsella, John Breslin, and Thomas Gängler. The Weighted Interests Vocabulary [online]. September 2010. Available from: http://purl.org/ontology/wi/weightedinterests.html [cited 10-12-2010].

[Boj07] Uldis Bojars. ResumeRDF Ontology Specification [online]. January 2007. Available from: http://rdfs.org/resume-rdf/ [cited 17-04-2011].

[Boo10] David Booth. URI Declaration in Semantic Web Architecture [online]. June 2010. Available from: http://dbooth.org/2007/uri-decl/ [cited 08-01-2011].

[BPSM+08] Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, and François Yergeau. Extensible Markup Language [online]. November 2008. Available from: http://www.w3.org/TR/xml/ [cited 08-01-2011].

[Bri04] Dan Brickley and R.V. Guha, editors. RDF Vocabulary Description Language 1.0: RDF Schema [online]. February 2004. Available from: http://www.w3.org/TR/rdf-schema/ [cited 10-07-2010].

[Bri07] Dan Brickley. FoafHistory [online]. 2007. Available from: http://wiki.foaf-project.org/w/FoafHistory [cited 17-04-2011].

[BRM+10] Dan Brickley, Dave Reynolds, Libby Miller, Toby Inkster, Yi Zeng, Yan Wang, Danica Damljanovic, Zhisheng Huang, Alexandre Passant, Sheila Kinsella, John Breslin, and Thomas Gängler. The Cognitive Characteristics Ontology Specification [online]. September 2010. Available from: http://purl.org/ontology/cco/cognitivecharacteristics.html [cited 10-12-2010].

[Bus45] Vannevar Bush. As We May Think. *The Atlantic Monthly*, 176:101–108, July 1945. Available from: http://www.theatlantic.com/past/docs/unbound/flashbks/computer/bushf.htm.

[Can06] Pedro Cano. *Content-based Audio Search: From Fingerprinting to Semantic Audio Retrieval*. PhD thesis, Technology Department of Pompeu Fabra University, Barcelona, 2006.

[Car96] B. Carpenter. Architectural Principles of the Internet [online]. June 1996. Available from: http://tools.ietf.org/html/rfc1958 [cited 07-01-2011].

[Car08] Jeremy Carroll. Named Graphs [online]. January 2008. Available from: http://www.w3.org/2004/03/trix/Overview.html [cited 12-01-2011].

[Cas10a] Norman Casagrande. Fingerprint API and app updated [online]. July 2010. Available from: http://blog.last.fm/2010/07/09/fingerprint-api-and-app-updated [cited 13-04-2011].

[Cas10b] Norman Casagrande. lastfm / Fingerprinter [online]. July 2010. Available from: https://github.com/lastfm/Fingerprinter [cited 13-04-2011].

[CBF06] Sally Jo Cunningham, David Bainbridge, and Annette Falconer. 'More of an Art than a Science': Supporting the Creation of Playlists and Mixes. In *Proceedings of the 7th International Conference on Music Information Retrieval*, Victoria, Canada, October 2006.

[CBHS05] Jeremy J. Carroll, Christian Bizer, Pat Hayes, and Patrick Stickler. Named graphs, provenance and trust. In *Proceedings of the 14th international conference on World Wide Web*, page 613, New York, New York, USA, 2005. ACM Press. Available from: http://portal.acm.org/citation.cfm?doid=1060745.1060835, doi:10.1145/1060745.1060835.

[CGD06] Nik Corthaut, Sten Govaerts, and Erik Duval. Moody Tunes: The Rockanango Project. In *Proceedings of the 7th International Conference of Music Information Retrieval*, Victoria, Kanada, Oktober 2006.

[CGVD08] Nik Corthaut, Sten Govaerts, Katrien Verbert, and Erik Duval. Connecting the Dots: Music Metadata Generation, Schemas and Applications. In *Proceedings of the Ninth International Conference on Music Information Retrieval*, Philadelphia, USA, September 2008.

[CHH09] Richard Cyganiak, Andreas Harth, and Aidan Hogan. N-Quads: Extending N-Triples with Context [online]. Novemer 2009. Available from: http://sw.deri.org/2008/07/n-quads/ [cited 12-01-2011].

[Cic09] Paolo Ciccarese. Collections Ontology Specification [online]. January 2009. Available from: http://swan.mindinformatics.org/spec/1.2/collections.html [cited 22-03-2011].

[CJJ04] Sally Jo Cunningham, Matt Jones, and Steve Jones. Organizing Digital Music for Use: An Examination of Personal Music Collections. In *Proceedings of the 5th International Conference on Music Information Retrieval*, Barcelona, Spain, October 2004.

[CK91] J. M. Crawford and Benjamin Kuipers. *Principles of Semantic Networks*, chapter ALL – Formalizing Access Limited Reasoning, pages 299–330. Morgan Kaufmann, 1991.

[Cla11] Clark & Parsia, LLC. Pellet: OWL 2 Reasoner for Java [online]. 2011. Available from: http://clarkparsia.com/pellet/ [cited 22-02-2011].

[CLP+07] Bin Cui, Ling Liu, Calton Pu, Jialie Shen, and Kian-Lee Tan. QueST: Querying Music Databases by Acoustic and Textual Features. In *MULTIMEDIA '07: Proceedings of the 15th international conference on Multimedia*, pages 1055–1064, New York, NY, USA, 2007. ACM. doi:http://doi.acm.org/10.1145/1291233.1291465.

[CO09] Nepomuk Consortium and OSCAF. OSCAF/NEPOMUK Ontologies [online]. 2009. Available from: http://www.semanticdesktop.org/ontologies/ [cited 21-04-2011].

[Col07] Robert M. Colomb. *Ontology and the Semantic Web*. IOS Press, 2007.

[Con07] Dan Connolly. Gleaning Resource Descriptions from Dialects of Languages [online]. September 2007. Available from: http://www.w3.org/TR/grddl/ [cited 11-01-2011].

[Cro06] D. Crockford. The application/json Media Type for JavaScript Object Notation [online]. July 2006. Available from: http://tools.ietf.org/html/rfc4627 [cited 08-01-2011].

[CS04] Jeremy J. Carroll and Patrick Stickler. TriX : RDF Triples in XML. Technical report, Digital Media Systems Laboratory, HP Laboratories Bristol, 2004.

[CV00] Wei Chai and Barry Vercoe. Using user models in music information retrieval systems. In *1st Symposium on Music Information Retrieval*, Plymouth, Massachusetts, USA, October 2000.

[Dav07] Ian Davis. 303 Asymmetry [online]. December 2007. Available from: http://blog.iandavis.com/2007/12/05/303-asymmetry/ [cited 08-01-2011].

[dB10] Jos de Bruijn. RIF RDF and OWL Compatibility [online]. June 2010. Available from: http://www.w3.org/TR/rif-rdf-owl/ [cited 08-01-2011].

[DCM08] DCMI Usage Board. DCMI Metadata Terms [online]. January 2008. Available from: http://dublincore.org/documents/dcmi-terms/ [cited 10-09-2010].

[DER11] DERI. Neologism - Easy Vocabulary Publishing [online]. February 2011. Available from: http://neologism.deri.ie/ [cited 02-03-2011].

[DG10] Nima Dokoohaki and Thomas Gängler. Beyond interests [online]. September 2010. Available from: http://lists.foaf-project.org/pipermail/foaf-dev/2010-September/010388.html [cited 22-04-2011].

[DHSW02] Erik Duval, Wayne Hodgins, Stuart Sutton, and Stuart L. Weibel. Metadata Principles and Practicalities. *D-Lib Magazine*, 8(4):–, April 2002.

[DJ10] Simon Dixon and Kurt Jacobson. LinkedBrainz - A project to provide MusicBrainz NGS as Linked Data [online]. November 2010. Available from: http://linkedbrainz.c4dmpresents.org/ [cited 05-04-2011].

[DN09] Ian Davis and Richard Newman. Expression of Core FRBR Concepts in RDF [online]. May 2009. Available from: http://vocab.org/frbr/core.html [cited 10-10-2010].

[DPdSM05] Li Ding, Yun Peng, Paulo Pinheiro da Silva, and Deborah L. McGuinness. Tracking RDF Graph Provenance using RDF Molecules. Technical report, UMBC, April 2005.

[DR08] T. Dierks and E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.2 [online]. August 2008. Available from: http://tools.ietf.org/html/rfc5246 [cited 22-02-2011].

[DS10] L. Dusseault and J. Shell. PATCH Method for HTTP [online]. March 2010. Available from: http://tools.ietf.org/html/rfc5789 [cited 16-02-2011].

[DSC10] Souripriya Das, Seema Sundara, and Richard Cyganiak. R2RML: RDB to RDF Mapping Language [online]. October 2010. Available from: http://www.w3.org/TR/r2rml/ [cited 11-01-2011].

[Dub10a] Dublin Core Metadata Initiative. Twitter / @DublinCore [online]. October 2010. Available from: http://twitter.com/DublinCore/status/26651006313 [cited 22-04-2011].

[Dub10b] Dublin Core Metadata Initiative. Twitter / @DublinCore [online]. December 2010. Available from: http://twitter.com/DublinCore/status/12506920367685633 [cited 22-04-2011].

[Dub11a] Dublin Core Metadata Initiative. DCMI Home: Dublin Core Metadata Initiative (DCMI) [online]. April 2011. Available from: http://dublincore.org/ [cited 22-04-2011].

[Dub11b] Dublin Core Metadata Initiative. Twitter / @DublinCore [online]. February 2011. Available from: http://twitter.com/DublinCore/status/35344721261891584 [cited 22-04-2011].

[Edl11] Stefan Edlich. NOSQL Databases [online]. January 2011. Available from: http://nosql-database.org/ [cited 12-01-2011].

[Eng62] Douglas C. Engelbart. Augmenting Human Intellect: A Conceptual Framework. Technical report, Air Force Office of Scientific Research, October 1962.

[FGM+99] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext Transfer Protocol [online]. June 1999. Available from: http://tools.ietf.org/html/rfc2616 [cited 07-01-2011].

[Fie00] R.T. Fielding. *Architectural styles and the design of network-based software architectures*. Phd thesis, University of California, Irvine, 2000.

[Fie08] Roy T. Fielding. A little REST and Relaxation. In *ApacheCon Europe 2008*, 2008. slides.

[Fie11] Benjamin Fields. *Contextualize Your Listening : The Playlist as Recommendation Engine*. PhD thesis, Department of Computing, Goldsmiths, University of London, 2011.

[Fin04] Michael Fingerhut. Music Information Retrieval, or how to search for (and maybe find) music and do away with incipits. In *International Association of Sound and Audiovisual Archives (IASA) Conference 2004*, Oslo, Norwegen, 2004.

[FL10] Ben Fields and Paul Lamere. Finding a Path Through the Juke Box: The Playlist Tutorial. In *11th International Society of Music Information Retrieval Conference (ISMIR 2010)*, August 2010. tuturial. Available from: http://www.slideshare.net/BenFields/finding-a-path-through-the-juke-box-the-playlist-tutorial.

[Fra10] Franz Inc. AllegroGraph RDFStore Web 3.0's Database [online]. 2010. Available from: http://www.franz.com/agraph/allegrograph/ [cited 15-01-2011].

[FSA10] B. Fitzpatrick, B. Slatkin, and M. Atkins. PubSubHubbub Core 0.3 – Working Draft [online]. February 2010. Available from: http://pubsubhubbub.googlecode.com/svn/trunk/pubsubhubbub-core-0.3.html [cited 22-02-2011].

[FT02] Roy T. Fielding and Richard N. Taylor. Principled design of the modern Web architecture. *ACM Transactions on Internet Technology*, 2(2):115–150, May 2002. Available from: http://portal.acm.org/citation.cfm?doid=514183.514185, doi:10.1145/514183.514185.

[Gal10] Ian Davis David Galbraith. BIO: A vocabulary for biographical information [online]. June 2010. Available from: http://purl.org/vocab/bio/0.1/ [cited 17-04-2011].

[Gar09] Garlik. 4store [online]. 2009. Available from: http://4store.org/ [cited 15-01-2011].

[Gar10] Garlik. 5store [online]. February 2010. Available from: http://4store.org/trac/wiki/5store [cited 22-01-2011].

[GCG+10] Yolanda Gil, James Cheney, Paul Groth, Olaf Hartig, Simon Miles, Luc Moreau, and Paulo Pinheiro da Silva. Provenance XG Final Report [online]. December 2010. Available from: http://www.w3.org/2005/Incubator/prov/XGR-prov/ [cited 12-01-2011].

[GD10] Frédérick Giasson and Bruce D'Arcus. The Bibliographic Ontology [online]. February 2010. Available from: http://bibliontology.com/ [cited 10-10-2010].

[Gdh07] J. Gregorio and B. de hOra. The Atom Publishing Protocol [online]. October 2007. Available from: http://tools.ietf.org/html/rfc5023 [cited 23-02-2011].

[GI10] Thomas Gängler and Toby Inkster. The Association Ontology Specification [online]. September 2010. Available from: http://purl.org/ontology/ao/associationontology.html [cited 10-08-2010].

[GJ10a] Thomas Gängler and Kurt Jacobson. Media Types [online]. October 2010. Available from: http://purl.org/ontology/mt/ [cited 19-04-2011].

[GJ10b] Thomas Gängler and Kurt Jacobson. The Play Back Ontology Specification [online]. September 2010. Available from: http://purl.org/ontology/pbo/playbackontology.html [cited 10-10-2010].

[GJ10c] Thomas Gängler and Kurt Jacobson. The Recommendation Ontology Specification [online]. August 2010. Available from: http://purl.org/ontology/rec/recommendationontology.html [cited 10-11-2010].

[GKP05] Maciej Gawinecki, Mateusz Kruszyk, and Marcin Paprzycki. Ontology-based Stereotyping in a Travel Support System. In *Proceeding of the XXI Autuumn Meeting of Polish Information Processing Society*, pages 73–85, 2005.

[GMF04] R Guha, R Mccool, and R Fikes. Contexts for the Semantic Web. In *Proceedings of the International Semantic Web Conference 2004*, volume 3298 of *Lecture Notes of Computer Science*, page 32 46. Springer, 2004.

[Gän09a] Thomas Gängler. Metadaten und Merkmale zur Verwaltung von persönlichen Musiksammlungen. Belegarbeit, Department of Comupter Science, University of Technology Dresden, September 2009.

[Gän09b] Thomas Gängler. Metadaten und Merkmale zur Verwaltung von persönlichen Musiksammlungen, November 2009. presentation. Available from: http://www.slideshare.net/zaziki/metadaten-und-merkmale-zur-verwaltung-von-persnlichen-musiksammlungen-low-res.

[Gän10a] Thomas Gängler. Association Ontology with Graphs [online]. August 2010. Available from: http://smiy.svn.sourceforge.net/viewvc/smiy/associationonto/branches/associationonto_w_graphs/ [cited 10-10-2010].

[Gän10b] Thomas Gängler. DBPedia [online]. June 2010. Available from: http://purl.org/ontology/is/inst/dbpedia [cited 19-04-2011].

[Gän10c] Thomas Gängler. Further Semantic Web ontologies for the T-Box of the Libre.fm knowledge representation model [online]. December 2010. Available from: http://lists.nongnu.org/archive/html/librefm-discuss/2010-12/msg00001.html [cited 22-04-2011].

[Gän10d] Thomas Gängler. My SpecGen version 6 [online]. July 2010. Available from: http://smiy.org/2010/07/13/my-specgen-version-6/ [cited 13-02-2011].

[Gän10e] Thomas Gängler. On Resources, Information Resources and Documents [online]. November 2010. Available from: http://infoserviceonto.smiy.org/2010/11/25/on-resources-information-resources-and-documents/ [cited 10-01-2011].

[Gän10f] Thomas Gängler. Ontologies and Vocabularies in the PMKB space [online]. December 2010. Available from: http://smiy.wordpress.com/2010/12/20/ontologies-and-vocabularies-in-the-pmkb-space/ [cited 06-03-2011].

[Gän10g] Thomas Gängler. Some More Individual - Webspace Index [online]. December 2010. Available from: http://purl.org/smiy/ [cited 22-03-2011].

[Gän10h] Thomas Gängler. The Association Ontology [online]. July 2010. Available from: http://smiy.org/2010/07/26/the-association-ontology/ [cited 10-08-2010].

[Gän10i] Thomas Gängler. The Cognitive Characteristics Ontology [online]. October 2010. Available from: http://smiy.org/2010/10/19/the-cognitive-characteristics-ontology/ [cited 2010-11-17].

[Gän10j] Thomas Gängler. The Counter Ontology [online]. July 2010. Available from: http://smiy.org/2010/07/26/the-counter-ontology/ [cited 10-08-2010].

[Gän10k] Thomas Gängler. The Info Service Ontology - Describing, Rating, Categorising Information Services with the Info Service Ontology [online]. July 2010. Available from: http://infoserviceonto.smiy.org/ [cited 10-06-2010].

[Gän10l] Thomas Gängler. The Info Service Ontology Specification [online]. July 2010. Available from: http://purl.org/ontology/is/infoservice.html [cited 10-06-2010].

[Gän10m] Thomas Gängler. The Ordered List Ontology [online]. July 2010. Available from: http://smiy.org/2010/07/15/the-ordered-list-ontology/ [cited 10-07-2010].

[Gän10n] Thomas Gängler. The Play Back Ontology [online]. July 2010. Available from: http://smiy.org/2010/07/27/the-play-back-ontology/ [cited 10-10-2010].

[Gän10o] Thomas Gängler. The Recommendation Ontology [online]. August 2010. Available from: http://smiy.org/2010/08/07/the-recommendation-ontology/ [cited 10-11-2010].

[Gän10p] Thomas Gängler. The Weighting Ontology Specification [online]. September 2010. Available from: http://purl.org/ontology/wo/weightingontology.html [cited 10-12-2010].

[Gän11a] Thomas Gängler. A Comparision of the SWAN Collections Ontology and the Ordered List Ontology [online]. March 2011. Available from: https://sourceforge.net/apps/mediawiki/smiy/index.php?title=A_Comparision_of_the_SWAN_Collections_Ontology_and_the_Ordered_List_Ontology [cited 22-03-2011].

[Gän11b] Thomas Gängler. A generalisation of the Linked Data publishing guideline [online]. February 2011. Available from: http://smiy.org/2011/02/17/a-generalisation-of-the-linked-data-publishing-guideline/ [cited 10-03-2011].

[Gän11c] Thomas Gängler. Mo - levels of abstraction - whole way down [online]. April 2011. Available from: http://wiki.musicontology.com/index.php/Mo_-_levels_of_abstraction_-_whole_way_down [cited 10–04-2011].

[Gän11d] Thomas Gängler. Property Reification Vocabulary Rules [online]. February 2011. Available from: http://purl.org/ontology/prv/rules# [cited 20-04-2011].

[Gän11e] Thomas Gängler. Reification and Provenance modelling [online]. January 2011. comment. Available from: http://lists.w3.org/Archives/Public/public-rdf-comment/2011Jan/0001.html [cited 13-02-2011].

[Gän11f] Thomas Gängler. smiy/remote-sitemap [online]. February 2011. Available from: http://smiy.svn.sourceforge.net/viewvc/smiy/remote-sitemap/ [cited 22-04-2011].

[Gän11g] Thomas Gängler. The common, layered Semantic Web technology stack [online]. 2011. Available from: http://smiy.org/2011/01/10/the-common-layered-semantic-web-technology-stack/ [cited 15-04-2011].

[God09] Andrew Godwin. LastGraph [online]. January 2009. Available from: http://lastgraph.aeracode.org/ [cited 13-04-2011].

[Goo11] Google. Social Graph API [online]. 2011. Available from: http://code.google.com/apis/socialgraph/ [cited 17-04-2011].

[GPM08] I. Goncalves, S. Pfeiffer, and C. Montgomery. Ogg Media Types [online]. September 2008. Available from: http://tools.ietf.org/html/rfc5334 [cited 10-03-2011].

[Gra10] Gracenote. Gracenote Playlist Plus [online]. May 2010. Available from: http://www.gracenote.com/products/playlistplus/ [cited 10-04-2011].

[Gru93] Thomas R. Gruber. Toward Principles for the Design of Ontologies Used for Knowledge Sharing. *Toward Principles for the Design of Ontologies Used for Knowledge Sharing*, 43:907–928, 1993.

[Gru09] Thomas R. Gruber. *Encyclopedia of Database Systems*, chapter Ontology. Springer-Verlag, 2009.

[Hal06] Harry Halpin. Representationalism: The Hard Problem for Artificial Life. In *Proceedings of the Tenth Conference on Artificial Life (ALife X)*, 2006.

[Hal09] Harry Halpin. *Sense and Reference on the Web*. Phd thesis, University of Edinburgh, School of Informatics, Institute for Communicating and Collaborative Systems, Edinburgh, UK, 2009.

[Har10] Toby Harris. BBC Stories [online]. September 2010. Available from: http://tobyz.net/tobyzstuff/projects/bbcstories [cited 21-04-2011].

[Hau10a] Michael Hausenblas. Dataset Dynamics (dady) vocabulary [online]. January 2010. Available from: http://vocab.deri.ie/dady [cited 23-02-2011].

[Hau10b] Michael Hausenblas. Linked Data Specifications [online]. November 2010. Available from: http://linkeddata-specs.info/ [cited 16-02-2011].

[Hau10c] Michael Hausenblas. Quality Web Data [online]. 2010. Available from: http://qualitywebdata.org/ [cited 16-02-2011].

[Haw10] Sandro Hawke. RIF In RDF [online]. June 2010. Available from: http://www.w3.org/TR/rif-in-rdf/ [cited 08-01-2011].

[Hay04] Patrick Hayes, editor. RDF Semantics [online]. February 2004. Available from: http://www.w3.org/TR/rdf-mt/ [cited 10-07-2010].

[Hay08] Pat Hayes. Re: Uniform access to descriptions, April 2008. Available from: http://lists.w3.org/Archives/Public/www-tag/2008Apr/0139.html [cited 10-01-2011].

[HB11] Tom Heath and Chris Bizer. Linked Data: Evolving the Web into a Global Data Space [online]. March 2011. Available from: http://linkeddatabook.com/editions/1.0/ [cited 02-03-2011].

[Hec05] Dominikus Heckmann. *Ubiquitous User Modeling*. PhD thesis, Saarland University, November 2005.

[Hep10] Martin Hepp. GoodRelations Language Reference [online]. September 2010. Available from: http://purl.org/goodrelations/v1 [cited 10-11-2010].

[Her08] Oscar Celma Herrada. *Music Recommendation and Discovery in the Long Tail*. PhD thesis, Department of Technology, Universitat Pompeu Fabra, Barcelona, Spain, 2008.

[Hew09] Hewlett-Packard Development Company. Jena Relational Database backend [online]. 2009. Available from: http://jena.sourceforge.net/DB/layout.html [cited 12-01-2011].

[HH08] Patrick J. Hayes and Harry Halpin. In Defense of Ambiguity [online]. 2008. Available from: http://www.ibiblio.org/hhalpin/homepage/publications/indefenseofambiguity.html [cited 21-03-2011].

[Hic10] Ian Hickson. HTML5 [online]. October 2010. W3C Working Draft. Available from: http://www.w3.org/TR/html5/ [cited 07-01-2011].

[Hob06] Jerry R. Hobbs and Feng Pan, editors. Time Ontology in OWL [online]. September 2006. Available from: http://www.w3.org/TR/owl-time [cited 2010-11-18].

[Hol05] Peter Holtz. *Was ist Musik? Subjektive Theorien Musikschaffender Künstler*. PhD thesis, Philosophischen Fakultät der Friedrich-Alexander-Universität Erlangen-Nürnberg, 2005.

[HS10] Steve Harris and Andy Seaborne. SPARQL 1.1 Query Language [online]. October 2010. Available from: http://www.w3.org/TR/sparql11-query/ [cited 17-02-2011].

[HS11] Michael Hausenblas and Henry Story. WebID Incubator Group Charter [online]. February 2011. Available from: http://www.w3.org/2005/Incubator/webid/charter [cited 22-02-2011].

[HSB+05] Dominik Heckmann, Tim Schwartz, Boris Brandherm, Michael Schmitz, and Margeritta von Wilamowitz-Moellendorff. Gumo - The General User Model Ontology. In *User Modeling 2005*, volume 3538, pages 428–432. Springer Berlin / Heidelberg, 2005.

[Ide11a] Kingsley Uyi Idehen. Twitter / @kidehen [online]. February 2011. Available from: http://twitter.com/kidehen/status/34733906351628288 [cited 22-04-2011].

[Ide11b] Kingsley Uyi Idehen. Twitter / @kidehen [online]. February 2011. Available from: http://twitter.com/kidehen/status/34732592959852544 [cited 22-04-2011].

[Ide11c] Kingsley Uyi Idehen. Twitter / @kidehen [online]. February 2011. Available from: http://twitter.com/kidehen/status/34733174512685056 [cited 22-04-2011].

[Ide11d] Kingsley Uyi Idehen. What is Linked Data, really? [online]. February 2011. Available from: http://www.openlinksw.com/dataspace/kidehen@openlinksw.com/weblog/kidehen@openlinksw.com%27s%20BLOG%20%5B127%5D/1639 [cited 15-02-2011].

[IJBV11] Robert Isele, Anja Jentzsch, Chris Bizer, and Julius Volz. Silk - A Link Discovery Framework for the Web of Data [online]. January 2011. Available from: http://www4.wiwiss.fu-berlin.de/bizer/silk/ [cited 20-04-2011].

[Imb10] Antoine Imbert. MongoDB-RDF [online]. December 2010. Available from: https://github.com/ant0ine/MongoDB-RDF [cited 12-01-2011].

[Ink10] Toby Inkster. Web page counter example in PHP [online]. June 2010. Available from: http://buzzword.org.uk/2010/counter/ [cited 10-08-2010].

[ITU11] ITU. X.509 [online]. February 2011. Available from: http://www.itu.int/rec/T-REC-X.509 [cited 22-02-2011].

[Jac10a] Kurt Jacobson. DBTune - Artists - Last.fm [online]. 2010. Available from: http://dbtune.org/artists/last-fm/ [cited 13-04-2011].

[Jac10b] Kurt Jacobson. Echonest Artist Similarity RDF Service [online]. 2010. Available from: http://dbtune.org/artists/echonest/ [cited 14-04-2011].

[Jac10c] Kurt Jacobson. LinkedBrainz [online]. September 2010. Available from: http://wiki.musicbrainz.org/LinkedBrainz [cited 12-04-2011].

[Jac10d] Kurt Jacobson. The Media Format Ontology [online]. October 2010. Available from: http://purl.org/ontology/mfo/ [cited 19-04-2011].

[Jon10] Richard Jones. Playdar - Music Content Resolver [online]. May 2010. Available from: http://www.playdar.org [cited 23-04-2011].

[JRG10] Kurt Jacobson, Yves Raimond, and Thomas Gängler. The Similarity Ontology [online]. October 2010. Available from: http://purl.org/ontology/similarity/ [cited 10-09-2010].

[JRH10] Michael O. Jewell, Paul Rissen, and Toby Harris. Stories Ontology [online]. September 2010. Available from: http://www.contextus.net/stories/ [cited 21-04-2011].

[JW04] Ian Jacobs and Norman Walsh. Architecture of the World Wide Web, Volume One [online]. December 2004. Available from: http://www.w3.org/TR/webarch/ [cited 10-01-2011].

[Kay07] Michael Kay. XSL Transformations (XSLT) Version 2.0 [online]. January 2007. Available from: http://www.w3.org/TR/xslt20/ [cited 11-01-2011].

[KB10] Michael Kifer and Harold Boley. RIF Overview [online]. June 2010. Available from: http://www.w3.org/TR/rif-overview/ [cited 08-01-2011].

[KC04] Graham Klyne and Jeremy J. Carroll. Resource Description Framework (RDF): Concepts and Abstract Syntax [online]. February 2004. Available from: http://www.w3.org/TR/rdf-concepts/ [cited 07-01-2011].

[KC08] Ian Knopke and Chris Cannam. *Sonic Annotator*, 2008. Available from: http://www.omras2.org/files/runner.pdf.

[Kly00] Graham Klyne. Contexts for RDF Information Modelling [online]. October 2000. Available from: http://www.ninebynine.org/RDFNotes/RDFContexts.html [cited 14-01-2011].

[Knu09] Holger Knublauch. SPIN - SPARQL Inferencing Notation [online]. October 2009. Available from: http://spinrdf.org/ [cited 18-02-2011].

[Koi05] Marja-Riitta Koivunen. Annotea Project [online]. October 2005. Available from: http://www.w3.org/2001/Annotea/ [cited 10-08-2010].

[Las11] Last.fm. Scrobbling 2.0 Documentation [online]. 2011. Available from: http://last.fm/api/scrobbling [cited 13-04-2011].

[LC07] Paul Lamere and Oscar Celma. Music Recommendation Tutorial. In *8th International Conference on Music Information Retrieval*, Vienna, Austria, September 2007.

[Les06] Micheline Lesaffre. *Music Information Retrieval - Conceptual Framework, Annotation and User Behaviour*. PhD thesis, Faculty of Arts and Philosophy, Department of Art, Music and Theatre Sciences Ghent University, 2006.

[LG11] Martin Lévesque and Thomas Gängler. Remote Sitemap Generator [online]. January 2011. Available from: http://sourceforge.net/projects/rsitemapgen/ [cited 22-04-2011].

[Lic60] J. C. R. Licklider. Man-Computer Symbiosis. *IRE Transactions on Human Factors in Electronics*, HFE-1:4–11, March 1960. Available from: http://groups.csail.mit.edu/medg/people/psz/Licklider.html.

[Lic63] J. C. R. Licklider. Memorandum For Members and Affiliates of the Intergalactic Computer Network, April 1963. Available from: http://www.kurzweilai.net/memorandum-for-members-and-affiliates-of-the-intergalactic-computer-network.

[Lie07] Anthony Liekens. Last.fm scripts - user tag clouds [online]. 2007. Available from: http://anthony.liekens.net/pub/scripts/last.fm/ [cited 14-04-2011].

[Lil08] Anita Shen Lillie. MusicBox: Navigating the space of your music. Master's thesis, School of Architecture and Planning, Massachusetts Institute of Technology, September 2008.

[Lou02] Michael J. Loux. *Metaphysics: A Contemporary Introduction*. Routledge contemporary introductions to philosophy. Routledge, London, UK, 2 edition, 2002.

[LVH05] Tuck W Leong, Frank Vetere, and Steve Howard. The serendipity shuffle. In *Proceedings of the 17th Australia conference on Computer-Human Interaction*, pages 1–4, Narrabundah, Australia, 2005. Computer-Human Interaction Special Interest Group (CHISIG) of Australia.

[Maz06] Stefano Mazzocchi. jpeg2rdf [online]. 2006. Available from: http://simile.mit.edu/repository/RDFizers/jpeg2rdf/.

[MB09] Alistair Miles and Sean Bechhofer. SKOS Simple Knowledge Organization System Reference [online]. August 2009. Available from: http://www.w3.org/TR/skos-reference/ [cited 01-03-2011].

[MBRI09] Libby Miller, Dan Brickley, Dave Reynolds, and Toby Inkster. Weighted Interest Vocabulary [online]. December 2009. Available from: http://xmlns.notu.be/wi/ [cited 10-12-2010].

[McC10] Shane McCarron. XHTML+RDFa 1.1 [online]. November 2010. Available from: http://www.w3.org/TR/xhtml-rdfa/ [cited 10-01-2011].

[Men09] Noah Mendelsohn. The Self-Describing Web [online]. February 2009. Available from: http://www.w3.org/2001/tag/doc/selfDescribingDocuments.html [cited 28-02-2011].

[Met10] Metadata Provenance Task-Group. vocabularies [dc-provenance] [online]. November 2010. Available from: http://wiki.bib.uni-mannheim.de/dc-provenance/doku.php?id=vocabularies [cited 22-04-2011].

[Mil10] Dan Brickley Libby Miller. FOAF Vocabulary Specification [online]. August 2010. Available from: http://xmlns.com/foaf/spec/ [cited 01-04-2011].

[MK03] Robert MacGregor and In-Young Ko. Representing Contextualized Data using Semantic Web Tools An Example of Contextualized Data. In *In Practical and Scalable Semantic Systems (workshop at 2nd ISWC)*, 2003.

[MK06] Vassil Momtchev and Atanas Kiryakov. ORDI : Ontology Representation and Data Integration Framework. Technical report, Ontotext Lab, Sirma Group Corp., 2006.

[MKC06] Meinard Müller, Frank Kurth, and Michael Clausen. Aktuelle Aspekte des Music Information Retrieval. *Datenbank-Spektrum*, 18/2006:24–32, 2006.

[MLL05] Alexandre Monnin, Freddy Limpens, and David Laniado. Nice Tag Ontology [online]. September 2005. Available from: http://ns.inria.fr/nicetag/2009/09/25/voc.html [cited 10-08-2010].

[MM04] Frank Manola and Eric Miller. RDF Primer [online]. February 2004. Available from: http://www.w3.org/TR/rdf-primer/ [cited 14-01-2011].

[Mot09] Boris Motik, Peter F. Patel-Schneider, and Bernardo Cuenca Grau, editors. OWL 2 Web Ontology Language Direct Semantics [online]. October 2009. Available from: http://www.w3.org/TR/owl2-direct-semantics/ [cited 07-01-2011].

[MSS03] B. S. Manjunath, Phillippe Salembier, and Thomas Sikora. *Introduction to MPEG-7*. Wiley, 2003.

[MSSvE07] Antoni Mylka, Leo Sauermann, Michael Sintek, and Ludger van Elst. NEPOMUK ID3 Ontology [online]. 2007. Available from: http://www.semanticdesktop.org/ontologies/nid3/ [cited 21-04-2011].

[New05] Richard Newman. Tag ontology design [online]. December 2005. Available from: http://www.holygoat.co.uk/projects/tags/ [cited 10-08-2010].

[Nil00] M. Nilsson. The audio/mpeg Media Type [online]. November 2000. Available from: http://tools.ietf.org/html/rfc3003 [cited 10-03-2011].

[Not10] M. Nottingham. Web Linking [online]. October 2010. Available from: http://tools.ietf.org/html/rfc5988 [cited 16-02-2011].

[Now09] Benjamin Nowack. The Semantic Web - Not a piece of cake... [online]. July 2009. Available from: http://bnode.org/blog/2009/07/08/the-semantic-web-not-a-piece-of-cake [cited 10-01-2011].

[Noy06] Natasha Noy and Alan Rector, editors. Defining N-ary Relations on the Semantic Web [online]. April 2006. Available from: http://www.w3.org/TR/swbp-n-aryRelations/ [cited 2010-11-17].

[NS05] M. Nottingham and R. Sayre. The Atom Syndication Format [online]. December 2005. Available from: http://tools.ietf.org/html/rfc4287 [cited 22-02-2011].

[NSMH04] Claudia Niederée, Avaré Stewart, Bhaskar Mehta, and Matthias Hemmje. A Multi-Dimensional, Unified User Model for Cross-System Personalization. In *Procceedings of the AVI 2004 Workshop On Environments For Personalized Information Access*, 2004.

[Ogb10] Chimezie Ogbuji. SPARQL 1.1 Uniform HTTP Protocol for Managing RDF Graphs [online]. October 2010. Available from: http://www.w3.org/TR/sparql11-http-rdf-update/ [cited 17-02-2011].

[Oli11] Silver Oliver. Twitter / @silveroliver [online]. February 2011. Available from: http://twitter.com/#!/silveroliver/status/40712231444287488 [cited 21-04-2011].

[O'N10] Dan O'Neill. Home - ID3.org [online]. November 2010. Available from: http://www.id3.org/ [cited 27-02-2011].

[Ont11] Ontotext. OWLIM Semantic Repository [online]. 2011. Available from: http://www.ontotext.com/owlim/index.html [cited 15-01-2011].

[Ope09] OpenLink Software. Virtuoso Server Extension Interface (VSEI) (C Interface) [online]. 2009. Available from: http://docs.openlinksw.com/virtuoso/cinterface.html [cited 20-04-2011].

[Ope10a] OpenLink Software. Virtuoso 6.2 – New Features [online]. 2010. Available from: http://virtuoso.openlinksw.com/whats-new/ [cited 22-01-2011].

[Ope10b] OpenLink Software. Virtuoso Universal Server [online]. 2010. Available from: http://virtuoso.openlinksw.com/ [cited 15-01-2011].

[Ope11] OpenID Foundation. OpenID Foundation website [online]. 2011. Available from: http://openid.net/ [cited 22-02-2011].

[Pac05] Francois Pachet. Knowledge Management and Musical Metadata. In *Enzyclopedia of Knowledge Management*, pages 672–677. Idea Group, September 2005.

[Pas10a] Alexandre Passant. dbrec - Music Recommendations Using DBpedia. In *Proceedings of the 9th International Semantic Web Conference (ISWC 2010)*, pages 209–224. Springer, 2010.

[Pas10b] Alexandre Passant. You need it, you want it, you get it ;) - The Recommendation Ontology [online]. July 2010. Available from: `http://groups.google.com/group/music-ontology-specification-group/msg/a7763c82c0457c57` [cited 22-04-2011].

[PCI10] Jirí Procházka, Richard Cyganiak, and Toby Inkster. Property Reification RDF Vocabulary [online]. June 2010. Available from: `http://esw.w3.org/PropertyReificationVocabulary` [cited 10-15-2010].

[PCIG10] Jirí Procházka, Richard Cyganiak, Toby Inkster, and Thomas Gängler. The Property Reification Vocabulary Specification [online]. September 2010. Available from: `http://purl.org/ontology/prv/propertyreification.html` [cited 10-15-2010].

[PH10] Niko Popitsch and Bernhard Haslhofer. DSNotify Eventsets: A vocabulary for change events in linked data sources [online]. May 2010. Available from: `http://dsnotify.org/vocab/eventset/v0.1/dsnotify-eventset.html` [cited 23-02-2011].

[PHG+08] Adrian Paschke, David Hirtle, Allen Ginsberg, Paula-Lavinia Patranjan, and Frank McCabe. RIF Use Cases and Requirements [online]. December 2008. Available from: `RIFUseCasesandRequirements` [cited 10-01-2011].

[PHMR10] N. Popitsch, B. Haslhofer, and E. Momeni Roochi. An Evaluation Approach for Dynamics-Aware Applications using Linked Data. In *9th International Workshop on Web Semantics (WebS '10)*, 2010.

[PKB10] Alexandre Passant, Sheila Kinsella, and John Breslin. The Interest Mining Ontology [online]. January 2010. Available from: `http://rdfs.org/imo/ns` [cited 10-12-2010].

[PM10] Alexandre Passant and Pablo Mendes. sparqlpush [online]. 2010. Available from: `http://code.google.com/p/sparqlpush/` [cited 01-03-2011].

[Pru10] Eric Prud'hommeaux. SPARQL 1.1 Federation Extensions [online]. June 2010. Available from: `http://www.w3.org/TR/sparql11-federated-query/` [cited 17-02-2011].

[PS10] Alexandre Passant and Balthasar A.C. Schopman. User (weighted) Interests Ontology [online]. May 2010. Available from: `http://vocamp.org/wiki/HypiosVoCampParisMay2010#User_.28weighted.29_Interests_Ontology` [cited 10-12-2010].

[RA07a] Yves Raimond and Samer A. Abdallah. The Event Ontology [online]. October 2007. Available from: `http://motools.sf.net/event/event.html` [cited 2010-11-16].

[RA07b] Yves Raimond and Samer A. Abdallah. The Timeline Ontology [online]. October 2007. Available from: `http://motools.sourceforge.net/timeline/timeline.html` [cited 2010-11-18].

[Rai06] Yves Raimond. AudioScrobbler RDF Service [online]. 2006. Available from: `http://dbtune.org/last-fm/` [cited 11-01-2011].

[Rai08a] Yves Raimond. *A Distributed Music Information System*. PhD thesis, Queen Mary, University of London, November 2008.

[Rai08b] Yves Raimond. Audio Features Ontology Specification [online]. June 2008. Available from: http://purl.org/ontology/af/ [cited 04-04-2011].

[Rai08c] Yves Raimond. Echonest Analyze XML to Music Ontology RDF [online]. July 2008. Available from: http://dbtune.org/echonest/ [cited 11-01-2011].

[Rai08d] Yves Raimond. Playcount Ontology [online]. June 2008. Available from: http://motools.svn.sourceforge.net/viewvc/motools/dbtune/playcount/playcount.n3 [cited 10-08-2010].

[RASG07] Yves Raimond, Samer Abdallah, Mark Sandler, and Frederick Giasson. The Music Ontology. In *Proceedings of the 8th International Conference on Music Information Retrieval*, pages 417–422, Vienna, Austria, September 2007.

[RBH10] Jonathan Rees, David Booth, and Michael Hausenblas. DRAFT AWWSW Progress Report [online]. May 2010. Available from: http://www.w3.org/2001/tag/awwsw/web-semantics-20100525.html [cited 10-01-2011].

[Rey10] Dave Reynolds. OWL 2 RL in RIF [online]. June 2010. Available from: http://www.w3.org/TR/rif-owl-rl/ [cited 08-01-2011].

[RG10a] Yves Raimond and Thomas Gängler. motools/lastfm [online]. August 2010. Available from: http://motools.svn.sourceforge.net/viewvc/motools/lastfm/ [cited 11-01-2011].

[RG10b] Yves Raimond and Thomas Gängler. motools/lastfm/branches/lastfm20 [online]. May 2010. Available from: http://motools.svn.sourceforge.net/viewvc/motools/lastfm/branches/lastfm20/ [cited 20-04-2011].

[RG10c] Yves Raimond and Thomas Gängler. motools/lastfm/branches/lastfm25 [online]. June 2010. Available from: http://motools.svn.sourceforge.net/viewvc/motools/lastfm/branches/lastfm25/ [cited 20-04-2011].

[RG10d] Yves Raimond and Thomas Gängler. Music Ontology 2.1 proposal [online]. November 2010. Available from: http://wiki.musicontology.com/index.php/MOProposal21 [cited 2010-11-15].

[RGG10] Yves Raimond, Frederick Giasson, and Thomas Gängler. Class Schemas [online]. November 2010. Available from: http://wiki.musicontology.com/index.php/Classes_Schemas [cited 04-04-2011].

[RGJ+10] Yves Raimond, Frédérick Giasson, Kurt Jacobson, George Fazekas, Thomas Gängler, and Simon Reinhardt. Music Ontology Specification [online]. November 2010. Available from: http://www.musicontology.com/ [cited 27-02-2010].

[RN10] Marko A. Rodriguez and Peter Neubauer. Constructions from Dots and Lines. *Bulletin of American Society for Information Science & Technology*, August/September:n/a, 2010.

[Roh10] Kurt Rohloff. SHARD [online]. 2010. Available from: http://www.dist-systems.bbn.com/people/krohloff/shard.shtml [cited 15-01-2011].

[Rov11] Rovi Corporation. Rovi Music - All Music Guide Data [online]. March 2011. Available from: http://www.rovicorp.com/products/online_stores_portals/insight_osp/rovi_music.htm [cited 10-04-2011].

[Row10] Matthew Rowe. A Proposed 5th Rule/Principle for Linked Data: Push [online]. June 2010. Available from: http://www.matthew-rowe.com/?q=node/37 [cited 15-02-2011].

[RSG10] Yves Raimond, Mats Skillingstad, and Thomas Gängler. The Counter Ontology Specification [online]. September 2010. Available from: http://purl.org/ontology/co/counterontology.html [cited 10-08-2010].

[RWO11] Jem Rayfield, Paul Wilton, and Silver Oliver. Sport Ontology [online]. February 2011. Available from: http://www.bbc.co.uk/ontologies/sport/ [cited 21-04-2011].

[San10] Robert Sanderson and Herbert Van de Sompel, editors. Open Annotation: Alpha2 Data Model Guide [online]. Juni 2010. Available from: http://annotation.lanl.gov/alpha2/ [cited 10-08-2010].

[Sch09] Michael Schneider, editor. OWL 2 Web Ontology Language RDF-Based Semantics [online]. October 2009. Available from: http://www.w3.org/TR/owl2-rdf-based-semantics/ [cited 08-01-2011].

[Sea10] Andy Seaborne. Jena, a Semantic Web Framework [online]. November 2010. Available from: http://wiki.apache.org/incubator/JenaProposal [cited 12-01-2011].

[Ser08] Xavier Serra. Sound and Music Description. Presentation at Music Technology Group, Universitat Pompeu Fabra, Barcelona, 2008.

[sev05] several. RecordDescription [online]. January 2005. Available from: http://esw.w3.org/RecordDescription [cited 14-01-2011].

[sev07a] several. CurriedFunction [online]. December 2007. Available from: http://esw.w3.org/CurriedFunction [cited 14-01-2011].

[sev07b] several. FollowYourNose [online]. August 2007. Available from: http://esw.w3.org/FollowYourNose [cited 23-01-2011].

[sev07c] several. OnTheFlyProperties [online]. December 2007. Available from: http://esw.w3.org/OnTheFlyProperties [cited 14-01-2011].

[sev08] several. The Tabulator Extension [online]. 2008. Available from: http://dig.csail.mit.edu/2007/tab/ [cited 20-04-2011].

[sev09a] several. Automated Generation of RDF Views over Relational Data Sources with Virtuoso [online]. 2009. Available from: http://virtuoso.openlinksw.com/dataspace/dav/wiki/Main/VOSSQL2RDF [cited 22-01-2011].

[sev09b] several. AwwswVocabulary [online]. April 2009. Available from: http://esw.w3.org/AwwswVocabulary [cited 10-01-2011].

[sev09c] several. Category:Programming Environment [online]. 2009. Available from: http://www.w3.org/2001/sw/wiki/Category:Programming_Environment [cited 22-01-2011].

[sev09d] several. Category:Triple Store [online]. 2009. Available from: http://www.w3.org/2001/sw/wiki/Category:Triple_Store [cited 22-01-2011].

[sev09e] several. Virtuoso Sponger [online]. 2009. Available from: http://virtuoso.openlinksw.com/dataspace/dav/wiki/Main/VirtSpongerWhitePaper [cited 21-04-2011].

[sev10a] several. Category:Relationship Type [online]. September 2010. Available from: http://wiki.musicbrainz.org/Category:Relationship_Type [cited 12-04-2011].

[sev10b] several. ConverterToRdf [online]. November 2010. Available from: http://esw.w3.org/ConverterToRdf [cited 11-01-2011].

[sev10c] several. Deploying Linked Data [online]. 2010. Available from: http://virtuoso.openlinksw.com/dataspace/dav/wiki/Main/VirtDeployingLinkedDataGuide [cited 20-04-2011].

[sev10d] several. Example of Linked Data Usage of PubSubHubbub Implementation [online]. 2010. Available from: http://ods.openlinksw.com/wiki/ODS/VirtPubSubHub [cited 20-04-2011].

[sev10e] several. How can I associate related rules to an ontology/RDF graph? [online]. November 2010. Available from: http://www.semanticoverflow.com/questions/2293/how-can-i-associate-related-rules-to-an-ontology-rdf-graph [cited 16-02-2011].

[sev10f] several. MusicBrainzIdentifier [online]. June 2010. Available from: http://wiki.musicbrainz.org/MBID [cited 10-04-2011].

[sev10g] several. OAuth Community Site [online]. 2010. Available from: http://oauth.net/ [cited 22-02-2011].

[sev10h] several. OpenLink-supplied Virtuoso Sponger Cartridges [online]. 2010. Available from: http://virtuoso.openlinksw.com/dataspace/dav/wiki/Main/VirtSpongerCartridgeSupportedDataSources [cited 11-01-2011].

[sev10i] several. SpecGen [online]. July 2010. Available from: http://forge.morfeo-project.org/wiki_en/index.php/SpecGen [cited 13-02-2011].

[sev10j] several. Virtuoso Open-Source Edition [online]. 2010. Available from: http://virtuoso.openlinksw.com/dataspace/dav/wiki/Main/ [cited 22-01-2011].

[sev10k] several. Virtuoso Sponger [online]. 2010. Available from: http://virtuoso.openlinksw.com/dataspace/dav/wiki/Main/VirtSponger [cited 11-01-2011].

[sev10l] several. WebAccessControl [online]. November 2010. Available from: http://www.w3.org/wiki/WebAccessControl [cited 22-02-2011].

[sev10m] several. WebID Protocol ACLs Tutorial [online]. 2010. Available from: http://virtuoso.openlinksw.com/dataspace/dav/wiki/Main/VirtAuthFOAFSSLACL.

[sev11a] several. ActivityStreams - an evolving format for syndicating social activities around the web [online]. 2011. Available from: http://activitystrea.ms/ [cited 17-04-2011].

[sev11b] several. DatasetDynamics [online]. February 2011. Available from: http://www.w3.org/wiki/DatasetDynamics [cited 23-02-2011].

[sev11c] several. Extension Modules [online]. April 2011. Available from: http://wiki.musicontology.com/index.php/Extension_Modules [cited 04-04-2011].

[sev11d] several. Long Tail [online]. April 2011. Available from: http://en.wikipedia.org/wiki/Long_Tail [cited 20-04-2011].

[sev11e] several. Music Ontology Wiki [online]. April 2011. Available from: http://wiki.musicontology.com/index.php/Main_Page [cited 04-04-2011].

[sev11f] several. Next Generation Schema [online]. March 2011. Available from: http://wiki.musicbrainz.org/NGS [cited 01-04-2011].

[sev11g] several. Publishing Rulesets in the Wild [online]. February 2011. Available from: http://www.semanticoverflow.com/questions/3130/publishing-rulesets-in-the-wild [cited 16-02-2011].

[sev11h] several. PushBackDataToLegacySources [online]. February 2011. Available from: http://www.w3.org/wiki/PushBackDataToLegacySources [cited 23-02-2011].

[sev11i] several. TF-JSON [online]. March 2011. Available from: http://www.w3.org/2011/rdf-wg/wiki/TF-JSON [cited 09-03-2011].

[sev11j] several. VorbisComment - XiphWiki [online]. January 2011. Available from: http://wiki.xiph.org/index.php/VorbisComment [cited 27-02-2011].

[sev11k] seversl. Information [online]. April 2011. Available from: http://en.wikipedia.org/wiki/Information [cited 18-04-2011].

[SGP10] Simon Schenk, Paul Gearon, and Alexandre Passant. SPARQL 1.1 Update [online]. October 2010. Available from: http://www.w3.org/TR/sparql11-update/ [cited 17-02-2011].

[Shi11] Joshua Shinavier. Graph databases and recommendation systems [online]. April 2011. Available from: http://groups.google.com/group/gremlin-users/msg/cb3de23778bff439 [cited 22-04-2011].

[SHJJ09] Henry Story, Bruno Harbulot, Ian Jacobi, and Mike Jones. FOAF+SSL: RESTful Authentication for the Social Web. In *6th European Semantic Web Conference*, June 2009.

[SIS+11] Web Identification and Discovery [online]. February 2011. Available from: http://www.w3.org/2005/Incubator/webid/spec/ [cited 22-02-2011].

[SKR06] Raj Sharman, Rajiv Kishore, and Ram Ramesh, editors. *Ontologies: A Handbook of Principles, Concepts and Applications in Information Systems*, volume 14 of *Integrated Series in Information Systems*. Springer New York, Secaucus, NJ, USA, 2006.

[Sos08] Sergey Sosnovsky. Ontological technologies for user modelling. working paper, March 2008.

[Sow91] John Sowa, editor. *Principles of Semantic Networks*. Morgan Kaufmann, 1991.

[SSvEH07] Simon Scerri, Michael Sintek, Ludger van Elst, and Siegfried Handschuh. NEPOMUK Annotation Ontology Specification [online]. August 2007. Available from: http://www.semanticdesktop.org/ontologies/nao/ [cited 10-08-2010].

[Sta11] Stanford Center for Biomedical Informatics Research. The Protégé Ontology Editor and Knowledge Acquisition System [online]. 2011. Available from: http://protege.stanford.edu/ [cited 02-03-2011].

[Sti05] Patrick Stickler. CBD - Concise Bounded Description [online]. June 2005. Available from: http://www.w3.org/Submission/CBD/ [cited 08-01-2011].

[SYS10] SYSTAP. Bigdata [online]. 2010. Available from: http://www.systap.com/bigdata.htm [cited 15-01-2011].

[TD09] Sam Tunnicliffe and Ian Davis. Changeset Vocabulary [online]. May 2009. Available from: http://vocab.org/changeset/schema.html [cited 23-02-2011].

[Tea11] Team Tomahawk. Tomahawk - The Social Music Player [online]. 2011. Available from: http://tomahawk-player.org/ [cited 24-04-2011].

[The11a] The Echo Nest. Artist API Methods [online]. 2011. Available from: http://developer.echonest.com/docs/v4/artist.html [cited 14-04-2011].

[The11b] The Echo Nest. Catalog API Methods [online]. 2011. Available from: http://developer.echonest.com/docs/v4/catalog.html [cited 14-04-2011].

[The11c] The Echo Nest. Echo Nest API Overview [online]. 2011. Available from: http://developer.echonest.com/docs/v4/index.html [cited 14-04-2011].

[The11d] The Echo Nest. Playlist API Methods [online]. 2011. Available from: http://developer.echonest.com/docs/v4/playlist.html [cited 14-04-2011].

[The11e] The Echo Nest. Song API Methods [online]. 2011. Available from: http://developer.echonest.com/docs/v4/song.html [cited 14-04-2011].

[The11f] The Echo Nest. Track API Methods [online]. 2011. Available from: http://developer.echonest.com/docs/v4/track.html [cited 14-04-2011].

[Tol04] Karsten Tolle. Understanding data by their context using RDF. In *International Conference on Advances in Intelligent Systems 2004*, Luxembourg, November 2004. IEEE.

[Top11] TopQuadrant. TopQuadrant | Products | TopBraid Composer [online]. 2011. Available from: http://www.topquadrant.com/products/TB_Composer.html [cited 02-03-2011].

[Vat11] Bernard Vatant. Twitter / @hubject [online]. January 2011. Available from: http://twitter.com/hubject/status/27748791150968832 [cited 22-04-2011].

[W3C09] W3C OWL Working Group. OWL 2 Web Ontology Language Document Overview [online]. October 2009. Available from: http://www.w3.org/TR/owl2-overview/ [cited 07-01-2011].

[Wan07] Xiaoshu Wang. URI Identity and Web Architecture Revisited [online]. November 2007. Available from: http://dfdf.inesc-id.pt/tr/web-arch [cited 10-01-2011].

[WAO08] Xiaoshu Wang, Jonas S. Almeida, and Arlindo M.L. Oliveira. Ontology of Ontology [online]. November 2008. Available from: http://dfdf.inesc-id.pt/ont/o3 [cited 06-01-2011].

[Whi10] Brian Whitman. The Echo Nest Musical Fingerprint (ENMFP) [online]. April 2010. Available from: http://blog.echonest.com/post/545323349/the-echo-nest-musical-fingerprint-enmfp [cited 14-04-2011].

[Wil10] Gregory Todd Williams. SPARQL 1.1 Service Description [online]. October 2010. Available from: http://www.w3.org/TR/sparql11-service-description/ [cited 17-02-2011].

[WS96] Richard Y. Wang and Diane M. Strong. Beyond accuracy : What data quality means to data consumers. *Journal of Management Information Systems*, 12(4):5–34, 1996. Available from: http://w3.cyu.edu.tw/ccwei/PAPER/ERP/dataquality%28JMIS%29.pdf.

[ZWDH10] Yi Zeng, Yan Wang, Danica Damljanovic, and Zhisheng Huang. E-foaf:interest Vocabulary Specification [online]. April 2010. Available from: http://wiki.larkc.eu/e-foaf:interest [cited 10-12-2010].

[ZX10] Hai Zhuge and Bei Xu. Basic operations, completeness and dynamicity of cyber physical socio semantic link network CPSocio-SLN. *Concurrency and Computation: Practice and Experience*, 1532-0634:n/a, 2010.