Converting and Integrating Vocabularies for the Semantic Web

Mark van Assem

SIKS Dissertation Series No. 2010-40

The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

Promotiecommissie:
prof.dr. A.Th. Schreiber (promotor)
dr. J.R. van Ossenbruggen (copromotor)
dr. T. Baker
prof.dr. L. Hardman
prof.dr. E. Hyvönen
prof.dr.ir. J.L. Top
prof.dr. P.Th.J.M. Vossen

ISBN 978-90-8659-483-2

Cover photo by Jasper Groen, inspired by Rembrandt's Anatomy Lesson of Dr. Nicolaes Tulp

VRIJE UNIVERSITEIT

Converting and Integrating Vocabularies for the Semantic Web

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad Doctor aan
de Vrije Universiteit Amsterdam,
op gezag van de rector magnificus
prof.dr. L.M. Bouter,
in het openbaar te verdedigen
ten overstaan van de promotiecommissie
van de faculteit der Exacte Wetenschappen
op vrijdag 1 oktober 2010 om 13.45 uur
in de aula van de universiteit,
De Boelelaan 1105

door
Marcus Franciscus Johannes van Assem
geboren te Amersfoort

promotor: prof.dr. A.Th. Schreiber
copromotor: dr. J.R. van Ossenbruggen

Contents

Preface

First of all I'd like to thank my promotor Guus Schreiber. Thank you for the opportunity to work with you and showing me how to keep research insightful and down-to-earth at the same time. I'm also indebted to my co-promotor Jacco van Ossenbruggen. His energetic attitude and support towards the end of the project helped me finish what I started. I am grateful for the critical reading and insightful remarks of my committee, consisting of Tom Baker, Lynda Hardman, Eero Hyvönen, Jan Top and Piek Vossen. The papers on which this thesis are based were made in close cooperation with Aldo Gangemi, Laura Hollink, Véronique Malaisé, Maarten Menken, Alistair Miles, Guus Schreiber, Jan Wielemaker and Bob Wielinga. The communities around the W3C WordNet Working Group and SKOS were also instrumental in shaping my research. Helpful proofreading of some of the chapters was done by Laura Hollink, Michel Klein and Stefan Schlobach. In this place I also like to thank NWO and MultimediaN, for making my work possible through funding of the CHIME and E-Culture projects. Thanks are also due to my current supervisor, prof. Jan Top for his flexibility in allowing me to finish this project.

Jasper, Joost and Laura, my holy trinity, thank you for always listening to my complaints and melting my worries away with loads of fun! I could not have done it without you. Laura, our many conversations about research (and the other stuff in life) helped me focus and learn. Anna, Veronique, Willem, Zhisheng, Wouter, Viktor, Antoine, thanks for sharing rooms, coffee, staplers, research ideas and your friendship. Big up to DJ Alistair, my main man John and dr. Marieke ("the kind that helps people"). Thanks to my many other friends including Sonja, Jan Jaap, Lieke, Evert-Jan and Armanda, Jos and Ilona, Ivo and Nienke, Matthijs and Marianne, Ana and Jirk, Lieke, Vincent, and the good old "VU-clan" for not caring at all about what the Dickens I was up to at work. Katharina, thank you for your support and all we shared.

During my PhD I had the privilege of working in one of the top cities in the world concerning Semantic Web research (formed by the VU and our cousins at CWI and UvA, most of whom have found their way to the VU now). It is hard to convey how truely important the ideas of others are in shaping your own. Because I enjoyed getting to know all of you I will break the tradition of not trying to mention all in fear of forgetting some. Thank you Stefano, Kathrin, Paul, Christophe, Frank, Rinke, Szymon, Spyros, Krystyna, Ruud, Gaston, Annette, Jacopo, Shenghui, Wouter, Henriette, Eyal, Antoine, Ronny, Machiel, George, Zharko, Marjolein, Andreas, Radu, Peter, Heiner, Marta, Jeen, Perry, Annerieke, Joost, Sander, Chris, Hans, Lora, Dick, Anton, Jaap, Marieke, Pieter, Zsofie, Luit, Sybren, Chide, Maksym, Vincent, Balthasar, Yiwen, Bob, Dan, Davide, Riste, Roxane, Lloyd and Vera for making Amsterdam such a great place to do research. Holger, Michel and Stefan, thanks for some fatherly advice here and there! Jan, Lourens, Borys, Edgar, thank you for letting me interrupt your more important activities with questions on programming and other things technical. Elly, Ilse and all people at the helpdesk, your office support has been exemplary. Jos, I was glad to get to know you in the short time you had left.

In the CHIME and E-Culture projects I had the privilege to work with (apart from those already mentioned above) Kateryna Falkovych, Frank Nack, Vadim Chepegin, Lora Aroyo, Paul de Bra, Geert-Jan Houben, Alia Amin and Michiel Hildebrand. Great fun was always to be had with the PhD students at the interesting SIKS courses and the VU PhD course, where I got to know many nice people including Karen and Peter-Paul. I also want to mention Jan Top's research group at Wageningen UR, where I've had the opportunity to investigate new avenues of research with Nicole, Jeen, Mari, and our "data monk" Hajo.

One of the perks of doing a PhD is undoubtedly the traveling. I learned a lot and had a great time at the conferences in Crete, Japan and Montenegro, and thoroughly enjoyed the Semantic Web Summer School near Madrid organized by Enrico Motta and Asun Goméz-Pérez. I also spent a few months with prof. Eero Hyvönen's research group in Helsinki. Contrary to popular belief, Finns are unbelievably sociable. Thanks are due to my three T's, Tomi, Tuukka and Thomas for making it the most interesting and fun summer of my career. It was also nice to meet the many visiting researchers at our own university, such as our Spanish guest Raul and our Finnish guest Tuukka, and introducing them to Amsterdam.

Als laatste wil ik mijn familie danken, in het bijzonder mijn naaste familie. Marloes en Gert-jan, dank je voor zoveel verschillende dingen als helpen klussen in mijn nieuwe appartement, "Kolonisten" spelen wanneer we als familie bijeen komen en gewoon omdat jullie zulke geweldige mensen zijn. Mijn lieve vader en moeder, jullie steun en liefde is de constante van mijn leven in zoveel kleine en grote dingen. Jullie hebben dit allemaal mogelijk gemaakt!

1

Introduction

In this chapter the reader is introduced to the topic of this thesis. The first section deals with the context: metadata collections and the Semantic Web. The sections after that deal with the specific research questions, the research methods and the contributions the thesis makes. This chapter concludes with an overview of the remaining chapters.

1.1 Context

Collections

The basic processes going on in a library have not changed since the days of the library of Alexandria. When a new book is added to the collection, librarians first assign terms to it that describe the subjects the book deals with, a process known as *indexing*. The terms (one or more words) are chosen to describe the subject in a short but intuitive way, e.g. "geology" or "bicycle manufacturing". To be able to look up the books on a certain subject, an *index* is made which lists the books categorized by their subjects. Essentially an index is a mapping from one dimension or attribute of the book to the books themselves. For example, a library also has an *author index* which categorizes the books by their author (usually in alphabetical order). A second process is called *searching*, which has different scenarios. Usually one is searching for either a book already known to exist, or books not known to the searcher that deal with a certain subject. The searcher first has to think of some terms that describe the subject best, and then start looking for it in the index.

Nowadays libraries make use of *vocabularies* to group the terms that are used for indexing and searching. For example, the Dewey Decimal Classification used by many libraries worldwide[1] divides book subjects into ten major categories (e.g. "Technology", "Literature") and further subcategories (e.g. "Engineering", "English & Old English literature"). Not only books but also artworks, scientific articles and TV programs are nowadays indexed with vocabularies. Usually several attributes of an indexed object are described, such as its author, date of creation and location of manufacture. One vocabulary usually focuses on only a few attributes and contains terminology from a specific domain such as art, medicine or geography.

Vocabularies embody some principles that are useful to simplify search. The first principle is to make the set of search terms finite. A list of all subjects that are used for indexing helps

[1] http://www.oclc.org/dewey/resources/summaries/deweysummaries.pdf

the searcher because s/he does not have to guess which subjects are available and which term has been used to signify the subject. A second principle is to group different synonyms together, so that there are several ways to find the subject from the subject list. A third principle is to give a unique identifier to the subject, so that indexing can be done with the identifier instead of with a term. This prevents problems related to homonyms (e.g. "bank" can refer to a riverbank or a financial institute). A fourth principle is subject generalization: a hierarchy is created between the subjects. This helps both in finding a subject (for indexing or for search) as well as in search itself, because the search can be made to include all subjects that are more specific in meaning. Thus a search on the term "sculpture" will also return books indexed with "equestrian statues". This is called *expansion search* or *hierarchical search*.

All vocabularies have in common that they are comprised of *concepts* (subjects) which have been selected by a specific *community*. Each concept (person, object, abstract idea) has one or more terms (a sequence of characters) recognized by that community to stand for the specific concept. Any set of concepts and terms in a particular domain that has been selected for use in indexing and search is called an *indexing* or *search vocabulary* in this thesis (or vocabulary for short). Different types of vocabularies exist which have an increasing complexity, including: subject heading lists, folksonomies, glossaries, classification schemes, terminologies, taxonomies and thesauri (Smith and Welty 2001). Vocabulary types such as thesauri and classification schemes are more complex than glossaries and subject heading lists because they have a hierarchical relation that orders concepts into a tree. An ontology may also be seen as a vocabulary, as it specifies a set of concepts and associated terms.

The community that builds a vocabulary usually works in a specific domain such as art, medicine, geography or audiovisual archives. Well-known examples of domain-specific vocabularies are the Art and Architecture Thesaurus (AAT), IconClass[2] and Medical Subject Headings (MeSH). Some vocabularies are an exception because of their broad scope, such as the Library of Congress Subject Classification (LCSC) and the Dewey Decimal System (DDC). Another interesting example with a broad scope is WordNet, a thesaurus of the English language. The examples mentioned all share a proven, decades long track record as search and indexing vocabularies. AAT and IconClass are used by many museums around the world[3] [4] , MeSH is used to index over 19 million biomedical articles[5] and the DDC is used by many libraries around the world. WordNet is a resource that was originally developed for research in computational linguistics, but now regularly used in information retrieval (Fellbaum 1998).

Vocabularies are used to specify *metadata* of objects such as books, videos and works of art in a collection. The metadata of an object concerns for example its author, its subject, date of creation, art style and object type. These atrributes are called *metadata elements*, and the values of these elements used in describing an object are often taken from vocabularies. The metadata of one object consists of all element-value pairs that apply to it (e.g. title=Anatomy Lesson, author=Rembrandt, ...). These different elements are needed to manage, display and

[2] http://www.iconclass.nl

[3] http://www.cwhonors.org/viewCaseStudy.asp?NominationID=112

[4] http://www.iconclass.nl/about-iconclass/what-is-iconclass

[5] http://www.ncbi.nlm.nih.gov/pubmed/

query collections. A *metadata element set* provides an agreed upon set of metadata elements for the description of an object. An example of a metadata element set is Dublin Core.[6] Different kinds of collections may need a metadata element set that is specialized for the particular type of object. For example, for scientific documents in PubMed[7] we would like to record in which periodical they were published, for paintings in the Rijksmuseum Amsterdam[8] we would like to record the associated art style. Usually, several vocabularies are needed to completely describe one object. For example, a painting's style can be described with AAT, its creator with the Union List of Artist Names (ULAN), and its pictorial content with IconClass. In summary, collections consist of many objects that are indexed with the help of a metadata element set and a set of vocabularies suitable for the specific type of object at hand.

The Semantic Web

The broad context of this thesis is the question how collections can be made available on the Semantic Web. The Semantic Web is an extension of the current Web proposed by Berners-Lee (1999). This extension has the aim of adding data to the Web that is *machine processable* and to *integrate* data from different sources. Both help computers to understand more of the information present on the Web and support users in finding and processing information. Some Semantic Web researchers apply techniques from the field of Knowledge Representation (KR) to make data machine processable. In KR, formalisms are designed that allow representation and automated reasoning. Pieces of knowledge such as "Van Gogh painted the Sunflowers" and "all persons have a name" are translated into logical statements that a computer can manipulate. Some of the statements describe a particular state of the world, such as the first statement. Other statements, such as the latter, encode constraints or reasoning rules of the world.

The efforts of these Semantic Web researchers have resulted in a family of *representation languages* called RDF, RDF(S), OWL Lite, OWL DL and OWL Full (RDF Core Working Group 2004a, **?**, Web Ontology Working Group 2004). The design of these languages is inspired by efforts in the KR community. This family has at its core the idea that knowledge can be represented as nodes connected by binary edges (i.e. a graph). The nodes in the graph can represent anything: documents, people, objects and abstract concepts. The edges are relationships between the nodes. Edges and nodes are identified by globally unique *Uniform Resource Identifiers* (URIs). The graphs can be uploaded to the Web. The nodes and edges can then be accessed through their URI, just as Web pages are accessed through their URL. The languages are grounded in logic, which enables one to make statements like "someone who is a painter is also an artist" and "a person who painted one or more paintings is called a painter". These statements allow a computer to infer new knowledge from the knowledge it already has.

More formally, RDF can be described as "*a version of existential binary relational logic in which relations are first-class entities in the universe of quantification*" (RDF Core Working Group 2004b). RDFS is a "*semantic extension*" of RDF. OWL DL and OWL Lite are based on description

[6] `http://dublincore.org/documents/dces/`
[7] `http://www.ncbi.nlm.nih.gov/pubmed/`
[8] `http://www.rijksmuseum.nl/`

logic (Baader et al. 2003). While RDF places almost no limitations on what can be expressed as long as it is a graph, OWL DL is much stricter. However, a computer can derive many new facts from an OWL DL model, while inferential capabilities of RDF(S) are more limited.

Representing knowledge in this way allows for answering complex queries like "Which paintings did Van Gogh make while he was in France?" that cannot be answered by search engines like Google. Google does not provide an answer, just a list of pages on which answers might be found. For example, none of the top ten results on Google on this query[9] is a list of exactly those paintings he made in France. Most pages are overviews of his life and works, which contain many paintings made in Belgium and the Netherlands. To ask these kinds of questions implies the need to communicate to the computer a specific model of the world in terms of nodes and edges. Concepts like Painter, Country and Painting and relations between concepts like painted (with domain Painter and range Painting) and madeIn (domain Painting and range Country) need to be represented. This allows the expression of statements like <Rembrandt, painted, AnatomyLesson> and <AnatomyLesson, madeIn, theNetherlands>. Existing vocabularies can play a role here because they already provide these concepts.

Integration of several sources is necessary to e.g. answer the earlier query about Van Gogh, because (metadata about) his paintings are distributed over different museums. The problem of data integration can be split into the problems of *syntactic data integration* and *semantic data integration*. Syntactic data integration means that a data format is used that can be parsed by the museum's applications. For example, syntactic integration between databases of museums can be reached by converting their databases to XML or RDF. Semantic integration means that the meaning of the concepts used in the data can be related to concepts that the processors already understand. For example, the attributes author (used in museum A's data) and creator (used in museum B's data) need to be *mapped* to each other. In this case the mapping relation is an equivalence relation. A query for "all artworks by the author Van Gogh" on museum A's collection can now automatically be translated to a query for "all artworks by the creator Van Gogh" on museum B's collection (assuming that the two concepts representing Van Gogh are also mapped). Museum A's applications will now be able to retrieve and interpret data provided by museum B as if it were already present in A's local repository (and vice versa). XML provides no built-in language features to express mappings, while RDFS and OWL do. This is one argument for the claim of some Semantic Web researchers that RDF(S) and OWL make it easier to integrate data from different sources (e.g. Decker et al. 2000). In this thesis we will assume that it makes sense to use RDF/OWL for integration tasks.

Semantic and syntactic integration are related to syntactic and semantic *interoperability* (e.g. Decker et al. 2000). Interoperability is usually defined as the ability of two or more applications to exchange and understand each other's data (see e.g. European Commission 2004). This definition implies that it is known which applications need to understand the data at the time the data is published. This is often not the case, especially in the Semantic Web setting. Thus the goal should be to represent and publish data in a way that makes that data as understandable as possible for *any* processor. In this thesis we will use this other view on the term interoperability. Thus, in this thesis

[9] Accessed on the 3rd of January 2010.

the term "interoperable vocabulary representation" is equivalent to "a vocabulary representation that promotes integration".

Besides the problems of syntactic and semantic integration, there is a a third obstacle to integration of collection metadata. Not all collections make use of vocabularies for values of metadata elements; in many cases a literal is used (e.g. "Paris" instead of a reference to the concept Paris in the Thesaurus of Geografic Names). The result is that the graph structure is implicit; there is no direct link with other paintings made in Paris. Moreover, homonymy can cause wrong search results (e.g. persons named "Paris"), and when the sources are multilingual then correct results can be missed (e.g. "Parijs" in Dutch, "Parigi" in Italian). In sum, during conversion and integration of collection metadata the values of database fields need to be interpreted as the appropriate vocabulary concepts.

The Semantic Web enterprise can be seen as a new paradigm that contains elements of the field of knowledge representation and of the Web. It moves from search in local databases of relatively flat structures to web resources connected through edges in a graph. Search is not hierarchical as in library systems, but focuses on selecting some relevant part of the graph; any type of edge ("relationship") can play a role.

Collections and the Semantic Web

The communities that create collections and the Semantic Web community can benefit from making collections available in an interoperable representation language as used by the Semantic Web. There is one main benefit for the former community: the Semantic Web provides methods and tools to integrate metadata. Integration of data enables cross-collection search and novel search strategies.

Conversely, collections are also useful for the Semantic Web community for several reasons. Firstly, the Semantic Web only works if data is available in the representation languages it promotes. For example, a semantic search engine does by definition not work on web pages written in natural language but only on representation formats such as RDF/OWL data. Research projects that develop such search engines often aim at developing novel search strategies. This is only possible if realistic datasets are available in RDF/OWL to test their effectiveness.

Data integration in the Eculture project

This thesis aims to help make collections available to the Semantic Web. The context of most of the work is the MultimediaN E-Culture project (Schreiber et al. 2006). This project, and similar projects like MuseumFinland (Hyvönen et al. 2005), have done extensive work in converting collections (metadata element sets, vocabularies, objects) provided by museums to RDF/OWL. These projects then provide semantic search facilities (e.g. faceted browsing) over the resulting data.

The projects used several conversion and integration techniques, including: (1) replacing literals with URIs from vocabularies; (2) mapping of vocabulary concepts. The first technique is used because collections often do not use concepts but literal values to denote e.g. Vincent van Gogh. Such literal values can be replaced with the concept for Van Gogh from ULAN. This helps prevent homonymous search results. The second technique entails establishing equivalence relations be-

tween concepts of different vocabularies. For example, the concepts for "impressionism" in both AAT and SVCN were mapped so that e.g. a query for all impressionist paintings will return results from different collections.

The result of applying these techniques is that the implicit connections between collections are made explicit in the form of an RDF graph. An explicated graph structure can enable novel search strategies such as Relation Search. It can describe to a user how two concepts are related. For example, Van Gogh and Gauguin are both impressionist, and both are related to Emile Bernard — Van Gogh as a student and Gauguin as collaborator of another of Bernard's students. This is still an experimental but interesting avenue of research.

Integration is also necessary at the level of metadata element sets, because different collections use different elements, but also similar elements with different names. The E-Culture project experimented with various ways to integrate metadata element sets. One way is to first create a representation of each collection's metadata element set in RDF. This results in several RDF schemas, with one element being represented by one RDF property in the schema (an example property might be paintedBy. Then an element set is selected that can cover all of the individual collection's element sets. An RDF schema is also created for the overarching (more general) set (this may contain the property creator. Finally, the properties of the individual collection's schemas are mapped to the properties of the overarching schema (e.g. a mapping from paintedBy to creator. All collections can now be queried as if they were one collection on the abstraction level of that overarching element set/schema.

In sum, the E-Culture project has developed a cross-collection search system where syntactic and semantic integration is based on vocabularies and metadata schemas. Vocabularies provide a ready-made set of concepts (Painter, Painting, creator) that can act as the "glue" that holds together the various collections. They form the nodes and edges that connect the collections in an explicated graph structure.

Data conversion

Before integration of metadata based on vocabularies and metadata element sets can take place, the vocabularies and element sets have to be converted to a representation in RDF/OWL (the family of Semantic Web representation languagues). However, a variety of syntactic representation formats such as XML, databases and Prolog are in use, inhibiting syntactic integration and subsequent semantic integration. There are several problems involved in conversion from these formats to RDF/OWL. Firstly, the syntax of the original representation has to be understood. Secondly, the link between the syntax and the conceptual model that is expressed in that syntax has to be understood. Thirdly, this conceptual model may differ from the conceptual model of RDF/OWL. These models have to be mapped before the actual data can be converted. There may be several alternative mappings with their own trade-offs. Lastly, the way the resulting RDF/OWL data is to be used in applications can also play a role. How to address these issues is one of the main topics of this thesis. We elaborate further in the next section.

1.2 Problem Statement and Research Questions

This thesis focuses on conversion of vocabularies for representation and integration of collections on the Semantic Web. A secondary focus is how to represent metadata schemas (RDF Schemas representing metadata element sets) such that they interoperate with vocabularies. The primary domain in which we operate is that of cultural heritage collections. The background worldview in which a solution is sought is that of the Semantic Web research paradigm with its associated theories, methods, tools and use cases. In other words, we assume the Semantic Web is in principle able to provide the context to realize interoperable collections. Interoperability is dependent on the interplay between representations and the applications that use them. We mean applications in the widest sense, such as "search" and "annotation". These applications or tasks are often present in *software* applications, such as the E-Culture application. It is therefore necessary that applications' requirements on the vocabulary representation are met. This leads us to formulate the following problem statement:

HOW CAN EXISTING VOCABULARIES BE MADE AVAILABLE TO SEMANTIC WEB APPLICATIONS?

We refine the problem statement into three research questions. The first two focus on the problem of conversion of a vocabulary to a Semantic Web representation from its original format. Conversion of a vocabulary to a representation in a Semantic Web language is necessary to make the vocabulary available to Semantic Web applications. In the last question we focus on integration of collection metadata schemas in a way that allows for vocabulary representations as produced by our methods.

Research Question 1: HOW CAN VOCABULARIES BE CONVERTED TO AN INTEROPERABLE REPRESENTATION IN AN APPLICATION-NEUTRAL WAY?

In this question we study how it is possible to convert vocabularies to an interoperable representation without focusing on any particular application. We answer the question by developing a method that enables such conversions. The intentions of the vocabulary creators themselves are central. The aim of conversion is to preserve all content (concepts, terms, definitions etc.) and intended semantics. The statements present in the vocabularies, both explicit and implied, remain the same in both representations. This is especially important because we cannot know beforehand what an application's requirements will be. The underlying assumption is that application-neutral conversion results in a representation that is complete and usable for a wide range of applications.

There are two main problems. The first is how to analyze a vocabulary such that a correct understanding of the link between its syntax and its semantics emerges. Vocabularies usually define their own syntax, so for each vocabulary this link needs to be established. The second problem is how the semantics of a vocabulary are best reflected in RDF/OWL. There are several ways in which to represent a piece of knowledge in RDF/OWL, and the most appropriate way has to be selected. We aim to find guidelines that assist practitioners in making these choices.

Conversion of a vocabulary's concepts and terms requires a target metamodel of vocabularies expressed in RDF/OWL. One approach would be to develop a new metamodel for each vocabulary.

Another approach is to use an existing (standard) metamodel as target. The first approach may hinder integration, but the second approach may need to sacrifice some of the vocabulary content or semantics because the metamodel does not fit the vocabulary. We investigate both approaches and their respective merits.

Research Question 2: HOW CAN VOCABULARIES BE CONVERTED TO AN INTEROPERABLE REPRESENTATION WITH GIVEN APPLICATION REQUIREMENTS?

With this question we focus on situations where application-specific requirements are known in advance and should be incorporated into the conversion process. These can be requirements from a generic application such as "search", or a specific software application such as the E-Culture project. An example is that a geographic vocabulary may separate the representation of coordinates in longitude and lattitude, while an application may require that coordinates are expressed as one literal. We answer the question by developing a method that enables such conversions. In the context of this research question the application requirements become the guiding principle of conversion instead of the preservation of content and semantics.

There are two main problems. The first is how application requirements can be related to requirements on the vocabulary representation. The second is how to deal with trade-offs on what to include and what not. Some particular details might be useful for the application, but also complicate algorithms that must query the representation.

Some applications require the use of several vocabularies in concord (e.g. the E-Culture application). This raises the problem of how to integrate several vocabularies (e.g. so that search across collections indexed with different vocabularies becomes possible). The solution proposed by the Semantic Web is *alignment* of vocabularies with so-called alignment tools. Such tools can be seen as a particular application, which place requirements on vocabulary representations, just as any other application. In this context the vocabularies are not delivered custom-made for the application as the output of a conversion process. The vocabulary representation may not match the requirements of the application. We investigate whether it is necessary to adapt parts of the method to cater for the vocabulary requirements posed by alignment applications. We also investigate whether alignment applications make use of all relevant features of vocabularies in the alignment task. If they do not, this indicates how alignment applications can be improved for the task of aligning vocabularies.

Research Question 3: HOW CAN METADATA SCHEMAS BE REPRESENTED IN A WAY THAT ALLOWS FOR INTEGRATION OF COLLECTIONS THAT USE DIFFERENT VOCABULARIES?

In the context of the Semantic Web, several collections with their own vocabularies will be present. The vocabulary concepts appear as values in metadata schemas used to represent collections. They are often controlled sets of values developed to describe objects. The role of the metadata schema is to represent the relevant attributes of objects.

The problem is that integration of collections is hindered because each uses its own metadata schema and indexing vocabularies. This makes it difficult to develop applications that query and visualize the data. One way to establish a unified view of the collections' metadata is to link each

individual schema to a more general schema. This solution can only be realized if it is possible to establish this link. We can formulate several requirements to this link. Firstly, the semantics of the elements in the individual schemas should be covered. Each element of each individual schema should be linked to an element of the generic schema that has the same or more generic meaning. Secondly, the generic schema should not impose specific vocabularies and datatypes on the individual schemas. Each collection has its own set of appropriate values for metadata elements, and the representation should reflect that. Thirdly, the representation of the individual and generic schemas should interoperate with vocabulary representations as produced by our methods. In other words, it should be possible to link metadata elements to (parts of) vocabularies represented in RDF/OWL. We term this "collection-specific value ranges" for metadata elements.

We study this problem with the E-Culture project as case study. The E-Culture project wishes to use a domain-specific metadata schema (VRA Core Categories for discribing visual art) to represent the content of several collections. Furthermore, it wishes to integrate data from cultural heritage collections with other types of metadata collections from other domains, such as TV programs. In the future, metadata about, e.g., a vase found in an excavation can then be coupled to a TV program about that excavation. For this purpose the E-Culture project wishes to use the Dublin Core metadata element set. In the case study we develop an RDF/OWL representation of VRA Core Categories and link it to the existing RDF representation of Dublin Core.

1.3 Research Design

In this thesis we consider a *method* to be a step-wise process that practitioners can follow to reach a particular goal (see Schreiber et al. 2000). Guidelines are provided to assist practitioners in performing the steps of the process. The output of two research questions consists methods. The research design to develop these methods contains the following steps. First an initial method is developed from scratch, based on literature and our own experiences and insights. In the second step the method is used to convert a set of vocabularies (case studies). In the third step the results of the case studies are used to generalize the method and improve it. A complicating matter in developing the methods is that the Semantic Web has more than one language as a basis for knowledge representation. We attempt to create guidelines that are appropriate for all these langugages, so that the practitioner may choose depending on his/her needs.

The usefulness and quality of the resulting methods is dependent on the case studies (i.e. the vocabularies that are converted). The vocabularies we have chosen tend to be well-known and often used. Some have a complex structure. With this selection we try to ensure that the resulting methods are useful for converting vocabularies in general. The vocabularies we selected are: WordNet, MeSH, AAT, TGN, GTAA, IPSV, ULAN. For one particular issue we also studied existing representations of SVCN and ARIA.

The answer to the last research question is not a method, but rather a way how specific requirements on combining metadata schemas and vocabularies can be fulfilled. To answer this question we have done a case study for a specific metadata schema that should be combined with vocabulary representations as developed earlier in the thesis.

1.4 Contributions

The contributions of this thesis to the state of the art can be summarized as follows: (a) the development of new conversion methods; (b) case studies that illustrate and validate the methods; (c) availability and usage of the developed conversions; (d) analysis of requirements on vocabulary representation by alignment tools, and identification of useful vocabulary features that are ignored by the tools; (e) a case study that shows how metadata schemas and vocabularies can be represented in a way that allows for integration of heterogeneous collections.

Firstly, we developed three conversion methods. We developed a generic method for converting vocabularies to RDF/OWL, comprised of several steps and guidelines. The method provides practical guidance on how to convert vocabularies such that the intentions of the original vocabulary authors are reflected in the RDF/OWL version as much as possible. Our second method is focused on converting vocabularies to SKOS. SKOS (Simple Knowledge Organization System) is a metamodel for expressing vocabularies in RDF standardized by the W3C (Semantic Web Deployment Working Group 2008a). We identified what information can be converted in a SKOS-compliant manner, when specialisations can be used and give examples of information that cannot be expressed in SKOS. Our third method is focused on converting vocabularies for an application. The method helps to translate application requirements into requirements on the vocabulary representation. Use cases are employed to determine the application requirements. An overview of all three methods is given in Appendix A.

Secondly, we performed case studies of several vocabularies to illustrate and validate the methods. The generic method is illustrated with case studies of WordNet and MeSH. The SKOS-specific method is illustrated with IPSV, GTAA and MeSH. The application-specific method is illustrated with a conversion of WordNet (for W3C) and with a conversion of the three Getty vocabularies in the context of the E-Culture application. All the conversions are available as downloads except for the Getty vocabularies (because of licence restrictions). The WordNet conversion for W3C can also be queried online because each WordNet URIs is a HTTP URI: at that location a set of triples that describes the URI is made available. WordNet is one of the earliest examples of a large vocabulary published according to the "linked open data" principles and the "Recipes for Publishing RDF Vocabularies" (Semantic Web Best Practices and Deployment Working Group 2006a).

Thirdly, our conversions contribute to the state of the art because they are successfully (re)used. This is visible specifically for MeSH and WordNet. Our MeSH-SKOS conversion was included in the Health Care and Life Sciences Knowledge Base[10] and its follow-up NeuroCommons.[11] Our online version of WordNet (in W3C webspace) was incorporated into the Linked Open Data cloud and is linked to DBpedia and OpenCyc.

Fourthly, we contribute an analysis of requirements that existing alignment tools place on vocabulary representations. This shows that current tools cannot handle the representations proposed in this thesis. Additionally, we contribute an analysis of which features that are present in vocabularies are useful for alignment. Some of these features are not yet incorporated into the tools'

[10] http://www.w3.org/TR/hcls-kb/

[11] http://neurocommons.org/page/Bundles/mesh/mesh-skos

algorithms.

Our fifth and last contribution is a number of techniques and lessons learnt on integrating metadata schemas and vocabularies from heterogeneous collections, based on a case study. We show how an existing metadata element set can be represented in an RDF/OWL schema that is compatible with this goal. We also show how the schema can be specialized for a particular collection using a representation pattern we propose, in an interoperable way.

A contribution that is not within the scope of the research questions is presented in Chapter 6. We present two alternative techniques for evaluating vocabulary alignments. Vocabulary alignments are central to applications such as search over heterogeneous collections (e.g. as in the E-Culture portal). Searches should cover results from all collections and therefore similar concepts in each collection's vocabularies should be aligned. After an alignment tool creates an alignment, it must be evaluated as to its suitability. We present two techniques that can provide a better estimation of the suitability of a vocabulary alignment for an application compared to traditional techniques.

1.5 A Few Notes on Terminology and RDF/OWL Notation

The reader should be aware that in some of the chapters that are based on early publications we only refer to vocabularies as thesauri (e.g. Chapter 2). As explained earlier, we consider vocabularies to encompass anything from a glossary to an ontology, in so far as that they all define concepts with associated terms. The number of features that vocabularies display grows from simple vocabularies to complex ones: a glossary is a list of unrelated concepts with definitions, a classification scheme defines a hierarchy between concepts, a thesaurus also defines related concepts. The usage of the term "thesaurus" instead of "vocabulary" in the earlier publications is a misnomer as our analysis focuses on any vocabulary's features.

This thesis uses the Turtle syntax[12] created by Dave Beckett to display RDF/OWL instead of the more widespread RDF/XML syntax, because the former is more compact and – we feel – more readable. Figures in papers included in the thesis which used RDF/XML have been changed to RDF/Turtle. Turtle is derived from the N3 syntax.[13] Both these syntaxes are not official standards within W3C as of yet. Triples in running text are displayed between angle brackets to make them stand out (e.g. <ex:a, ex:p, ex:b>).

In some chapters a hierarchical relationship between two concepts is indicated in the text by using an arrow between the concepts; e.g. Object ◁– Chair means that Chair is hierarchically located below Object.

1.6 Chapter Overview of the Thesis

This thesis is divided into eight chapters. **Chapter 2** focuses on research question 1. It introduces a generic method to convert vocabularies from their original format to RDF/OWL while preserving

[12] http://www.w3.org/TeamSubmission/turtle/

[13] http://www.w3.org/DesignIssues/Notation3.html

the content and semantics. Application concerns are not taken into account. The chapter is based on (Van Assem et al. 2004) and uses WordNet and MeSH as case studies. Later chapters make a few modifications to the generic method. An overview is presented in Appendix A.

Chapter 3 also focuses on research question 1. It presents a method to convert vocabularies to a specific metamodel, in this case the W3C's Simple Knowledge Organization System (SKOS). SKOS proposes a theory of vocabulary structure and content, supported by a specific RDFS schema. The aim of SKOS is to promote interoperability between vocabularies used on the Semantic Web. It is intended to be suitable for a wide range of vocabularies and applications. An overview of the method is given in Appendix A. The chapter is based on (Van Assem et al. 2006b) and uses GTAA, IPSV and MeSH as case studies.

Chapter 4 focuses on research question 2. It describes the conversion of WordNet by the W3C's Best Practices and Deployment Working Group and compares it to the conversion presented in Chapter 2. This comparison allows us to conclude whether a conversion method that disregards application concerns produces usable conversions. Additionally, the chapter adds a fourth step to the generic method, and presents practical details of publishing a large-scale vocabulary on the Semantic Web. The chapter is based on (Van Assem et al. 2006a) and (Semantic Web Best Practices and Deployment Working Group 2006c).

Chapter 5 continues investigating research question 2. A method is proposed for application-centric vocabulary conversion. The method is adapted from the method in Chapter 2 and takes the results from Chapter 4 into account. An overview of the method is given in Appendix A. The case study concerns the conversion of the so-called Getty vocabularies (AAT, TGN, ULAN) in the context of the E-Culture application built by the MultimediaN E-Culture project.

In **Chapter 6** the focus is on research question 2. The role of vocabulary representations in the creation of vocabulary alignments is studied. We assess the requirements that alignment tools place on vocabulary representations. We also assess what vocabulary features alignment tools can use to create better alignments. The context of these assessments is a study of alignment evaluation strategies. We propose two new application-dependent evaluation strategies and compare them to existing application-independent strategies. We study the alignments between three "source" vocabularies (ARIA, SVCN, WordNet) to one "target" vocabulary (AAT). The chapter is an extended and adapted version of (Hollink et al. 2008).

In **Chapter 7** we focus on research question 3 about integration of metadata schemas. The context is the E-Culture project. We develop an RDF/OWL representation of a metadata schema for cultural heritage called VRA Core, and show how this can be integrated with a generic metadata schema called Dublin Core. The schema is defined in such a way that it can be combined with vocabularies as produced by our methods. An overview of VRA and the actual schema produced is provided in Appendix C.

Finally, **Chapter 8** provides conclusions to our research questions and a discussion.

2

A Method for Converting Vocabularies to an Interoperable Representation

In this chapter we focus on research question 1: "How can vocabularies be converted to an interoperable representation in an application-neutral way?" A method is developed for conversion of vocabularies to RDF/OWL with the aim of preserving the content and semantics of the original sources. No particular application or metamodel is presupposed. First a proposal for a methd is created, based on the authors' experience in conversion activities. The method identifies four steps in the conversion process, where each step refines the vocabulary representation of the previous step. Guidelines are provided to assist in producing the conversion. The method's scope is then tested by applying it in two use cases: WordNet and MeSH. Several guidelines were added to be able to handle these cases.

Later chapters modify parts of the method presented here. An overview of the changes can be found in Appendix C.

This chapter was published in the Proceedings of the Third International Semantic Web Conference, and was co-authored with Maarten Menken, Guus Schreiber, Jan Wielemaker and Bob Wielinga (Van Assem et al. 2004).[1]

2.1 Introduction

Thesauri are controlled vocabularies of terms in a particular domain with hierarchical, associative and equivalence relations between terms. Thesauri such as National Library of Medicine's Medical Subject Headings (MeSH) are mainly used for indexing and retrieval of articles in large databases (in the case of MeSH the MEDLINE/PubMed database containing over 14 million citations[2]). Other resources, such as the lexical database WordNet, have been used as background knowledge in several analysis and semantic integration tasks (Fellbaum 1998). The native format of such resources is often a proprietary XML, ASCII or relational schema. They are generally not available in the Semantic Web languages RDF(S) and OWL. This paper describes a method for converting thesauri to RDF/OWL and illustrates it with conversions of MeSH and WordNet.

The main objective of converting existing resources to the RDF data model is that these can then be used in Semantic Web applications for annotations. Thesauri provide a hierarchically

[1] Changes that have been made with respect to the original publication are the following: the title was changed, some sentences changed to improve readability, broken URLs were updated, some markup of inline triples homogenized with the rest of the thesis, and a figure which used the RDF/XML syntax was changed to use the RDF/Turtle syntax.

[2] `http://www.ncbi.nlm.nih.gov/pubmed`

structured set of terms about which a community has reached consensus. This is precisely the type of background knowledge required in Semantic Web applications. One insight from the submissions to the Semantic Web challenge at ISWC'03[3] was that these applications typically used simple thesauri instead of complex ontologies.

Although conversions of thesauri have been performed, currently no accepted methodology exists to support these efforts. This paper presents a method that can serve as the starting point for such a methodology. The method and guidelines are based on the authors' experience in converting various thesauri. This paper is organized as follows. Section 2.2 provides introductory information on thesauri and their structure. In Section 2.3 we describe our method and the rationale behind its steps and guidelines. Sections 2.4 and 2.5 each discuss a case study in which the conversion method is applied to MeSH and WordNet, respectively. Additional guidelines that were developed during the case studies, or are more conveniently explained with a specific example, are introduced in these sections. Related research can be found in Section 2.6. Finally, Section 2.7 offers a discussion.

2.2 Structure of Thesauri

Many thesauri are historically based on the ISO 2788 and ANSI/NISO Z39.19 standards (International Organization for Standardization 1986, ANSI/NISO 2003). The main structuring components are terms and three relations between terms: Broader Term (BT), Narrower Term (NT) and Related Term (RT). *Preferred terms* should be used for indexing, while *non-preferred terms* are included for use in searching. Preferred terms (also known as *descriptors*) are related to non-preferred terms with Use For (UF); USE is the inverse of this relation. Only preferred terms are allowed to have BT, NT and RT relations. The Scope Note (SN) relation is used to provide a definition of a term (see Figure 2.1).

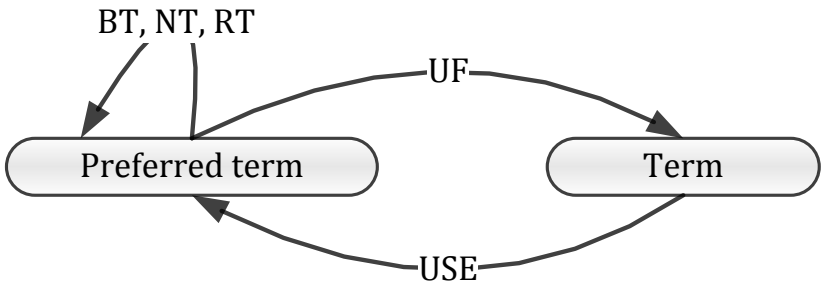

Figure 2.1 The basic thesaurus relations. Scope note is not shown.

[3] http://challenge.semanticweb.org/

Two other constructs are *qualifiers* and *node labels*. Homonymous terms should be supplemented with a qualifier to distinguish them, for example "Beams (radiation)" and "Beams (structures)". A node label is a term that is not meant for indexing, but for structuring the hierarchy, for example "Knives By Form". Node labels are also used for organizing the hierarchy in either *fields* or *facets*. The former divides terms into areas of interest such as "injuries" and "diseases", the latter into more abstract categories such as "living" and "non-living" (International Organization for Standardization 1986).

The standards advocate a *term-based* approach, in which terms are related directly to one another. In the *concept-based* approach (Miles and Matthews 2004), concepts are interrelated, while a term is only related to the concept for which it stands. A term is a *lexicalization* of a concept (Soergel et al. 2004). The concept-based approach may have advantages such as improved clarity and easier maintenance (Johnston et al. 1998).

2.3 Method Description

The method is divided into four steps: (0) a preparatory step, (1) a syntactic conversion step, (2) a semantic conversion step, and (3) a standardization step. The division of the method into four steps is an extension of previous work (Wielinga et al. 2004).

Step 0: Preparation

In the preperatory step an analysis is made of the thesaurus. To perform this step (and therefore also the subsequent steps) correctly, it is essential to contact the original thesaurus authors when the documentation is unclear or ambiguous. An analysis of the thesaurus contains:

- Conceptual model (the model behind the thesaurus is used as background knowledge in creating a sanctioned conversion);
- Relation between conceptual and digital model;
- Relation to standards (aids in understanding the conceptual and digital model);
- Identification of multilinguality issues.

Although we recognize that multilinguality is an important and complicating factor in thesaurus conversion (see also (Miles et al. 2004a)), it is not treated in this paper.

Step 1: syntactic conversion

In this step the emphasis lies on the syntactic aspects of the conversion process from the source representation to RDF(S). Typical source representations are (1) a proprietary text format, (2) a relational database and (3) an XML representation. This step can be further divided into two substeps.

Step 1a: structure-preserving translation.

In Step 1a, a *structure-preserving* translation between the source format and RDF format is performed, meaning that the translation should reflect the source structure as closely as possible. The translation should be complete, meaning that all semantically relevant elements in the source are translated into RDF.

Guideline 1: USE A BASIC SET OF RDF(S) CONSTRUCTS FOR THE STRUCTURE-PRESERVING TRANSLATION. Only use constructs for defining classes, subclasses, properties (with domains and ranges), human-readable rdfs:labels for class and property names, and XML datatypes. These are the basic building blocks for defining an RDF representation of the conceptual model. The remaining RDF(S) and OWL constructs are used in Step 2 for a semantically oriented conversion. However, one might argue that the application of some constructs (e.g. domains and ranges) also belongs to semantic conversion.

Guideline 2: USE XML SUPPORT FOR DATATYPING. Simple built-in XML Schema datatypes such as **xsd:date** and **xsd:integer** are useful to supply schemas with information on property ranges. Using user-defined XML Schema datatypes is still problematic[4] ; hopefully this problem will be solved in the near future.

Guideline 3: PRESERVE ORIGINAL NAMING AS MUCH AS POSSIBLE. Preserving the original naming of entities results in more clear and traceable conversions. Prefix duplicate property names with the name of the source entity to make them unique. The meaning of a class or property can be explicated by adding an rdfs:comment, preferably containing a definition from the original documentation. If documentation is available online, rdfs:seeAlso or rdfs:isDefinedBy statements can be used to link to the original documentation and/or definition.

Guideline 4: TRANSLATE RELATIONS OF ARITY THREE OR MORE INTO STRUCTURES WITH BLANK NODES. Relations of arity three or more cannot be translated directly into RDF properties. If the relation's arguments are independent of each other, a structure can be used consisting of a property (with the same name as the original relation) linking the source entity to a blank node (representing the relation), and the relation's arguments linked to the blank node with an additional property per argument (see examples in Section 2.4).

Guideline 5: DO NOT TRANSLATE SEMANTICALLY IRRELEVANT ORDERING INFORMATION. Source representations often contain sequential information, e.g. ordering of a list of terms. These may be irrelevant from a semantic point of view, in which case they can be left out of the conversion.

Guideline 6: AVOID REDUNDANT INFORMATION. Redundant information creates representations which are less clear and harder to maintain. An example on how to avoid this: if the Unique Identifier (UI) of a resource is recorded in the rdf:ID, then do not include a property that also records the UI.

[4] http://www.w3.org/2001/sw/WebOnt/webont-issues.html#I4.3-Structured-Datatypes

Guideline 7: AVOID INTERPRETATION. Interpretations of the meaning of information in the original source (i.e., meaning that cannot be traced back to the original source or documentation) should be approached with caution, as wrong interpretations result in inconsistent and/or inaccurate conversions. The approach of this method is to postpone interpretation (see Step 2b).

Instead of developing a new schema (i.e., thesaurus metamodel), one can also use an existing thesaurus schema, such as SKOS (see Section 2.3), which already defines "Concept", "broader", etc. This may be a simpler approach than to first develop a new schema and later map this onto the SKOS. However, this is only a valid approach if the metamodel of the source and of SKOS match. For thesauri with a (slightly) different metamodel, it is recommended to develop a schema from scratch, so as not to lose the original semantics, and map this schema onto SKOS in Step 3. A drawback is that the naming of the original metamodel is lost (e.g. "BT" instead of "broader").

Step 1b: explication of syntax.

Step 1b concerns the *explication* of information that is implicit in the source format, but intended by the conceptual model. The same set of RDF(S) constructs is used as in Step 1a. For example, the AAT thesaurus (Peterson 1994) uses node labels (called "Guide Terms" in AAT), but in the AAT source data these are only distinguished from normal terms by enclosing the term name in angle brackets (e.g. <Knives by Form>). This information can be made explicit by creating a class GuideTerm, which is an rdfs:subClassOf the class AATTerm, and assigning this class to all terms with angle brackets. Other examples are described in Sections 2.4 and 2.5.

Step 2: Semantic Conversion

In this step the class and property definitions are augmented with additional RDFS and OWL constraints. Its two substeps are aimed at explication (Step 2a) and interpretation (Step 2b). After completion of Step 2a the thesaurus is ready for publication on the Web as an "as-is" RDF/OWL representation.

Step 2a: explication of semantics.

This step is similar to Step 1b, but now more expressive RDFS and OWL constructs may be used. For example, a broaderTerm property can be defined as an owl:TransitiveProperty and a relatedTerm property as an owl:SymmetricProperty.

A technique that is used in this step is to define certain properties as specializations of predefined RDFS properties, e.g. rdfs:label and rdfs:comment. For example, if a property nameOf is clearly intended to denote a human-readable label for a resource, it makes sense to define this property as a subproperty of rdfs:label. RDFS-aware tools will now be able to interpret nameOf in the intended way.

Step 2b: interpretations.

In Step 2b specific *interpretations* are introduced that are strictly speaking not sanctioned by the original model or documentation. A common motivation is some application-specific requirement, e.g. an application wants to treat a broaderTerm hierarchy as a class hierarchy. This can be stated as follows: <broaderTerm, rdfs:subPropertyOf, rdfs:subClassOf>. Semantic Web applications using thesauri will often want to do this, even if not all hierarchical links satisfy the subclass criteria. This introduces the notion of metamodeling. It is not surprising that the schema of a thesaurus is typically a metamodel: its instances are categories for describing some domain of interest.

Guideline 8: CONSIDER TREATING THE THESAURUS SCHEMA AS A METAMODEL. The instances of a thesaurus schema are often general terms or concepts, that occur as classes in other places. RDFS allows one to treat instances as a classes: simply add the statement that the class of those instances is a subclass of rdfs:Class. For example, an instance i is of class C; class C is declared to be an rdfs:subClassOf rdfs:Class. Because instance i is now also an instance of rdfs:Class, it can be treated as a class.

The above example of treating broader term as a subclass relation is similar in nature.

A schema which uses these constructions is outside the scope of OWL DL. Application developers will have to make their own expressivity vs. tractability trade-off here.

The output of this step should be used in applications as *a specific interpretation of the thesaurus*, not as a standard conversion.

Step 3: Standardization

Several proposals exist for a standard schema for thesauri.[5] Such a schema may enable the development of infrastructure that can interpret and interchange thesaurus data. Therefore, it may be useful to map a thesaurus onto a standard schema. This optional step can be made both after Step 2a (the result may be published on the web as a standard conversion) and Step 2b (the result may be published on the web as an interpretation). Unfortunately, a standard schema has not yet been agreed upon.[6] As illustration, the SKOS schema developed by the W3C Semantic Web Advanced Development for Europe project[7] is mapped to MeSH in Section 2.4.

The SKOS schema is concept-based, with class Concept and relations narrower, broader and related between Concepts. A Concept can have a prefLabel (preferred term) and altLabels (non-preferred terms). Also provided is a TopConcept class, which can be used to arrange a hierarchy under special concepts (such as fields and facets, see Section 2.2). TopConcept is a subclass of Concept. Note that because SKOS is concept-based, it may be problematic to map term-based thesauri to SKOS.

[5] http://www.w3.org/2001/sw/Europe/reports/thes/thes_links.html

[6] Since the time of writing SKOS has been developed into a W3C Recommendation, see http://www.w3.org/TR/2009/REC-skos-reference-20090818/

[7] http://www.w3.org/2001/sw/Europe/reports/thes/1.0/guide/

2.4 Case One: MeSH

This section describes how the method has been applied to MeSH (version 2004[8]). The main source consists of two XML files: one containing so-called *descriptors* (228 MB), and one containing *qualifiers* (449 Kb). Each has an associated DTD. A file describing additional information on descriptors was not converted. The conversion program (written in XSLT) plus links to the original source and output files of each step can be found at `http://thesauri.cs.vu.nl/`. The conversion took two people approximately three weeks to complete.

Analysis of MeSH

The conceptual model of MeSH is centered around Descriptors, which contain Concepts (U.S. National Library of Medicine 2001). In turn, Concepts consist of a set of Terms. Exactly one Concept is the preferred Concept of a Descriptor, and exactly one Term is the preferred Term of a Concept. Each Descriptor can have Qualifiers, which are used to indicate aspects of a Descriptor, e.g. "ABDOMEN" has the Qualifiers "pathology" and "abnormalities". Descriptors are related in a polyhierarchy, and are meant to represent broader/narrower *document retrieval sets* (i.e., not a subclass relation). Each Descriptor belongs to one (or more) of fifteen Categories, such as "Anatomy" and "Diseases" (U.S. National Library of Medicine 2004). The Concepts contained within one Descriptor are also hierarchically related to each other.

This model is inconsistent with the ISO and ANSI standards, for several reasons. Firstly, the model is concept-based. Secondly, Descriptors contain a set of Concepts, while in the standards a Descriptor is simply a preferred term. Thirdly, Qualifiers are not used to disambiguate homonyms.

Converting MeSH

Step 1a: structure-preserving translation.

In the XML version of MeSH, Descriptors, Concepts, Terms and Qualifiers each have a Unique Identifier (UI). Each Descriptor also has a TreeNumber (U.S. National Library of Medicine 2004). This is used to indicate a position in a polyhierarchical structure (a Descriptor can have more than one TreeNumber), but this is implicit only. Relations between XML elements are made by referring to the UI of the relation's target (e.g. `<SeeRelatedDescriptor>` contains the UI of another Descriptor). In Step 1a, this is converted into instances of the property hasRelatedDescriptor. The explication of TreeNumbers is postponed until Step 1b.

Most decisions in Step 1a concern which XML elements should be translated into classes, and which into properties. The choice to create classes for Descriptor, Concept and Term are clear-cut: these are complex, interrelated structures. A so-called `<EntryCombination>` relates a Descriptor-Qualifier pair to another Descriptor-Qualifier pair. Following guideline 4, two blank nodes are created (each representing one pair) and related to an instance of the class EntryCombination. As already mentioned, relations between elements in XML MeSH are made by referring

[8] `http://www.nlm.nih.gov/mesh/filelist.html`

to the UI of the target. However, each such relation also includes the *name* of the target. As this is redundant information, the name can be safely disregarded.

Guideline 9: GIVE PREFERENCE TO THE RELATION-AS-ARC APPROACH OVER THE RELATION-AS-NODE APPROACH. In the relation-as-arc approach, relations are modeled as arcs between entities (RDF uses "properties" to model arcs). In the relation-as-node approach, a node represents the relation, with one arc relating the source entity to the relation node, and one arc relating the relation node to the destination entity (Miles and Matthews 2004). The relation-as-arc approach is more natural to the RDF model, and also allows for definition of property semantics (symmetry, inverseness, etc.) in OWL.

It is not always possible to follow guideline 9, e.g. in the case of MeSH `<ConceptRelation>`. A ConceptRelation links two concepts and an additional attribute. The additional attribute prevents the usage of a relation-as-arc. A blank node is used to represent the relationship according to guideline 4.

Guideline 10: CREATE PROXY CLASSES FOR REFERENCES TO EXTERNAL RESOURCES IF THEY ARE NOT AVAILABLE IN RDF. If the thesaurus refers to external resources which are not described further, create a proxy class for those references. For example, each MeSH Concept has an associated SemanticType, which originates in the UMLS Semantic Network.[9] This external resource is not available in RDF, but might be converted in the near future. In MeSH, only the UI and name of the SemanticType is recorded. One could either use a datatype property to relate the UI to a Concept (again, the redundant name is ignored), or create SemanticType instances (empty proxies for the actual types). We have opted for the latter, as this simplifies future integration with UMLS. In this scenario, either new properties can be added to the proxies, or the existing proxies can be declared owl:sameAs to SemanticType instances of a converted UMLS.

Guideline 11: ONLY CREATE RDF:IDS BASED ON IDENTIFIERS IN THE ORIGINAL SOURCE. A practical problem in the syntactical translation is what value to assign the rdf:ID attribute. If the original source does not provide a unique identifier for an entity, one should translate it into blank nodes, as opposed to generating new identifiers. A related point is that if the UI is recorded using rdf:ID, additional properties to record an entity's UI would introduce redundancy, and therefore shouldn't be used.

Guideline 12: USE THE SIMPLEST SOLUTION THAT PRESERVES THE INTENDED SEMANTICS. In XML MeSH, only one Term linked to a Concept is the preferred term. Some terms are permutations of the term name (indicated with the attribute isPermutedTermYN), but unfortunately have the same UI as the Term from which they are generated. A separate instance cannot be created for this permuted term, as this would introduce a duplicate rdf:ID. Two obvious solutions remain: create a blank node or relate the permuted term with a datatype property permutedTerm to Term. In the first solution, one would also need to relate the

[9] `http://www.nlm.nih.gov/pubs/factsheets/umlssemn.html`

node to its non-permuted parent, and copy all information present in the parent term to the permuted term node (thus introducing redundancy). The second solution is simpler and preserves the intended semantics.

Step 1b: explication of syntax.

In Step 1b, three explications are made. Firstly, the TreeNumbers are used to create a hierarchy of Descriptors with a subTreeOf property (with domain and range Descriptor). Secondly, the TreeNumber starts with a capital letter which stands for one of fifteen Categories. The class Category and property inCategory are introduced to relate Descriptors (domain) to their Category (range). Thirdly, the ConceptRelations are translated into three properties, brd, nrw and rel, thus converting from a relation-as-node to a relation-as-arc approach (see Guidelines 12 and 9). This requires two observations: (a) the values NRW, BRD and REL of the attribute relationName correspond to narrower, broader and related Concepts; and (b) the relationAttribute is not used in the actual XML, and can be removed. Without the removal of the relationAttribute, the arity of the relation would have prevented us from using object properties.

Some elements are not explicated, although they are clear candidates. These are XML elements which contain text, but also implicit information that can be used to link instances. For example, a Descriptor's `<RelatedRegistryNumber>` contains the ID of another Descriptor, but also other textual information. Splitting this information into two or more properties changes the original semantics, so we have chosen to create a datatype property for this element and copy the text as a literal value.

Step 2a: explication of semantics.

In Step 2a, the following statements are added (a selection):

- The properties brd and nrw are each other's inverse, and are both transitive, while rel is symmetric;
- A Concept's scopeNote is an rdfs:subPropertyOf the property rdfs:comment;
- All properties describing a resource's name (e.g. descriptorName) are declared an rdfs:subPropertyOf the property rdfs:label;
- Each of these name properties is also an owl:InverseFunctionalProperty, as the names are unique in the XML file. Note that this may not hold for future versions of MeSH;
- All properties recording a date are an owl:FunctionalProperty;
- The XML DTD defines that some elements occur either zero or once in the data. The corresponding RDF properties can also be declared functional;
- As a Term belongs to exactly one Concept, and a Concept to exactly one Descriptor, hasTerm as well as hasConcept is an owl:InverseFunctionalProperty (former has domain/range Concept/Term, latter has domain/range Descriptor/Concept).

Unfortunately, the relation represented by class EntryCombination cannot be supplied with additional semantics, e.g. that it is an owl:SymmetricProperty (see guideline 9).

Step 2b: interpretations.

In Step 2b, the following *interpretations* are made, following guideline 8. Note that these are examples, as we have no specific application in mind.

- brd is an rdfs:subPropertyOf rdfs:subClassOf;
- Descriptor and Concept are declared rdfs:subClassOf rdfs:Class.

Step 3: standardization.

In Step 3, a mapping is created between the MeSH schema and the SKOS schema. The following constructs can be mapped (using rdfs:subPropertyOf and rdfs:subClassOf):

- mesh:subTreeOf onto skos:broader;
- mesh:Descriptor onto skos:Concept;
- mesh:hasRelatedDescriptor onto skos:related;
- mesh:descriptorName onto skos:prefLabel.

There is considerable mismatch between the schemas. Descriptors are the central concepts between which hierarchical relations exist, but it is unclear how MeSH Concepts and Terms can be dealt with. SKOS defines datatype properties with which terms can be recorded as labels of Concepts, but this cannot be mapped meaningfully onto MeSH' Concept and Term classes. For example, mesh:conceptName cannot be mapped onto skos:prefLabel, as the former's domain is mesh:Concept, while the latter's domain is skos:Concept (skos:Concept is already mapped onto mesh:Descriptor). Furthermore, the mesh:Category cannot be mapped onto skos:TopCategory, because skos:TopCategory is a subclass of skos:Concept, while mesh:Category is not a subclass of mesh:Descriptor.

2.5 Case Two: WordNet

This section describes how the method has been applied to WordNet release 2.0. The original source consists of 18 Prolog files (23 MB in total). The conversion programs (written in Prolog) plus links to the original source as well as the output files of each step can be found at `http://thesauri.cs.vu.nl/`. The conversion took two persons approximately three weeks to complete. Note that Step 3 for WordNet is not discussed here for reasons of space, but is available at the forementioned website.

Analysis of WordNet

WordNet (Fellbaum 1998) is a concept-based thesaurus for the English language. The concepts are called "synsets" which have their own identifier. Each synset is associated with a set of lexical representations, i.e. its set of synonyms. The synset concept is divided into four categories, i.e. nouns, verbs, adverbs and adjectives. Most WordNet relations are defined between synsets. Example relations are hyponymy and meronymy.

There have been a number of translations of WordNet to RDF and OWL formats. Dan Brickley[10] translated the noun/hyponym hierarchy directly into RDFS classes and subclasses. This is different from the method we propose, because it does not preserve the original source structure. Decker and Melnik[11] have created a partial RDF representation, which does preserve the original structure. The conversion of the KID Group at the University of Neuchatel[12] constitutes an extension of representation both in scope and in description of semantics (by adding OWL axioms). We follow mainly this latter conversion and relate it to the steps in our conversion method. In the process we changed and extended the WordNet schema slightly (and thus also the resulting conversion).

Converting WordNet

Step 1a: structure-preserving translation.

In this step the baseline classes and properties are created to map the source representation as precisely as possible to an RDF representation:

- Classes: SynSet, Noun, Verb, Adverb, Adjective (subclasses of SynSet), AdjectiveSatellite (subclass of Adjective);
- Properties: wordForm, glossaryEntry, hyponymOf, entails, similarTo, memberMeronymOf, substanceMeronymOf, partMeronymOf, derivation, causedBy, verbGroup, attribute, antonymOf, seeAlso, participleOf, pertainsTo.

Note that the original WordNet naming is not very informative (e.g. "s" represents synset). For readability, here we use the rdfs:labels that have been added in the RDF version. All properties except for the last four have a synset as their domain. The range of these properties is also a synset, except for wordForm and glossaryEntry. Some properties have a subclass of SynSet as their domain and/or range, e.g. entails holds between Verbs.

The main decision that needs to be taken in this step concerns the following two interrelated representational issues:

1. Each synset is associated with a set of synonymous "words". For example, the synset 100002560 has two associated synonyms, namely nonentity and nothing. Decker and Melnik represent these labels by defining the (multi-valued) property wordForm with a literal

[10] http://lists.w3.org/Archives/Public/www-rdf-interest/1999Dec/0002.html
[11] Originally published at http://www.semanticweb.org/library/, not available anymore.
[12] http://www2.unine.ch/imi/page11291_en.html

value as its range (i.e. as an OWL datatype property). The Neuchatel approach is to define a word as a class in its own right (WordObject). The main disadvantage of this is that one needs to introduce an identifier for each WordObject as it does not exist in the source representation, and words are not unique (homonymy).

2. The last four properties in the list above (antonymOf, etc.) do not represent relations between synsets but instead between *particular words* in a synset. This also provides the rationale for the introduction of the class WordObject in the Neuchatel representation: antonymOf can now simply defined as a property between WordObjects.

We prefer to represent words as literal values, thus avoiding the identifier problem (see guideline 11). For handling properties like antonymOf we defined a helper class SynSetWord with properties linking it to a synset and a word. For each subclass of SynSet, an equivalent subclass of SynSetWord is introduced (e.g. SynSetVerb). A sample representation of an antonym is depicted in Figure 2.2.

```
[] a wn:SynSetWord ;
    wn:inSynset wn:100017087 ;
    wn:relevantWord "natural_object" ;
    wn:antonymOf [
        a wn:SynSetWord ;
        wn:inSynSet wn:100019244 ;
        wn:relevantWord "artifact" ;
    ] .
```

Figure 2.2 An RDF/Turtle representation of the antonym relation between natural object and artifact.

In this example, the word natural object in synset 100017087 is an antonym of the word artifact in synset 100019244.

Step 1b: explication of syntax.

The source representation of WordNet does not contain many implicit elements. The only things that need to be added here are the notions of hypernymy and holonymy (three variants). Both are only mentioned in the text and are apparently the inverse[13] of respectively the hyponym relation and the three meronym variants. Consequently, these four properties were added to the schema.

Step 2a: explication of semantics.

In this step additional OWL axioms can be introduced to explicate the intended semantics of the WordNet classes and properties. A selection:

- Noun, Verb, Adverb, and Adjective together form disjoint and complete subclasses of SynSet;

[13]The WordNet documentation uses the term "reflexive", but it is clear that inverseness is meant.

- hyponymOf and hypernymOf are transitive properties;
- hyponymOf only holds between nouns or verbs;[14]
- hyponymOf/hypernymOf and the three variants of meronymOf/holonymOf are inverse properties;
- verbGroup and antonymOf are symmetric properties.

In addition, we defined the properties wordForm and glossaryEntry as subproperties of respectively rdfs:label and rdfs:comment.

From the WordNet documentation it is clear that these properties have this type of intended semantics. The alternative for defining these as subproperties would have been to use rdfs:label and rdfs:comment directly in the RDF representation, thus dropping the original names. This makes the traceability of the conversion less clear.

Step 2b: interpretations.

We have used WordNet heavily in semantic annotation of images (see e.g. (Hollink et al. 2003)). In that application we used the WordNet hierarchy as an RDFS subclass tree by adding the following two metastatements:

- <wn:SynSet, rdfs:subClassOf, rdfs:Class>;
- <wn:hyponymOf, rdfs:subPropertyOf, rdfs:subClassOf>.

Tools such as Triple20[15] will now be able to visualize the synset tree as a subclass tree. The repercussions of this type of metamodeling for RDF storage and retrieval are discussed in (Wielemaker et al. 2003).

2.6 Related Research

Soualmia et al. (2004) describe a migration of a specialized, French version of MeSH to an OWL DL representation. Their goal is to improve search in a database of medical resources. We mention a few of their modeling principles. Firstly, they "clean" the taxonomy by distinguishing between part-of and is-a relationships (the former type are translated into a partOf property, the latter into rdfs:subClassOf). Secondly, qualifiers are translated into properties, and their domains are restricted to the union of the descriptors on which they may be applied. The properties are hierarchically organized using rdfs:subPropertyOf according to the qualifier hierarchy.

Wroe et al. (2003) describe a methodology to migrate the Gene Ontology (GO) from XML to DAML+OIL. Their goal is to "support validation, extension and multiple classification" of the

[14] In the Neuchatel representation an intermediate class NounsAndVerbs is introduced to express these constraints. This is not needed, as OWL supports local property restrictions which allow one, for example, to state that the value range of the hyponymOf property for Noun must be Noun.

[15] http://www.swi-prolog.org/web/Triple20.html

GO. In each step, the converted ontology is enriched further. For example, three new part_of relations are introduced. Also, new classes are added to group part_of instances under their parent components, which enables visualization of the hierarchy. MeSH and the KEGG enzyme database are used to enrich class definitions so a new classification for Gene enzyme functions can be made by a reasoner. Additional modeling of class restrictions allowed the same reasoner to infer 17 is-a relationships that were omitted in the original source.

Goldbeck et al. (2003) describe a conversion of the NCI (National Cancer Institute) Thesaurus from its native XML format to OWL Lite. Their goal is to "make the knowledge in the Thesaurus more useful and accessible to the public". A mapping of XML tags onto OWL classes and properties is defined, based on an analysis of the underlying conceptual model and NCI's thesaurus development process. rdf:IDs are created using a transformation of the original concept names (spaces removed, illegal characters substituted). This is a reasonable approach, under the assumption that names are indeed and will remain unique (see guideline 11).

There are two main differences with our work. Firstly, the forementioned projects do not separate between "as-is" conversion and enrichment steps, as our method does. Therefore, the conversions may only be usable for the project's own goals. Secondly, we try to generalize over specific conversions and aim to define a more general conversion process.

In the SWAD-Europe project a schema is being developed to encode RDF thesauri. Additionally, work is in progress to produce a guideline document on converting multilingual thesauri (Miles et al. 2004a). The development of a standard schema influences our standardization step, and guidelines on converting multilingual thesauri may be incorporated to broaden the scope of our method.

2.7 Discussion

This paper has presented a method to convert existing thesauri to RDF(S) and OWL, in a manner that is sanctioned by the original documentation and format. Only in a separate step may interpretations be made for application-specific purposes.

Two additional aims of converting existing resources to the RDF model may be identified. Firstly, the quality of a thesaurus can be improved using the semantics of RDF(S), as some thesauri use relations with weak or unclear semantics, or apply them in an ambiguous way (Soergel et al. 2004). Secondly, converted thesauri can be checked using standard reasoners, identifying e.g. missing subsumption and inverse relations (e.g. BT/NT).

Recently the W3C has installed the Semantic Web Best Practices and Deployment (SWBPD) Working Group[16] , which aims to provide guidelines for application developers on how to deploy Semantic Web technology. This method may serve as input for and might be extended by this Working Group.

Several issues remain as future research in developing a methodology. Firstly, translating between the source model and the RDF model is a complex task with many alternative mappings, especially for thesauri that do not conform to or define extensions of the ISO and ANSI standards.

[16] http://www.w3.org/2001/sw/BestPractices/

Secondly, more guidelines are required on how to convert multilingual thesauri. Thirdly, a standard thesaurus schema is required to perform step three of our method. It is clear that the current SWAD-E proposal does not completely cover complex thesauri such as MeSH. An open question is whether the proposal might be extended or that this type of thesaurus is simply outside the scope, as MeSH deviates from the ISO and ANSI standards.

Acknowledgements

This work was partly supported by the IST project IST-2002-507967 “HOPS” and the CHIME project, part of the NWO ToKeN2000 programme. The authors are grateful for the useful discussions with and feedback from their colleagues.

3

A Method for Converting Vocabularies to the SKOS Metamodel

In this chapter we again address research question 1, "How can vocabularies be converted to an interoperable representation in an application-neutral way?" In this chapter a predefined metamodel called SKOS is used, which requires a dedicated method (different from the generic one presented in the previous chapter). The SKOS metamodel and associated schema is a developing standard for representing vocabularies. A standard schema simplifies sharing and reusing vocabularies, but places limitations on what can be represented. Specifically, some structural features of more complex vocabularies cannot be represented, so that content and semantics cannot be preserved. Our approach is the same as in the previous chapter: a method outline is built based on past experiences, which is then applied to new cases (here: IPSV, GTAA and MeSH). The cases are used to evaluate our method and also the applicability of SKOS for representing vocabularies.

An overview of the method is given in Appendix C.

This chapter was published in the Proceedings of the Third European Semantic Web Conference, and was co-authored with Véronique Malaisé, Alistair Miles and Guus Schreiber (Van Assem et al. 2006b).[1]

3.1 Introduction

Thesauri and thesauri-like resources such as MeSH (Johnston et al. 1998) and the Art and Architecture Thesaurus (Peterson 1994) are controlled vocabularies developed by specific communities, often for the purpose of indexing (annotation) and retrieval (search) of resources (images, text documents, web pages, video, etc.). They represent a valuable means for indexing, retrieval and simple kinds of reasoning on the Semantic Web. Most of these resources are represented in databases, as XML files, or some other special-purpose data format. For deployment in Semantic Web applications an RDF/OWL representation is required. Thesauri can be converted to RDF/OWL in different ways. One conversion might define a thesaurus metamodel which represent terms as instances of a class Term, while another converts them into literals contained in a property term. This can introduce structural differences between the conversions of two thesauri which have the same semantics. Using a common framework for the RDF/OWL representation of thesauri (and

[1] The following changes were made: a diagram with an example of SKOS was added, a footnote with reference to the current version of SKOS was added, wording of some sentences was changed.

thesauri-like resources) either enables, or greatly reduces the cost of (a) sharing thesauri; (b) using different thesauri in conjunction within one application; (c) development of standard software to process them (because there is no need to bridge structural differences with mappings). However, there is a significant amount of variability in features of thesauri, as exemplified by the case studies presented here. The challenge for a common metamodel such as SKOS is to capture the essential features of thesauri and provide enough extensibility to enable specific, locally-important thesaurus features to be represented.

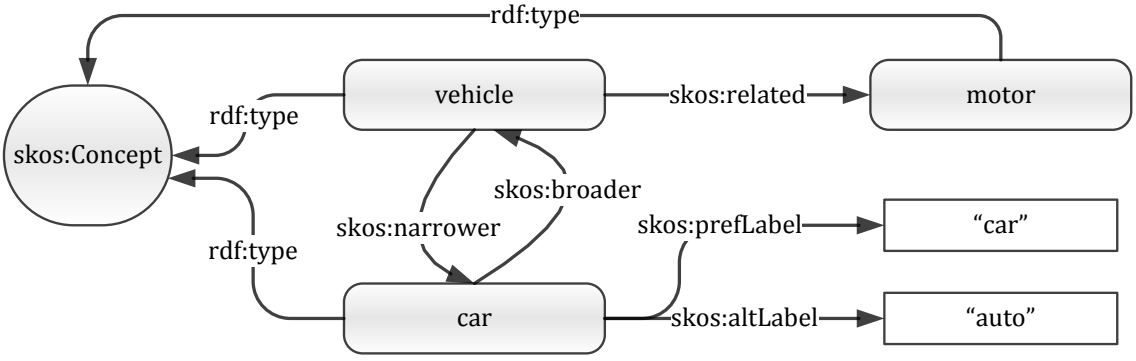

Figure 3.1 An example of three thesaurus concepts and their relationships in the SKOS metamodel. Thesaurus concepts (instances) are shown in boxes with round corners, instances in boxes with straight corners.

The SKOS Core Guide (Semantic Web Best Practices and Deployment Working Group 2005b) and the SKOS Core Vocabulary Specification (Semantic Web Best Practices and Deployment Working Group 2005c) are currently Working Drafts for W3C Working Group Notes[2]. They present the basic metamodel consisting of an RDF/OWL schema, an explanation of the features that the properties and classes of the schema represent. Guidelines and examples for extending SKOS Core are given by a proposed draft appendix to the SKOS Core Guide[3] and another draft proposes additional properties for representing common features in thesauri.[4] Because they are at the proposal stage they have no formal status within W3C process as yet. For the purpose of this paper we take these four documents to represent the SKOS metamodel and guidelines. Together they define (in a non-formal way) what constitutes a "correct" SKOS RDF document. SKOS models a thesaurus (and thesauri-like resources) as a set of skos:Concepts with preferred labels and alternative labels (synonyms) attached to them (skos:prefLabel, skos:altLabel). Instances of the Concept class represent actual thesaurus concepts can be related with skos:broader, skos:narrower and skos:related properties (see Figure 3.1 for an example). This is a departure from the structure of many existing thesauri that are based on the influential ISO 2788 standard published in 1986, which has *terms* as the central entities instead of concepts. ISO 2788 defines two types of terms (preferred and non-preferred) and five relations between terms: broader, narrower, related, use and

[2]Since this chapter was written SKOS has become a W3C Recommendation, see `http://www.w3.org/TR/skos-reference/`

[3]`http://isegserv.itd.rl.ac.uk/cvs-public/~checkout~/skos/drafts/appextensions.html`

[4]`http://www.w3.org/2004/02/skos/extensions/spec/2004-10-18.html`

use for. "Use" and "use for" are allowed between preferred and non-preferred terms, the others only between preferred terms (International Organization for Standardization 1986). More recent standards such as ANSI/NISO Z39-19 acknowledge that terms are "lexical labels" representing concepts, but still use a term-based format (ANSI/NISO 2003). Often it is possible to convert a term-based thesaurus into a concept-based one, but sometimes information is lost (examples appear in the paper). The standards (including SKOS) allow polyhierarchies, i.e. a term/concept can have more than one broader term/concept.

Careful analysis of a thesaurus may still not be enough to produce an errorless, interoperable conversion to SKOS. To help ensure the quality and utility of conversions a structured method is required. This paper addresses a methodological research question: given the SKOS metamodel for thesauri, can a step-wise method be developed that assists in converting thesauri to this metamodel in a correct manner? The method should be able to guide the development of a deterministic program (i.e. does not require human intervention) that generates correct SKOS RDF for a specific thesaurus. We address the research question by first by examining existing thesaurus conversion methods in Section 3.2. Secondly, we develop our method by refining an applicable existing method in Section 3.3. Thirdly, we apply our method to three thesauri in Sections 3.4 through 3.6. Fourthly, we evaluate our method and the SKOS metamodel in Section 3.7.

3.2 Existing Thesaurus Conversion Methods

This section discusses existing methods to convert thesauri. We distinguish conversion methods for specific thesauri, methods that convert thesauri to ontologies and methods that convert any thesaurus to RDF/OWL.

A first stream of research presents methods to convert one specific thesaurus from its native format to RDF/OWL, such as for MeSH (Soualmia et al. 2004) and the NCI thesaurus (Goldbeck et al. 2003). Although the steps and techniques developed for these methods are useful in thesaurus conversion, it is not clear if they can be applied to other thesauri because only features that appear in the specific thesaurus are covered. We do not consider these methods when choosing a method to base ours on.

A second stream of research presents methods with the goal to convert any thesaurus into an *ontology*, such as the work of Soergel et al. (2004). A major difference between thesauri and ontologies is that the latter feature logical is-a hierarchies, while in thesauri the hierarchical relation can represent anything from is-a to part-of. Their method has three steps: (1) define ontology metamodel; (2) define rules to convert a traditional thesaurus into the metamodel, introducing more specific kinds of relationships; and (3) manual correction. The main goal of the method is to refine the usual thesaurus relationships into more specific kinds of relationships such as "causes", "hasIngredient" and "growsIn". The method does not target a specific output format, although hints are given for conversion to RDFS. It is not clear if the method would convert thesaurus concepts into rdfs:Classes with rdfs:subClassOf and other relations between them, or rather as instances of a class skos:Concept as in SKOS.

An elaborate 7-step method is defined by (Hyvönen et al. 2005)[5] with the goal of creating a true ontology consisting of an RDFS or OWL class hierarchy. Thesaurus concepts are converted into instances of a metaclass (a subclass of rdfs:Class) so that they are simultaneously instances and classes. The main goal of the method is that conversion refines the traditional BT/NT relationships into rdf:type, rdfs:subClassOf or partOf. Another goal is to rearrange the class hierarchy to better represent an ontological structure, e.g. to ensure only the real root concepts do not have a parent. Besides refining the relations it retains the original structure by also converting the BT/NT/RT relations into equivalent RDFS properties. It does not currently use SKOS.

A third stream of research presents methods to convert thesauri into RDF/OWL without creating an ontology (i.e. an "is-a" hierarchy"). Earlier work by Van Assem et al. (2004) describes a method to convert thesauri in four steps: (1) preparation; (2) syntactic conversion; (3) semantic conversion; and (4) standardization. In the first step, an analysis is made of the thesaurus and its digital format. This is used in step two to convert to very basic RDF, after which it is converted to more common modeling used in RDF and OWL in step three. In the last step the RDF/OWL metamodel developed for the specific thesaurus is mapped to SKOS. This method has the goal of preservation of the thesaurus' original semantics. The approach is step-wise refinement of the thesaurus' RDF/OWL metamodel.

Work by Miles et al. (2004b) defines a method to convert thesauri to an earlier version of SKOS in three steps: (1) generate RDF encoding; (2) error checking and validation; and (3) publishing encoding on the web. Three case studies illustrate the method. It is based on two goals: (a) supporting thesaurus interoperability through usage of SKOS; and (b) preservation of all information encoded in the thesaurus. The first step is separated into conversion of thesauri with a "non-standard structure" or "standard structure". Thesauri with "standard structure" are based on the ISO 2788 standard. Such thesauri can be converted into instances of the SKOS schema without loss of information. Thesauri with "non-standard structure" are those who have "structural features that are not described by the standard ISO 2788". The recommendation is to develop an extension of the SKOS schema using rdfs:subClassOf and rdfs:subPropertyOf to support non-standard features as this solution ensures that both method goals are met. The method and described cases does not admit of a third category of thesauri, namely those with non-standard structure which cannot be defined as a strict specialization of the SKOS schema (this paper shows examples of these). The second step comprises error checking and validation using the W3C's RDF validator, while the third step is not discussed further.

3.3 Development of Conversion Method

The development of our method is based on a tentative process with the following components: (a) defining goals of the method; (b) comparing to existing methods and choosing an applicable one; (c) developing the steps of our method; (d) applying the method; and (e) evaluating the method. This section presents the first three components. We apply the method in Sects. 3.4 through 3.6 and evaluate in the discussion. We restrict the scope of our method to monolingual thesauri and do not

[5] In Finnish, our understanding is based on correspondence with the author.

discuss thesaurus metadata. We also ignore some practical issues such as defining an appropriate namespace for the converted thesaurus.

Method goals

The overall goal of the method is to support interoperability of thesauri encoded in RDF/OWL. The first subgoal of the method is to produce conversion programs that convert the digital representations of a specific thesaurus to SKOS. The underlying assumption is that converting to SKOS provides interoperability. A subgoal that follows is that the resulting conversion program should produce correct SKOS RDF. The second subgoal of the method is that the converted thesaurus is complete (i.e. has all information that is present in the original) as long as this does not violate the previous goal. For this method we value the goal of interoperability higher than the goal of being complete.

Comparison with existing methods

Here we compare the goals to those of existing methods to choose a suitable one to use as a basis for our own. The method by Soergel et al. does not have interoperability of thesauri as a goal. For each thesaurus a new metamodel is developed. Its main goal is to produce a more refined version of the thesaurus. This is not in opposition to our goal of completeness, but does introduce more work than necessary to achieve our main goal and may also introduce incorrect interpretations of the thesaurus' relations.

In Hyvönen's method the thesaurus is converted into a rearranged class hierarchy. It does not use a standard metamodel such as SKOS to promote interoperability and it rearranges the thesaurus' original structure. The method by Van Assem et al. also does not have interoperability of thesauri as a goal. The metamodels of different thesauri converted using this method may have structural differences. The method by Miles et al. has the same goal as ours: interoperability of thesauri in RDF/OWL. The stated goals of using SKOS and of completeness also match. A difference is that it does not acknowledge possible conflicts between these goals.

Developing steps of the method

The method by Miles et al. has comparable goals and therefore we take their method as a starting point and adapt it. We focus here on working out the first step of the method, namely producing a conversion ("encoding") of the thesaurus in correct SKOS RDF. We do not adapt and discuss steps two and three.

The first step in the method by Miles et al. is split in two different processes depending on whether the thesaurus is "standard" or "non-standard". This requires an analysis of the thesaurus, so we include this as a separate activity in our method. Furthermore, the two processes only differ on whether they convert directly to instances of the SKOS schema or into extensions of the SKOS schema (defined with rdfs:subPropertyOf and rdfs:subClassOf). We decide to merge the two processes, and for each thesaurus feature in the analysis we determine whether to use a class/property from the SKOS schema or define a new subclass/subproperty.

Substep	Activity	Output
(A) thesaurus analysis	analyze digital format, analyze documentation	catalogue of data items and constraints, list of thesaurus features
(B) mapping to SKOS	define data item to SKOS schema mapping	tables mapping data items to schema items
(C) conversion program	develop algorithm	conversion program

Table 3.1 Substeps and activities of step 1.

We analyzed which activities need to be performed in the step, starting with its inputs and outputs. The input of the step is the thesaurus digital format, and its documentation (including interviews with experts and applications that use the thesaurus such as websites). The output of the step should be a program that transforms the data from the original digital format to SKOS RDF. In some cases the output of the step will also include an extension of the SKOS schema. There are three activities to be performed that link output to input: (1) creating an (algorithm for the) transformation program; (2) defining a mapping between input data items and output SKOS RDF as a basis for the algorithm; and (3) analyzing the thesaurus. We split the last activity into two parallel analyses: an analysis of the digital format and of the documentation. Both are helpful to understand which features the thesaurus has and how they are encoded. This results in the substeps and activities summarized in Table 3.1.

For the thesaurus analysis, we have listed the set of features that appear in common thesauri. We derived this set from studying thesaurus standards (International Organization for Standardization 1986, ANSI/NISO 2003) and the SKOS documentation listed earlier. There are three sets: one specific to term-based thesauri, one specific to concept-based thesauri and one set that is used in both. *Term-based features* are: term, compound term (combination of two or more terms), "use" relation, "use for" relation, broader term relation between preferred terms, narrower relation between preferred terms, scope note attached to preferred term (indicates scope for which term can be used in indexing), documentation attached to terms such as definitions and historical notes. *Concept-based features* are: concept, compound concept, preferred labels, non-preferred labels, broader concept relation, narrower concept relation, documentation attached to concepts such as definitions and historical notes. General features are: node labels (explained later), facets (a top-level named group of terms or concepts that is not meant for use in indexing itself). SKOS is a concept-based model. Therefore, any feature that cannot be converted into a concept-based or generic feature falls outside the scope of the SKOS schema and thus of SKOS interoperability. Although most term-based features in their most basic form can be converted into concept-based features, there are exceptions.

A sub-activity we would like to highlight here is the identification of unique identifiers in the source to generate the `rdf:ID`s of skos:Concepts. Some thesauri like MeSH already provide unique identifiers, but others like GTAA do not provide one. Two options are: (a) generate completely new identifiers which have no relation to the terms or concepts themselves; (b) use the name of the preferred term if it is unique (replacing illegal URI characters). The first option has the disadvantage of additional management (a mapping between source terms and identifiers

needs to be maintained). The second option has the disadvantage that a concept is not independent of its name. Additional programming is required to ensure that when a term changes name, the corresponding skos:Concept's label is changed, instead of its URI. Currently we have not found a particular reason to prefer one option over the other.

In the next three sections we apply the method to three thesauri. We have chosen IPSV, GTAA and MeSH because they (a) are used in practice; and (b) represent progressively complex thesauri (i.e. non-standard features). The progressive complexity allows us to explore the limitations of our method and of SKOS.

3.4 Case Study: IPSV

The Integrated Public Sector Vocabulary (IPSV) is a thesaurus developed in the UK for indexing government documents.[6] It is modeled with the ISO2788/BS5723 standards in mind and contains 2732 preferred terms and 4230 non-preferred terms. The IPSV is a result of the merger of three thesauri. The sources and results of the conversion are available on-line.[7]

Step A: analyze thesaurus.

We used the XML version[8] in our analysis as it is the most complete. IPSV-XML has a DTD which provides the catalogue of data items and their constraints. IPSV-XML is a reasonably standard term-based thesaurus with preferred and non-preferred terms both called `<Item>`s in the XML data. Columns one and three of Table 3.2 list the data items and the features (for non-standard features we describe the function instead). IPSV provides unique identifiers for its terms and has a polyhierarchy.

Step B: map data items to SKOS.

We have analyzed which data items correspond to which SKOS features or specializations of them (column three of Table 3.2). Although polyhierarchies are not allowed in ISO 2788, this is allowed in SKOS so this does not hinder a correct conversion. We were not able to find appropriate (specializations of) SKOS properties for the last four data items in the table. The two data items that indicate version information for terms cannot be made subproperties of skos:altLabel or skos:prefLabel as done for the AToZ attribute, because there is no place to store the version number (only literals are allowed for the label properties). A solution would be to attach two new properties to skos:Concept that have instances of a class Term as range. To these instances we can then attach a property that repeats the term name and then another property with the version number. Although this solution represents the information correctly, it introduces redundancy into the conversion (it repeats the term name with non-SKOS classes and properties). If this is not an issue this solution can be used to remain complete. However, it is a structural work-around

[6] `http://www.esd.org.uk/standards/ipsv/`

[7] `http://thesauri.cs.vu.nl/eswc06/`

[8] Also available in other formats, see `http://www.esd.org.uk/documents/IPSVVersionsAndFormats.pdf`

Data Item	Feature/function	Property/class
<Item Id="*A*" ConceptId="*B*" Preferred="True"> <Name: >*X*	Preferred Term	skos:Concept with `rdf:ID`=A, skos:prefLabel=X attached to it
<Item Id="*A*" ConceptId="*B*" Type="Synonym"> <Name: >*X*	Non-Preferred Term	skos:prefLabel=X attached to concept with `rdf:ID`=B
<Item Type="misspelling"> <Name: >*X*	common misspelling of a (non)preferred term	skos:hiddenLabel=X
<UseItem>	USE relation	none required
<ScopeNote>*X*	ScopeNote	skos:scopeNote=X attached to concept created for surrounding `<Item>`
<BroaderItem Id="*X*">	Broader Term	skos:broader to Concept with `rdf:ID`=X
<RelatedItem ConceptId="*X*">Y	Related Term	skos:related to Concept with `rdf:ID`=X
<BroaderItem Id="*X*" Default="true">	default broader term	ipsv:broaderDefault (subproperty of skos:broader) to Concept with `rdf:ID`=X
<Item AToZ="true" Preferred="*Y*">Z	term should be displayed on websites	ipsv:displayableAltLabel=Z (subproperty of skos:altLabel) when Y=false, ipsv:displayablePrefLabel=Z (subproperty of skos:prefLabel) when Y=true
<Item Obsolete="true">X	obsolete term	ipsv:obsoleteTerm=X (subproperty of skos:hiddenLabel
<Item AddedInVersion="*X*">	X is a real indicating in which IPSV version the term was added	
<Item LastUpdatedInVersion="*X*">	X is a real indicating in which IPSV version the term was last changed	
<Shortcut>*X*	X is a letter; keyboard shortcut for an application	ipsv:shortcut attached to concept created for surrounding `<Item>`

Table 3.2 Mapping of IPSV Data Items to features and RDFS property/classes. The upper part lists standard features, the middle part specializations and the lower part non-standard features. Closing tags in Data Item column are omitted.

because SKOS does not have the ability to attach information on specific skos:prefLabels and skos:altLabels directly.

Items that are Obsolete are removed from the actual thesaurus but are retained to be able to retrieve documents that were indexed with older versions of the thesaurus. The skos:hiddenLabel is intended to contain labels that should not be displayed to users but should be available for retrieval purposes, so we create an ipsv:obsoleteTerm that is a subproperty of skos:hiddenLabel. Shortcuts are attached to terms in the XML, but are actually meant to be able to insert a whole concept within an application, so it is attached to skos:Concept as a non-standard feature without a SKOS superproperty.

Step C: create conversion program.

We created a SWI-Prolog program that parses the IPSV-XML file and converts it to SKOS RDF using the mappings from step 1b. The program takes an <Item> and applies the matching mappings between data items and SKOS RDF. There is no need for any other information external to the <Item> to generate the triples for that Item. For example, because non-preferred Items also contain the identifier of their preferred Item (in the ConceptId attribute), we can generate the skos:altLabel triple even if the preferred Item that is used to generate the skos:Concept is not yet processed.

Case study summary.

The case study took one analyst approximately two weeks to perform and was not very complex as the thesaurus is not complicated and clearly documented. For a few issues one of the original developers was contacted. A lesson learnt is that it is not always possible to perform a complete information-preserving conversion. Some information on terms was lost.

3.5 Case Study: GTAA

The GTAA thesaurus is the controlled vocabulary used at The Netherlands Institute for Sound and Vision[9] , which archives and indexes most of the public broadcasted TV and radio programs of the Netherlands.[10] GTAA stands for *the Common Thesaurus for Audiovisual Archives*; it is the result of the collaborative work of different institutions concerned with audiovisual document indexing, including the FilmMuseum of Amsterdam. It contains 159,831 preferred terms, 1,900 non-preferred terms, and 88 categories. A sample of the source file, the conversion program and the resulting RDF are available on-line.[11]

[9] http://www.beeldengeluid.nl/

[10] Of the estimated 850,000 hours of audio-visual material that is preserved in the Netherlands, around 700,000 hours is archived by Sound and Vision.

[11] http://thesauri.cs.vu.nl/eswc06/

Data Item	Feature/function	Property/class
Term A	Preferred Term	skos:Concept with `rdf:ID`=A, skos:prefLabel=A attached to it
US Term B	Non-Preferred Term	skos:altLabel=B attached to concept
CC Category C	Grouping of Preferred Terms by Categories	skos:member between a skos:Collection (with `rdf:ID`=C) and a skos:Concept
BT Term A	Broader Term	skos:broader
NT Term A	Narrower Term	skos:narrower
RT Term A *or* See also	Related Term	skos:related
SN X *or* (X)	ScopeNote	skos:scopeNote=X attached to concept created for surrounding Preferred Term
LT	relationship between terms from different facets	gtaa:hasLinkedTerm (subproperty of skos:related)
DL	relationship between terms within a certain time period	gtaa:hasDebateLine (subproperty of skos:related)

Table 3.3 Mapping of GTAA Data Items to features and RDFS property/classes. Upper part lists standard features, the lower part specializations. "Term A" is an actual term in the thesaurus such as "Boat"

Step A: analyze thesaurus.

We had access to GTAA documentation and data as text files with an ISO-style formatting. This thesaurus is a faceted term-based thesaurus, where only one facet (the Subject facet, used to describe the content of a program) is organized with the ISO 2788 broader term/narrower term hierarchy. The other facets are alphabetical controlled lists, with some scope notes (lists of people's names, geographical location, etc.). The Subject facet contains one non-standard feature called Category. Each term is supplied with at least one Category, providing an alternative way to the normal NT/BT hierarchy for indexers to find them. We list GTAA data items in column one of the upper part of Table 3.3 and the features they represent in column two.

Step B: map data items to SKOS.

Two issues arose in this step. The first one concerns the GTAA BT relationship. In the documentation of the thesaurus, the BT and NT relationships are stated to be each other's inverse. In the data itself, two or more preferred terms can have a NT link with the same narrower term. However, this narrower term has only one BT link to one of the broader terms (instead of multiple BT links). There are two options: either the missing BT links are intended but omitted in the data, or the BT link has a special status, e.g. it is a defaultBroader such as in IPSV. After discussion with GTAA experts, and according to the fact that this defaultBroader relationship does not appear in the documentation, we mapped the GTAA BT to skos:broader (see column three of Table 3.3).

Secondly, there are two ways to interpret the CC relationship. Either it is meant to disambiguate different aspects of a term (as in "Church-institution" *vs.* "Church-building"), or it is a way of grouping terms sharing a specific aspect (as with "Milk_by_animal" and "Cow-milk", "Buffalo-milk", etc.). In the second case, "Milk_by_animal" is called a node label: it is a way of

grouping terms, but it should not be used for indexing. These node labels are usually part of the term hierarchy. The experts indicated that this option was the intended usage of Categories: to provide a grouping of terms under a label that is not used in the indexing process. Nevertheless, they are meant to provide an *alternative* grouping of the GTAA terms, and thus are not part of the BT/NT hierarchy. Although we mapped the Categories to an existing SKOS construct, namely the skos:Collection (see column three of Table 3.3), this modelling remains a non-standard feature that cannot be processed by SKOS software. The Categories have explicit identifiers, from which we could infer their hierarchy (01 stands for Philosophy, and 01.01 is one of its subdivisions, for instance).

GTAA does not include identifiers for its terms, so we used the preferred term's name as the `rdf:ID` of concepts.

Step C: create conversion program.

As our source for the GTAA data was plain text, we created a Perl program to convert it according to the mappings in Table 3.3. We also had to make some manual corrections for reference errors introduced by thesaurus maintenance. Some relationships were referring to terms of the thesaurus that became obsolete, to terms which changed spelling, or to terms that became non-preferred terms. We corrected the references, or suppressed the relationships when no reference could be found; as these are relatively straightforward decisions no expert involvement was necessary.

Case study summary.

The conversion could be made by direct mapping to or by extension of the SKOS schema, except for the Categories. In the conversion process, understanding the GTAA model from textual resources and experts interview, and converting the Categories into a SKOS construct took the longest time. Including programming, the process took about two weeks for one analyst.

3.6 Case Study: MeSH

The Medical Subject Headings (MeSH) is a large thesaurus-like vocabulary developed by the U.S. National Library of Medicine and used to index millions of biomedical article citations.[12] It contains 22,997 "descriptors", most of which are used to index the subject of articles (two of the sixteen trees do not contain subjects but publication types and geographical regions). MeSH is the result of a merger of many different sources. The input data files and results of the conversion are available on-line.[13]

Step A: analyze thesaurus.

MeSH is available in different formats which contain the same information. We chose the XML version[14] because it is easier to analyze and convert. MeSH-XML has a DTD which provides us

[12] `http://www.ncbi.nlm.nih.gov/pubmed`

[13] `http://thesauri.cs.vu.nl/eswc06/`

[14] `http://www.nlm.nih.gov/mesh/filelist.html`

with the data catalogue and constraints. MeSH is a concept-based thesaurus without facets. Concepts are called "Descriptors" in MeSH terminology. The MeSH structure is complicated: "Descriptors" contain "Concepts", "Concepts"contain "Terms". Each has a name and a unique identifier, and to each entity documentation is attached such as its date of introduction and historical notes. Descriptors are hierarchically related: each MeSH Descriptor has one or more "TreeNumbers", which implicitly encode its position in a polyhierarchy (e.g. A01.456 is a child of A01). Each Descriptor has a preferred Concept, and each Concept has a preferred Term. MeSH Concepts that appear within one Descriptor can be related to each other with relations "brd", "nrw" and "rel". MeSH has fifteen trees with top-concepts named e.g. "organisms" or "diseases". These appear to be facets, but they are used in indexing articles so we interpret them as normal thesaurus Concepts.

As the MeSH DTD defines almost 90 tags[15] and for each tag different attributes, we only list the exemplary and special data items in column one of Table 3.4 (the corresponding feature, or function if it is a non-standard feature, is in column two). MeSH Descriptors have a redundant <DescriptorName> and <ConceptName> as these strings are the same as the name of the preferred Concept and Term, respectively.

MeSH has two non-standard features that require special attention. Firstly, so-called Qualifiers are used to indicate specific aspects of Descriptors, such as "pathology" or "abnormalities". They are combined with Descriptors to enable more specific article indexing (e.g. "Abdomen/abnormalities"). Secondly, so-called EntryCombinations relate a non-preferred Descriptor/Qualifier pair to a preferred Descriptor/Qualifier pair (or preferred Descriptor without Qualifier). This is comparable to but slightly different from the ISO 2788 "USE" relation, which can be used to point from a non-preferred non-compound term to a preferred compound term. The difference is that in MeSH the non-preferred concept is a compound.

Step B: map data items to SKOS.

We mapped Descriptor to skos:Concept instances and sub-tags to properties of skos:Concept (see Table 3.4). Each child Descriptor is linked to its parent(s) - stated implicitly in the <TreeNumber> tag(s) - with skos:broader. We only map Descriptor names (DescriptorName, ConceptName) one time, removing the redundancy.

Because the MeSH Concepts and Terms are converted into skos:prefLabel and skos:altLabels, information about the Concepts and Terms themselves is lost. One example is the Concept's "brd", "nrw" and "rel" relations. These cannot be mapped to the broader/narrower concept feature, because the Descriptor hierarchy is already mapped to that. Two more examples are the Term's <Abbreviation> and <LexicalTag>. Only in cases where it is valid to attach information about a Concept or Term to the Descriptor can this information be preserved by attaching it to the skos:Concept, which is not the case for a number of Concept and Term tags. An example where this *is* possible is with a preferred Concept's <ScopeNote>.

To support the use of Descriptor/Qualifier pairs in indexing we introduced classes mesh:Qualifier and mesh:CompoundConcept as subclass of skos:Concept. Qualifiers are a special class

[15] An overview of their meaning is given in: http://www.nlm.nih.gov/mesh/xml_data_elements.html

Data Item	Feature/function	Property/class
<DescriptorRecord> <DescriptorName> <String>*X* <DescriptorUI>*Y*	Concept	skos:Concept with `rdf:ID`=Y and skos:prefLabel=X
<Concept PreferredConceptYN="Y"> <ScopeNote>*X*	Scope Note	skos:scopeNote=X attached to concept created for surrounding <DescriptorRecord>
<TreeNumber>*X*	implicitly indicates Broader Concept	skos:broader to concept with `rdf:ID`=X
<Term RecordPreferredTerm="N"> <String>*B*	Non-preferred Label	skos:altLabel=B attached to concept with `rdf:ID` found in surrounding Descriptor
<SeeRelatedDescriptor> <DescriptorReferredTo> <DescriptorUI>*X* <DescriptorName>*Y*	Related Concept	skos:related to Concept with `rdf:ID`=X
<HistoryNote>X	Historical Note	mesh:historyNote=X (subproperty of skos:historyNote)

Data Item	Feature/function	Property/Class
<EntryCombination> <ECIN> *X* <ECOUT> *Y*	Compound Concept and special relation (see text). X and Y contain tags with the identifiers of one Descriptor/Qualifier pair in them	mesh:CompoundConcept mesh:Qualifier mesh:main mesh:qualifier (subclasses and subproperties of skos:Concept and skos:broader) mesh:preferredCombination (no parent)
<PublicMeSHNote>X	Note mixing historical and see also information	mesh:publicMeSHNote=X (subproperty of skos:note)
<PreviousIndexing>X	Historical Note	skos:historyNote
<ConsiderAlso>X	textual reference to other possible records	mesh:considerAlso=X (subproperty of skos:note)
<ActiveMeSHYear>X	Year in which the Descriptor was part of MeSH	mesh:activeMeSHYear=X (subproperty of skos:editorialNote)
<RecordOriginator>X	Thesaurus where the Descriptor comes from	mesh:recordOriginator (suproperty of skos:note)
<DateCreated>X	Date Descriptor was first created	mesh:dateCreated=X (subproperty of skos:editorialNote

Data Item	Feature/function	Property/class
<ActiveMeSHYear>	Year in which Descriptor was present in MeSH	mesh:activeMeSHYear
<DescriptorClass>	Classifies Descriptor into one of four numbered categories, including "topical descriptor" and "publication type"	mesh:descriptorClass
<RunningHead>	page header used in printed MeSH versions	mesh:runningHead
<LexicalTag>	lexical category of a <Term>	
<Abbreviation>	abbreviation of a <Term>	

Table 3.4 Mapping of representative MeSH Data Items to features and RDFS property/classes. Upper part lists standard features, the middle part specializations and lower part non-standard features. Omitted closing tags in Data Item column.

of Concepts because they do not have broader/narrower relations themselves. The properties mesh:main and mesh:qualifier are used to attach a Descriptor (skos:Concept) and Qualifier (mesh:-Qualifier) to the CompoundConcept. By making the properties a subproperty of skos:broader, the CompoundConcepts become narrower concepts of their contained concept, so that queries for documents with that concept as subject will also return documents indexed with the CompoundConcept. For the `rdf:ID` of the CompoundConcept the unique Descriptor and Qualifier identifiers are concatenated. We used the same CompoundConcept class to represent `<EntryCombination>`s which we link with mesh:preferredCombination. This last property does not have a SKOS parent. The only candidate skos:related has a different semantics: it links preferred concepts that are related in meaning (a symmetric relation), while mesh:preferredCombination links a non-preferred concept to a preferred concept (asymmetric relation).

Step C: create conversion program.

We created a SWI-Prolog program that parses the MeSH-XML file and converts it to SKOS RDF using the mappings from step B. The program takes a DescriptorRecord tag and converts it into a skos:Concept. It also converts the non-standard features of MeSH.

Case study summary.

The case study took one analyst approximately two weeks to perform and was relatively complex because of the many non-standard features and ambiguities. We have not yet been able to confirm our decisions with MeSH experts. We learned that some thesauri have complex structures for which no SKOS counterparts can be found (e.g. information on Terms) and that for some features care is required in converting them in such a way that they are still usable for their original purpose (e.g. the CompoundConcepts).

3.7 Discussion and Evaluation

In this section we first evaluate our method and then discuss the applicability of the SKOS metamodel for representating thesauri. The case studies showed that the method gives appropriate guidance in identifying common features of thesauri. However, we found that two of our three cases had non-standard features which our method cannot anticipate. Further case studies should increase the number of identified non-standard features to be incorporated into the method. For the analysis of the meaning of some features it is necessary to investigate how the feature is used in practice (e.g. GTAA Categories). Conversion of concept-based thesauri should be simpler than term-based thesauri as SKOS is concept-based, but we cannot confirm this as MeSH is not a typical example of the first category. Although MeSH was not a good choice as a case study in this respect, it did help us in identifying the boundaries of applicability of SKOS (see below). A problematic type of feature are textual notes that mix several kinds of knowledge (e.g. `<PublicMeSHNote>` contains historical and see also information). Our method does not investigate if it is possible to separate them. We are currently unsure whether such an investigation will result in generic rules that can be incorporated in our method.

The SKOS metamodel itself seems applicable for representing resources which have considerable resemblance to the ISO 2788 standard. From the MeSH case we learned that SKOS does not have a standard class to represent compound concepts, although this is a feature that is defined in ISO 2788. A related ISO feature, the USE relation from non-preferred compound terms to preferred ones has no SKOS counterpart either. Thesauri such as IPSV and MeSH also represent management information about their terms (e.g. date of term creation) which cannot be represented within SKOS itself.[16] One might argue that this information is not relevant to a thesaurus' content. It may represent information on a higher level of abstraction that should not be considered for conversion. However, SKOS does partly supports representing other types of management information e.g. with the skos:changeNote and skos:editorialNote. Besides management information, there is also additional content information on terms that cannot be represented in SKOS, such as the MeSH `<LexicalTag>`. If it is appropriate to represent additional information on terms, a solution is to introduce into SKOS a new class skos:Term as the range of skos:prefLabel and skos:altLabel. This would enable terms to be entities in themselves to which additional properties can be attached.

Lastly, we note that it is difficult to confirm whether or not a given RDF document is valid SKOS RDF. The draft SKOS Test Set[17] and implementation[18] can simplify this in the future.

Acknowledgements

This work was partly supported by the BSIK Multimedian e-Culture project, and NWO's CHIME and CHOICE projects. The authors wish to thank Stella Dextre Clarke for providing information concerning the IPSV and Eero Hyvönen for correspondence on his method. The authors also thank all participants of *public-esw-thes@w3c.org* who have contributed to the development of SKOS.

[16] This problem has been resolved through an extension called SKOS XL after this chapter was written. See Section 4.9 for more information.

[17] `http://isegserv.itd.rl.ac.uk/cvs-public/~checkout~/skos/drafts/integrity.html`

[18] `http://www.w3.org/2004/02/skos/core/validation`

4

Case Study: WordNet

In this chapter we turn to research question 2: "How can vocabularies be converted to an interoperable representation with given application requirements?" An implicit assumption of the method presented in Chapter 2 is that concentrating on only the source without considering potential applications results in a conversion that is useful for a broad range of applications. We test this assumption by comparing the conversion made Chapter 2 to a recent conversion by the W3C's Semantic Web Best Practices and Deployment Working Group. It set up a Task Force (TF) to create a standard conversion of WordNet covering the already known and expected use cases for WordNet on the Semantic Web. By comparing the conversion of the TF to our previous conversion we discovered that the earlier conversion does not cover the use cases. Our assumption does not hold. This chapter adapts parts of our method. Guidelines are changed and an additional step is introduced concerning publishing a vocabulary on the Web. In Chapter 5 we complete the new method.

This chapter is based on material from (Van Assem et al. 2006a) and (Semantic Web Best Practices and Deployment Working Group 2006c). The former was published in the Proceedings of the Fifth International Conference on Language Resources and Evaluation, the latter was published as a Working Draft of W3C's Semantic Web Best Practices and Deployment Working Group. Both were co-authored with Aldo Gangemi and Guus Schreiber.

4.1 Introduction

In this chapter we describe a conversion of WordNet based on (expected) application requirements, instead of application-neutral conversion as advocated in Chapter 2. This conversion was made by the WordNet Task Force[1] of W3C's Semantic Web Best Practices Working Group[2] and published as a Working Draft (Semantic Web Best Practices and Deployment Working Group 2006c). First use cases and requirements were identified. These were then used to motivate design decisions in the conversion. Our method for vocabulary conversion presented in Chapter 2 implicitly assumes that it is not necessary to consider application requirements. A conversion process that focuses solely on faithfully representing the source will result in a conversion that is useful for a broad range of applications. This assumption is put to the test here by comparing the WordNet conversion from Chapter 2 with that made by the TF. If they differ then this is evidence that the implicit assumption does not hold. A new application-centric method will need to be developed.

[1] http://www.w3.org/2001/sw/BestPractices/WNET/tf

[2] http://www.w3.org/2001/sw/BestPractices/

In the following section we first examine the use cases and requirements for WordNet. Subsequent sections analyze which decisions the TF made and compares them to choices in the steps of our earlier conversion. Although the TF did not use our method, we present the choices made within our step-by-step method because this allows for a structured presentation of the comparsion. To improve the readability of this chapter the analysis of existing conversions is not treated in Step 0 but moved to Section 4.12.

4.2 Procedure, Use Cases and Requirements

The Task Force set out to create a new WordNet conversion because of a perceived need for a standard that can be applied in the growing number of Semantic Web applications. A standard has as benefit that interoperability between applications and data is simplified. W3C's Working Group process aims at establishing a consensus of the participating experts and uses a peer-review. The Task Force also looked at how already existing conversions were constructed. Some of these conversions are not complete and the TF took different design decisions for reasons detailed later.

The TF approached this task using the normal iterative design process of W3C. First a proposal and documentation in a draft Working Group Note is made by the TF members. Intermediate drafts are published through the public mailing list of the WG and commented upon by the WG members and any other interested party. Membership to the mailing list is open and any suggestion is discussed freely with the TF members who actively participated. When deemed ready a formal review of the proposal and draft Working Group Note are solicited by the TF. The WG selects the parties who conduct the review. Specific activities that the TF performed were (a) analysis of existing conversions; (b) finding use cases and formulating requirements; (c) analysis of source files and documentation; (d) design of RDF/OWL schema; (e) design of conversion program of Prolog data to RDF/OWL.

Use cases

The TF and the participating community identified use cases for WordNet. There are two types: use cases in existing projects and expected future use cases where a specific representation would be required. Cases of the latter type are marked "expected" below. Based on the use cases we formulate requirements to the representation.

Use Case 1: ANNOTATION WITH SYNSETS. In the MIA project (Hollink et al. 2003), the Codepict[3] system and the E-Culture project (Schreiber et al. 2006) synsets are used to annotate the subject content of images. In the OntoSeek system, synsets are used to annotate product descriptions in catalogs (Guarino et al. 1999). These uses require a representation where synsets have a URI, because blank nodes in a vocabulary cannot be referred to in annotations.

[3] `http://rdfweb.org/2002/01/photo/`

Use Case 2: ANNOTATION WITH WORD SENSES (EXPECTED). In computational linguistics WordNet is often used for sense disambiguation or "semantic concordance building": the annotation of lexical forms in texts with a word sense. In this way the meaning of the lexical form is recorded; see e.g. Ide and Véronis (1998), Fellbaum (1998, p.199). The disambiguation process consists of selecting the appropriate sense. Concordances can be used on the Semantic Web for understanding and annotation of natural language texts. This use case indicates a preference for a representation where word senses have a URI. Without a URI it is still possible to uniquely identify the sense by combining the synset URI plus the word form, but this is less convenient.

Use Case 3: GROUNDING OTHER SCHEMAS WITH WORDNET (EXPECTED). Other schemas can use WordNet as a way to "ground" their own concepts. With grounding we mean providing semantics of a concept by referring to an existing, accepted set of concepts. We expect that this use case will occur in the future. Some researchers are already experimenting with this idea. For example, one version[4]of the FOAF schema (Brickley and Miller 2005) uses Synsets as superclass of its own classes (e.g. foaf:Person and foaf:Document are linked to wn:Person and wn:Document). DBpedia uses WordNet to provide types for instances. For example, dbp:Federer is typed as a synset-tennis_player-noun-1. These examples require that at least Synsets are available with their own URI.

Use Case 4: TRANSITIVE QUERYING. In projects such as (Hollink et al. 2003), it is necessary that queries for objects annotated with synset A also returns objects annotated with hyponyms of synset A. To reach this goal, the project used a subclass representation of WordNet's hyponym hierarchy. The project did not use OWL and RDFS does not provide a means to state that the hypernym relation is transitive. The hypernym relation was replaced with the subclass relation to achieve the desired behaviour.[5] In OWL this workaround is not necessary because any property can be made transitive. This use case shows that it should be possible to either make the hyponym relation transitive, or turn the hyponym hierarchy into a subclass hierarchy.

Use Case 5: INTEGRATION OF WORD NETS (EXPECTED). The Global WordNet Association collects WordNets in many languages and aims at standardizing and mapping them.[6] This potentially requires a representation where Synsets, Word Senses and Words have a URI.

Use Case 6: ENRICHMENT OF WORDNET (EXPECTED). Future applications may need enrichments of WordNet. Examples include pronunciation information and lexical categories such as singular-plural relations between Words. This requires a representation where as many entities as possible have URIs (especially Synsets, WordSenses and Words) so that the additional information can be attached to WordNet.

[4] http://xmlns.com/foaf/spec/20071002.rdf

[5] This only works if the query language and/or storage software supports RDFS entailment of the rdfs:subClassOf property.

[6] http://www.globalwordnet.org/

Use Case 7: OBTAINING INFORMATION ON ENCOUNTERED ANNOTATIONS (EXPECTED). When a processor encounters a resource that is annotated with a WordNet URI, it may not know its meaning. It may also not have a WordNet version available (e.g. because the necessity of WordNet was not foreseen at design time). To let the processor obtain descriptive information on the resource to (partially) understand the meaning of the encountered URI, it is necessary that it can be dereferenced on the Web. This use case also holds for humans exploring an RDF source annotated with WordNet. Both situations suggest that there should be a version of WordNet available on-line that provides appropriate descriptive information. This use case is similar to the motivation for the `DESCRIBE` query form in SPARQL (RDF Data Access Working Group 2008). What exactly constitutes appropriate descriptive information about a resource is subject of debate.

Requirements

We formulate the following requirements to the representation of WordNet in RDF/OWL that are based on the use cases (coded with "RU"):

- RU1: Each Synset should have a URI (see use case 1);
- RU2: Each WordSense should have a URI (use cases 2 and 5);
- RU3: Each Word should have a URI (use case 5 and 6);
- RU4: Serve WordNet online, with descriptive information for each URI (use case 7);

Additionally we have some generic requirements that are not based on specific use cases (coded with "RG"):

- RG1: it should be a full conversion;
- RG2: it should be convenient to work with;
- RG3: it should as much as possible reflect the original structure of WordNet (i.e. avoid interpretation);
- RG4: it should provide OWL semantics while still being intepretable by pure RDFS tools (i.e. the OWL semantics can be used but can also be ignored); and

Requirements RG1-3 are requirements that were implicit in our previous work, we merely make them explicit here. For example, RG2 is implicit in guidelines 5 and 9 (leave out irrelevant ordering information and prefer the relation-as-arc approach over the relation-as-node approach).

The first requirement implies that the content of the original source should be preserved. The second requirement means that design choices should also take into account how the representation format (in this case RDF/OWL) is used in practice and what kinds of operations are difficult to perform on it. By its nature this is a vague requirement that can only be measured against user's perceptions. The third requirement implies that its semantics should be preserved. We simply

want to change the representation format of WordNet without changing its conceptual model. The conversion should stay neutral to possible interpretations of e.g. WordNet relations (i.e. not add any unintended semantics). For example, the hyponym relation is sometimes interpreted as similar or equal to the rdfs:subClassOf relation, but there are cases for which this interpretation does not hold. RG4 is a new requirement. Before the author of this thesis became involved in the TF the WordNet metamodel was specified in OWL only. However, the community pointed out that this renders the conversion unusable for RDF(S) processors.[7] Ideally a conversion caters to both RDF(S) and OWL users. Our method already partially addresses this requirement: the outcome of Step 1b is a complete vocabulary schema in RDFS, the outcome of Step 2a consists of additional statements in the OWL schema.

There may be tension between the requirements. For example, while one RDF/OWL structure may reflect the WordNet structure more appropriately than another, this structure may be less convenient to work with. In such cases a suitable trade-off needs to be made between the requirements.

In the following sections we compare the results of the TF's conversion step-by-step with our earlier conversion. At the end of each step we compare the two and note the differences.

4.3 Step 0: Preparation - Conceptual and Digital Model

In this step we study both the conceptual structure of WordNet as well as how it is syntactically represented. The conceptual model of WordNet (Fellbaum 1998) contains three core entities: words, word senses and synonym sets. Words are lexical units such as "car". Each lexical unit may have different meanings in different contexts. Each specific meaning is a word sense. For example, one sense of "car" is an automobile, while another is a railway car. Word senses with synonymous meanings are then grouped in synonym sets (also known as "synsets"), combining senses such as "car", "automobile" and "machine". Figure 4.1 provides an example of words, word senses and synsets.

There are five different kinds of word senses in WordNet: nouns (e.g. "car"), verbs ("to drive"), adverbs ("beautifully"), adjectives ("fast", "slow") and adjective satellites ("swift", "leasurely"). Adjective satellites form a subcategory of adjectives that do not have a direct antonym. For example, "swift" and "leasurely" are not antonyms, although "fast" and "slow" are. Satellites are related to the "head" adjective (e.g. "fast") through a similarity relationship. This is actually a specialization relationship: "swift" can be applied to a subset of the nouns to which "fast" can be applied (Fellbaum 1998, p.51). Another distinction that can be made between senses is those that consist of one word, and those that are collocations of several words, e.g. "living thing". For each synset a definition is given ("gloss"), and for each verb word sense a "frame sentence" that outlines a sentence construction with the verb (e.g. for "suffice" the frame is "It —-s that CLAUSE"). For each word sense a "tag count" is provided, which is the frequency of this sense in a particular text corpus (the largest count is 10,720 for the verb "be").

Ten relationships are defined between synsets (e.g. hyponomy), and five between word senses

[7] See `http://lists.w3.org/Archives/Public/public-swbp-wg/2005Sep/0033.html`.

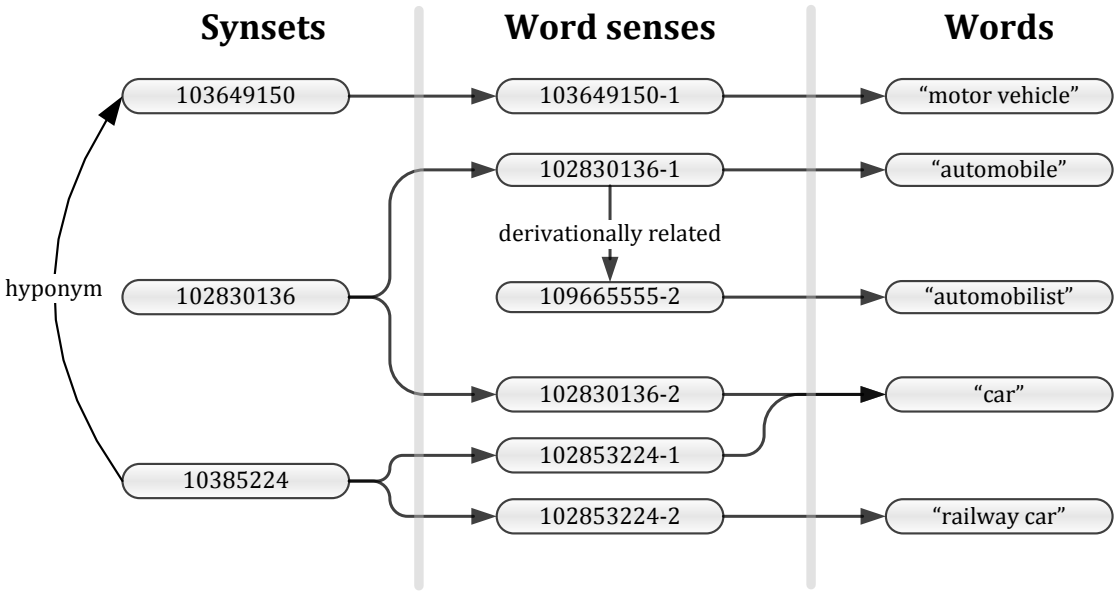

Figure 4.1 Example of the three-layered structure of WordNet. Horizontal arrows represent the containment relationship. The synsets are displayed with the synset ID, senses with the synset ID and word number.

(e.g. antonymy). WordNet's three-layered structure deviates from the standards for representing thesauri, where concepts have labels (usually nouns or noun phrases) and labels do not have relationships attached to them (ANSI/NISO 2003). The three-layered structure has been used for wordnets in many other languages and is widely publicized.[8]

To study the digital model we used the Prolog distribution of WordNet for this conversion (as in Chapter 2). It consists of eighteen files: one file that represents synsets, word senses and words, fifteen files that each represent a relationship, and two that record synset definitions and frames (more information follows). Two example clauses from the first file are:

```
s(100003009,1,''living_thing'',n,1,1).
s(100003009,2,''animate_thing'',n,1,0).
```

Each fact denotes exactly one word sense. The word senses with the same synset ID together form a synset. For example, the two clauses above together form the synset with the ID 100003009. The first argument in the predicate is an ID for the synset, the second gives a number to the word sense within the synset. The third argument is the lexical form (word) of the word sense and the fourth argument encodes the word sense's type (noun in this case). The fifth argument is the sense number, which gives a number to the sense in which the lexical form is used (e.g., the different senses of “car” each have a different number). The last argument is the tag count. Relations are identified by lists of clauses like the following three (each from its own source file):

[8] http://www.globalwordnet.org/gwa/wordnet_table.htm

```
hyp(100002056,100001740).
mp(100004824,100003226).
ant(100017087,1,100019244,1).
```

The first states a hyponymy relation between two synsets, the second states part meronymy between synsets, the third states antonymy between two word senses (second and fourth argument are word numbers). A detailed description of the Prolog syntax and its mapping to RDF/OWL is provided in (Semantic Web Best Practices and Deployment Working Group 2006c).

Comparison

No differences between the conversion in Chapter 2 and the current conversion were encountered in this step. (Do note that the analysis presented there was less detailed because of reasons of space.)

4.4 Step 1a: Structure-Preserving Translation

In this step the basic classes and properties are defined, resulting in the hierarchy in Figure 4.2 and properties in Table 4.1.

Classes and Properties

The basic classes are Synset, WordSense and Word. The need for these classes comes from (a) the conceptual model; and (b) requirements RU1-3. The requirements state that a URI for these types of entities is needed, so it is logical to also define of which type those entities are. The membership of word senses to synsets is encoded in the Synset ID used in the Prolog predicate `s/6`. The different subtypes of WordSense are encoded in the fourth argument of `s/6`, with the following meaning:

- n= NounWordSense;
- v = VerbWordSense;
- a = AdjectiveWordSense;
- s = AdjectiveSatelliteWordSense;
- r = AdverbWordSense.

For each of these we define a subclass of WordSense. Each WordNet relationship is a binary relation, so each can be represented with a property. Some additional properties were needed to link instances to each other and to record attributes of the classes. Some peculiarities are explained below. Table 4.1 splits the properties into four categories: properties that

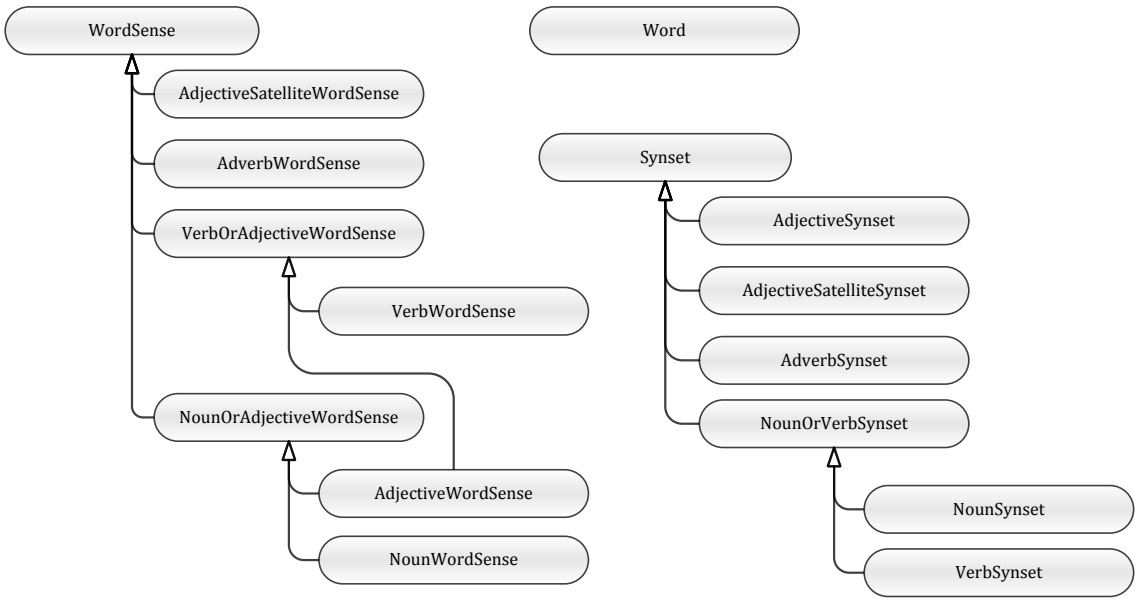

Figure 4.2 The class hierarchy after Step 1a.

- connect instances of the main classes to each other;
- describe an attribute of these classes in the form of XML Schema Datatypes (e.g. tagCount)
- represent WordNet relations between Synsets (e.g. hyponymOf);
- represent WordNet relations between WordSenses (e.g. antonymOf),

The domains and ranges of the properties that model relationship properties follow almost directly from the source documentation. For example, the documentation states that the"entails" relation "*only holds for verbs*". A few relations have two classes as domain and/or range. For example, hyponymOf holds between one noun or verb synset and another noun or verb synset. From the documentation of the predicates concerned we conclude two things. Firstly, it is necessary to model subclasses of Synset which contain only one type of wordsense (e.g. NounSynset). Secondly, it is necessary to include classes that represent a union of two other classes (e.g. union of NounSynset and VerbSynset). Although this is most appropriately modeled with owl:unionOf, requirement RG4 means that a representation is needed that can also be processed by RDF(S) infrastructure. Therefore, "union classes" such as NounOrVerbSynset are introduced. Here we diverge from our earlier conversion which only represented the "union classes" in OWL restrictions.[9]

Guideline 13: CREATE SEPARATE "UNION CLASSES" FOR RDFS PROCESSING. If a class is needed to represent a union of two other classes to e.g. define a domain or range, create a separate named class to represent the union and make each class in the union a subclass

[9]The current schema available at the W3C website erroneously omits these classes.

of it. This enables RDFS processors to interpret the union, which is not the case when only owl:unionOf is used.

Property	Domain	Range	Char.	Prolog clause
synsetContainsWordSense	Synset	WordSense		s
word	WordSense	Word		s
lexicalForm	Word	xsd:string		s
synsetId	Synset	xsd:string		s
frame	VerbWordSense	xsd:string		fr
gloss	Synset	xsd:string		g
tagCount	WordSense	xsd:integer		s
hyponymOf	NounOrVerbSynset	NounOrVerbSynset	trans	hyp
entails	VerbSynset	VerbSynset	trans	ent
similarTo	AdjectiveSynset	AdjectiveSatelliteSynset	trans	sim
memberMeronymOf	NounSynset	NounSynset		mm
substanceMeronymOf	NounSynset	NounSynset		ms
partMeronymOf	NounSynset	NounSynset		mp
classifiedByTopic	Synset	NounSynset		cls
classifiedByUsage	Synset	NounSynset		cls
classifiedByRegion	Synset	NounSynset		cls
causes	VerbSynset	VerbSynset		cs
sameVerbGroupAs	VerbSynset	VerbSynset	sym	vgp
attribute	NounSynset	AdjectiveSynset		at
adjectivePertainsTo	AdjectiveWordSense	NounOrAdjectiveWordSense		per
adverbPertainsTo	AdverbWordSense	AdjectiveWordSense		per
derivationallyRelated	WordSense	WordSense	sym	der
antonymOf	WordSense	WordSense	sym	ant
seeAlso	WordSense	WordSense		sa
participleOf	AdjectiveWordSense	VerbWordSense		ppl

Table 4.1 Properties defined in Step 1a. The column "Prolog clause" indicates the main Prolog clause used to generate instances of the properties. The "Char." column indicates OWL property characteristics defined in step 2a (trans=transitive, sym=symmetric).

The `cls` predicate has three arguments: two synset IDs and one character for encoding that the first synset is classified into the second synset as a topic, a specific usage or a specific region. Because it has three arguments, it seems impossible to convert this to a binary property. However, we have devised a solution applicable only to a limited set of cases; see guideline 14. The predicate was converted to three properties between synsets: classifiedByTopic, classifiedByUsage and classifiedByRegion.

Guideline 14: FACTOR THE THIRD ARGUMENT OF A RELATION INTO SUBPROPERTY NAMES. Following guideline 9 but not losing the third argument of a relationship is possible by factoring the third argument into the names of new subproperties. For example, if a relation (represented with property rel) has a third argument of type C with possible values X and Y, create properties relX and relY . This is only sensible for arguments with a limited set of non-numerical values.

For the three classifiedBy... properties and the derivationallyRelated no exact domain/range was indicated in the documentation for the Prolog predicates `cls` and `der`, so this had to be established from the instances.

The choice for the property synsetContainsWordSense represents a trade-off between requirements RG2 (convenient to work with) and RG3 (reflect the original structure). The original source defines a *word number* for each word sense in the synset (see Section 4.3). Although it is possible to define a strict sequence in RDF using rdf:Seq, the community is not fond of this construct because it is awkward to process. As we did not find evidence that the word numbers are intended to signify a strict sequence, we decided to interpret this structure as a set. Sets can either be represented with the equally awkward to process rdf:Bag or by a custom property that represents a one-to-many membership between Synset and WordSenses.[10] We chose the second approach for the sake of convenience. We also do not record the word number, as this would only be useful information if the word number indeed signifies a sequence. The word number can only be preserved with a 3-aried relation between Synset, WordSense and the number, which requires a non-standard representation in RDF. We would like to avoid such structures (see also Guideline 9).

The property `synsetId` is introduced to record the local identifier of Synsets. This may be necessary when e.g. interfacing with sources in other formats that use the IDs as annotations. This is a use case that was not identified by the community but still one that can be expected to occur. This propery was not present in the earlier conversion; the local IDs were present within the URIs but this is not convenient: the URIs would need to be processed with e.g. a regular expression to obtain the IDs. Moreover, obtaining the local IDs from the URIs violates the principle of URI opacity (W3C Technical Architecture Group 2004, Sec. 2.5). In guideline 6 and guideline 11 we explicitly recommended not to store local IDs in a separate property, with the argument that it introduces redundancy. However, the above explains why we must modify those earlier guidelines.

Guideline 15: AVOID REDUNDANT INFORMATION (MODIFIED VERSION OF GUIDELINE 6, PAGE 16). Redundant information creates representations which are less clear and harder to maintain. This guideline as formulated in guideline 6 is correct, but the example given is not. Storing the local ID of a resource in both the URI and a separate property is not redundant because the local ID may be necessary for integration purposes.

Guideline 16: CONSIDER STORING THE ORIGINAL IDS. (REPLACES PART OF GUIDELINE 11, PAGE 20). Guideline 11 states that "properties to record an entity's UI would introduce redundancy, and therefore shouldn't be used". However, the identifiers used in the original source may be used in other datasources. The identifiers can therefore be useful for integration purposes. In cases where it can be expected that the IDs were used in other sources or applications, the IDs should be recorded in the conversion in a separate (datatype) property.

Guideline 16 is particularly relevant in the Cultural Heritage sector, where experts tend to refer to objects by their identifiers when communicating with other experts. They will want to see the

[10] In this representation one can query for all WordSenses in a single SPARQL statement. To obtain the same list of resources from a rdf:Seq a new SPARQL statement is needed to query for each next element of the rdf:Seq until the end of the sequence is encountered.

object or concept's ID in displays, which is an additional motivation to store the ID in a separate property.

URIs for WordNet resources

Above, it was decided which classes and properties were needed. Here we discuss what the syntax should be for URIs for those classes and properties, but in particular for their instances. In our earlier guideline we gave the advise to only create URIs based on a unique identifier in the source, and else use a blank node. However, blank nodes make it difficult to address a particular node, which is contrary to the goals of many use cases. Therefore, we replace guideline 11 (page 20):

Guideline 17: DEFINE URIS FOR ALL MAJOR ENTITIES (REPLACES GUIDELINE 11, PAGE 20). Guideline 11 states that "If the original source does not provide a unique identifier for an entity, one should translate it into blank nodes". However, for most purposes it is necessary to be able to directly reference a resource. Because direct reference is not possible with blank nodes, it is safer to use a URI when in doubt. Whether the original source specifies a unique identifier that can be used in the URI is not relevant to this decision.

We extend our earlier method with concrete advice on the syntax of the URIs. We adopt the principles given in[11] that URIs should be: *descriptive* (mnemonic; easy to remember), *persistent* (do not change), *dereferencable* and *unambiguous*. The last two are discussed in Section 4.10.

Guideline 18: CREATE URIS THAT ARE DESCRIPTIVE AND PERSISTENT. Always prefer (a combination of) human-readable data to generate URIs, even if a locally unique ID is provided by the vocabulary. Choose a name that is unlikely to need to change.

The wish for descriptive URIs came directly from the participating community.[12] Previous conversions used synset IDs as the local part of Synset URIs, which tells a human developer nothing about its meaning. Descriptive (human-readable) URIs do not violate the principle of URI opacity (W3C Technical Architecture Group 2004, Sec. 2.5) as long as the information contained in them is not (ab)used for automated reasoning.

Pattern	Example
wn20instances:synset- + $lexform$- + $type$- + $sensenr$	wn20instances:synset-bank-noun-2
wn20instances:wordsense- + $lexform$- + $type$- + $sensenr$	wn20instances:wordsense-bank-noun-1
wn20instances:word- + $lexform$	wn20instances:word-bank
wn20schema:$propertyOrClassName$	wn20schema:hyponymOf

Table 4.2 Patterns for WordNet URIs with an example URI.

Note that guideline 18 also holds for classes and properties, so *we drop guideline 3* (basing property/class names on the names used in the original source; page 16). The motivation is

[11] `http://esw.w3.org/topic/GoodURIs`

[12] Through personal correspondence between the TF and contributors to the mailing list.

that the link between source and conversion is not needed by application developers that use the conversion. When it is needed to understand the link, good and detailed documentation is required anyway, and syntactic name preservation is not enough to acquire this understanding. The TF chose a system for creating the local parts of instance URIs, which makes them unique and provides a short summary of the resource. Table 4.2 gives the patterns plus an example. Two namespaces were made: one for instances and one for the schema. The initial reason was to prevent URI clashes (e.g. property "word" and synset "word"), but it was decided later to use prefixes in the local names to prevent clashes.

The *lexform* is the lexical form of a WordSense or Word, *type* is one of noun, verb, adjective, adjective satellite and adverb, *sensenr* is the the number in the fifth argument of `s/6`, The patterns for synsets and word senses are similar because a word sense uniquely identifies a synset. The lexform of a Synset was generated by copying the lexform from its WordSense with word number "1". This is a somewhat arbitrary choice, but it seems that the first sense is the more simple one lexically speaking. Another choice might have been to use the sense with the highest tag count. An alternative choice for uniquely identifying a WordSense is to combine the synset ID and the wordnumber (as in `ant(100017087,1,100019244,1)`). However, the synset ID is not descriptive so the lexical form and type would have to be added anyway.

Using information like the class name and the lexical type is also required to guarantee unicity. It prevents *URI clashes* for e.g. the word "word", the property "word" and the synset "word". Some lexical forms contain characters that are not allowed in XML NCNames (XML Core Working Group 2006). In order to generate a permissible URI we substituted (back)slashes, parentheses and spaces with underscores. For example, the URI for the word "read/write memory" becomes "*word-read_write_memory*".

It is accepted practice to use for the URI namespace a HTTP namespace that is under the creator's control. This partly satisfies the requirement for avoiding ambiguity (it avoids *URI clashes*). We found the W3C willing to provide the namespaces to host the conversion online. The namespaces provided are:

```
wn20instances = http://www.w3.org/2006/03/wn/wn20/instances/
wn20schema =  http://www.w3.org/2006/03/wn/wn20/schema/
```

The W3C is strongly committed to maintaining any URIs it publishes, which guarantees persistency.[13]

Version information in URIs

If multiple editions or versions of a vocabulary are converted and published, URI clashes may occur if the URIs for a new version are generated in the same manner as the previous version. Or the old version is lost when the new version is published at the same HTTP URIs (see Section 4.10 for more explanation). This is problematic because e.g. two synset instances cannot be assumed to have the same meaning in a new version. Meaning changes not only when the content of a synset changes, but also when e.g. a hypernym of a synset changed in meaning. Furthermore, whether or

[13] See `http://www.w3.org/Consortium/Persistence` for W3C's persistence policy.

not two synsets from different versions are "equal" or not will probably depend on the application, so equality should not be decided by those converting the vocabulary.

However, there are also disadvantages to publishing separate versions. Firstly, it requires resources to keep all versions online. Secondly, in the situation where one version is published at the HTTP URIs where the old version was available, applications get an automatic upgrade. If there are separate versions, application developers need to change all references in their code from the old to the new URIs in order to upgrade. Thirdly, if the old version becomes unavailable, the application cannot retrieve vocabulary content until the references are updated (it may even malfunction because it needs the vocabulary content). Fourthly, applications need to be aware of all different versions of all relevant vocabularies to be able to retrieve content. They also need to be aware of all versions to succesfully process data that is annotated with those vocabulary versions. This requires a bookkeeping of vocabulary versions which complicates the application.

Guideline 19: CONSIDER INCLUDING A VERSION-SPECIFIC COMPONENT IN THE URI. If the original source is expected to have more than one edition (which may be expected for almost all vocabularies), it may be necessary to make them distinct. This can be done by including the version name or number in the URI

In sum, whether or not to publish separate versions of a vocabulary under distinct URIs presents a trade-off. A compromise is to publish the versions under version-specific URIs, but have a separate set of URIs without version specification that always point to the newest version. The TF decided to use version-specific URIs.

Comparison

Major differences with our earlier conversion are: instances of WordSense have URIs instead of blank nodes, a new class Word, and a property synsetID. The "union classes" to define ranges consisting of multiple classes in RDFS are also new and the syntax of the URIs is different. The "pertains to" relation is split into two properties in this conversion, and relations were added that were missing by mistake in the previous conversion (frame, tag count, classified by). Lastly, we added/modified seven guidelines, and dropped one.

4.5 Step 1b: Explication of Syntax

In step 1b four explications can be made. Firstly, the notion of word collocations is present in the conceptual model, but only implicitly indicated by the underscores in word forms. To make this explicit a class Collocation is introduced as a subclass of Word and all words that have underscores are made instance of Collocation. Without this class each query that needs to process queries needs to use a pattern match on the lexical forms to select only collocations.

A second explication is that AdjectiveSatelliteWordSense is a subclass of AdjectiveWordSense. Both types are explicitly present in the Prolog source, but it is not explicitly given that the former is a specific type (i.e. a subset) of the latter. These first two explications result in an adapted class hierarchy as depicted in Figure 4.3.

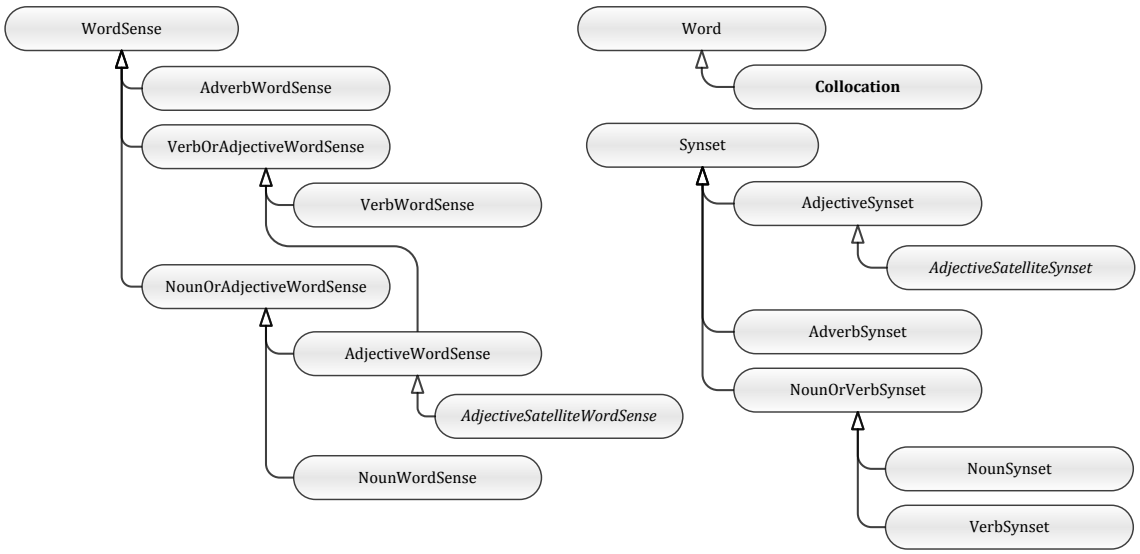

Figure 4.3 The complete class hierarchy of the conversion. In comparison with Figure 4.2, a new class Collocation has been added (in bold) and two classes have a different hierarchical position (in italic).

Thirdly, the properties gloss and lexicalForm are made a subproperty of rdfs:comment and of rdfs:label, respectively. These statements have to be removed when WordNet is to be used in an OWL DL environment (rdfs:comment and rdfs:label are owl:AnnotationProperty; an owl:AnnotationProperty is not allowed in a property axiom). We have left them in because they are valuable for applications that displays RDF/OWL datasources (it allows the application to decide which description to show and in what context).

Fourthly, two superproperties are introduced to group together several closely related properties that represent relationships. The property meronymOf is superproperty of memberMeronymOf, substanceMeronymOf and partMeronymOf; classifiedBy is superproperty of classifiedByTopic, classifiedByUsage and classifiedByRegion.

Guideline 20: INTRODUCE SUPERPROPERTIES TO GROUP PROPERTIES WITH SIMILAR SEMANTICS. To make the semantics of several similar properties explicit, introduce a superproperty for them. This makes it possible to query them without considering the finer distinctions that the subproperties represent, which can be convenient. It also helps humans in understanding a schema.

Comparison

Differences with our earlier conversion are the following: introduction of a Collocation class and a guideline that suggests to introduce superproperties. This was done in WordNet for the properties representing the classification and meronymy relations. In the previous conversion the subclass relationship between adjectives and adjective satellites was defined in step 1a. However, decisions on subclass relations are taken in step 1b (see Section 2.3).

4.6 Step 2a: Explication of Semantics

Requirement RG4 states that the conversion should provide the appropriate OWL semantics while still being intepretable by pure RDFS tools. the OWL semantics can be used but can also be ignored. A straightforward strategy to fulfill the latter is to define each class in the schema as both rdfs:Class and owl:Class, and each property as an rdf:Property and either owl:ObjectProperty or owl:DatatypeProperty as appropriate. This specifically refers to compatibility with OWL DL, because in OWL Full rdfs:Class and owl:Class are equivalent. owl:DatatypeProperty is a subclass of owl:ObjectProperty, and owl:ObjectProperty is equivalent to rdf:Property. To conform to the former requirement, restrictions, property characteristics and disjointness statements between classes were added to the schema.

Firstly, we analyzed which OWL property characteristics are appropriate for the properties. For each we investigated which characteristics can be added (see Table 4.1). In most cases this is straightforward as the characteristic is already described in the source documentation. Sometimes the characteristic is implicit in the documentation as is the case in the `sameVerbGroup` relationship:

> " *The vgp operator specifies verb synsets that are similar in meaning and should be grouped together when displayed in response to a grouped synset search.* "

This implies that if `vgp(A,B)` then also `vgp(B,A)`, which makes this a symmetric relation. Another case is hyponymy. The source documentation does not provide evidence for characteristics such as transitivity, but Fellbaum (1998, p.25) does provides evidence: the hyponym relation "*is the transitive, asymmetric, semantic relation that can be read 'is-a' or 'is-a-kind-of'*".

A second candidate for explication is property inverseness. The data itself does not contain relations that are each other's inverse, but we can add inverse properties for those already present in our schema for maximum explication of the meaning of those properties. There are two groups of owl:ObjectProperty for which inverseness can be relevant: properties that link the main classes Synset, WordSense and Word, and properties that represent WordNet relations. Inverses for the former are useful in querying and in defining restrictions. Inverses for the latter are only useful for querying. Defining the inverse properties and owl:inverseOf statements is not enough for RDF(S) infrastructure to use them, because they do not automatically derive the inverse triples based on the owl:inverseOf statements. There are four alternative solutions:

1. explicitly add the inverse properties and inverse triples;
2. add the inverse properties but not the inverse triples. Add a warning in the conversion documentation for users of RDF(S) infrastructure stating that the inverse properties are not instantiated;
3. add the inverse properties as owl:ObjectProperty but not as rdf:Property, do not add inverse triples. Use of owl:ObjectProperty serves as an additional warning in the schema that the inverse properties are only available in OWL;

4. add the inverse properties and inverse triples in separate schema and instance files. Users can make their own choice.

Discussion in the community indicated a disfavor of the first option[14] , because it adds data for which a syntactic alternative is available (querying for e.g. `X hypernymOf Y` is just a syntactic alternative to querying for `Y hyponymOf X`). The second and third options have the disadvantage that RDF(S) users, being unaware of the warning, may query one of the "empty" inverse properties. They would wrongly get an empty result set. The fourth option is the most flexible, but also the unsafest as what can or cannot be queried depends on which files are loaded. The second and third option are safer. The community also indicated that it is comfortable with using the syntactic alternative for inverse properties. In the end, we chose the third option. It is slightly better than the second because of the additional "flagging of the issue" that owl:ObjectProperty provides.

For property characteristics such as transitivity there are only two possible alternatives: either add the implied triples or not. Because of the community's dislike of adding more data, we chose for the second alternative. When the characteristic is needed in an RDFS application this has to be realized with a local rule. No "hiding" is necessary in this case: each property with a characteristic is both instance of rdf:Property and the appropriate OWL class (e.g. owl:SymmetricProperty). The characteristic is automatically "hidden" from RDFS.

Thirdly, we can identify the following metamodel constraints. These constraints can be modelled in OWL restrictions:

- each Synset contains at least one WordSense;
- each WordSense belongs to exactly one Synset;
- each WordSense has exactly one associated Word;
- each Word belongs to at least one WordSense;
- each subclass of Synset (e.g. AdjectiveSynset) contains only WordSenses of one type (e.g. AdjectiveWordSense);

The first four follow directly from the structure of the `s/6` predicate. For example, each `s/6` clause defines one WordSense, and contains exactly one Synset identifier, from which follows that a WordSense belongs to exactly one Synset. We derive the last constraint from the conceptual model. These constraints can be represented one-to-one with OWL restrictions as depicted in Figure 4.4. The exception is the last constraint, which has to be stated with one restriction for each WordSense/Synset type. For this set of restrictions we give only one example (for class AdjectiveSynset). The second and third constraints are combined into one class definition (for class WordSense).

The classes are only defined as partial instead of complete because there is no indication that such a strong interpretation of the constraints is warranted, i.e. we leave open the possibility that there are instances that have a WordSense attached to them using containsWordSense but are not

[14] `http://lists.w3.org/Archives/Public/public-swbp-wg/2005Nov/0149.html`

```
@prefix rdfs:    <http://www.w3.org/2000/01/rdf-schema#> .
@prefix rdf:     <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix wns:     <http://www.w3.org/2006/03/wn/wn20/schema/> .
@prefix owl:     <http://www.w3.org/2002/07/owl#> .


wns:Synset a owl:Class;
  rdfs:subClassOf
    [ a owl:Restriction;
      owl:onProperty containsWordSense;
      owl:someValuesFrom wns:WordSense
    ].


wns:WordSense a owl:Class;
  rdfs:subClassOf
    [ a owl:Restriction;
      owl:onProperty inSynset;
      owl:cardinality "1"^^<http://www.w3.org/2001/XMLSchema#int>
    ];
  rdfs:subClassOf
    [ a owl:Restriction;
      owl:onProperty word;
      owl:cardinality "1"^^<http://www.w3.org/2001/XMLSchema#int>
    ].

wns:Word a owl:Class;
  rdfs:subClassOf
    [ a owl:Restriction;
      owl:onProperty sense;
      owl:minCardinality "1"^^<http://www.w3.org/2001/XMLSchema#int>
    ].

wns:AdjectiveSynset a owl:Class;
  rdfs:subClassOf
    [ a owl:Restriction;
      owl:onProperty containsWordSense;
      owl:allValuesFrom wns:AdjectiveWordSense
    ].
```

Figure 4.4 OWL restrictions on WordNet classes to represent constraints in the metamodel.

instance of Synset. Additionally, all classes are made pairwise disjoint, except of course a class and its subclasses. The "union classes" defined using rdfs:subClassOf in the previous section are now made explicit by stating owl:unionOf between the union class (e.g. NounOrVerb) and the classes it combines (e.g. Noun and Verb).

Comparison

The OWL semantics provided in this step are compatible with the choices made in our earlier conversion, but the provisions to enable usage by both RDF(S) and OWL users are new additions. The cardinality restrictions were omitted in the previous conversion.

4.7 Syntax and Documentation Errors

In steps 1a-2a we found several errors in the data and documentation. Sometimes the Prolog code and its documentation was erroneous or in conflict with each other, which presented difficulties in understanding and converting WordNet. Four types of errors and omissions were discovered: order and meaning of arguments, relation characteristics, missing symmetric clauses and missing frames. In most cases we fixed the error in our representation.

Firstly, for many predicates representing relationships the documentation gives the wrong argument order; see (Semantic Web Best Practices and Deployment Working Group 2006c). This was discovered when we reviewed the entities that the clauses related. For example, the Prolog fact `cs(200020689,200014429)` represents a "causative" relation between the synsets for "anaestesize" and "sleep" (in that order). However, the documentation states that "*The cs operator specifies that the second synset is a cause of the first synset.*" In some cases the source documentation also wrongly asserted that the related entities were word senses instead of synsets, or the other way around.

Secondly, some cases were found in which a wrong term was used to describe a relation characteristic. For example, it is stated that the the hypernym relation has a "reflexive" relation called hyponym, while the correct term is "inverse". Another example is the derivational related relation, that is documented as "reflexive", while "symmetry" is meant.

Thirdly, sometimes the source is not "complete". For example, the antomym relation is symmetric, and the documentation claims that the symmetric clause is also present in the source file (e.g. if `ant(a,b)` is in the source then also `ant(b,a)`). For some predicates, such as `ant` for antonyms, the symmetric clause is missing. In such cases we decided not to convert the clause because it is unsure whether the missing symmetric clause is an error or the one that is present.

Fourthly, clauses of the form `fr(SynsetId, PatternId)` relate a VerbWordSense to a verb construction pattern (e.g. "Somebody —-s that CLAUSE"). However, the actual patterns were not present in the Prolog distribution. We converted the data from the proprietary format to Prolog clauses `sen(PatternId, String)` to be able to convert the frames to RDF. For more details on how the original format was converted we refer to Appendix D in (Semantic Web Best Practices and Deployment Working Group 2006c).

In Table 4.1 we defined domains and ranges for each property based on the WordNet documentation. This does not guarantee that each Prolog clause we converted to these properties has a subject and object that conform to the domain/range definition. We wrote a small program to check for triples that do not conform to the definition, but found none. Note that performing this type of check is generic; it can be performed on any conversion. A software package to perform this type of checks would be useful for quality control of conversions.

4.8 Step 2b: Interpretation

The interpretative steps that can be taken on WordNet range from simple mappings to implement a class hierarchy to a complete restructuring into an ontology as described in (Gangemi et al. 2003a). Chapter 2 describes how the hyponym hierarchy can be interpreted as a class hierarchy, by making

hyponymOf a subproperty of rdfs:subClassOf. This is a valid solution to realize use case 4 (using WordNet for transitive querying), but this interpretation is not included in the conversion because it conflicts with Requirement RG3 (avoid interpretation).

4.9 Step 3: Standardization

This step discusses how to map WordNet to the SKOS standard for representing vocabularies on the Semantic Web. This mapping was the work of the author and not of the TF as such a mapping was not needed for the TF's goals. The discussion below uses a newer version of SKOS than previous chapters (Semantic Web Deployment Working Group 2008b).

There are several decisions that have to be made when mapping WordNet to SKOS. The first concerns what the actual mappings should be. There are four main mapping targets: (1) skos:Concept; (2) skos:broader/narrower; (3) skos:related; and (4) other SKOS properties.

Mapping to skos:Concept

We map Synset to skos:Concept. We do not map WordSense and Word to skos:Concept, although it can be argued that they are also concepts. There are several reasons. Firstly, only Synset is used to represent a concept hierarchy. Secondly, skos:Concepts typically have several synonymous lexical labels. A WordSense rather represents one meaning of one term; it does not group synonyms. Thirdly, SKOS has defined an extension called SKOS XL (eXtension for Labels) to identify, describe and link lexical entities. Therefore, we decided to map WordSenses and their relations using this extension, and not to overload skos:Concept (see Section 4.9).

Mapping to skos:broader

The second mapping choice is what to map to skos:broader. The only property in WordNet that unambiguously indicates that one concept is more general than another is hyponymOf. The hyponym relation is stated to be transitive and irreflexive. In the new SKOS specification there is a distinction between skos:broaderTransitive and its subproperty skos:broader, with the remark that only the latter should actually be used to make assertions. The reason is that SKOS prefers to leave out transitive statements in a vocabulary conversion. It should be up to the application whether these statements should be present or not. With this in mind we choose to map hyponymOf to skos:broader instead of skos:broaderTransitive, even though hyponymOf has transitive semantics.

Mapping to skos:related

The third mapping choice is what to map to skos:related. This property models symmetric, associative (i.e. non-hierarchical) relations between concepts. Subproperties of symmetric relations can be non-symmetric in OWL (e.g. sisterOf and brotherOf are valid non-symmetric subproperties of the symmetric siblingOf). Therefore, non-symmetric WordNet properties that relate entities mapped to skos:Concept can be mapped to skos:related. This means that almost all remaining properties that model relations can be mapped to skos:related. The exceptions are relations such

as antonymOf and seeAlso which relate word senses. As it was decided that word senses would not be mapped, it is not possible to map relations that have word sense as domain/range.

We include a mapping from meronymOf to skos:related. Although the meronym properties do seem to indicate some narrower-broader semantics[15], in general it is not clear if one would like to retrieve documents indexed with the narrower concept when querying for the more generic concepts. Some examples illustrate this: linebacker-football team, cell-organism, oxtail-oxtail soup. While some vocabularies do organize parts under wholes, others do not (e.g. AAT separates chairs and chair parts), so any choice we make here can be debated. We have chosen here for a solution that does not commit to semantics that enlarges the retrieval set of hierarchical searches.

We choose not to map classifiedByUsage because it represents a lexical (meta-level) classification of concepts (e.g. firewall-colloquialism). The other two, classifiedByTopic and classifiedByRegion are however mapped to skos:related (e.g. Kaaba-Islam, tempura-Japan).

Mapping to other SKOS properties

The fourth mapping choice is what to map to other SKOS properties. The gloss can be mapped to skos:definition. The remaining issue is how to map literalForm to skos:prefLabel and/or skos:altLabel. This issue is discussed below together with the schema-level mapping vs. data transformation mapping issue. A schema-based mapping of the choices discussed so far is given in Figure 4.5.

Schema-based mapping vs. data transformation-based mapping

When a vocabulary can be seen as a strict specialization of SKOS, it can be mapped in a schema (by using declarative rdfs:subClassOf and rdfs:subPropertyOf statements). In other cases a data transformation (using a more expressive language than RDF/OWL) is necessary. The mappings discussed earlier can be realized in a schema, but the mapping of lexicalForm to skos:prefLabel / skos:altLabel is problematic for two reasons. Firstly, lexicalForms are attached to instances of Word, not Synset. This prohibits a subproperty mapping of lexicalForm to the two label properties of SKOS. A second problem is that SKOS prescribes that concepts have exactly one preferred label, while WordNet makes no obvious distinction between the senses of a synset. For these reasons, a data transformation is required. See Figure 4.6 for an example implemented in SWI-Prolog. The heuristic used in this implementation is to choose as preferred label the sense that is most frequently used in English texts, according to the tagCount.

In summary, it is possible to map WordNet to SKOS using declarative statements in an RDF/OWL schema, except for the labels. A data transformation specified in a more expressive language can overcome the label problem. Note that SKOS recommends to prevent two concepts in the same SKOS concept scheme to have the same preferred label (Semantic Web Deployment Working Group 2008a). This recommendation has to be violated when converting WordNet, because there are many homonymous words in WordNet. WordNet uses word senses to represent the distinction, instead of qualifiers in the concept labels.

[15]The ISO2788 and ANSI/NISO Z39-19 standards even define a partitive relation (BTP) that is a subcategory of the generic hierarchical relation (BT)

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix my: <http://www.example.com/my#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix wn20schema: <http://www.w3.org/2006/03/wn/wn20/schema/> .

wn20schema:NounSynset               rdfs:subClassOf skos:Concept .
wn20schema:VerbSynset               rdfs:subClassOf skos:Concept .
wn20schema:AdverbSynset             rdfs:subClassOf skos:Concept .
wn20schema:AdjectiveSynset          rdfs:subClassOf skos:Concept .
wn20schema:AdjectiveSatelliteSynset rdfs:subClassOf skos:Concept .

wn20schema:hyponymOf rdfs:subPropertyOf skos:broader .

wn20schema:meronymOf          rdfs:subPropertyOf skos:related .
wn20schema:entails            rdfs:subPropertyOf skos:related .
wn20schema:similarTo          rdfs:subPropertyOf skos:related .
wn20schema:classifiedByTopic  rdfs:subPropertyOf skos:related .
wn20schema:classifiedByRegion rdfs:subPropertyOf skos:related .

wn20schema:causes          rdfs:subPropertyOf skos:related .
wn20schema:sameVerbGroupAs rdfs:subPropertyOf skos:related .
wn20schema:attribute       rdfs:subPropertyOf skos:related .

wn20schema:gloss rdfs:subPropertyOf skos:definition .
```

Figure 4.5 Schema-based mapping of WordNet to SKOS. Mapping of lexical forms is a data transformation, see Figure 4.6.

Use original URIs for SKOS version or not

The last decision is whether the mapped version should use the original WordNet URIs or not. In other words: should the mapping of Synset to skos:Concept be realized by creating new instances of skos:Concept for each instance of Synset, or should Synset be made a subclass of skos:Concept so that each instance of the former is also an instance of the latter (use the existing URIs)?

Guideline 21: CREATE A NEW NAMESPACE FOR THE SKOS REPRESENTATION IF ADDITIONAL TRIPLES MIGHT BE ADDED THAT ONLY HOLD FOR THE SKOS REPRESENTATION. Using the existing URIs keeps the link between the original and the SKOS version intact. However, if annotations are made to the SKOS version that would not be applicable to the complete conversion, create URIs in another namespace.

As we do not foresee enrichments for the SKOS representation, it is OK to reuse the original WordNet URIs in the SKOS representation.

Mapping to SKOS XL

The SKOS community recently defined an extension to their basic model, called SKOS XL (Semantic Web Deployment Working Group 2008b, App. A). The motivation is that some vocabularies contain relationships between terms, e.g. to indicate that one is an abbreviation of another (Semantic Web Deployment Working Group 2007). A class xl:Label is introduced so that terms

```
% synset_label_to_skos(+SynsetURI, +ConceptURI)
%    SynsetURI:  synset that is being converted to SKOS
%    ConceptURI: URI of the skos:Concept for which
%                pref/altLabels need to be asserted

synset_label_to_skos(SynsetURI, ConceptURI) :-

   % get all labels belonging to synset, select one preferred
    select_pref_label(SynsetURI, Pref, Rest),

    % attach  pref/altLabels to the concept
    rdf_assert(ConceptURI, skos:prefLabel, Pref),
    rdf_assert(ConceptURI, rdf:type, skos:'Concept'),
    maplist(rdf_assert(ConceptURI, skos:altLabel), Rest).

% select_pref_label(+SynsetURI, -Pref, -Rest)
%    Pref: label of wordsense attached to synset
%          with highest tag count
%    Rest: list of other wordsenses' labels attached to synset

select_pref_label(SynsetURI, Pref, Rest) :-

    % create pairs Tagcount-Label, then sort them
    get_senses(SynsetURI, WSURIs),
    maplist(create_pair, WSURIs, Pairs),
    keysort(Pairs, P),

    % keysort outputs lowest first, need highest first
    reverse(P, [_Tag-Pref|RestPairs]),
    % get labels from the remaining pairs
    strip_labels(RestPairs, Rest).

% create_pair(+WSURI, -TagCount-Label)
%    WSURI:   URI of a word sense
%    TagCount-Label: tagcount of word sense,
%  label attached to sense

create_pair(WSURI, TagCount-Label) :-
    rdf(WSURI, wn20schema:tagCount, TagCount),
    rdf(WSURI, wn20schema:word, Word),
    rdf(Word, wn20schema:lexicalForm, Label).
```

Figure 4.6 Data transformation in SWI-Prolog for mapping WordNet lexicalForms to SKOS preferred and alternative labels. Slightly edited and a few predicates not shown to improve readability.

can be instances. Properties xl:prefLabel and xl:altLabel connect xl:Label to skos:Concept. Properties between Labels can be made subproperty of xl:labelRelation.

This extension enables us to map WordSenses to SKOS, which is not possible with the basic SKOS schema. The statements in Figure 4.7 can be added to those in Figure 4.5.

Comparison

The output of this step cannot be compared directly with our earlier conversion, as this step was not made for WordNet there. However, the choice concerning which concepts (not) to map is

```
@prefix xl: <http://www.w3.org/2008/05/skos-xl#> .

wn20schema:NounWordSense               rdfs:subClassOf xl:Label .
wn20schema:VerbWordSense               rdfs:subClassOf xl:Label .
wn20schema:AdverbWordSense             rdfs:subClassOf xl:Label .
wn20schema:AdjectiveWordSense          rdfs:subClassOf xl:Label .
wn20schema:AdjectiveSatelliteWordSense rdfs:subClassOf xl:Label .

wn20schema:adjectivePertainsTo   rdfs:subPropertyOf xl:labelRelation .
wn20schema:adverbPertainsTo      rdfs:subPropertyOf xl:labelRelation .
wn20schema:derivationallyRelated rdfs:subPropertyOf xl:labelRelation .
wn20schema:antonymOf             rdfs:subPropertyOf xl:labelRelation .
wn20schema:seeAlso               rdfs:subPropertyOf xl:labelRelation .
wn20schema:participleOf          rdfs:subPropertyOf xl:labelRelation .
```

Figure 4.7 Additional statements for a schema-based mapping of WordNet to SKOS XL.

similar to the mapping described for MeSH in Section 2.3. In our previous description of Step 3, we have only considered schema-based mappings, but here we extend the method to include rule-based ones. We have also made a more detailed analysis of the possible mappings and their consequences. Furthermore, we have added two new guidelines.

The output of this step is made available at `http://thesauri.cs.vu.nl/wnskos/`. It is not available in the W3C version, as a mapping to SKOS was not required for the goals of W3C's TF.

4.10 Step 4: Publishing on the Web

Requirement RU4 states that WordNet should be served online, with descriptive information for each URI. We also adopted four principles for "good URIs", two of which are that ambiguous URIs should be avoided and URIs should be dereferencable[16]. Dereferencability means that relevant content should be served when an HTTP Get is done on that URI. This is actually exactly the same as requirement RU4.

How to avoid ambiguity is a complex goal, but at the very least it entails choosing URIs in a namespace that is under the vocabulary publisher's control. Given that such URIs are chosen, a simple strategy to comply with the principle of dereferencability is to upload the RDF/OWL files to a Web location owned by the publisher. However, this straighforward solution actually does not avoid ambiguity and causes practical usage problems which we will detail now.

We decided to include a new step in our method that specifically deals with issues surrounding publication.

[16]We base ourselves on `http://esw.w3.org/topic/GoodURIs`. There the word "navigable" is used, but we feel this implies that only human users should be able to retrieve a representation of the URI's meaning. We use the term "dereferencable" which applies to both humans and computers.

Dereferencability and Ambiguity

URIs can be used to identify any resource: concrete objects, digital documents, abstract concepts. A subset of all URIs is the set of HTTP URIs, which are those URIs that yield a *representation* of a resource when dereferenced on the web. URIs should be *dereferencable*, which means that relevant, descriptive information concerning the URI should be available when it is dereferenced (with an HTTP GET). This requirement codifies the established community practice of placing the RDF/OWL document describing the URIs in the same HTTP namespace. For example, the resource rdfs:Class with namespace rdfs equal to `http://www.w3.org/2000/01/rdf-schema#` is in a document at the HTTP URI `http://www.w3.org/2000/01/rdf-schema#`. This is a useful practice as it allows both humans and agents to look up information on a previously unknown URI.

Two types of URIs can be distinguished: *hash URIs* that use the '#' to separate the global namespace from the local name (the *fragment identifier*), and *slash URIs* which use '/' instead. Although the community usually uses hash URIs, these have a problematic aspect when they are dereferenced. When a hash URI is dereferenced, not just the resource but the *whole document* is returned, which in the case of WordNet would be a document of 166 MB. Therefore, if the general requirement of convenience and the specific requirement of URI dereferencability is to be fulfilled, slash URIs are preferable.

A new problem arises when considering the requirement of avoiding ambiguity. While a hash URI can refer to abstract concepts[17], the specifications are unclear on whether a slash URI can refer to anything else but a document. Hence a triple ex:Concept dc:author ex:TimBL might not mean that TimBL was the creator of Concept, but merely that he authored a particular piece of RDF, depending on whether the namespace **ex** ends with a hash or a slash. This ambiguity was known in W3C as the *httpRange-14 issue*[18] where it was formulated as follows: should the set of URIs used to identify documents be disjoint from URIs that identify other resources? After a three year debate, the compromise[19] was that the HTTP response code should be used to disambiguate a HTTP URI: if a 2xx response is returned (and a document along with it), then the resource is an "information resource" (i.e. a document). If it returns a 303 redirect to a second URI, then the first URI can identify any resource (including abstract concepts).

Guideline 22: USE SLASH URIS AND 303 REDIRECTS WHEN PUBLISHING LARGE VOCABULARIES ON THE WEB. To avoid overloading agents with unnecessary data, use slash URIs. To avoid the ambiguity whether the URI refers to a document or other resource, use 303 redirects.

We recommend especially those organizations which publish large vocabularies to set up a server in their own domain and use redirects. This practice is not possible for each vocabulary

[17] The meaning of the fragment ID depends on the MIME media type of the retrieved object (Internet Engineering Task Force 2005). RDF's MIME type describes its semantics (Internet Engineering Task Force 2004), which in turn refers to (RDF Core Working Group 2004c). This last specification states that the part before the fragment is assumed to be an RDF document, even when the document does not exist at that location. Furthermore, it states that the fragment identifier can be used to identify anything.

[18] `http://www.w3.org/2001/tag/issues.html#httpRange-14`

[19] See `http://www.w3.org/DesignIssues/HTTP-URI.html` and `http://www.w3.org/DesignIssues/HTTP-URI2` by Tim Berners-Lee for a summary of different positions and the resolution.

publisher, as control of a web server is needed in order to configure redirects. An intermediate solution is to use a redirection service such as `http://thing-described-by.org/` or PURL (`http://purl.org/`). Using the former, one can prepend the document URL with the domain, e.g. `http://thing-described-by.org?http://en.wikipedia.org/wiki/Bank` and thus attach a web document (e.g. the Wikipedia entry for Bank) to a URI for an abstract concept (e.g. the WordNet synset "synset-bank-noun-2"). However, if the whole community would rely on only a few redirection services, this would represent an unacceptable small number of points of failure in the Semantic Web. The resources made available to the purl.org server have already been increased to be able to handle the frequent requests of PURL URIs from the Dublin Core namespace.[20] If uptake of services like PURL is large, other solutions should be explored.

Concise Bounded Descriptions

Dereferencability requires that relevant, descriptive content be returned when a URI is dereferenced. However, what is relevant or appropriate varies for machines and humans: for the former it is RDF/OWL, for the latter a natural language representation in HTML. It is possible to provide both depending on context using the *content negotiation mechanism* (W3C Technical Architecture Group 2004). Content negotiation allows providing different representations of the same resource. In HTTP this is implemented by the *Accept* header which contains the MIME type(s) the agent can process. There are six basic "recipes" defined in (Semantic Web Best Practices and Deployment Working Group 2006a) for publishing vocabularies on the Web of which four use content negotiation. The Recipes vary on whether hash or slash URIs are used and how much human-readable documentation is provided (none, single file for the whole vocabulary, file per resource).

Guideline 23: IMPLEMENT RECIPE FOUR, FIVE OR SIX FOR PUBLISHING VOCABULARIES. In case of large-scale vocabularies such as WordNet, implement Recipe four, five or six because these use slash URIs. Provide appropriate triples and human-readable content to satisfy the requirement of dereferencability, ideally Recipe six.

We implemented the sixth Recipe[21] by providing relevant triples for all WordNet URIs.[22] There are different options concerning what triples to return:

- a graph that contains a pre-defined set of properties if the resource has values for them (e.g. rdf:type, rdfs:subClassOf);
- all statements connected to the resource with some offset, e.g. everything connected in at most two steps;

[20] `http://lists.w3.org/Archives/Public/public-lod/2010Mar/0293.html`

[21] This conversion was developed in parallel with the Recipes document. The sixth Recipe was at the time a yet uncompleted extension of Recipe five that stated "*the RDF content being made available [...] such that clients can obtain a partial RDF description of the vocabulary as appropriate*" which is exactly our purpose.

[22] The human-readable descriptions are not yet available but the redirects are in place. Currently the RDF is also served on requests for HTML. A solution would to provide the URI to a generic RDF browser such as Tabulator (Berners-Lee et al. 2006) to automatically generate the description.

- the Concise Bounded Description of the URI (see below);
- the Symmetric Concise Bounded Description of the URI.

Ultimately it depends on the application context which set of vocabulary triples is an appropriate response to the application's request. As we cannot predict this we can only provide a reasonable "default". We think that Concise Bounded Descriptions are such a default. The CBD of a URI is calculated as follows (Stickler 2005):

> " *1. Include in the subgraph all statements in the source graph where the subject of the statement is the starting node;*
>
> *2. Recursively, for all statements identified in the subgraph thus far having a blank node object, include in the subgraph all statements in the source graph where the subject of the statement is the blank node in question and which are not already included in the subgraph.*
>
> *3. Recursively, for all statements included in the subgraph thus far, for all reifications of each statement in the source graph, include the concise bounded description beginning from the rdf:Statement node of each reification. "*

An example CBD for the noun "bank" is depicted in Figure 4.8. The *Symmetric* CBD not only includes statements for which the URI is the subject, but also those for which the URI is the object. The Symmetric CBD offers a more complete overview of a resource, because it also includes "inbound" triples. For example, the CBD of the synset "bank" will contain a hyponymOf triple with the synset for "slope" as an object, while the Symmetric CBD would also include hyponymOf triples with the synsets for "riverbank" and "waterside" as subjects and "bank" as object. Without those last two triples it is more difficult to traverse the graph (starting from "bank"). Typically the Symmetric CBD of a WordNet URI will only contain a few more triples, so we would advice to use Symmetric CBDs. At the time of writing, the W3C version only provides asymmetric CBDs.

```
wn20instances:synset-bank-noun2 a wn20schema:NounSynset ;
  wn20schema:synsetId "108639924" ;
  rdfs:label "bank";
  wn20schema:containsWordSense wn20instances:wordsense-bank-noun-2 ;
  wn20schema:gloss "(sloping land (especially the slope beside a body of water);
                   ''they pulled the canoe up on the bank'';
                   ''he sat on the bank of the river and watched the currents'')" ;
  wn20schema:hyponymOf wn20instances:synset-slope-noun-1 .
```

Figure 4.8 Concise Bounded Description for the URI *http://www.w3.org/2006/03/wn/wn20/instances/synset-bank-noun-2* as available online (online version uses RDF/XML instead of RDF/Turtle)

The CBDs can be calculated either by querying the RDF on-line, or by precomputing them off-line. The former has the advantage of always being up to date, the latter that requests can

be answered more quickly. As it is unlikely that the WordNet data will change often, the TF chose the second approach. The TF also chose to serve the "Full" version online, instead of the stripped-down "Basic" version (see Section 4.11).

Comparison

The earlier conversion is only provided as a downloadable file, so it does not conform to the requirement of dereferencability. Additionaly, the namespace it uses is in Princeton's web domain while (a) the files are not served there; and (b) there was no contact with Princeton to prevent URI clashes. This situation violates the requirements for dereferencability and avoiding ambiguity.

4.11 WordNet Basic and WordNet Full

The complete WordNet in RDF/OWL version described here is 166 MB uncompressed RDF/XML in size. The required memory footprint when loading all files into software such as SWI-Prolog's Semantic Web library[23] may be double that amount (figures vary for different software). However, it is clear that some of the use cases will require only parts of the complete WordNet. For example, for use cases such as use case 4 (transitive reasoning), only the hyponym relation is needed. From feedback of the community it proved that the size of WordNet poses real problems in terms of needed memory and query response times.[24] To mitigate these problems we have made separate files for each WordNet relation. The required footprint can be diminished by loading only those files that are required for the application at hand.

Furthermore, in use cases similar to use case 1 (annotation with synsets) only the synsets and the lexical forms of the senses are needed. The RDF for the WordSense and Word instances and the properties connecting them adds memory footprint which is not used. To keep the footprint small for such applications we provide WordNet Basic. WordNet Basic has adapted schema and instance files. The schema is a stripped-down version of the Full schema, as it does not have classes WordSense and Word, and the properties to connect them to each other and to Synset. It has one additional property in comparison to Full called senseLabel. This property is attached to Synset instances, and contains the values of all strings that are connected to the synset in the Full version through the chain $<$ Synset, synsetContainsWordSense, WordSense, word, Word, lexicalForm, xsd:string $>$ The instance RDF file that belongs to WordNet Basic does not have instances of WordSense and Word, and each Synset has a set of senseLabels.

The instance files for the WordNet relations can be loaded individually as for the Full version, with the caveat that only properties (relations) between Synsets are sensible to load. The relations antonymOf, seeAlso, participleOf and derivationallyRelated are between WordSenses, and can therefore not be used in WordNet Basic. All files referred to in this section are available online.[25]

[23] http://www.swi-prolog.org/pldoc/package/semweb.html

[24] http://lists.w3.org/Archives/Public/public-swbp-wg/2005Dec/0089.html

[25] http://www.w3.org/2006/03/wn/wn20/

Guideline 24: CONSIDER CREATING A FULL AND BASIC VERSION. When a vocabulary is large, and some use cases require only part of the vocabulary, consider creating Basic version and a Full version of the representation. Also consider publishing parts of the vocabulary in separate files so that users can load only the necessary part of the data.

A similar issue as presented here was long debated by the designers of SKOS. There discussion centered around whether whether terms could be concepts with URIs themselves or not. (Translated to the WordNet case: is a word sense just a literal attached to a synset or are word senses instances of a class WordSense, so that other properties can be attached to them.) Introducing terms as concepts in their own right would make the metamodel a bit more complex, while only a small number of vocabularies make use of this feature. The compromise was that the standard SKOS metamodel focuses on "basic" vocabulary representations: terms are just literals attached to concepts. The SKOS XL extension allows "full" representations where terms are instances of a class Label.

Mapping to SKOS

The mapping of WordNet Basic to SKOS is almost the same as that of WordNet Full. The difference is that for WordNet Basic it is possible to map the lexical labels to SKOS with a schema-based mapping. The senseLabel can be used as source of the mapping. The remaining problem is to which of the two SKOS label properties to map to (prefLabel or altLabel). Because SKOS disallows to have more than one preferred label per concept (in one language), skos:prefLabel is not an option. Therefore, we choose to map senseLabel to skos:altLabel.[26]

4.12 Comparison to Other Existing Conversions

Our conversion is based on our own analyses as well as studying existing conversions. To the best of our knowledge there are four other conversions: by Dan Brickley[27], Decker & Melnik[28], the University of Neuchatel[29] and the University of Chile (Graves and Gutierrez 2006). The last one was performed in parallel with the activities of the WN TF without both parties initially being aware of it.

The conversion by Brickley is a partial conversion, as only the noun-part of WordNet is converted. Of the relations only the hypernym relation is converted. Brickley converts the noun hierarchy into rdfs:Classes and the hyponym relationship into rdfs:subClassOf. This is an attractive interpretation, but we argue that not all hyponyms can be interpreted in that way. For example, the synset denoting the city "Paris" is a hyponym of the synset denoting "capital", but "Paris" should be an instance of "capital" instead of a subclass. An attempt to provide a consistent semantic

[26] In previous versions of SKOS, each concept had to have *exactly* one skos:prefLabel. In that case, a schema-based mapping of WordNet Basic to SKOS is not possible.

[27] See `http://lists.w3.org/Archives/Public/www-rdf-interest/1999Dec/0002.html`. Originally available at `http://xmlns.com/2001/08/wordnet/`, but not available anymore.

[28] Originally published at `http://www.semanticweb.org/library/`, not available anymore.

[29] `http://www2.unine.ch/imi/page11291_en.html`

translation of hyponymy has been done by Gangemi et al. (2003a), but in this work we explicitly avoid semantic translation of the intended meaning of WordNet relations.

The conversion by Decker & Melnik is also a partial one. It does convert all synset types, but only three of the WordNet relations. Another difference is that it attaches lexical forms as labels to the Synset instances. Hence WordSenses and Words do not have a URI.

The two previous conversions are based on an older version of Princeton WordNet and are not updated as far as the TF can tell. Both provide RDFS semantics only.

The conversion of Neuchatel is close to the one in this document. It has roughly the same class hierarchy, with one exception. It contains a class to represent word senses, but does not have a separate class for words. It does provide "union classes" like "Nouns_and_Adjectives" with subclasses Noun and Adjective like our conversion. However, it does not provide owl:unionOf statements to express OWL semantics of this class.

Anonther difference with this conversion is that Neuchatel is in OWL (e.g. all properties are either owl:ObjectProperty or owl:DatatypeProperty, while the conversion of the TF is both in RDFS and OWL (e.g. each OWL property is also defined to be an rdf:Property. The conversion by the TF splits some relations into sub-relations, because their semantics warranted such a separation. For example, the Prolog predicate `per` denotes (a) a relation between an adjective and a noun or adjective or (b) a relation between an adverb and an adjective. We convert `per` into `adjectivePertainsTo` and `adverbPertainsTo`. The Neuchatel conversion does not provide sub-relations, and omits relations "derivation" and "classification", and also does not provide inverses for all relationships. The conversion uses hash URIs, while the TF's uses slash URIs. The main advantages of the conversion by the TF in comparison to the Neuchatel conversion is that it is more complete, uses slash URIs, is interpretable by both RDFS and OWL infrastructure, and represents Words as first-class citizens.

Representing words as first-class citizens allows fine-grained mappings to WordNets in other languages. Future integration of WordNet with WordNets in other languages can be done on three levels: relating Synsets, relating WordSenses and relating Words from the different WordNets to each other. As the other conversions do not provide URIs for words, these only allow integration on the first two levels. For future integration of WordNet with other multilingual resources it is essential that one can refer to two different words with the same lexical form, or two words with a different lexical form but similar meanings.

The conversion by University of Chile was made in parallel to the efforts of this TF.[30] It has almost the same class hierarchy as this conversion; only the class Collocation is not present. The schema is modelled in RDFS, so it does not define restrictions, disjointness axioms, property characteristics and inverse properties. It does not have the superproperties for WN relations that we have introduced, and it uses hash URIs. The main technical advantages of the version by the TF is that it includes OWL semantics and that it uses slash URIs.

The previously mentioned conversions do not convert the frame sentences, while the TF's conversion and the conversion of University of Chile include them. A practical advantage of the TF's conversion over the other conversions is the availability of a Basic and Full version and

[30] `http://lists.w3.org/Archives/Public/public-swbp-wg/2006Jan/0048`

separate files for the WN relations. In summary, the advantages of the TF's conversion over other versions are that it is complete, uses slash URIs, provides OWL semantics while still being interpretable by RDF(S) infrastructure, provides a Basic and Full version, and provides URIs for words.

4.13 Discussion

In this chapter we addressed research question 2: "How can vocabularies be converted to an interoperable representation with given application-constraints?". The question is whether the method from Chapter 2 is suitable for application-specific conversions. To test this we made a new conversion of WordNet directly based on application requirements. The conversion differed from the previous one, the most prominent differences being URIs for WordSenses and Words, and the Full/Basic version. This shows that a new, application-specific method is necessary. This method should include an analysis of use cases and requirements. The use cases also showed that several guidelines from the earlier method are not sensible for any conversion. For example, it is usually best to provide a URI instead of a blank node, even when the source provides no identifier to base the URI on. A URI allows the node to be used in annotatations, which is what the concepts of a vocabulary are meant for.

Some differences we found, such as the missing relations and Collocation class, simply result from taking a more detailed look at WordNet. No particular methodological guidelines can prevent such oversights.

The conversion effort was useful to reveal errors and ambiguities in WordNet's Prolog representation. In other words, conversion can help in quality control. Additional error checking can be performed using the OWL restrictions, but this remains to be done.

We provided a WordNet Basic version to fulfill an implicit application requirement: limiting the size/content of the vocabulary to what is sufficient for the application's goals. We also mapped both Basic and Full versions to SKOS, to provide interoperability through a standard metamodel. The Full version was mapped to the SKOS XL extension. The usefulness of SKOS XL for WordNet is limited. It does allow mapping of WordSenses to skosxl:Label so that these can be used in e.g. annotation applications based on SKOS. However, it does not provide a standard set of relationships to which e.g. wn:antonymOf and derivationallyRelated can be mapped. Applications that wish to use these relationships need to interpret them and cannot rely on the SKOS metamodel alone.

The impact of our WordNet conversion is most apparent when looking at the "LOD cloud", the network of interlinked datasets produced by the Linked Open Data project. After our WordNet version went online, this project has produced datasets based on roughly the same principles as presented in this chapter (CBDs, use of the Recipes, decriptive and dereferencable URIs). WordNet itself plays a central role in the LOD cloud. This is evidenced by the fact that WordNet is strongly linked to the central hub of the cloud, DBPedia (an RDF conversion of Wikipedia), with 338,061 links (Hausenblas et al. 2008). Only the Flickr dataset has more links to DBPedia.

We feel that online publishing of vocabularies (with dereferencable URIs) deserves more at-

tention in the community. It is not enough to annotate data with URIs. A description of this URI should also be available to other users of your data (both computers and humans) so that when they encounter an unknown URI they can automatically look up its meaning.

Acknowledgements

We thank Aldo Gangemi and Guus Schreiber for co-authoring the conversion and the W3C Group Note. We also thank the reviewers of the W3C Note (Jeremy Carroll, Brian McBride, John McClure, Benjamin Nguyen and Jacco van Ossenbruggen) and the many contributors to the W3C Best Practices mailing list for their comments and suggestions. Dan Brickley and Brian McBride have contributed to the WordNet conversion described in the Note through their work in the WordNet Task Force and additional comments and suggestions. Special thanks to Ralph Swick for help in generating CBDs and setting up the conversion in W3C webspace. We also thank the MultimediaN e-Culture team, in particular Jan Wielemaker, for important usage comments. We acknowledge Antoine Isaac for cooperation in defining the WordNet-SKOS mapping.

5

Case Study: the Getty Vocabularies and the E-Culture project

In this chapter we focus on the same research question as in the previous chapter: "How can vocabularies be converted to an interoperable representation with given application-constraints?" In Chapter 4 it was shown that the generic method presented in Chapter 2 does not always result in a conversion that is suitable for applications. Some guidelines were added to the generic method. However, that chapter also indicated that some applications may have special needs not covered by a generic method. Therefore, this chapter investigates an adaptation of the generic method that focuses on application needs instead. We perform a case study where the requirements of an application are taken into account in the coversion process. The application is the MultimediaN E-Culture search and browsing application, the vocabularies to be converted are the Getty vocabularies (AAT, TGN, ULAN). The result is an adapted method for application-specific conversion.

An overview of the method is given in Appendix C.

5.1 Introduction

In this chapter we focus on research question 2: "How can vocabularies be converted to an interoperable representation with given application-constraints?" from the point of view of applications. We approach this research question by studying two issues: (1) how should the generic method be adapted to cater for application-specific conversions; and (2) how helpful is our adapted method for application-specific conversions? To answer these two questions we utilize a case study: conversion of the Getty vocabularies (a set of three vocabularies in the domain of cultural heritage) for the E-Culture search and browsing application. The E-Culture team did not follow a specific step-wise method in their conversion activities. By comparing the actual result of their conversion to our method we can determine which additional guidelines and/or steps are needed for a conversion of this type. By including a post-hoc analysis of the requirements of the E-Culture application and comparing that with the conversion results, we can determine if there are any discrepancies between the required and the actual representation. If guidelines and/or steps can be included or are already present in the method that help prevent these discrepancies, we may conclude that our method is useful in that respect.

A first modification to the method is that we drop the generic method's requirement on preservation of content and semantics. An application may require a representation that violates this

principle. A consequence is that the results of such a conversion cannot be published as a generic conversion.

A second set of modifications comes from reflecting on the overall steps before we start with the E-Culture case. We use as a reference our experiences in converting WordNet which was based on a set of use cases; this is an abstract form of application-oriented conversion (see Chapter 4). For example, use case 2 (annotation with word senses) showed that URIs for word senses are needed in the conversion. From that experience we deduce that determining a set of use cases is useful in guiding the conversion. We should also analyze the functionality the application needs to support, so that we can derive requirements on the vocabulary representation. To assist in this process we have to understand the goals of the application. To this end we add an "application description". The application description, requirements and use cases together form Step 0a.

In this chapter, we do not focus on Step 3 (standardization by mapping to SKOS). If the application developers wish to have a SKOS-based conversion (e.g. because their existing software works with SKOS-based representations) then this simply becomes one of the vocabulary requirements. After the vocabulary has been converted, it is still possible to conduct Step 3, e.g. to simplify integration of the vocabulary with other applications. We also do not discuss Step 4 (publication on the web) in this chapter, because the Getty vocabularies cannot be published online due to license restrictions. The step itself is still useful for application-oriented conversions. It is likely that the results of performing this step will not result in changes to the method, because the guidelines for that step are independent of the exact content and metamodel to be published. Finally, we keep the remaining steps: Step 0 contains a description of the digital and conceptual model (now called Step 0b), Step 1a concentrates on the basic conversion, Step 1b identifies explications of the syntax on the basis of the conceptual model, Step 2a adds OWL semantics and Step 2b specific interpretations.

5.2 Step 0a: Case Study Description

The case study description is part of step 0. It consists of an application description, application requirements and application use cases. The following is based on (Schreiber et al. 2006), the application[1] and unstructured interviews with the E-culture Project team. In this case it is a post-hoc analysis of the motivations of the application designers, but in general it should be performed before conversion.

Application Description

The E-Culture application is a cross-collection search and browsing application. It offers three types of search interfaces: keyword-based (Google-like), time-based search (user enters a structured query such as "Works of Picasso in the late period of his life") and faceted search (user progressively constrains the total set of works by selecting values from facets such as *creator*, *style* and *place*, see e.g. (Hildebrand et al. 2006)). Search results are displayed in a *result view*, which shows the image and title of each artwork, grouped in categories (e.g. "Works by Picasso",

[1] `http://e-culture.multimedian.nl/demo/search`

"Works by a colleague of Picasso". Each result view also displays (a) all artwork's locations on a Google map; and (b) a timeline which shows year of the artworks' creation and lifespan of artists). By clicking on an artwork, a *detail view* is opened, which shows all information associated with the work (i.e. all RDF triples displayed in a human-friendly format using the resources' labels). Keyword and time-search use a graph selection algorithm which first matches all literals in the RDF store to the relevant keyword(s). It returns all nodes that are a number of steps away from these literals. Each property has a weight, and when the total weight is above a threshold the next node is not included anymore. The grouping of results is also done through the graph selection algorithm. The backend of the application is written in SWI-Prolog.

The application contains different datasets, obtained from the Rijksmuseum Amsterdam (the ARIA masterpieces, about 750 objects), the Artchive.org website (about 4,000 works, mainly paintings) and Rijksmuseum Volkenkunde Leiden (about 80,000 ethnographic objects). The datasets were converted to RDF, resulting in almost 30,000 triples.[2] The ARIA dataset has its own ARIA vocabulary, the RVL its own SVCN vocabulary. For integration purposes the E-Culture team decided to use the Getty vocabularies as a target for mapping literals (e.g. creator values in Artchive) and collection-specific vocabulary concepts (e.g. object types in SVCN). Because there is no existing standard conversion of the Getty vocabularies available for use or adaptation, the E-Culture team made its own from scratch.

The application's backend is programmed in SWI-Prolog on top of its Semantic Web library (Wielemaker et al. 2003). At the time this case study was done the application ran on a 32-bit server with 8GB memory. In that configuration the theoretical maximum size of the triple base is 49M triples.

Application Use Cases

The following basic use cases were identified. Each use case describes a specific sequence of interactions of actor(s) with the system.

Use Case 1: KEYWORD SEARCH. User types in keyword(s). A *result view* is returned with a set of works from the dataset. (The keyword will usually refer to an artist, art style, artwork or geographic location. Less usual keywords will refer to materials, year (of manufacture), biographical information on artists, etcetera.)

Use Case 2: TIME-BASED SEARCH. User types in a structured query such as "Picasso in late period of life". A *result view* is returned where each work conforms to the time constraints. (Some background knowledge is used to translate time indicators to concrete years.)

Use Case 3: BROWSING TIMELINE. User browses the timeline to get an overview of the artists in the result view.

Use Case 4: BROWSING MAP. User browses the map to get an overview of the geographical location of artworks in the result view.

[2] A considerable set of works was not converted because they lacked an image or semantic description.

Use Case 5: FACET SEARCH. User limits the values in a facet (e.g. *location*) to constrain the result view. A new result view is produced that only contain works that have an annotation with the chosen value (e.g. *Europe*). All works with a subordinate value for the facet (e.g. *Germany*) are included.

Use Case 6: EXPLORATIVE BROWSING. User browses through the result sets and click through several Detailed Views to learn about: an artwork, art style, artist, relations between artists, etcetera.

The team also mentioned two use cases that are explicitly not supported: (1) updating of the vocabularies and datasets by museum specialists; and (2) finding provenance information (e.g. who entered the description of this artwork).

Application Requirements & Vocabulary Requirements

Based on the requirements posed on the application as a whole we can determine the requirements on the vocabularies. The main application requirement is that the application should support search and browsing over an integrated dataset of CH objects. Based on the application description the generic requirement can be refined. Firstly, to integrate the datasets it is necessary to (1) replace literals in the datasets with appropriate URIs; and (2) have one common metadata schema over which can be searched. Secondly, the E-Culture application should be multilingual, which means that the user should be able to switch between interface languages. Thirdly, the response times for the various types of searches should be acceptable to users. Based on the use cases and application requirements we can determine the following *vocabulary requirements* which are to be taken into account during conversion:

1. all time information should be converted (for use cases 2 and 6);
2. all geographic information should be converted (for use cases 1 and 4);
3. all hierarchical information should be converted (for use case 5);
4. all language information should be converted;
5. all concepts should have a URI.

These are *minimal requirements* that will satisfy the functional application requirements; if additional functionality should be supported this can result in additional vocabulary requirements. A general guideline that the team used during conversion was to keep the resulting metamodel as simple as possible and not to use reification. We come back to this last point later.

5.3 Step 0b: Digital and Conceptual Model

In this substep of Step 0 the the Getty vocabularies are analyzed to determine the content, semantics and the digital representation. This will assist accurate conversion based on the requirements

formulated in the previous substep. For this analysis we study the 2004 edition of the Getty vocabularies in the XML format, as this is the version used by the E-Culture project. First we give an overall introduction to the size and purpose of the vocabularies, then for each we discuss their conceptual and digital models.

Introduction to the Getty vocabularies

The Art and Architecture Thesaurus (Peterson 1994) defines 30,984 concepts in the domain of art and architecture that are related with broader/narrower/related relationships. AAT has seven facets such as "Materials", "Styles and Periods" and "Objects" .[3] Each facet has one or more named *hierarchies*; e.g. the facet "Agents" has the hierarchies "People" and "Organizations". It is used in art institutes around the world for indexing some of the metadata fields, such as the style or period and the type of objects.[4] The Union List of Artist Names is a list of 115,652 visual artists (e.g. painters, printmakers) and corporate bodies (e.g. publishers, architectural firms).[5] Its main information content is biographical. The Thesaurus of Geographical Names contains the names and geographical locations of 892,361 places.[6] It does not only contain current but also historic places and information on the type of place.

AAT

Digital model

Each of the three vocabularies has one set of XML files which contain the actual records and additional files that provide more information on parts of a record. To realize this, some of the XML subtags inside a record tag have identifiers which are used as a reference into the additional file. Subtags include those for representing contributors, languages and sources for terms. The total amount of uncompressed XML in the main files of AAT is 236MB. Figure 5.1 shows a sample record (original size over 400 lines) which has been simplified so that only the most interesting features remain (see Appendix B for the complete record).

Each record (tag `<Subject>`) has an ID, a record type, a descriptive note, exactly one preferred parent, zero or more associative relations with other records, one preferred term and zero or more non-preferred terms. The `<Record_Type>` can take values {Concept, Facet, Guide Term, Hierarchy Name}. Terms have an ID, are written in a particular language, can be vernacular[7], and have a "display order" (it numbers the terms of one concept starting from "1"). This is the preferred order to display terms in digital interfaces.

[3] `http://www.getty.edu/research/conducting_research/vocabularies/aat/about.html`
[4] `http://www.cwhonors.org/viewCaseStudy.asp?NominationID=112`
[5] `http://www.getty.edu/research/conducting_research/vocabularies/ulan/about.html`
[6] `http://www.getty.edu/research/conducting_research/vocabularies/tgn/about.html`
[7] A vernacular term is the name given to a concept in a local language, e.g. "Roma" for "Rome".

```
<Subject Subject_ID="300000206">
  <Record_Type>Concept</Record_Type>
  <Descriptive_Note> Complexes where plants or animals are raised
                    for livelihood or commerce.
  </Descriptive_Note>

  <Preferred_Parent>
    <Parent_Subject_ID>300125766</Parent_Subject_ID>
    <Relationship_Type>Parent/Child</Relationship_Type>
  </Preferred_Parent>

  <Associative_Relationship>
    <Relationship_Type>2000/related to</Relationship_Type>
    <Related_Subject_ID>300192802</Related_Subject_ID>
  </Associative_Relationship>

  <Preferred_Term>
    <Term_Type>Descriptor</Term_Type>
    <Term_Text>farms</Term_Text>
    <Term_ID>1000000206</Term_ID>
    <Display_Order>1</Display_Order>
    <Vernacular>Undetermined</Vernacular>
    <Preferred_Language>70052/American English</Preferred_Language>
  </Preferred_Term>

  <Non-Preferred_Term>
    <Term_Type>Alternate Descriptor</Term_Type>
    <Term_Text>farm</Term_Text>
    <Term_ID>1000289951</Term_ID>
    <Display_Order>2</Display_Order>
    <Vernacular>Undetermined</Vernacular>
    <Non-Preferred_Language>70052/American English</Non-Preferred_Language>
  </Non-Preferred_Term>
</Subject>
```

Figure 5.1 Simplified XML record for the AAT concept “farm”.

Conceptual model

AAT is a thesaurus designed according to the principles layed down in several standards such as the ANSI Z39.19-1980 and the ISO 2788-1986 (Peterson 1994). It is a monohierarchy of indexing concepts which have preferred and non-preferred terms. Usually the plural form of a term is the preferred term, while the singular form and spelling variants are non-preferred terms. The set of terms of one concept should be shown in a certain order in displays to humans (such as on the Getty website). At the top of the hierarchy there are seven *facets* such as Objects and Agents. *Hierarchies* are the next level divisions, e.g. the Agent facet has hierarchies People and Organizations. Below the hierarchies follow the actual indexing concepts. Between concepts in the hierarchy there can be so-called *guide terms* (called *node labels* in the standards). These terms are only meant to structure the hierarchy, not for indexing. For example, between the AAT concepts Chairs and its child Armchairs is the guide term <chairs by form> (guide terms are usually displayed with angle brackets). The designers have taken much care not to include part-of or instantive relations in the hierarchy, so that the broader/narrower hierarchy can usually be

interpreted as subclass.[8] AAT has some features that are not typical of thesauri in general, such as thirty types of associative relations between concepts (e.g. user/producer, field of study/practice), and language and display information provided for terms.

TGN

Digital model

The structure of the TGN XML sources is more complex than that of AAT and the files are also larger (the main files amount to 2.68GB). Figure 5.2 shows a sample record (original size over 350 lines) which has been simplified even more than the AAT record. For example, information on associative relations has been left out.

The TGN records generally use the same XML subtags as AAT's for a record's ID, type, descriptive note, associative and hierarchical relationships, and preferred and non-preferred terms. One of the few exceptions is the two additional tags for terms: the lexical category (noun, adjective or both) and the <Historical_Flag> (notes whether a term is used in current parlance or not). Another slight difference with AAT is that there are non-preferred hierarchical relationships.

The <Record_Type> can take values {Facet, Physical, Adminstrative, Both}. TGN also has XML structures that are not in AAT to represent *Place Types* (e.g. city, river) and coordinates. A coordinate is given in lattitude and longitude. In some cases the *bounding coordinates* are given, consisting of four coordinates contained in tags <Lattitude_Least>, <Lattitude_Most>, <Longitude_Least> and <Longitude_Most>. Within a <Coordinates> tag, the tags <Elevation_feet> and <Elevation_meters> may occur.

Conceptual model

TGN can also be seen as a thesaurus: it provides a set of indexing concepts that are hierarchically and associatively related, with one facet World as its single root. Each place can be either Physical (e.g. a mountain), Administrative (e.g. Amsterdam), or Both. The difference between the first two types is the basis for determining the boundaries: on some physical boundary (e.g. continent) or on human administrative choices (e.g. city limits). One of the three actual occurrences of type Both is The Low Countries, which describes both a geographic region as well as the political association formed by Belgium, The Netherlands and Luxembourg. TGN is different from AAT in that its concepts are structurally more similar to each other: all concepts are named geographic regions with coordinates, while AAT contains concepts as diverse as art styles and colours.

The hierarchical relation should not be interpreted as subclass like in AAT, but as geographic containment: a place is hierarchically related to another if it is located inside the bounding box of the other place. If e.g. a river runs through two countries, then the river is placed below the direct parent of the two countries. An administrative place can have a physical parent and the other way around (see the example in Figure 5.3). The hierarchy is also different from that of AAT in that

[8] As far as we can tell from our unstructured survey of the AAT hierarchy it can be safely interpreted as a subclass hierarchy. Of course some choices can be debated. E.g. according to the OntoClean methodology the class Furniture should be interpreted as a type restriction instead of a proper superclass of Chairs, Tables etcetera.

```
<Subject Subject_ID="7000354">
  <Record_Type>Administrative</Record_Type>
  <Descriptive_Note>Located on fertile plain; one of Morocco's four imperial...
  </Descriptive_Note>

  <Coordinates>
     <Standard>  <Latitude>
       <Degr>31</Degr> <Min>49</Min> <Sec>00</Sec> <Dir>North</Dir>
       <Decimal>31.8167</Decimal>
     </Latitude>
     <Longitude>
       <Degr>008</Degr> <Min>00</Min> <Sec>00</Sec> <Dir>West</Dir>
       <Decimal>-8</Decimal>
     </Longitude>  </Standard>
  </Coordinates>

  <Preferred_Place_Type>
    <Place_Type_ID>83002/inhabited place</Place_Type_ID>
    <Display_Order>1</Display_Order>
    <Historic_Flag>Current</Historic_Flag>
    <Display_Date>founded by Yusuf ibn-Tashfin in 1062</Display_Date>
    <Start_Date>800</Start_Date>  <End_Date>9999</End_Date>
  </Preferred_Place_Type>

  <Non-Preferred_Place_Type>
    <Place_Type_ID>83110/capital</Place_Type_ID>
    <Display_Order>10</Display_Order>
    <Historic_Flag>Historical</Historic_Flag>
    <Display_Date>of Almoravid dynasty, until 1147; of Morocco, 1550-1660
    </Display_Date>
    <Start_Date>1062</Start_Date>  <End_Date>1660</End_Date>
  </Non-Preferred_Place_Type>

  <Preferred_Term>
    <Term_Type>Noun</Term_Type>
    <Term_Text>Marrakech</Term_Text>
    <Term_ID>92316</Term_ID>
    <Display_Order>1</Display_Order>
    <Historic_Flag>Current</Historic_Flag>
    <Vernacular>Vernacular</Vernacular>
  </Preferred_Term>

  <Non-Preferred_Term>
    <Term_Type>Noun</Term_Type>
    <Term_Text>Marrakesh</Term_Text>
    <Term_ID>169061</Term_ID>
    <Display_Order>2</Display_Order>
    <Historic_Flag>Current</Historic_Flag>
    <Vernacular>Vernacular</Vernacular>
  </Non-Preferred_Term>
</Subject>
```

Figure 5.2 Simplified XML for TGN record "Marakesh".

places can have multiple parents. For example, Milan has Italy as preferred parent and the Roman Empire as non-preferred parent.

The location of a place is given as a single coordinate in lattitude and longitude (indicating its

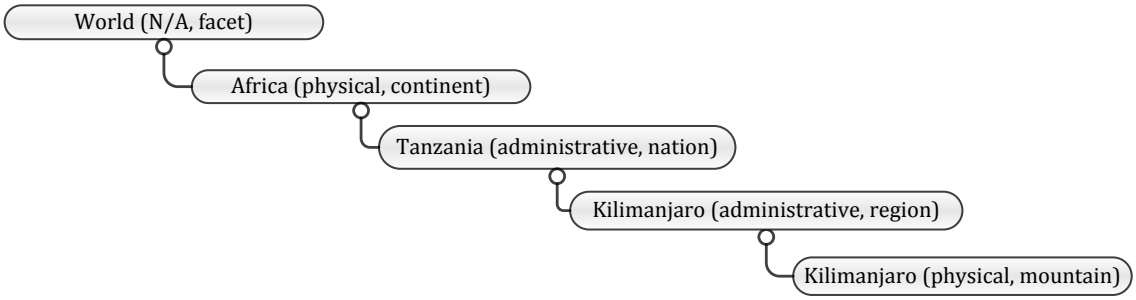

Figure 5.3 Part of the TGN hierarchy. An arrow with a round head is used to indicate hierarchical containment. In parenthesis after a place are its record type and placetype. The facet World does not have a record type.

center or source in case of e.g. a river), or as a bounding box[9] defined by the minimal and maximal longitude, and minimal and maximal lattitude.

Additionally, each place can have different types (e.g. capital, religious center) during its lifespan. Terms can have a start and end year when they were in use. The end year 9999 in the XML means that the term is still in use today. In contrast to AAT, TGN is a multilingual thesaurus. Besides giving terms in different languages it also indicates the lexical type of terms and which term within one language is the preferred one.

ULAN

Digital model

The structure of ULAN's XML is also more complex than that of AAT, and larger in size (787MB uncompressed XML). In Figure 5.4 is a portion of the record for the painter "Rembrandt" (original size over 5,000 lines, of which around 90% consists of information on editorial changes). Information on preferred and non-preferred terms, hierarchical relations and display order (e.g. for roles) has been left out. The compositional structure of tags that are also found in AAT is generally the same; exceptions include optional start and end years for terms.

The <Record_Type> can take values {Person, Corporate Body}. ULAN-specific tags specify artist biographies, roles, events and nationality. They have preferred and non-preferred versions. Events have a type (e.g. active, exhibition, baptism). The tag comes in a preferred and non-preferred version and includes subtags to represent a place and time period.

Conceptual model

As the name suggests, ULAN is a list of concepts instead of a thesaurus. It has little hierarchical structure. The parent relation is used to relate a concept to a facet (either Person or Corporate Body), which is the same as the usage of the record type: it gives typing information, not hi-

[9] The box with the minimum volume that still encloses all points of which the place consists. See http://en.wikipedia.org/wiki/Minimum_bounding_rectangle

```
<Subject Subject_ID="500011051">
  <Record_Type>Person</Record_Type>
  <Descriptive_Note> Rembrandt was one of the most popular and influential ...
  </Descriptive_Note>

  <Associative_Relationship>
    <Relationship_Type>2602/influenced</Relationship_Type>
    <Related_Subject_ID>500027532</Related_Subject_ID>
  </Associative_Relationship>

  <Associative_Relationship>
    <Relationship_Type>1101/teacher of</Relationship_Type>
    <Related_Subject_ID>500015747</Related_Subject_ID>
    <Display_Date>between 1648 or 1650 and 1653 in Amsterdam</Display_Date>
    <Start_Date>1648</Start_Date>
    <End_Date>1653</End_Date>
  </Associative_Relationship>

  <Preferred_Biography>
    <Biography_ID>4000028251</Biography_ID>
    <Biography_Text>Dutch painter, draftsman and printmaker, 1606-1669</Biography_Text>
    <Birth_Place>4390330011/Leyden (South Holland, Netherlands)</Birth_Place>
    <Birth_Date>1606</Birth_Date>
    <Death_Place>4390000029/Amsterdam (North Holland, Netherlands)</Death_Place>
    <Death_Date>1669</Death_Date>
    <Sex>Male</Sex>
  </Preferred_Biography>

  <Preferred_Event>
    <Event_ID>12002/active</Event_ID>
    <Place>4390000029/Amsterdam (North Holland, Netherlands)</Place>
    <Start_Date>1631</Start_Date>
    <End_Date>1669</End_Date>
  </Preferred_Event>

  <Preferred_Nationality>
    <Nationality_Code>905020/Dutch</Nationality_Code>
  </Nationalities>

  <Preferred_Role>
    <Role_ID>31100/artist</Role_ID>
  </Preferred_Role>
  <Non-Preferred_Role>
    <Role_ID>31442/etcher</Role_ID>
  </Non-Preferred_Role>
  <Non-Preferred_Role>
    <Role_ID>31261/painter</Role_ID>
   </Non-Preferred_Role>
</Subject>
```

Figure 5.4 Simplified XML for ULAN record "Rembrandt". The second <Associative_Relationship> models a teacher/student relationship between Rembrandt and Willem Drost (Subject ID 500015747).

erarchical information. The record type contains the same information, so the parent relation is redundant. The associative relation of type member of (which relates a person to a corporate body)

can be interpreted as a partitive relation. It occurs 2100 times. This does not provide a real hierarchy like in TGN, because the parent corporate bodies do not have parents themselves, i.e. there is no tree structure.

Associative relations in ULAN have two additional arguments compared to AAT: a place and time. Events are happenings in the life of persons or corporations such as baptisms, positions held and contests won. The difference between an event and an associative relation is that the latter relates two persons or corporations to each other (e.g. teacher/student), while the former only includes one ULAN person or corporation.

Concepts can have several roles throughout their life, which in ULAN are typically art-related (e.g. painter, etcher, art academy). For each role a location and start/end year are provided. It also contains biographical information in the form of a person's nationality, gender, and year and place of birth/foundation and death/termination. Like TGN, ULAN is multilingual. Besides the "overall" preferred term there is also one term per language that is the preferred term *within that language*.

5.4 Step 1a: Structural Translation

In this section we describe which basic conversion choices were made, which information was not converted, and how n-ary relationships were converted. Note that the step's name was changed to reflect that not all information is necessarily preserved in this adapted step.

Basics: record types, hierarchy and URIs

The tag <Subject> contains one record, which represents one specific vocabulary concept. E-Culture introduced a vp:Subject and made all concepts an instance of this class. A namespace **vp** (for Getty *vocabulary program*) was introduced for classes and properties that represent the shared basic structure of all three vocabularies. The specific namespaces **aat**, **tgn** and **ulan** contain those specializations and extensions required for the separate vocabularies. Adding the subclasses aat:Subject, tgn:Subject and ulan:Subject allows for separating the vocabulary records from each other when needed. E-Culture chose to use the unique IDs to create URIs. Additional typing of concepts is contained in <Record_Type>. For example, in the record representing Rembrandt the <Record_Type> has the string "Person" as value. The tag can be seen as a binary relationship between a record and a string. Similar to other conversions presented in this thesis, the E-Culture team interpreted the strings as subclasses of the main record class, as shown in Figure 5.5.

Instances of vp:Subject and its subclasses are hierarchically related through vp:prefParent and vp:nonPrefParent (domain and range vp:Subject). Properties are introduced to represent simple data attached to records (e.g. vp:descriptiveNote and vp:Id for the corresponding tags). As these choices are similar to those of conversions described earlier we do not elaborate on these any further.

In this phase, none of the described properties and classes can be left out (e.g. the hierarchy is required for faceted search). However, the E-Culture team observed that it did not need all the records from TGN and ULAN, because its datasets do not mention all places and all artists.

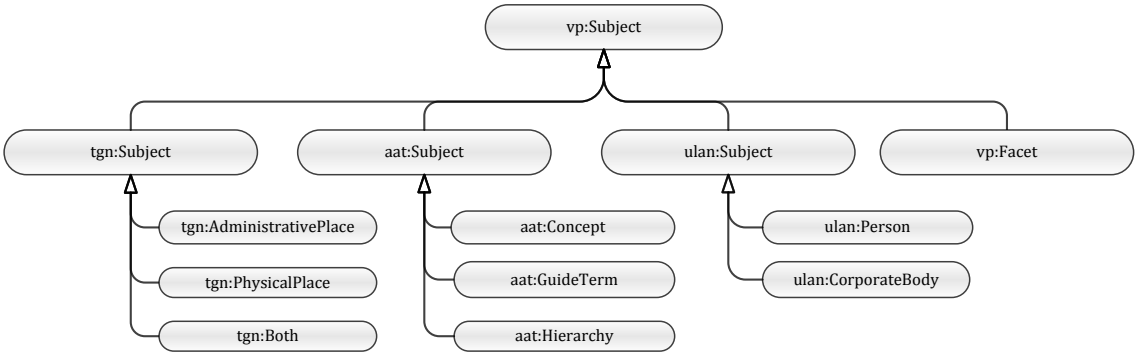

Figure 5.5 Main class hierarchy of the Getty vocabularies.

Therefore the coverted records were split into subsets and only those subsets that were required were actually loaded into the application. This results in a smaller triple base on the server which helps to improve response times.

Information not converted

The E-Culture team decided not to convert seven tags and their subtags at all as they were judged not to contain information relevant to the application requirements. The meaning of the tags is summarized in Table 5.1. The tags can be divided into three groups: tags related to the management and provenance of the vocabulary (the revision history, note contributors, merged status, facet code, legacy ID), tags that provide redundant information (hierarchy tag) and tags related to display (sort order). These choices are consistent with the requirements and use cases formulated earlier.

Tag	Meaning
<Merged_Status>	proposed merge with other concept performed
<Facet_Code>	code of containing facet
<Sort_Order>	order of display for sibling concepts
<Legacy_ID>	identifier in now superseded ID system
<Note_Contributor>	source of the descriptive note
<Hierarchy>	all labels of all parent concepts in one string
<Revision_History>	record of all modifications of concept
<Contributor>	ID, brief and full name of contributor
<Language>	ID and long name for language of terms
<Merged_Subject>	maps IDs of two deprecated concepts to new merged concept
<Source>	ID and bibliographic information on a cited source for years, concepts, etc.

Table 5.1 Tags that were not converted by the E-Culture team.

Conversion patterns for N-ary relationships

The Getty vocabularies have n-ary relations (i.e. relations with arity higher than two). For example, the tag `<Preferred_Place_Type>` models a relation between a a place, a type, a textual note and a start and end year. N-ary relations are problematic in RDF/OWL as they cannot be converted to a (binary) property. This can be solved with the *relation instance pattern* (RI): create an instance for the relation and attach all arguments to that instance using one new property per argument. The *relation instance pattern* was already mentioned in guideline 4 and is described in detail in (Semantic Web Best Practices and Deployment Working Group 2006b). Another option for lossless conversion of n-aries is to use RDF reification. However, E-Culture explicitly refrains from reification as it contends that reification will not be a part of the RDF standard in a few years. Furthermore, the reified arguments have a different status compared to the usual two property arguments: they should be *about* the statement, not part of the statement itself (Semantic Web Best Practices and Deployment Working Group 2006b). It is unclear in the E-Culture case for which arguments this would hold, and also unclear what practical benefit the application could have from separating the two kinds of arguments using reification. When faced with this type of uncertainty it is advisable not to distinguish the arguments from each other, and so the relation instance pattern is the safest option.

Guideline 25: USE THE RELATION INSTANCE PATTERN (RI) FOR LOSSLESS CONVERSION OF N-ARY RELATIONS. In order to follow guideline 4 (translate relations of arity three or more into structures with blank nodes) we recommend applying the pattern for representing n-ary relations as described in detail in (Semantic Web Best Practices and Deployment Working Group 2006b), and the additional advice described below. This results in a lossless conversion. However, we recommend not to use blank nodes as prescribed by Semantic Web Best Practices and Deployment Working Group (2006b) and guideline 4, because this prohibits later use of the node in annotation scenarios.

Alternatively, it is possible to drop all of a relation's arguments except two and then convert to a property. We call this the *reduction pattern* (RED). Reduction is a way to simplify the metamodel of the conversion, but results in information loss. Because E-Culture desires a simple model it made use of this pattern for most of the n-aries; information that was deemed unnecessary could be dropped. An additional benefit is that the application logic does not have to be aware of instances that actually represent relationships (as is the case with the relation instance pattern).

Guideline 26: USE THE REDUCTION PATTERN (RED) TO SIMPLIFY N-ARY RELATIONS. To simplify an n-ary relation, select from the arguments the two central ones and convert to a property. The loss of semantics is higher when the relation is essentially between three (or more) entities, e.g. a purchase-relation between a product, a seller and a buyer. Other arguments such as notes may represent a smaller information loss.

In Table 5.2 the n-ary relations in the Getty vocabularies are listed with their arguments. Some are compulsory in the XML, some are optional. Also listed is which conversion pattern the E-Culture team has used in their conversion.

One special case is the tags which model a biography (<Preferred_Biography> and <Non-Preferred_Biography>). They contain items such as year and place of birth. However, these items do not together form an n-ary relation. They are simply grouped together because ULAN lists multiple biographies which come from different sources. The "arguments" can be converted separately, into properties such as ulan:birthDate and ulan:birthPlace. However, because there can be multiple biographies for one person, this results in ambiguity: it becomes unclear which birthdates and birthplaces belong together. Because the E-Culture team decided only to convert the *preferred* biography, this ambiguity does not occur.

Relation	**Vocab.**	**Compulsory Args.**	**Optional Args.**	**Pattern**
Term	all	*term type*, *term ID*, term, *display order*, *vernacular flag*, *display name*	*language*, *start/end year*, *note*	RED
Associative Relation	AAT	relation type, related Subject ID	*place*, *start/end year*, *historic flag*, *note*	RED
Event	ULAN	event type, place, *display order*	*start/end year*	RI
Placetype	TGN	place type, *display order*, *historic flag*	*note*, *start/end year*	RED
Role	ULAN	role, *display order*, *historical flag*		RED
Nationality	ULAN	nationality, *display order*		RED

Table 5.2 Table of n-ary relations and conversion pattern results. An argument is compulsory when the XML schema enforces the presence of the tag that encodes it. An implicit argument we do not put in the table is the record in which the relation occurs. In italic are arguments that are lost in the E-Culture conversion.

As can be seen in Table 5.2, E-Culture used the reduction pattern for all relations except the Event. For Events the relation instance pattern was used instead, but only one of the additional arguments was converted: the place. Most relations come in a preferred and non-preferred version in the XML, which is reflected in the property names chosen by E-Culture (for example, vp:prefTerm and vp:nonPrefTerm). Table 5.3 shows the properties E-Culture created for this step. In the subsection below an example is given of how a relationship can be converted with either pattern. Also, some advice complementary to the Working Group's pattern is given.

Example: converting the Term relation

In some vocabularies terms are literals on which no more information is provided than whether they are preferred or non-preferred. In the Getty vocabularies however, the relation between a concept and a literal is more appropriately interpreted as an n-ary relation. A term-relation is contained in the tags <Preferred_Term> and <Non-Preferred_Term> which exhibit the same compositional structure. The subtags encode the following compulsory arguments: term type, term ID, the term string itself, display order, vernacular flag, display name flag. Optional arguments (subtags) are: language, start/end year, and a note. The type can have the values {Descriptor, Alternate Descriptor, Used For Term} in AAT. A Descriptor is simply a preferred term. If the preferred term is in plural form, then its singular form is included as a non-preferred term of type

Property	Domain	Range
vp:descriptiveNote	vp:Subject	rdfs:Literal
vp:id	vp:Subject	rdfs:Literal
vp:prefParent	vp:Subject	vp:Subject
vp:nonPrefParent	vp:Subject	vp:Subject
vp:prefTerm	vp:Subject	rdfs:Literal
vp:nonPrefTerm	vp:Subject	rdfs:Literal
ulan:birthDate	ulan:Subject	**xsd:gYear**
ulan:deathDate	ulan:Subject	**xsd:gYear**
ulan:birthPlace	ulan:Subject	tgn:Subject
ulan:deathPlace	ulan:Subject	tgn:Subject
ulan:event	ulan:Subject	ulan:Event
ulan:eventPreferred	ulan:Subject	ulan:Event
ulan:eventNonPreferred	ulan:Subject	ulan:Event
ulan:eventPlace	ulan:Event	tgn:Subject
ulan:gender	ulan:Subject	ulan:GenderValue
ulan:nationalityPreferred	ulan:Subject	ulan:Nationality
ulan:nationalityNonPreferred	ulan:Subject	ulan:Nationality
ulan:rolePreferred	ulan:Subject	ulan:Role
ulan:roleNonPreferred	ulan:Subject	ulan:Role
ulan:prefBioNote	ulan:Subject	rdfs:Literal
ulan:nonPrefBioNote	ulan:Subject	rdfs:Literal
tgn:elevation	tgn:Subject	**xsd:decimal**
tgn:placeTypePreferred	tgn:Subject	tgn:PlaceType
tgn:placeTypeNonPreferred	tgn:Subject	tgn:PlaceType
tgn:standardLattitude	tgn:Subject	**xsd:decimal**
tgn:standardLongitute	tgn:Subject	**xsd:decimal**
tgn:boudingLatitudeLeast	tgn:Subject	**xsd:decimal**
tgn:boundingLongitudeLeast	tgn:Subject	**xsd:decimal**
tgn:boundingLatitudeMost	tgn:Subject	**xsd:decimal**
tgn:boundingLongitudeMost	tgn:Subject	**xsd:decimal**

Table 5.3 Properties in the E-Culture conversion. Not shown are properties that model associative relations in AAT and ULAN (e.g. aat:user/producer), and those that model ULAN event types (e.g. ulan:baptism).

Alternate Descriptor. Used For Terms are non-preferred terms (i.e. their usage is redundant to the `<Non-Preferred_Term>` tag). In TGN and ULAN the tags for language come in a preferred and non-preferred version. This indicates whether the term is the preferred term *within that language*.

The relation can be converted using either of the two generic conversion patterns for n-ary relationships. The E-Culture team used the reduction pattern, so that the relation between a vp:Subject and a literal can be represented as an rdf:Property. The resulting properties are vp:prefTerm and vp:nonPrefTerm. The `xml:lang` attribute can be used to encode the language argument. However, E-Culture did not use `xml:lang` in its ULAN and TGN conversions at all; all language information was lost. Consultation with the E-Culture team made it clear that this was an unintended oversight.

If the relation instance pattern was applied, then the preferred term in Figure 5.1 would be

```
@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#> .
@prefix vp: <http://e-culture.multimedian.nl/ns/getty/vp/> .
@prefix aat: <http://e-culture.multimedian.nl/ns/getty/aat/> .

aat:Concept a rdfs:Class .
vp:Term a rdfs:Class .
vp:TermType a rdfs:Class .
vp:TermRelation a rdfs:Class .
vp:Vernacular a rdfs:Class .

aat:descriptor a vp:TermType .
vp:undetermined a vp:Vernacular .

aat:1000000206 a Term ;
    vp:termText       "farms"@en-US ;
    vp:id             "1000000206"^^xsd:string .

aat:300000206 a aat:Concept ;
  vp:prefTerm farms-term-1 .

aat:farms-term-1 a vp:TermRelation ;
  vp:termType       vp:descriptor ;
  vp:term           aat:1000000206 ;
  vp:displayOrder   "1"^^xsd:positiveInteger ;
  vp:vernacular     vp:undetermined ;
  vp:prefLanguage   "true"^^xsd:boolean ;
  vp:id       "300000206"^^xsd:string .
```

Figure 5.6 RDF representation of the <Preferred_Term> in Figure 5.1 after applying the relation instance pattern. A concept is related to the term farms. The definitions of properties have been left out for brevity.

converted into the RDF in Figure 5.6. We have split the language tag into two separate components in the RDF. The reason is that it is beneficial to use the standard XML support for languages. In this case, the remaining piece of information (the preferred status within the language) can be encoded with a boolean property. Applying the relation instance pattern would require 10-15 triples for each occurrence[10] of a term tag, depending on which optional arguments are present.

Summary and Evaluation

Here we summarize the choices E-Culture made in this first step and evaluate if these choices were appropriate given the requirements and use cases. Firstly, the basic classes and properties are no different from those in a "full" conversion. E-Culture also decided to only use the records that are needed by the application, which is a reasonable decision.

Secondly, some XML tags were not converted: tags related to the management and provenance of the vocabularies, tags that provide redundant information and tags related to display order.

[10] We do not count schema-level triples that only need to be added once, such as <vp:undetermined, rdf:type, vp:Vernacular>

Relation	Arguments	Requirements
Term	(pref)language,	R4 (multilinguality)
AssociativeRelation	start/end year, place	R1 (time); R2 (geographic)
Events	start/end year	R1 (time)
Place type	start/end year	R1 (time)
Roles	-	-
Nationality	-	-

Table 5.4 Violations of requirements and use cases in relation conversion. The column "Arguments" lists relationship arguments that were not converted by E-Culture, "Requirements" lists which violations this causes.

When reviewing the requirements and use cases there is none that would require this information. Provenance information is even explicitly mentioned in a use case that will not be supported.

Relation	Occurences	RED	RI	Total RED	Total RI
Term	1,462,200	1	15	1,462,200	21,933,000
Association	66,393	1	9	66,393	597,537
Event	10,882	1	9	10,882	97,938
Place-type	955,408	1	8	955,408	7,963,264
Role	192,760	1	5	192,760	963,800
Nationality	134,163	1	3	134,163	402,489
Total	2,821,806			2,821,806	31,638,028

Table 5.5 N-ary relationships and triples needed to represent them. The column "RED" states the triples needed for the reduction pattern and the column "RI" for the relation instance pattern. Triples for optional arguments are also counted. If the optional triples are subtracted the total number of triples for the RI pattern is 21,096,193.

Thirdly, the E-Culture team converted most of the n-ary relationships with the *reduction pattern* (RED) to simplify the metamodel. In the majority of cases this is in opposition with a vocabulary requirement (see Table 5.4). For example, excluding the start/end year of Events violates Requirement 1.

We highlight two other unsuitable choices. Firstly, E-Culture used blank nodes to represent Events, which is in opposition with Requirement 5. Secondly, E-Culture decided not to include a term's *display label*, which is used to give the "normal order" of a term (e.g. "Rembrandt van Rijn") instead of the "inverted order" ("Rijn, Rembrandt van"). In hindsight the team thinks it needs both labels. The "normal order" is useful for displaying e.g. an alphabetical list of artists, the "inverted order" for displaying e.g. the name of an artist below a painting he made.

The consequences of applying the reduction pattern vs. the relation instance pattern for the size of the triple base is shown in Table 5.5. Multiplying the required triples with the actual occurrence of the relation in the XML shows that the difference ranges from about twenty to thirty million triples (depending on how many times the optional attributes occur). The total size of the whole conversion is even higher, because each record also has simple binary properties (e.g. to

represent its type, notes, biography, coordinates etcetera). As the demonstrator contained about nine million triples in total (including the collections) at the time of the Semantic Web Challenge, a full conversion clearly puts a considerable burden on the infrastructure. This is without considering the tags that were not converted at all, such as the concept's revision history.

5.5 Step 1b: Explication of Syntax

The E-Culture team made four syntactic explications. The first was to introduce a superclass tgn:Place for tgn:AdministrativePlace and tgn:PhysicalPlace. Because both classes model regions of the earth (only with different criteria for determining boundaries of instances) this is a valid explication of the shared semantics. The second explication concerns the property vp:termType that is used to indicate a type of term (in AAT) and the lexical category of the term itself (in TGN). To separate these two meanings a property tgn:lexicalType with range tgn:LexicalType was introduced. The third explication was to declare vp:parentPrefered and vp:parentNonPreferred subproperties of a generic vp:parent property (and likewise for all other properties that have preferred/non-preferred versions), and to make the properties vp:labelPreferred and vp:labelNonPreferred subproperties of rdfs:label. The class tgn:Place and property vp:parent are used in the application code to simplify querying the triple base. The other explications have no support in any use case (e.g. the term type is not of interest to users).

A fourth explication is to remove the class tgn:Both from the schema and to convert each into two separate instances: one of tgn:AdministrativePlace and one of tgn:PhysicalPlace. The reason is that a political entity (e.g. The BeNeLux) is conceptually different from the geographic space it occupies; they have different *identity criteria* (Guarino and Welty 2004). In other words, the type Both is an error from an ontological perspective. However, as there are only three instances of Both, the effect on the application of this explication is negligable.

An explication that the E-Culture team could have made but did not, is to distinguish between the different hierarchical relations in the Getty vocabularies. In AAT the parent relation represents a subclass relation; the concepts are classes. For example, the concept armchair can have concrete instances and is a proper subclass of chair. In TGN the parent relation represents a partitive relation[11] called the *place-area relation*, because the hierarchy represents spatial containment. In ULAN the parent relation has no hierarchical meaning, but the associative relation of type member of can be interpreted as a *member-bunch relation*; e.g. Tanzania is member of United Nations. However, the E-Culture use cases only need a generalized hierarchical relation to support facet browsing (see Requirement 3): the nature of the hierarchy is not important, as long as it is clear that selecting a child concept reduces the number of matches. Separating the geographic hierarchy of TGN from the conceptual one of AAT might be used to improve the generic graph selection algorithm, but no studies have been done to confirm this. In summary, this explication has no proven practical value in the E-Culture case.

[11] See Odell (1994) for a list of partitive relations and their meaning.

5.6 Step 2a: Explication of Semantics

The E-Culture team made two semantic explications. The first explication that E-Culture made concerns the transitivity of hierarchical relations vp:prefParent and vp:nonPrefParent. These can correctly be interpreted as transitive relations. However, E-Culture declared only the vp:prefParent to be transitive. The motivation of the E-Culture team is that only the main hierarchy is important in search. This may not be entirely justified. For example, a searcher may be interested in all objects found within the boundaries of the Roman Empire, but will then not get back art objects which are annotated with its non-preferred grandchild Rome.

A second explication is that the vp:prefParent property is an owl:FunctionalProperty, because in any context its cardinality is exactly one.

Again, both explications have only limited value in the E-Culture case. Transitivity is only used by the Facet Browser. For the graph selection algorithm it is even desirable to be able to distinguish how many steps two nodes are removed from each other, because this is used to determine relevancy. It stands to reason that when one is looking for images of France, then an image of Paris is more relevant than an image of its subordinate concept Montmartre. Because OWL inferencing would assert the transitive triples the algorithm cannot count the distance between subordinate nodes without additional processing. Declaring a property as functional also currently has little benefit for E-Culture. If one concept accidentally has two parents, this should result in a report on erroneous data, not in inferring that the two parents are actually the same concept. Other possible explications, such as declaring tgn:Place an owl:unionOf the classes tgn:AdministrativePlace and tgn:PhysicalPlace and declaring inverse properties are also not useful for E-Culture.

5.7 Step 2b: Interpretation

The E-Culture team has made no interpretations similar to those described in Step 2b of conversions presented earlier in this thesis; e.g. the parent relation is not interpreted as subclass. However, analysis shows that some concepts in each separate vocabulary are close in meaning or overlap with concepts in another vocabulary (e.g. places in ULAN's events and places in TGN). A possible explanation is that the Getty editors used each other's results in the construction of the separate vocabularies, but copied concepts to their own editing space instead of maintaining a link by reference.[12] E-Culture has attempted to reconstruct the implicit links so that the graph-based selection algorithm can traverse from one vocabulary to another. Without restoring this link, it would not be possible for the algorithm to find a relation between e.g. art objects annotated with tgn:Amsterdam and artists in ULAN who were involved in Events in ulan:Amsterdam. The following implicit links have been identified:

- ULAN Places (objects in ulan:birthPlace and ulan:eventPlace) to instances of tgn:AdministrativePlace (birthplace/eventplace is usually a city);

[12] The current on-line edition does have some of the explicit links between the vocabularies, e.g. between ULAN birthplace and TGN place. We have no knowledge on how the links were re-established.

- instances of ulan:Role to subconcepts of <aat:people by occupation> (this AAT guide term has roles such as painter and sculptor below it);
- ulan:Nation to tgn:Places that represent nations (if a person has a nationality, s/he is a citizen of the nation).

The first two are equivalence links, the third concerns highly related concepts. The E-Culture project attempted to automatically reconstruct the links with heuristics as described below. The results are summarized in Table 5.6.

Source instances	Target instances	Matched src.	Unmatched src.
ulan:Place (6,005)	tgn:AdministrativePlace (364,262)	5,995	10 (0.002%)
ulan:Nationality (1,856)	tgn:AdministrativePlace nations (480)	400	1,456 (72%)
ulan:Role (882)	concepts below <aat:people by occupation> (950)	637	245 (28 %)

Table 5.6 Restored links between the Getty vocabularies. In parenthesis are the number of instances that are candidates for the particular mapping. The third and fourth column state how many source instances could be matched and how many remained unmatched.

Matching ULAN places to TGN uses the structure of the text values in ULAN, consisting of the place itself and its parents in parenthesis, e.g. "Amsterdam (North Holland, Netherlands)". In the first step the place itself is matched to the terms in TGN, which delivers a set of candidate TGN instances. In the second step a score is calculated for each candidate based on different factors, such as the number of matches from terms in parenthesis to TGN terms. The highest scoring candidate is chosen. Each matching ULAN term was removed from the ULAN conversion and replaced with the URI for the matching TGN place. If no match could be found, ULAN's string was preserved in the conversion.

The matching of instances of ulan:Nation to those tgn:Place instances that represent nations is based on direct comparison of their (non)preferred terms, as TGN also includes the nationality (e.g. "Dutch") as a non-preferred term of the nation (e.g. "The Netherlands"). Only tgn:AdministrativePlace's with one of the following placetypes are considered possible candidates: nation, state and nation division. Matches were related with the property vp:nation.

The matching of ulan:Roles to subconcepts of <aat:people by occupation> was performed by comparing different subsets of terms. ULAN contains three types of roles, signified in their terms: singular roles (e.g. architect), plural roles (a group of persons, e.g. architects), and roles that can denote either kind (e.g. animator(s). Role concepts in AAT always stand for a single person. However, the preferred term of an AAT role concept is in plural form, while one of its non-preferred terms contains the singular form. The E-Culture team only matched the first two kinds of ULAN roles to AAT roles. Matches between singular ULAN roles and AAT roles are represented with owl:sameAs. Matches between plural ULAN roles and AAT roles are represented with skos:related. After the automatic matching an additional 30 owl:sameAs and 17 skos:related were added manually.

To determine the accuracy of these heuristics an evaluation is necessary, which is outside the scope of the present study. However, the precision can be expected to be high because of the use of exact lexical matching only in combination with the topical overlap between the matched parts of the vocabularies. An explanation of the high percentage of unmatched nationalities is that this concept is much broader in meaning than that of nation (in TGN). For example, two categories of nationalities that cannot be matched represent tribes (e.g. Kikuyu) and areas (e.g. Newfoundlander, Québecois). The 28% unmatched roles is in part caused by not considering roles with the singular/plural combination (e.g. animator(s)). Another cause is some irregularies in the plural terms in AAT (e.g. bankers' instead of bankers).

5.8 Case Study Discussion

In this section we discuss the results of the case study. The use case showed that the E-Culture team made several choices in their conversion which are in conflict with their use cases and requirements. For example, the language tags were not converted, which conflicts with Requirement 4. In part this may be caused by incremental development: the original conversion was made in 2005 when it was not yet clear which direction the research on which the application is based would take. However, there are also cases where the need for a specific feature could have been foreseen. One such case is using blank nodes for Events. In both the WordNet case study as in this case study, we have not found any advantage from using a blank node to represent a vocabulary concept, while the clear disadvantage is that the concept cannot be used in annotations.

One conclusion from studying the simplification technique for n-ary relationships is that while information is lost, sometimes simplification may be necessary. We can see three arguments for simplification: (1) it is cheaper than buying a server with higher capacity; (2) search through the triple base will be faster; (3) graph selection algorithms over n-ary relations are more complex and thus slower. To summarize: complete conversions are not always usable. Another technique to decrease the size of the triple base is to only load the parts of the vocabularies that are necessary for the application. This may counter the need for simplification. An interesting topic for further research is to determine how these techniques interact. On the basis of such a study more accurate conclusions can be drawn on how to estimate the trade-off between completeness and size, and in turn on how an a priori estimate can be made of how much information can be included in a application-specific conversion. Such a study is dependent on factors such as the application's functionality and the efficiency of the infrastructure, and is beyond the current scope.

A specific circumstance in this case study was the absence of an existing (complete) conversion of the Getty vocabularies. If such a conversion had existed, the E-Culture project could have used that conversion as a basis for their own; this would save effort in interpreting the conceptual structure and syntactic conversion of the XML to RDF. However, the decisions that need to be taken on which information to discard and which to simplify would not be different than in the present study. Therefore we think a method as presented here would still be useful to guide the adaptation of the existing conversion for the purposes of E-Culture.

The choice for the reduction pattern by E-Culture leads to triples such as <ulan:Rembrandt,

ulan:rolePreferred, ulan:painter> (where painter is an instance of ulan:Role). However, a more usual interpretation of the relation between the concepts Rembrandt and Painter is the instance-class relationship (<ulan:Rembrandt, rdf:type, ulan:Painter>. This case can also be made for place types, and possibly nationalities. The decision of the E-Culture team not to convert roles and place types to subclasses (e.g. painter as a subclass of ulan:Person) is therefore not a common choice in the Semantic Web community. Not having classes for concepts such as painter, city, and mountain can also be limiting in the integration of different datasets and vocabulary (e.g. it is not possible to integrate two painter classes with owl:equivalentClass). However, no investigation of the repercussions of these decisions has been made, so no conclusion regarding the consequences is made in the present study.

A last observation regarding the case study is that E-Culture required few semantic (OWL) features to satisfy functional application requirements. This seems in line with the requirements that other Semantic Web applications pose (see the applications entered in the Semantic Web Challenge[13]). For most purposes it seems that RDFS plus some OWL features such as transitivity are sufficient.

5.9 Conclusion

In this chapter we focused on research question 2: "How can vocabularies be converted to an interoperable representation with given application-constraints?". To answer this research question we have investigated two issues: (1) how should the generic method be adapted to cater for application-specific conversions; and (2) how helpful is a structured method for application-specific conversions?

The answer to the first question is threefold. Firstly, some additions to Step 0 were needed. We added an application description, use cases and requirements to the analysis. These are needed to understand what needs the application has with respect to the vocabulary representation and content. Secondly, Step 3 is not a required step anymore. This step should only be performed if the application wishes to base its infrastructure on the standard (SKOS) metamodel. Thirdly, we introduced more guidelines to handle n-ary relationships. Guideline 25 recommends a specific representation pattern that allows lossless conversion of n-ary relations to RDF. We also developed the "reduction pattern" (guideline 26, page 89) to allow simplification of n-ary relations. This can help to reduce the size of the vocabulary and simplify algorithms that operate on the vocabulary. Besides these guidelines, the case study itself did not reveal the need for changes in the method. However, more case studies will be necessary to determine if these changes are enough to support a broad range of application-oriented conversion projects.

The answer to the second question is based on discrepancies observed between the requirements and the actual conversion. Firstly, several conversion choices were identified that were not consistent with the stated requirements and use cases of the application. For the Terms the language tag and display name should have been converted; for the AssociativeRelation the note, start/end year and place should have been converted; for the Events the start/end year; for Place-

[13] http://challenge.semanticweb.org/

Types the note and start/end year. For several tags the notes might also serve a useful function in browsing for learning. Secondly, blank nodes were used for Events while URIs can be needed in annotation. Based on these two categories of discrepancies we conclude that a structured method is indeed helpful in application-oriented conversions for applications similar to the E-Culture case. A caveat to these results is that foreseeing application development within a research project is difficult, as the exact research objectives may change. In these situations the chances of observing discrepancies are higher than in more traditional settings of application development.

Acknowledgements

The author would like to thank the E-Culture team for their cooperation, especially Jan Wielemaker.

6

Vocabularies in Alignment

In this chapter we return to research question 2: "How can vocabularies be converted to an interoperable representation with given application-constraints?" One such application is the alignment of several vocabularies. Alignment tools take as input the representations of two or more vocabularies, and produce an alignment between them. They are applications that place requirements on the representations just as search applications do. This chapter investigates what requirements they place and how existing vocabularies can meet them. Additionally, it investigates what features vocabularies have that the tools do not take advantage of in the process of finding alignments. The issues above are discussed in the context of a study that compares vocabulary alignment evaluation techniques. The techniques under study cover both application-dependent and independent evaluation. The target vocabulary of each alignment is AAT, the source vocabularies are SVCN, WordNet and ARIA.

This chapter is an extended and adapted version of (Hollink et al. 2008). The paper was published in the Proceedings of the Fifth European Semantic Web Conference, and was co-authored with Laura Hollink, Shenghui Wang, Antoine Isaac and Guus Schreiber. It was extended with material on vocabulary representations and tool requirements. Some material was reordered to better suit the goals of this thesis.

6.1 Introduction

The rise of the Semantic Web has led to a large number of different and heterogeneous ontologies. This has created a need to interconnect the ontologies. Tools have emerged that automatically produce correspondences between concepts in the ontologies (see e.g. (Shvaiko and Euzenat 2005) or (Kalfoglou and Schorlemmer 2003) for an overview). The set of correspondences found are together called an alignment (Euzenat and Shvaiko 2007). These correspondences can be used for various tasks, such as ontology merging, query answering, data translation, or navigation.[1] Tools produce alignments by reading the ontology representations and analyzing their conceptual overlap. In terms of the application-specific method developed in Chapter 5: the tools place requirements on the vocabulary representation. They can only process specific representations. We investigate which requirements the tools pose by looking at an exemplar of state-of-the-art alignment tools called Falcon-AO. We then analyze the impact on our conversion method. We also analyze the role of vocabulary features in the alignment process. Some features are present in the vocabularies but were not used by the tool (e.g. guide terms), other features were present but

[1] http://www.ontologymatching.org/

not made explicit in the vocabulary representations (e.g. qualifiers). We investigate which features could have helped in the alignment process.

Existing alignment evaluation strategies

After the tool produced an alignment its quality must be assessed before it can be employed in an application. Since 2004, the Ontology Alignment Evaluation Initiative (OAEI) organizes evaluation campaigns.[2] This has led to the development of mature alignment evaluation strategies. There are two main strategies: (1) assessing the alignment itself by judging the correctness of each correspondence and (2) comparing the alignment to a gold standard called a *reference alignment*. However, this type of evaluation does not guarantee that an application which uses the alignment will perform well. Evaluating the application that uses the alignment – commonly referred to as end-to-end evaluation – will provide a better indication of the value of the alignment (Van Hage et al. 2007).

However, end-to-end evaluation is time consuming, real-world applications that use alignments are as yet scarce, and associated data on user behaviour and user satisfaction is even more rare. A more feasible alternative is to take into account some characteristics of a particular usage scenario without doing a full-fledged end-to-end evaluation (Isaac et al. 2007). The OAEI more and more incorporates usage scenarios in the evaluation. For example, in the Anatomy track of OAEI 2007, tools were asked to return a high-precision and a high-recall alignment, supporting the respective usage scenarios of fully automatic alignment creation and suggestion of candidate alignments to an expert (Euzenat et al. 2007). Also, the number of tracks and test cases has increased every year (Euzenat et al. 2005, 2006, 2007), recognising the need for matching ontologies with different characteristics, such as size, richness and types of relations (e.g. rdfs:subClassOf, part-of), depth of the hierarchy, etc. Many existing datasets are not indexed with formal ontologies but with vocabularies such as thesauri and classification schemes. The need to align these is also recognised and is targeted in the OAEI Library and Food tracks.

Proposed evaluation strategies and case studies

We propose two new evaluation strategies. The first takes into account that some correspondences influence application performance more than other correspondences (the correctness of some correspondences is more important than that of other correspondences). The second takes into account that a correspondence may only be slightly wrong. It was not aligned to the best matching concept, but it is not entirely wrong either.

We perform three case studies in which we evaluate an alignment using all four strategies (the two existing strategies and our two proposed strategies). Each case study compares a different pair of vocabularies. Since the vocabularies are structured differently (e.g. more or less ontology-like), this allows us to discuss the effect of the characteristics of the vocabularies on the types of alignment errors that occur. In addition, we perform an end-to-end evaluation and compare this to the outcome of the four evaluation strategies in a qualitative way.

[2] http://oaei.ontologymatching.org/

We stress that it is not the purpose of this chapter to evaluate a particular alignment tool with respect to the quality of the alignments it produces.

Chapter outline

Section 6.2 describes the case vocabularies, while Section 6.3 discusses tool requirements. In Section 6.4 we discuss the proposed alignment evaluation strategies, together with related work concerning these strategies. The three subsequent sections deal with one case each. Section 6.8 describes a small end-to-end evaluation we did, after which Section 6.9 provides an interpretation of the results from all cases and the end-to-end evaluation. Section 6.10 provides a discussion of the role that vocabulary features and their representations play in the alignment process of Falcon-AO. The section also provides conclusions.

6.2 Case Study Vocabularies

Each of the three cases in this chapter consists of the alignment of one (source) vocabulary to the Art and Architecture Thesaurus (target). The three source vocabularies are SVCN, WordNet and ARIA. Ontology alignment tools can be applied to these vocabularies, although the looser semantics and differing representational choices may influence the quality of the alignment.

To limit the size of the study only the vocabulary parts that pertain to physical object types were used. The three source vocabularies differ in size, granularity, structure, and topical overlap with AAT, which allows us to investigate the role of vocabulary features in the evaluation strategies.

Vocabulary descriptions

The Getty Institute's AAT[3] is used by museums around the world for indexing works of art (Peterson 1994). Its concepts have English labels and are arranged in seven facets including Styles and Periods, Agents and Activities. In our study we concentrate only on the Objects facet, which contains 16,436 concepts ranging from types of chairs to buildings and measuring devices, arranged in a monohierarchy with a maximum depth of 17. Directly below facets are divisions called Hierarchy. In the Objects facet there are six, including Object Genres and Components. The broader/narrower hierarchy of this facet is ontologically clean. To prevent long lists of e.g. chair types, so-called *guide terms* are inserted to split on relevant characteristics (e.g. <chairs by form> and <chairs by location or context>). Concepts in AAT typically have many labels, e.g. "armchair", "armchairs", "chairs, arm", "chaises à bras". Labels can also include so-called qualifiers to distinguish them from homonymous concepts, e.g. credences (sideboards) and credences (tables). We added Dutch labels obtained from a translation of AAT[4] . This was required to align AAT to SVCN, because SVCN is available in Dutch only.

SVCN is a thesaurus developed and used by several Dutch ethnographic museums.[5] It has

[3] See http://www.getty.edu/research/conducting_research/vocabularies/aat/. The AAT is a licensed resource.

[4] http://www.aat-ned.nl/

[5] http://www.svcn.nl/

four facets, of which the Object facet has 4,200 concepts (making it four times smaller than the Object facet of AAT). SVCN's Object facet was originally created by selecting AAT concepts and translating the labels to Dutch. However, over time intermediate and leaf concepts have been inserted and removed, resulting in a hierarchy with a maximum depth of 13. The broader/narrower hierarchy is well-designed.

WordNet is a freely available thesaurus of the English language developed by Princeton.[6] It has three top concepts: Physical entity, Abstraction and Thing. We only used the hierarchy below Physical entity ◁– Physical object, which contains 31,547 concepts. Each concept has multiple synonymous terms. The main hierarchy is formed by the polyhierarchic hyponym relation which contains more ontological errors than AAT's hierarchy.[7] The topical overlap with AAT is reasonable, depending on the part of the hierarchy. WordNet's Physical object hierarchy covers, for example, also biological concepts such as people, animals and plants while the Object facet of AAT does not. Other parts of WordNet are very similar to AAT. For example, the hierarchy from Furniture down to Chesterfield sofas is almost identical to that in AAT. The maximum depth of the Physical entity hierarchy is the same as AAT' Object Facet: 17 nodes.

ARIA is a set of vocabularies developed by the Dutch Rijksmuseum for a website that showcases some 750 masterpieces of the collection.[8] We use one of the constituent vocabularies that is intended to describe the object type. It contains 491 concepts. There are 26 top concepts such as Altarpieces, Household scenes and Clothing, only half of which have subconcepts. Each concept has one term in Dutch and one in English. Its hierarchy is at most 3 concepts deep and is arranged in a polyhierarchy; e.g. Retables is subordinate to Altarpieces and Religious paraphernalia. The broader/narrower relation used in ARIA can in many cases not be interpreted as rdfs:subClassOf. For example, Costumes and textiles has a grandchild Portable altars. ARIA is the smallest and most weakly structured of the three source vocabularies.

Vocabulary representations

As a basis for our work the representations of AAT, SVCN and ARIA provided by E-Culture (Schreiber et al. 2006), and WordNet's representation by the W3C (Semantic Web Best Practices and Deployment Working Group 2006c) were used. The E-Culture representation of AAT has been discussed extensively in Chapter 5, we mention the more relevant points here. In step 1a E-Culture did not convert editorial information. Additionally, information was lost because attributes of n-ary relations were dropped (including start/endyear for events and associative relations). In step 1b an explication the E-Culture project did not make is the type of hierarchical relation (subclass in AAT, place-area in TGN). In step 2a the hierarchical relation was defined as a transitive property. In addition to what is described in Chapter 5, E-Culture's AAT now also has Dutch labels, obtained

[6] http://wordnet.princeton.edu/

[7] Two examples of hierarchical chains below Physical object with errors are: (1) Location ◁– North; and (2) Whole ◁– Natural object ◁– Plant part ◁– Lobe. The first confuses objects with the space in which they occur. The second confuses wholes with parts of wholes. In general the WordNet hierarchy is rife with roles such as Je ne sais quoi, Somewhere and Trivia.

[8] http://www.rijksmuseum.nl/aria/

from a translation of AAT.[9] This is crucial for the alignment of SVCN to AAT, as SVCN is only available in Dutch. The integration of the Dutch AAT is an interpretation that can be situated in step 2b. Also not discussed in Chapter 5 is that the qualifiers could have been represented explicitly (in step 2b). The guide terms do have an explicit representation (class aat:GuideTerm).

The representation of WordNet is discussed extensively in Chapter 4. In this chapter the Basic version of that conversion is used, because its (Full) three-leveled structure would complicate the comparison with SVCN and ARIA in the case studies.

SVCN and ARIA both use the SKOS schema in the representation. The SVCN representation is fully SKOS compliant and consists of skos:Concepts with a skos:prefLabel and zero or more skos:altLabels, related through skos:broader statements. Although the original SVCN representation separates guide terms from normal concepts, the E-Culture conversion does not reflect this difference. The ARIA representation uses skos:narrower to relate its concepts (represented with the class aria:Term; this class is not explicitly related to skos:Concept). Furthermore, the concepts are labelled with rdfs:label instead of with the SKOS label properties. We have no further knowledge on the conversion process and the choices made.

6.3 Tool Requirements and Vocabulary Interpretations

In the OAEI contests — that arguably represent the state-of-the-art — no tools participate that can handle SKOS or SKOS-like vocabularies.[10] To investigate this issue further we focus on one concrete tool. We selected Falcon-AO (Hu and Qu 2007), because it is consistently among the best performing tools. We employed it as an off-the-shelf tool. Like most tools Falcon-AO only produces equivalence mappings.[11]

Falcon-AO places specific requirements on the vocabulary representation. Concepts need to be represented with owl:Class, a concept hierarchy needs to be represented with the rdfs:subClassOf relation, and labels and comments with rdfs:label and rdfs:comment. A specific interpretation of the vocabulary into the OWL metamodel is necessary. This situation is not covered by our application-specific method, as that method assumes the application developer is in control of the vocabulary representation. An interpretation in the style of the generic method's Step 2b is necessary. The vocabulary interpretations appropriate for each case study vocabulary are given in Table 6.1. However, Falcon-AO cannot handle metamodelling interpretations such as <svcn:Subject, rdfs:subClassOf, owl:Class>. Therefore, a useful interpretation entails creating a new representation (a new file) where original properties and classes are replaced with those accepted by Falcon-AO: owl:Class, rdfs:subClassOf, rdfs:label, rdfs:comment and remaining non-hierarchical object properties.[12]

Because ARIA and SVCN have no concept definitions, no rdfs:comments could be generated

[9] `http://www.aat-ned.nl/`

[10] Personal communication with Food and Library track organizers, see also `http://oaei.ontologymatching.org/2008/skos2owl.html`.

[11] The exceptions are SCARLET and TaxoMap, of which the performance is ambiguous (Euzenat et al. 2007).

[12] Besides the interpretation, all `xml:lang` tags had to be removed from the newly produced file, because for some unknown reason Falcon-AO produced no mappings when they were present.

	owl:Class	rdfs:subClassOf	rdfs:label
SVCN	svcn:Subject	skos:broader	skos:prefLabel, skos:altLabel
WordNet	wn:Synset	wn:hyponymOf	wn:senseLabel
ARIA	rijks:Term	skos:narrower	rdfs:label
AAT	aat:Concept, aat:HierarchyName, aat:GuideTerm	vp:parentPreferred	aat:labelPreferred, aat:labelNonPreferred

Table 6.1 Interpretation of vocabulary schemas as OWL ontology schemas. Each row lists RDF properties and concepts of one vocabulary that are sources for the interpretation. The concepts/properties in a column are mapped to the property/concept mentioned in the column's table head.

for the OWL interpretation. WordNet does have definitions (gloss) but we chose not to include them in the interpretation to keep the cases comparable. Although they probably help in alignment, here we focus on the hierarchy and concept terms only.

6.4 Alignment Evaluation Strategies

Ideas have been put forward to find feasible alternatives to end-to-end evaluation. In this section we discuss two strategies that each take into account one characteristic of an application that uses an alignment. The application scenario that we focus on is a query reformulation scenario, in which users pose a query in terms of one vocabulary in order to retrieve items that are annotated with concepts from another vocabulary. We assume that there is a partial alignment between the two vocabularies, which is a realistic assumption given the state-of-the art of matching tools.

Strategy 1: Weighting by importance

If an alignment is large, evaluating all correspondences can be a time consuming process. A more cost-effective option is to evaluate a random sample of all correspondences, and generalize the results to get an estimate of the quality of the alignment as a whole.

An alternative to taking a random sample is purposefully selecting a sample. Van Hage et al. (2007) note that in a particular application some correspondences affect the result (and thus user satisfaction) more than others. An end-to-end evaluation can take this into account, but the evaluation of individual correspondences as it is currently performed does not. Evaluating the most important correspondences would better approximate the outcome of an end-to-end evaluation. The notion of "importance" can mean different things in different application contexts. Here we propose to use the estimated frequency of use of each correspondence as a weighting factor in the computation of performance measures. To this end, we divide all correspondences into strata based on their frequency of use. The overall performance of the correspondences can then be calculated based on the performance of each stratum. Van Hage et al. (2007) shows how stratified samples can be aggregated:

$$\hat{P} = \sum_{h=1}^{L} \frac{N_h}{N} \hat{P}_h \tag{6.1}$$

where $\hat{P}$ is the estimated performance of the entire population, $\hat{P}_h$ is the estimated performance of stratum h, $\frac{N_h}{N}$ is a weighting factor based on the relative size of the stratum, where N_h is the size of the stratum, and N is the total population size.

Instead of weighting the strata based on their size, we propose to weight them based on their expected frequency of use:

$$\hat{P} = \sum_{h=1}^{L} \frac{\sum_{a \epsilon H} \text{freq}(a)}{\sum_{a \epsilon A} \text{freq}(a)} \hat{P}_h \tag{6.2}$$

where freq(a) is the frequency of use of correspondence a, H is the total set of correspondences in stratum h, and A is the total set of correspondences in the alignment. In our experiments we shall use two strata: frequent and infrequent correspondences, but the approach also works for more strata.

Selecting the most frequently used correspondences for evaluation is beneficial in two situations. First, if there is a difference in quality between the frequently used correspondences and the infrequently used correspondences, the frequency-weighted precision will give a more reliable estimate of the performance of the application using the alignment. Second, in a semi-automatic matching process in which suggested correspondences are manually checked and corrected by an expert, the frequency provides an ordering in which to check. This kind of scenario is targeted in the Anatomy track of OAEI 2007 (Euzenat et al. 2007) by asking participants to generate a high-recall alignment. Ehrig and Euzenat (2005) consider the semi-automatic matching process by measuring the quality of an alignment by the effort it will take an expert to correct it. We argue that correction of a number of frequently used correspondences will positively affect the performance of the application more than correction of the same number of randomly selected correspondences.

Implementation of strategy 1

Ideally, query logs can be used to determine the frequencies. However, logs are not always available. We propose a way to estimate frequency. We assume here that each concept in source vocabulary X has an equal probability of being selected as a query by a user. For each query concept x, we determine the closest concept x' in X that has a correspondence to a concept y in vocabulary Y (the target vocabulary with which items are annotated). Closeness is determined by counting the number of steps in the (broader/narrower) hierarchy between x and x'. If a query concept x does not itself have a correspondence to Y, the correspondence of x' to vocabulary Y is used to answer the query, thus adding to the frequency count of correspondence $\{x', y\}$. Our estimation is biased, because in practice some query concepts are more often used than others.

Strategy 2: Graded incorrectness

Comparing an alignment A to a reference alignment R gives precision as well as recall scores. Precision is the proportion of correspondences in A that are also found in reference alignment R, while recall is the proportion of the reference alignment R that is covered by A.

Incorrect correspondences negatively affect the performance of an application. However, this effect varies depending on how incorrect the correspondence is. Performance of an application will drop steeply if a correspondence links two completely unrelated concepts, while it may drop only slightly if a correspondence links two closely related concepts. In other words, the correctness of a link can be graded between 0 and 1.

The idea of a more nuanced precision and recall measure has been proposed before. Ehrig and Euzenat (2005) propose to include a proximity measure in the evaluation of alignments. They suggest to use the effort needed by an expert to correct mistakes in an alignment as a measure of the quality of an alignment. In the same paper, they propose to use the proximity between two concepts as a quality measure. A very simple distance measure is used as an example.

Implementation of strategy 2

We investigate the use of a semantic distance measure to capture this gradation. More specifically, we use semantic distance to represent the distance between a correspondence in A and a correspondence in a reference alignment R. This allows us to distinguish between correspondences that cause incorrect results, and correspondences that are misaligned but still produce an acceptable result in the application.

We implement this idea by using the semantic distance measure of Leacock and Chodorow (1998). This measure scored well in a comparative study of five semantic distance measures by Budanitsky and Hirst (2001), and has the pragmatic advantage that it does not need an external corpus. The measure by Leacock and Chodorow, sim_{LC}, actually measures semantic proximity:

$$sim_{LC} = -\log \frac{len_{(c_1,c_2)}}{2D}$$

where $len_{(c1,c1)}$ is the shortest path between concepts $c1$ and $c2$, which is defined as the number of nodes encountered when following the (broader/narrower) hierarchy from $c1$ to $c2$. D is the maximum depth of the hierarchy.

In our case studies, we compare each correspondence $\{x, y\}$ in A to a correspondence $\{x, y'\}$ in a reference alignment R. We use the semantic distance between y and y' as a relevance measure for the correspondence $\{x, y\}$. To calculate precision and recall, we normalize the semantic distance to a scale from 0 to 1.

A side effect of using a semantic distance measure is that the assessments are no longer dichotomous but are measured on an interval level. Common recall and precision measures are not suited for this scale. Therefore, we use *Generalised Precision* and *Generalized Recall* as proposed by Kekäläinen and Järvelin (2002):

$$gP = \sum_{a \epsilon A} \frac{r(a)}{|A|} \qquad gR = \frac{\sum_{a \epsilon A} r(a)}{\sum_{a \epsilon R} r(a)} \tag{6.3}$$

where $r(a)$ is the relevance of correspondence a, A is the set of all correspondences found by the matching tool, and R is the set of all correspondences in the reference alignment. A similar notion of this measure was later described by Euzenat (2007). The latter measure is more general since it is based on an overlap function between two alignments instead of distances between individual correspondences.

In the following sections, we discuss the case studies.

6.5 Case 1: Alignment Between SVCN and AAT

We study the alignment Falcon-AO made between SVCN and AAT. We first discuss the two strategies that evaluate individual correspondences (the normal strategy and our alternative based on frequency). Then the strategies that evaluate against a reference alignment are discussed (the normal strategy and our alternative based on semantic distance).

Evaluating individual correspondences

Falcon produced 2,748 correspondences between SVCN and AAT. We estimated the frequency of use of each correspondence, as described in Section 6.4. Figure 6.1 displays cumulative percentages of these frequencies against cumulative percentages of the number of correspondences; all correspondences were ordered according to their frequency and displayed so that infrequent correspondences appear on the left side of the figure and the most frequent correspondences appear on the right. If each correspondence was used equally frequently, the graph would show a straight line from the origin to the top-right corner. In the case of the SVCN-AAT alignment, the graph does not deviate much from this straight line (Figure 6.1).

All correspondences were divided over two strata: frequently used and infrequently used correspondences. The size of the frequent stratum was set to 80 (3% of all correspondences), which are responsible for 20% of the use in the application scenario (Figure 6.1). The choice for a size of 80 is pragmatic: it is a low number of correspondences that can be evaluated but still reflects a large frequency percentage. We evaluated all correspondences in the frequent stratum and a random sample of 200 from the infrequent stratum. Table 6.2 shows that the precision of the two strata differs, but not significantly so (0.93 and 0.89). We then weighted the outcomes of these evaluations in two ways: (1) according to the sizes of the strata (80 and 2,668) as in Equation 6.1 and (2) according to the frequency of use of the correspondences in the strata as in Equation 6.2. Both weighting schemes gave a precision of 0.89 (see Table 6.2).

Since in this use case the size of the population of all correspondences is large compared to the sample sizes, we used the binomial distribution to approximate the margins of error (shown in Table 6.2). The margin of error of a binomial distribution is given by:

$$\text{Margin of error} = 1.96\sqrt{\frac{p(1-p)}{n}} \tag{6.4}$$

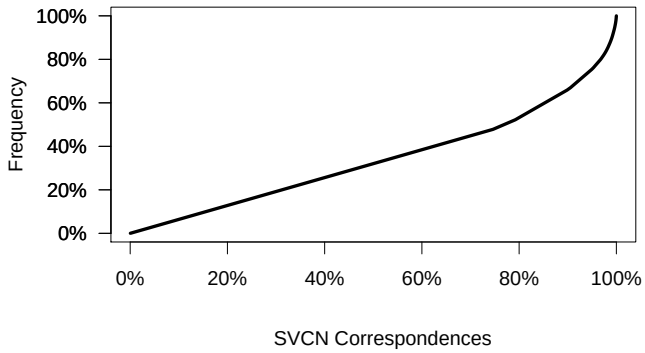

Figure 6.1 Cumulative percentage of estimated use of SVCN-AAT correspondences in the application scenario. The total number of correspondences is 2,748.

One reason for taking into account the frequency of use of correspondences is that it gives an order in which to manually check and correct the correspondences. We corrected all 80 correspondences in the frequent stratum and then recalculated the precision of the alignment, weighted by frequency of use. This gave a precision of 0.91, which is not a significant increase. After manual correction of a random sample of 80 correspondences the precision rises to 0.93, which is higher but again not a significant increase. A possible reason for the finding that random correction gives a better precision than correction of frequent correspondences, is the fact that there were more wrong correspondences in the random sample. Another factor is that the contribution to the total frequency of correspondences in the two strata is similar.

Evaluation Type	**Precision**	**Recall**
Random sample of infrequent stratum	0.89±0.04	
Frequent stratum	0.93	
Weighted based on stratum size	0.89±0.03	
Weighted based on frequency of use	0.89±0.03	
After correction of frequent stratum	0.91±0.03	
After random correction	0.93±0.03	
Comparison to a reference alignment	0.84±0.07	0.80±0.08
Semantic distance to a reference alignment	0.90±0.06	0.86±0.07

Table 6.2 Evaluation of the alignment between SVCN and AAT.

Comparison to a reference alignment

Reference alignment evaluation has the advantage that both precision and recall can be determined, but it is more costly because two vocabularies have to be aligned completely. Instead of aligning all concepts, we took a random sample of 100 concepts from SVCN and aligned those to AAT.

Based on this partial reference alignment, Falcon's alignment has a precision of 0.84 and a recall of 0.80 (Table 6.2). As an alternative, we employ a semantic distance measure to compare the correspondences to the reference alignment; each correspondence $\{x,y'\}$ in the reference alignment is compared to a correspondence $\{x, y\}$ delivered by Falcon. We use the sim_{LC} measure between y and y', which results in a scaled value (0-1). Generalized precision and recall can then be calculated over these values (see Table 6.2). In the case of SVCN, the semantic distance based precision and recall are higher that the 'traditional' precision and recall, but the differences lie within the margins of error.

6.6 Case 2: Alignment Between WordNet and AAT

Evaluating individual correspondences

Falcon produced 4,101 correspondences between WordNet and AAT. Applying our frequency estimation gives a distribution depicted in Figure 6.2. In this case the contribution of the most frequent correspondences is much greater; the top 20% of correspondences is already responsible for 70% of expected usage (reminiscent of Zipf's law).

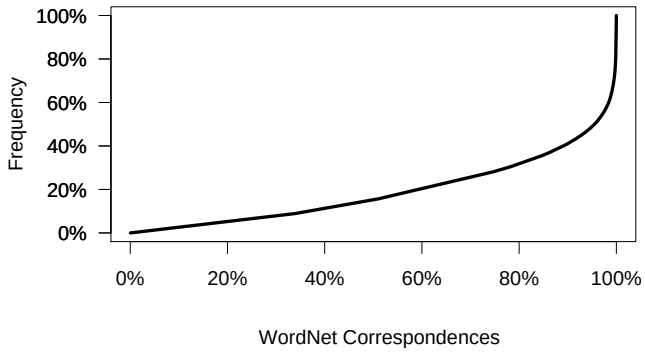

Figure 6.2 Cumulative percentage of estimated use of WN-AAT correspondences in the application scenario. The total number of correspondences is 4, 101.

We performed the same evaluation procedures as for the SVCN case, except that the size of the frequent stratum was set to 30 (0.7% of all correspondences). This is possible because here the contribution of the top correspondences is greater; the top 30 is responsible for 33% of total frequency. This reduction saves us a considerable evaluation effort. The results of the different evaluation strategies are presented in Table 6.3. In the case of WordNet, weighting based on stratum size gives a slightly higher precision than weighting based on frequency (0.71 and 0.68, respectively).

Manual correction of all 30 frequent correspondences gives a higher precision than correcting

Evaluation Type	Precision	Recall
Random sample of infrequent stratum	0.72±0.06	
Frequent stratum	0.60	
Weighted based on stratum size	0.71±0.05	
Weighted based on frequency of use	0.68±0.04	
After correction of frequent stratum	0.81±0.04	
After random correction	0.72±0.04	
Comparison to a reference alignment	0.62±0.10	0.45±0.10
Semantic distance to a reference alignment	0.64±0.09	0.47±0.10

Table 6.3 Evaluation of the alignment between WordNet and AAT.

30 randomly selected correspondences from the complete set of correspondences (0.81 and 0.72, respectively, calculated by frequency-based weighting). This shows that in the WordNet case, it is sensible to prioritize correction of the most frequent correspondences.

Comparison to a reference alignment

We performed a sample reference alignment evaluation in the same manner as for the SVCN case (n=100). The results are much lower than those for SVCN, as can be seen in Table 6.3. The margins of error are somewhat higher because the sample size is smaller. The effect of applying semantic distance is smaller than the effect we saw for SVCN.

6.7 Case 3: Alignment Between ARIA and AAT

Evaluating individual correspondences

Falcon produced 278 correspondences between ARIA and AAT. Figure 6.3 shows the results of applying our frequency estimation. In this case the contribution of the most frequent correspondences is large; the top 20% of correspondences is responsible for 50% of expected usage.

Again we opted for a size of 30 for the frequent stratum (6% of all correspondences), which are responsible for 42% of the use in the application scenario. In this case, weighting according to the size of the stratum gave a precision of 0.74, while weighting according to the frequency of use gave a precision of 0.70.

Since the sample size is large compared to the size of the population of all correspondences in this case, we cannot approximate the margin of error with a binomial distribution. Instead, we used the following equation to compute the margin of error for a hypergeometric distribution (den Brink and Koele 2002):

$$\text{Margin of error} = 1.96\frac{N-n}{N-1}m(1-\frac{m}{n}) \tag{6.5}$$

where N is the size of the population, n is the size of the sample, and m is the number of correct correspondence found in the sample.

Manual correction of the most frequent stratum gives a precision of 0.85, which is again higher than random correction (0.74).

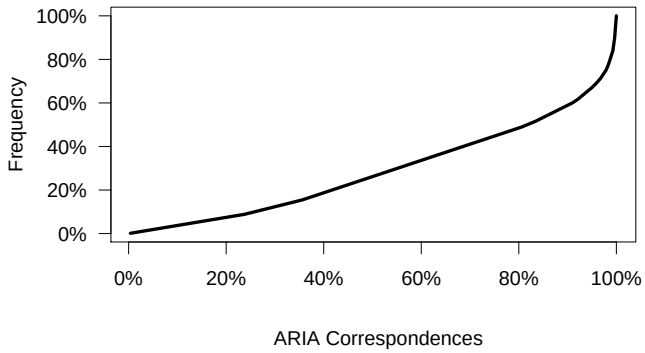

Figure 6.3 Cumulative percentage of estimated use of ARIA-AAT correspondences in the application scenario. The total number of correspondences is 278.

Evaluation Type	Precision	Recall
Random sample of infrequent stratum	0.75±0.03	
Frequent stratum	0.63	
Weighted based on stratum size	0.74±0.03	
Weighted based on frequency of use	0.70±0.03	
After correction of frequent stratum	0.85±0.02	
After random correction	0.74±0.03	
Comparison to a reference alignment	0.66±0.09	0.63±0.09
Semantic distance to a reference alignment	0.80±0.08	0.76±0.08

Table 6.4 Evaluation of the alignment between Aria and AAT.

Comparison to a reference alignment

We performed a sample reference alignment evaluation in the same manner as for SVCN and WordNet (n=100). The recall and precision measures as shown in Table 6.4 are in between those for SVCN (highest) and WordNet (lowest). The effect of applying semantic distance is considerable.

6.8 End-to-end Evaluation

In this section we present an end-to-end evaluation performed using the three alignments from the case studies. The application scenario that we focus on is a query reformulation task for information retrieval: a user query for concept $x \in$ vocabulary X is transformed into a concept $y \in$ vocabulary Y. We queried a dataset of 15,723 art objects indexed with AAT provided by E-Culture. Objects annotated with concepts from Y are returned to the user and the relevance of these objects to the query x is rated. We used 20 randomly selected query concepts from

each source vocabulary[13] and evaluated two different strategies of reformulation for cases where a query $x \in X$ has no direct correspondence to Y: (1) find a concept x' in the hierarchy below x that has a correspondence to a concept $y \in Y$ (strategy "downward"); or (2) find a concept x' above x with a correspondence to a concept $y \in Y$ (strategy "upward").

The effectiveness of the reformulation was evaluated by assessing the relevance of objects annotated with concept y (or subconcepts of y) on on a six-point scale ranging from "very relevant" to "not relevant at all". Generalized precision and recall were calculated from these ordinal assessments (see Table 6.5). For comparison we also calculated precision and recall based on dichotomous (0/1) assessments by rescaling the ordinal values 0-2 to 0 and 3-5 to 1. Recall was calculated based on a recall pool.[14]

		Precision		Recall	
Vocabulary	**Strategy**	**Binary Scale**	**6-point Scale**	**Binary Scale**	**6-point Scale**
ARIA	upward	0.27	0.37	0.83	0.88
	downward	0.70	0.66	0.49	0.43
SVCN	upward	0.46	0.48	0.93	0.96
	downward	0.79	0.76	0.42	0.36
WordNet	upward	0.46	0.48	0.80	0.81
	downward	0.63	0.67	0.18	0.18

Table 6.5 Precision and Recall of end-to-end evaluation for a six-point scale and a binary scale. Results are shown for two different query reformulation strategies.

We stress that the precision and recall figures presented in Table 6.5 refer to *relevancy* of the returned objects, instead of *correctness* of correspondences. This means that it is not possible to directly compare the results from reference alignment evaluations with the results from end-to-end evaluation. This is a general methodological difficulty when comparing evaluation strategies, not only for the scenario presented in this paper.

6.9 Case Study Results

The three case studies have illustrated differences between the evaluation strategies and between the aligned vocabularies. The results show that the different evaluation strategies stress different properties of an alignment. First we discuss how the scores varied between the two pairs of evaluation strategies. Secondly, we discuss how the different vocabularies influenced the score.

The results suggest that for both WordNet and Aria, an evaluation that takes into account the frequency of use will result in a more realistic estimation of application performance than an evaluation that does not take this into account. For SVCN, the frequency based weighting did not make a difference, nor did the correction of frequent correspondences. This lack of effect

[13] We excluded concepts that were too general such as Physical object

[14] A recall pool consists of the union of all objects returned by any of the systems; objects not in the pool are considered irrelevant. This strategy is regularly used in evaluation of text retrieval systems where evaluating all documents in the collection is practically infeasible.

can be explained from two observations: (a) the precision of SVCN is already high and therefore correction will have less effect; and (b) the frequency distribution of SVCN correspondences is relatively gradual, so that the most frequent stratum has less influence than in the WordNet case.

The variations in frequency of use of correspondences are most pronounced in WordNet. This can be explained from the fact that the proportion of WordNet concepts that has a correspondence to AAT is relatively small (13%). Queries for concepts without a correspondence to AAT will be reformulated to related concepts that do have a correspondence to AAT. This causes concepts that are central nodes in the hierarchy to get potentially high frequency counts. In line with this finding, correcting the most frequent correspondences gives a significantly higher precision than correcting randomly selected correspondences.

We conclude that in cases where only a small portion of a vocabulary can be aligned to a target vocabulary, for example when topical overlap is small, an estimation of the most frequently used correspondences gives a realistic image of application performance. In these cases it will be cost-effective to manually correct (only) the frequently used correspondences.

When comparing the "traditional" precision and recall scores to those based on semantic distance, we see a clear difference between the two measures in the results of ARIA (an average difference of 7%). A difference is notable for SVCN (average of 4%) although less clear, and almost no difference is visible for WordNet (1%). This is mirrored in the end-to-end evaluation, where the differences between a binary scale and a 6-point scale show the same trend: large differences for ARIA (an average of 13%), small differences for SVCN (6%) and no differences for WordNet (2%). An explanation is that ARIA returns many results that are only moderately relevant, while WordNet returns mainly highly relevant results. For applications in which users expect to see also moderately relevant results, an evaluation based on semantic distance better reflects the quality of the alignment.

The alignment of SVCN to AAT scores higher than those from WordNet and ARIA to AAT in all evaluations including the end-to-end evaluation. One exception is the result for recall in the downward strategy of the end-to-end evaluation; ARIA performs slightly better. The high scores of SVCN can be explained from its reasonably clean hierarchy and high similarity to the target vocabulary AAT. Evaluation of individual correspondences gives SVCN a precision of around 0.90 for all different weighting schemes. The different precision numbers lie around 0.70 for ARIA and WordNet. This suggests that a weakly structured, small vocabulary such as ARIA can be aligned with approximately the same precision as a large, richly structured vocabulary such as WordNet.

For ARIA, correcting frequent correspondences showed a clear improvement of the results. This is not entirely expected, since ARIA has relatively many correspondences and ARIA's frequency distribution is less pronounced. The effect is partly due to the fact that the precision of the frequent stratum is lower than the precision of the infrequent stratum.

The comparison against a reference alignment produces a clear ordering of the three alignments, in both precision and recall: SVCN is best, followed by ARIA and finally WordNet. The alignment of WordNet has a low recall (0.45 and 0.47) compared to the other vocabularies. A possible cause is the size of WordNet and the relatively low number of correspondences that was found. Although we have no clear explanation, the effect is reflected in the end-to-end evaluation; WordNet has a remarkably low recall when using the downward strategy.

The use of three vocabularies with varying degree of structure and topical overlap with AAT lead us to two observations. First, we conclude that two vocabularies that show a stark resemblance to each other with respect to structure and topical overlap, can be aligned with such high precision and recall that manual creation or correction of the alignment has little added value. This holds in particular for vocabularies that share a common source, such as SVCN and AAT. Second, a vocabulary with a weak structure is no impediment for a high-quality alignment. The ontological flaws of ARIA did not result in a worse alignment than the reasonable structure of WordNet.

6.10 Discussion and Conclusions

Vocabulary representations play a central role in alignment applications. We already observed in Section 6.3 that SKOS-like representations of vocabularies cannot be handled by tools such as Falcon-AO. An interpretation of the vocabulary is necessary. Step 2b of our method suppports the creation of an interpretation.

In this discussion we investigate how representing specific vocabulary features can help in alignment. Firstly, we discuss which vocabulary features are ignored by Falcon-AO but can actually help in the alignment process. Secondly, we discuss which vocabulary features were not made explicit in the case study vocabularies, and if they could have helped in alignment. Thirdly, we discuss how two particular features might be represented.

Vocabulary features not used by Falcon-AO

Falcon-AO uses owl:Class, rdfs:subClassOf, object properties (for graph-based matching), rdfs:label and rdfs:comment (for text-based matching). Falcon-AO does not use any other features present in vocabularies. This means Falcon-AO does not use e.g. meta-information present in the case vocabularies such as qualifiers and guide terms. The distinction between preferred and non-preferred labels is also not taken into account (non-preferred labels are not always exact synonyms but e.g. broader in meaning). Other vocabularies (e.g. TGN) include a distinction between preferred and non-preferred hierarchical relations which Falcon-AO is not aware of. Falcon-AO does not have knowledge of the specific meaning of guide terms (class aat:GuideTerm). However, the alignment it produced did contain mappings between SVCN guide terms and AAT guide terms. Almost all were correct (i.e precision was high; we have not investigated recall). A probable reason is that these matches were found because of literal matches. Lastly, Falcon-AO does not detect and handle n-ary relationships (either represented with the W3C pattern or with reification) as present in e.g. AAT.

Vocabulary features not made explicit in vocabulary representations

Besides features that Falcon-AO is not aware of, there are also features which were simply not explicitly represented in the vocabulary representations of our cases. (The vocabulary has a particular feature, but it is not reflected in the representation.) These features include editorial information and arguments of n-ary relations (the AAT Term relation and Associative relation). For the former

it is not very likely that it may help in alignment. For the latter this is more likely, but not in the cases studied here. Firstly, the source vocabularies do not have n-ary relations themselves, so that they cannot be compared to the n-ary relations of AAT. Secondly, most of the arguments of AAT's n-ary relations are not useful in alignment, e.g. the vernacular flag of the Term-relation. Possible exceptions are the notes and start/end years attached to Associative relations (see Table 5.2 on page 90).

The representations also do not separate guide terms into their components (a concept and an attribute of the concept). In some cases it can be beneficial to be aware of these components so that e.g. "cars by motor type" can be matched to "automobiles by engine".

A vocabulary feature that does affect alignment in our cases is the qualifier. For example, Falcon-AO did not match aria:table to aat:tables (support furniture), but instead (wrongly) to aat:sound table and aat:table boards. It is probable that the qualifier prevents a lexical match to be found. One approach might be to exclude qualifiers before matching. This is not a solution, as aat:tables (documents) will then also match. The qualifiers and the parents of the concepts might be used to disambiguate. For example, aria:table has aria:moveable furnishings as parent. The parent might be used as supporting evidence for choosing aat:tables (support furniture). This can be done by matching aria:moveable furnishings to the qualifier of aat:tables (support furniture), or by matching it to a parent of aat:tables (support furniture) such as aat:furnishings (artifacts). In any case, knowledge of the role of qualifiers is required to devise an improved alignment strategy.

Representing guide terms and qualifiers

Guide terms and concepts with qualifiers are examples of concepts that are composed of other concepts. There are several ways to deal with compound concepts. A first option is to ignore the distinction and include them as normal (SKOS) concepts (as done in SVCN, see Section 6.2). As we argue above this means that valuable information for alignment tools is not made explicit. In Section 5.4 we "flagged" guide terms as special cases of normal concepts by creating a subclass of skos:Concept called aat:GuideTerm. This solution still does not make explicit which part of the composition is the concept and which the guide term. A benefit is that it can make application-specific treatment of guide terms possible (e.g. suppress them in autocomplete fields) for a minimum cost in triples. In Section 3.6 we flagged concepts with qualifiers with the class mesh:CompoundConcept, but we also created its consituants (mesh:Concept and mesh:Qualifier) and linked those to mesh:CompoundConcept. A similar solution was presented in Section 4.9: each synset is composed of several word senses, represented as a skos:Concept and a skosxl:Label, respectively. SKOS proposes a different solution than ours: for representing guide terms it introduces the class skos:Collection which is *not* a subclass of skos:Concept. Guide terms thus do not appear in the skos:broader/narrower hierarchy as in the previous representations.

Although we argued before that alignment applications can benefit from an explict representation of compound concepts, this does not mean that such a representation is always preferable. Current alignment applications do not understand the extensions, which may result in worse alignments, e.g. because the tool aligns word senses to AAT concepts instead of synsets to AAT concepts. A solution to this problem is to postpone this type of explication until step 2b. The ex-

plication is realized as an "add-on" to the basic representation and can be loaded when necessary.

The choice whether to represent a compound concept as (a subclass of) "standard" concepts has repercussions for particular use cases. For example, if guide terms are standard concepts, a tree browser for selecting concepts can be generated easily by querying the broader/narrower hierarchy. This query becomes more difficult if the SKOS representation is used where guide terms are skos:Collections which are not in the skos:broader/narrower hierarchy. Another example favours solutions where guide terms are *not* standard concepts. Consider an annotation application which allows users to add concepts through an autocomplete field. The autocomplete field should hide the guide terms because these are not appropriate annotation concepts.

In sum, the representation of compound concepts such as guide terms and qualified concepts presents a trade-off with respect to size and supported use cases. Depending on the application a particular option may be preferable. How to deal with this trade-off is a subject for future research.

Conclusions

In this chapter we returned to research question 2: "How can vocabularies be converted to an interoperable representation with given application-constraints?" Here we investigated the constraints imposed by applications that produce alignments, where we took Falcon-AO as a representative. We found that alignment tools require an interpretation of the vocabulary because they cannot handle SKOS-like representations. Now that the SKOS standard is becoming more accepted, and more and more vocabularies are made available in SKOS, we would expect that alignment tools support SKOS. This is not the case, and we recommend that tool builders start to do so. In the meantime, an interpretation of the vocabulary suitable for alignment tools can be realized in step 2b of our method. The interpretation must not rely on metamodelling through rdfs:subClassOf and rdfs:subPropertyOf, as these tools do not understand these constructs. We also found that Falcon-AO does not take advantage of specific vocabulary features which can help in creating better alignments, such as qualifiers and guide terms. We have discussed several ways to represent these features and noted that each representation provides a different level of support for particular use cases. The choice for a particular representation is not clear-cut.

The comparison of evaluation strategies shows that in some cases a weighted evaluation strategy gives a better impression of the quality of an alignment than an unweighted strategy. It also helps to prioritize manual checking of correspondences. Evaluation based on semantic distance for the task of document retrieval is useful when not only highly relevant results are desired.

Acknowledgements

The authors would like to thank Willem van Hage, Alistair Vardy, Tom Moons, Niels Schreiber and the members of the E-Culture project for their help and advice. We also thank Wei Hu for information regarding Falcon-AO. The authors were supported by the NWO projects: CHIME, CHOICE and STITCH and the TELplus project.

7

Case Study: Representing a Metadata Element Set for Visual Art

This chapter focuses on research question 3: "How can metadata schemas be represented in a way that allows for integration of collections that use different vocabularies?" We investigate this question in the context of E-Culture project, which aims to provide browse and search facilities across several collection's metadata. The project has chosen VRA Core Categories as its main domain-specific metadata element set. We investigate if an RDF/OWL representation can be created as a specialization of Dublin Core. A link with Dublin Core should allow integration with other types of collections. We also analyze whether individual collections can be represented as specializations of our VRA Core Categories representation. The type of specialization we focus on is representing which vocabularies a collection uses for which metadata elements (e.g. an archaeological museum requires different vocabularies than a modern art museum). Representing such "collection-specific value ranges" for each element are useful for generating annotation interfaces and for data checking. We develop a pattern that allows representing collection-specific value ranges in combination with metadata schemas and vocabulary representations.

7.1 Introduction

In this chapter we broaden our perspective from interoperable representation of *vocabularies* to interoperable representation of *collection metadata*. Collection metadata is formed by the metadata of individual objects in the collection. An object's metadata usually consists of a set of attributes such as creator and subject, and values for these attributes. The attributes are usually called *metadata elements*. Each element describes which *role* a vocabulary value plays in the context of that object. For example, the value *Van Gogh* can be either the creator or the subject of an object.

A coherent set of elements used for the description of objects in a collection is called a *metadata element set*. We distinguish three types of element sets that are increasingly generic. Firstly, there are sets tailored to individual collections. Such a set contains exactly those elements that are applicable to that collection. Secondly, domain-specific sets are targeted at groups of similar collections (e.g. cultural heritage collections). They only contain those elements that are judged to be generic enough (i.e. occur in most collections) and relevant enough in that domain. Thirdly, domain-independent sets can be applied to many different types of collection metadata. Each of these three types of element set can be specified in RDF/OWL as a schema. Ideally, the schema of one type is a strict specialization of the schema of the next type. Software that processes data

in one schema can then also handle data of more specific schema(s). This simplifies integration of data and processing of data. In this chapter we study the specialization relationship between a domain-specific element set called VRA Core Categories and the domain-independent element set Dublin Core. We use the existing RDF representation of Dublin Core and provide a representation for VRA Core Categories as VRA does not define one itself.

Each collection uses its own set of vocabularies to provide values for particular metadata elements. For example, several Dutch ethnographic museums use SVCN (a vocabulary derived from the AAT) as the source for object type. Sometimes only a *part* of a vocabulary is relevant. For example, the objects in a museum of modern art will have values for the style/period element that are located below AAT's aat:modern European styles and movements concept. Other parts of the aat:Styles and Periods facet are not relevant. Indicating vocabularies used helps in annotation (the appropriate part of the vocabulary can be shown to annotators) and in checking data for correctness (values outside the specified vocabulary part are likely erroneous). This places requirements on the representation of the element sets. Firstly, the domain-specific and domain-independent element sets should not prescribe the allowed vocabularies. Secondly, the element set representations should allow the vocabulary parts to be indicated for a particular collection. We term this feature "collection-specific value ranges". It can be seen as another type of specialization relationship between element sets.

The context of our research is the E-Culture project. It aims at developing a system with semantic browse and search facilities for collections of visual art objects (e.g. paintings, spears). Over time it should evolve into a more generic system so that other types of resources (e.g. videos) and other domains (e.g. TV broadcasts) may be included.

This chapter is structured as follows. In the next section we introduce Dublin Core and VRA Core Categories. In Section 7.3 we introduce the E-Culture project, which forms the context of our RDF/OWL representation of VRA Core Categories. Section 7.4 describes the RDF/OWL representation. In Section 7.5 we discuss how collection-specific value ranges can be specified with the RDF/OWL representation. Section 7.6 discusses if and how VRA Core Categories can be interpreted as a specialization of Dublin Core. Section 7.7 analyzes the correctness of our representation's subproperty hierarchy, including the relationship between VRA Core Categories and Dublin Core. Section 7.8 discusses the consequences of the analysis: how and why should we incorporate the results into the RDF/OWL representation? We end this chapter with a summary in Section 7.9.

7.2 Dublin Core and VRA Core Categories 3.0

Dublin Core

The Dublin Core Metadata Initiative (DCMI) has been working since 1995 to develop and promote international standards for metadata description.[1] A broad range of institutions such as museums and libraries participate. One of DCMI main standards is the DCMI Metadata Element Set[2] (in the

[1] http://dublincore.org/about/

[2] http://dublincore.org/documents/dces/

remainder of this chapter we use "Dublin Core" to refer to the Element Set, not the organization). Dublin Core has as goals simplicity, genericity and extensibility. It provides fifteen "core" elements expected to be useful for describing almost any resource in any collection (Hillman 2005). The elements Creator, Contributor, Publisher and Rights are used to indicate actors involved in the creation and management of the resource. The elements Subject, Description, Title, Coverage and Language indicate aspects of the intellectual content of the resource. Format and Type describe the physical or digital appearance of the resource. The remaining elements are Date, Identifier, Source (points at a resource from which this resource is derived) and Relation (a placeholder for relationships with other resources). DCMI later added forty new elements. Most of these refine existing elements (e.g. DateAccepted, DateCopyrighted), some do not (e.g. InstructionalMethod, AccrualPolicy).

Dublin Core has been widely adopted and an RDF version is available since 2002 (Beckett et al. 2002, Nilsson et al. 2008b). The schema[3] represents each element as an rdf:Property. None have a domain, some have a range (e.g. dc:creator has range dc:Agent). All the fifty-five elements are currently available from one namespace: **http://purl.org/dc/terms/**.

DCMI has since then adopted RDF as its underlying conceptual model, and adopted some of its terminology including "property" and "range". For example, element Medium "refines" Format, and has domain dc:PhysicalResource and range dc:PhysicalMedium. Refinements are represented with rdfs:subPropertyOf. The schema contains twenty-two classes in total for defining domains and ranges.

VRA Core Categories 3.0

VRA Core Categories 3.0 is a standard defined by the Visual Resource Association (an organization in the of visual art resources with over 600 active members[4]). It defines seventeen elements (and *qualifiers* for those elements) with which to describe visual works of art and specific images of those works. VRA Core Categories was explicitly defined without an accompanying syntactical format, with the stated motivation that different "bindings" to a syntax may be developed and used by institutions. We describe an RDF/OWL representation of Core Categories developed for the E-Culture project in 2005. (The newer VRA Core Categories 4.0 was released as a beta in late 2005, definitively in 2007.[5] The changes[6] from 3.0 were the following: three of the elements were split them into new elements, one element was renamed, two new "main" elements were added.)

The VRA Core Categories 3.0 specification[7] describes a metadata element set for the description of *visual art*. It makes a distinction between two types of resources for which metadata can be provided: Works and Images. The former is a work of visual art, which can be almost anything visual ranging from a painting or a statue to a book or an opera performance. The latter is an image depicting a work. There can be many different images that represent the same work: Rembrandt's

[3] http://dublincore.org/2008/01/14/dcterms.rdf

[4] http://www.vraweb.org/about/index.html

[5] http://en.wikipedia.org/wiki/Visual_Resources_Association

[6] http://www.vraweb.org/projects/vracore4/VRA_Core4_Intro.pdf

[7] http://www.vraweb.org/resources/datastandards/vracore3/index.html

Nightwatch can be represented with different pictures of the painting, a detail of the painting, or an X-ray of it. Figure 7.1 shows the description of one of the seventeen main elements from the VRA Core Categories specification. A fixed set of attributes is used to define an element, such as *Description* and *Data Values*. The elements and their description resembles the description of Dublin Core elements by the Dublin Core Metadata Initiative (2008). Some elements have the attribute *Qualifiers* (e.g. "Title" has the qualifier "Title.Translation"). The main elements and some of their qualifiers are listed in Table 7.1. A complete list can be found in Appendix C.

```
MATERIAL
Qualifiers:
  Material.Medium
  Material.Support
Description: The substance of which a work or an image is composed.
Data Values (controlled): AAT
VRA Core 2.0: W5 Technique
CDWA: Materials and Techniques-Processes or Techniques- Name
Dublin Core:  FORMAT
```

Figure 7.1 Original description of VRA element "Material" in the VRA Core Categories specification. Mappings to elements of three other element sets are given: VRA Core Categories 2.0, Dublin Core and CDWA.

The accompanying text explains that each element can be used as many times as appropriate on one Work or Image. No element is mandatory, i.e. no value has to be given for any element. The VRA specification gives a set of examples of how to apply the elements. In the examples referencing to other objects is done by their name (i.e. no identifiers are used). Also, there is no element to link an Image to its Work. This is considered a local implementation issue. For relating Works to each other, either the "Title.Larger Entity" should be used (for physical or logical relationships) or the "Relation" element. Relation is a kind of catch-all category for anything which cannot be specified by other means. The type of relation should be indicated. The specification gives examples such as "Relation.derived from" and "Relation.source for".

7.3 Context: the E-Culture project

The aim of the E-Culture project is to provide semantic browse and search facilities across cultural heritage collections. Each collection has its own syntax for representing metadata, and uses different vocabularies (Schreiber et al. 2006). All of the collection's metadata is converted, where VRA Core elements are used as much as possible instead of converting the original metadata elements as-is. Elements that do not have a VRA Core equivalent are expected to be incorporated as specializations of VRA Core elements. If most of the data fits into VRA Core, this simplifies the development of search and browse infrastructure, as the infrastructure only has to manipulate a limited set of properties and classes. The project is also interested in linking its metadata schema with the Dublin Core metadata schema. Firstly, this allows the project to make its data available to a wider audience. Secondly, it would like to be able to incorporate data from other types of

VRA element	Range	Dublin Core	Example/Meaning
Record Type	{work,image}	Type	type of record
Type	AAT	Type	“print”, “sculpture”, “digital”
Title Title.Translation	formatted text	Title	“This is how it happened” “As Sucedi”
Measurements Measurements.Dimensions	formatted text	Format	 “24.5 x 35 cm”
Material Material.Medium	AAT AAT	Format	Material object is made of “ink”
Technique	AAT	Format	“etching”, “cabinet making”, “scanning”
Creator Creator.Role	ULAN, AAAF Controlled list	Creator, Contributor	names, appellations, or other identifiers assigned to creator or contributor “sculptor”
Date Date.Creation	formatted text	Date, Coverage	“1985”, “5th century”, “ca. 1990”
Location Location.Current Site Location.Current Repository	BHA, AAAF	Contributor, Coverage	geographic location and/or name of the repository, building, or site-specific work “Ann Arbor (MI,USA), University of Michigan Museum of Art”
ID Number		Identifier	The unique identifiers assigned to a Work or an Image
Style/Period Style/Period.Dynasty	AAT	Coverage, Subject	style, historical period, group, school, dynasty, movement, etc. whose characteristics are represented “Vakataka dynasty”
Culture	AAT, LCSH	Coverage	“Indian”
Subject	AAT, TGM, other	Subject	Terms/phrases that describe the object and what it depicts or expresses
Relation Relation.Type Relation.Identity		Relation	relationship between the Work being catalogued and the related work the type of relationship undocumented
Description	text	Description	free-text note about the Work or Image, including comments, description, or interpretation
Source		Source	reference to the source of the information recorded
Rights		Rights	Information about rights management

Table 7.1 Summary of the VRA Core Categories specification. All seventeen main elements are included plus a selection of qualifiers (complete table in Appendix C). The column “Dublin Core” states the mapping of elements to Dublin Core suggested by the specification. The column “Range” states the vocabularies or values that the specification recommends.

collections through the use of a generic schema. It should be possible to e.g. link an object from an archeaological museum to a TV documentary about the excavation. Data on TV documentaries will not use VRA Core Categories but a different specialization of Dublin Core.

The E-Culture project’s collections use different vocabularies to specify the range of values allowed in a particular metadata element. For example, the RMV and ARIA collections both use an in-house vocabulary to record the type of object (painting, vase, spear, etc.). Their experts tailor the vocabulary to the specific kinds of objects these collections have. The object type is listed in a specific part of the vocabulary, separated from e.g. subject matter concepts. Other collections do not specify a particular vocabulary and use literals instead.

The project finds it useful to be able to represent which vocabulary is used for which metadata element in a specific collection. Firstly, “collection-specific value ranges” enables generation of

an annotation interface that is tailored to the particular collection. For example, if a particular record does not have a value for the object type (or a wrong value), the interface can help the user pick a new value by listing values from the appropriate part of the vocabulary. Such an interface is discussed in (Hollink 2006, Ch.6). Secondly, it allows checking for errors in the original data (e.g. to detect that a value given in the "object type" element is in fact not an object type). We expect that in many cases the situation described above indicates some problem in either the annotation or the model, and the data checking tool that is used should make implementers aware of this.

7.4 RDF/OWL Representation of Core Categories

In this section we present a representation of VRA Core Categories in RDF/OWL. The resulting RDF/OWL representation can be found in Appendix C, a small part is shown in Figure 7.2. It was used by the E-Culture project to represent their collections.[8]

Our basic representation is derived from three interpretations of the VRA Core specification. Firstly, we interpret "metadata elements" as rdf:Property's. The only element we do not interpret as a property is "RecordType". Its function is to indicate if a VRA "record" is a work or an image, which is more appropriately represented with rdf:type.

Secondly, we interpret each "qualifier" as an rdfs:subPropertyOf the element. This is based on the observation that most qualifiers seem typical subproperties of their element (e.g. "Title.Translation" qualifier/subproperty of "Title"). This is consistent with Powell et al. (2007), who state that the now defunct Dublin Core term "qualifier" should be interpreted as either

- a specific range of a property (a vocabulary);
- a subproperty of the property ("element") it qualifies; or
- a syntax encoding scheme (datatype).

The first and third interpretations make no sense in this case, so we choose the second interpretation. Two exceptions are qualifiers "Relation.Type" and "Relation.Identity". The former can be realized by making a new property with a particular name (the "type") a subproperty of vra:relation. For the latter there is no description whatsoever as to its function. We suspect it is intended to provide a code for the relationship or referred Work. This can already be represented in RDF/OWL through the property's URI, so we did not include "Relation.Identity" in our schema.

Thirdly, we interpret the record types "Works" and "Images" as classes vra:Work and vra:Image. Their superclass vra:VisualResource is an addition of our own to enable all properties to have those record types as domain or range without resorting to owl:unionOf. We avoid owl:unionOf because it cannot be interpreted by by non-OWL infrastructure.

For the majority of elements the specification recommends vocabularies to be used in actual annotations (e.g. ULAN for vra:creator). We have chosen not to represent these as (global) RDFS ranges because it would prohibit other choices (e.g. use a literal value or a different vocabulary than the one suggested). See the next section for details.

[8] It was also used in a use case description by the W3C Multimedia Semantics Incubator Group, see `http://www.w3.org/2005/Incubator/mmsem/XGR-image-annotation/#solution_culture`.

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix vra: <http://www.vraweb.org/vracore3.htm#> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .

@base <http://www.vraweb.org/vracore/vracore3#> .

<http://www.vraweb.org/vracore/vracore3> rdf:type owl:Ontology .

<VisualResource> a rdfs:Class ;
  rdfs:label "VisualResource" ;

<Work> a rdfs:Class ;
  rdfs:label "Work" ;
  rdfs:subClassOf <VisualResource> .

<Image> a rdfs:Class ;
  rdfs:label "Image" ;
  rdfs:subClassOf <VisualResource> ;


<material> a rdf:Property ;
  rdfs:label "Material" ;
  rdfs:subPropertyOf dc:format ;
  rdfs:domain <VisualResource> .

<material.medium> a rdf:Property ;
  rdfs:label "Material.Medium" ;
  rdfs:subPropertyOf <material> ;
  rdfs:domain <VisualResource> .

<creator> a rdf:Property ;
  rdfs:label "Creator" ;
  rdfs:subPropertyOf dc:creator ;
  rdfs:domain <VisualResource> .

<creator.role> a rdf:Property ;
  rdfs:label "Creator.Role" ;
  rdfs:subPropertyOf <creator> ;
  rdfs:domain <VisualResource> .
```

Figure 7.2 Part of the RDF/OWL representation of VRA Core Categories. Shown are the class VisualResource and its two subclasses. Then follow the properties material and creator, and a subproperty of each. The domain of each property is VisualResource.

As VRA does not define a specific way to link an image to its work, we introduce vra:relation.depicts with domain vra:Image and range vra:Work. It is a subproperty of the generic vra:relation. Other specializations required for an application can also be defined as subproperties of vra:relation. An example would be a property ex:detail, which links an Image to a detail of an Image.

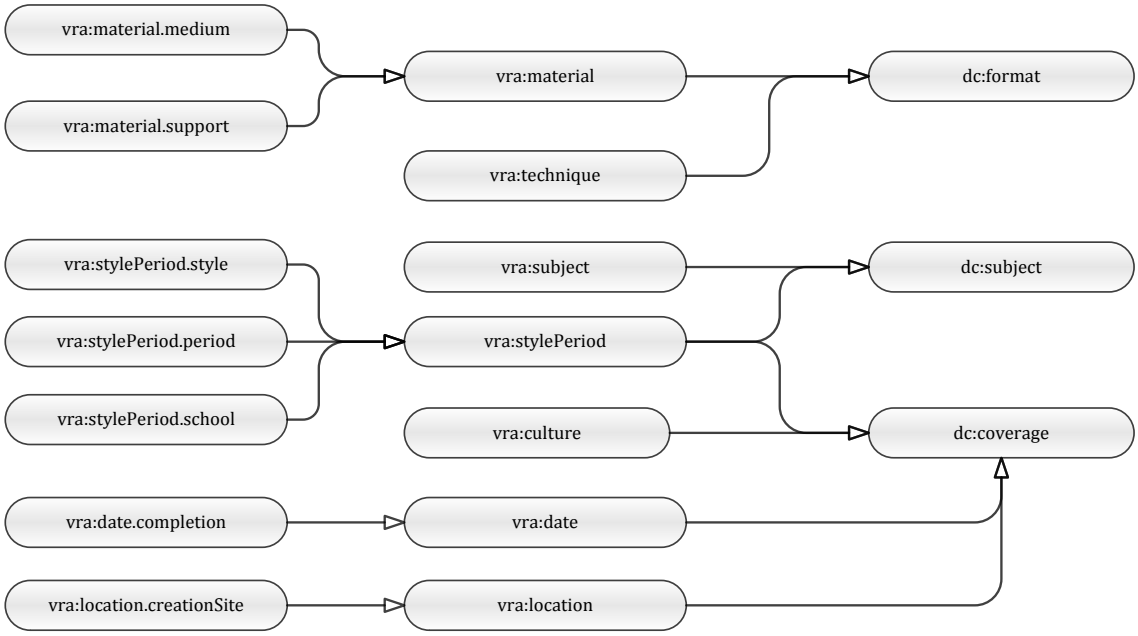

Figure 7.3 Part of the subproperty hierarchy of the RDF/OWL representation of VRA Core Categories. Shown is a selection of subproperties of dc:format, dc:coverage and dc:subject. Arrows represent rdfs:subPropertyOf relations.

Compatibility with OWL DL

As in previous chapters, we try to cater for software that expects representations that only use OWL vocabulary (mainly description logic classifiers).

Firstly, we type each rdfs:Class also as an owl:Class. Secondly, OWL DL requires that each property should be typed as either an owl:DatatypeProperty or owl:ObjectProperty. This creates a dilemma, because it cannot be known beforehand whether a particular collection uses a vocabulary or literals as values for metadata properties. Consider for example two repositories that contain paintings by Rembrandt. One repository does not use a vocabulary to record who made a painting. It uses the string "Rembrandt van Rijn" in the database field. Another repository uses a reference to a vocabulary, e.g. "painter123". The consequence is that vra:creator should be allowed to contain both URIs and literals, which is explicitly forbidden in OWL DL. Furthermore, the Dublin Core specification defines ranges for properties, including the class dc:Agent for dc:creator. The forced choice between owl:ObjectProperty and owl:DatatypeProperty is an overcommitment in the design of OWL DL, and the ranges in Dublin Core are an overcommitment in the design of the Dublin Core element set. It makes it impossible for us to comply with the criterium of *minimal ontological commitment* defined by Gruber (1994, p.3) in our RDF/OWL specification of VRA:

> " *An ontology should make as few claims as possible about the world being modeled, allowing the parties committed to the ontology freedom to specialize and instantiate the ontology as needed.* "

We have chosen not to type the properties in our VRA Core RDF/OWL schema as owl:DatatypeProperty or owl:ObjectProperty. If one uses our schema together with an RDFS representation of the Dublin Core element set, then VRA subproperties of Dublin Core elements in our schema will inherit their ranges. We feel this hinders integration of cultural heritage collections. The collections that for example the MultimediaN E-Culture project aims to integrate are not under the project's control, and not all conversions are done by the project team. It is inevitable that not all data will conform to the separation in owl:ObjectProperty and owl:DatatypeProperty that is prescribed by Dublin Core and enforced by OWL DL. However, OWL is not always an obstacle to integration. In the next section we examine a situation where the design of OWL supports minimal commitment while RDFS hinders it.

OWL property characteristics

As in previous chapters, for each property a decision has to be made as to which OWL property characteristics it has. Almost all elements/properties in VRA Core are asymmetric, meaning that if $R(a, b)$ then not $R(b, a)$ and also not $R(a, a)$. For example, <:NightWatch, vra:creator, :Rembrandt> is a reasonable statement, but the statements <:Rembrandt, vra:creator, :NightWatch> (inverse) and <:NightWatch, vra:creator, :NightWatch> (reflexive) are not. In OWL 2 asymmetry can be specified with owl:AsymmetricProperty (Web Ontology Working Group 2009). The only property that is not clearly asymmetric is vra:relation. It is a typically symmetric property: if $R(a, b)$ then also $R(b, a)$. Just like skos:related (Semantic Web Deployment Working Group 2008a) it can be specified to be owl:SymmetricProperty. It is safe to specify symmetry even though vra:relation may have non-symmetric subproperties, as symmetry is not inherited in OWL.

The property vra:relation is the only candidate for transitivity. However, the semantics of vra:relation are probably intentionally left underspecified as it is meant as a placeholder for specializations. Any relationship that one wishes to express (e.g. "Relation.source for") should fit. If someone wishes to use vra:relation directly (instead of specifying a subproperty), then transitivity of this relation should not be enforced. Properties that specialize it (with rdfs:subPropertyOf) can be declared transitive if this conforms to their semantics.

No VRA Core property can be functional, as the VRA Core Categories specification specifically states that all elements may be repeated as many times as necessary to describe a vra:Work or vra:Image. Inverses can be stated for all properties, e.g. vra:isTitleOf and vra:isLocationOf.

Refining the mapping to Dublin Core

The VRA Core specification maps its elements to the original fifteen Dublin Core elements only. It does not make use of the additional forty elements and element refinements. Consequently, our schema does not include mappings to the additional elements either. However, we have analyzed which elements and refinements can be used to create a more precise mapping of VRA Core to Dublin Core. We found that nine of in total fifty VRA properties could be mapped to six additional Dublin Core properties:

- vra:material to dc:medium ("The material or physical carrier of the resource");
- vra:measurements.dimensions and vra:measurements.resolutions to dc:extent ("The size or duration of the resource");
- vra:title.variant and vra:title.translation to dc:alternative ("An alternative name for the resource");
- vra:largerEntity and vra:series to dc:isPartOf, ("A related resource in which the described resource is physically or logically included");
- vra:creation to dc:created ("Date of creation of the resource");
- vra:date.alteration to dc:modified ("Date on which the resource was changed").

These mappings are available in a separate schema[9] that can be used as an add-on to the VRA Core Categories schema discussed so far. Only a small subset of the additional Dublin Core properties are applicable to VRA Core. Many new properties address aspects not directly relevant to VRA Core. For example, three of the new properties are dedicated to when and how new items are added to a collection of resources (dc:accruelMethod, dc:accrualPolicy, dc:accrualPeriodicity). Another set of examples is dc:dateAccepted (e.g. accepting a thesis), dc:dateCopyRighted and dc:dateSubmitted (e.g. submitting a thesis). These notions are not present in VRA Core.

7.5 Collection-Specific Value Ranges

In the context of the Semantic Web we should consider how to integrate collections, and at the same time retain the specific semantics of each collection. Because there are no "universal" vocabularies that are appropriate for all collections, each will use a set of vocabularies specific to its situation. We have to be able to indicate which collection uses which vocabulary for which metadata element. For example, we might want to state that Rijksmuseum uses ULAN for element "creator". We term this "collection-specific value ranges". An obvious way to do this is that each institution indicates a its own rdfs:range for vra:creator. However, this leads to unwanted inferences because multiple ranges are interpreted as intersection in RDFS and OWL. For example, assume that two museums use two different vocabularies to indicate creators. The first indicates that the range of vra:creator is v1:Artist and the second that the range of vra:creator is v2:Creator. Now any instance of these classes that is used in the range of vra:creator will be inferred to belong to the intersection of these two classes, i.e. it becomes an instance of *both* classes. The concepts from separate vocabularies are inferred to belong to other vocabularies.

A solution that does not cause these unwanted inferences is to create subclasses of vra:Work and vra:Image tailored to the specific collection. Attached to the subclasses are new subproperties of VRA properties that define local rdfs:ranges. However, this solution would cause a multitude

[9] http://thesauri.cs.vu.nl/vra/

of subproperties that actually have the same or very similar meaning. It also causes a blurred distinction between properties that model a specialized relationship between concepts and properties that only restrict the values to a particular vocabulary.

We conclude that RDFS does not allow us to specialize the ontology as advocated by Gruber's criterium of minimal commitment. OWL does provide a solution. OWL restrictions can be used to restrict the values a property may take to any class of values that can be specified with OWL class expressions (Dean et al. 2004). Restrictions can help us here by attaching them to subclasses of Work/Image. These subclasses group together collection-specific combinations of metadata properties and vocabularies. An example in Figure 7.4 shows how the property vra:creator is restricted to values of the class vocab:Artist. The example shows a collection-specific specialization of vra:Work called ex:MyWorkClass (the triple <ex:MyWorkClass, rdfs:subClassOf, vra:Work>). Then follow four statements that together define the restriction (from rdfs:subClassOf to vocab:Artist). It specifies that ex:MyWorkClass is a subclass of another class. This other class is defined as the class of things which have particular values for the property creator (owl:onProperty vra:creator). These values should come exclusively from the class Artist (owl:allValuesFrom vocab:Artist). In other words: ex:MyWorkClass is a subclass of the class of things which have only vocab:Artists as values for the property vra:creator.

```
@prefix ex: <http://www.example.com/#> .
@prefix vocab: <http://www.vocabulary.com/#> .

ex:MyWorkClass a owl:Class ;
  rdfs:subClassOf vra:Work ;
  rdfs:subClassOf [
    a owl:Restriction ;
    owl:onProperty vra:creator ;
    owl:allValuesFrom vocab:Artist
  ] .
```

Figure 7.4 Example of restricting the possible values of a property to a particular class. Only instances of vocab:Artist are allowed for property vra:creator. The class ex:MyWorkClass is defined as a specialization (subclass of) vra:Work.

This approach works fine if the goal is to restrict a property's values to *instances* of a class (instances of vocab:Artist in Figure 7.4). However, in the E-Culture case we would like to restrict the values to part of a hierarchy defined between instances (e.g. skos:broader hierarchy between instances of skos:Concept). For example, the restriction for vra.material should be defined on the instance aat:material. The allowed values are not instances of aat:material, but instances of aat:Concept that are located below aat:material in the vp:parent hierarchy (see Chapter 5 for information on the AAT representation).

We can solve this problem if we can create a class that contains the required values, and then make our ex:MyWorkClass a subclass of that class of values. The required values consist of all subconcepts of aat:material. This class can be defined as an OWL restriction (see Figure 7.5). The restriction states that the property vra:material can only have values that are related to mate-

```
vp:parent a owl:TransitiveProperty .

ex:MyWorkClass a owl:Class ;
   rdfs:subClassOf vra:Work ;

   rdfs:subClassOf
     [  a owl:Restriction ;
        owl:onProperty vra:material ;
        owl:allValuesFrom
             [ a owl:Restriction ;
               owl:onProperty vp:parent;
               owl:hasValue aat:material
             ]
     ] .
```

Figure 7.5 Example of a collection-specific value range for the property vra:material. A restriction is added to the collection-specific class ex:MyWorkClass that states that only values below aat:material are allowed. This only works if the hierarchical relation (in this case vp:parent) is transitive.

rial through the vp:parent relation. Note that vp:parent has to be declared transitive to allow all concepts below aat:material to be directly related to aat:material through vp:parent. This solution is probably applicable to other cases than just the E-Culture project, so we have described it as a reusable pattern:

OWL Pattern for collection-specific value ranges. USE THIS PATTERN TO RESTRICT THE VALUES OF A METADATA ELEMENT TO PART OF A VOCABULARY. THE PATTERN IS APPLICABLE WHEN THE VOCABULARY IS DEFINED AS A HIERARCHY BETWEEN INSTANCES. THE ELEMENT IS REPRESENTED BY PROPERTY P, THE HIERARCHICAL PROPERTY BY H, THE HIERARCHY PART HAS CONCEPT C AS MOST GENERIC CONCEPT.

- create a class W that will act as placeholder for collection-specific value ranges;
- define a restriction on P that states that all values are allowed that are related by property H to concept C;
- make W a subclass of this restriction;
- define that H is transitive.

This pattern happens to be almost identical to one discussed in W3C's "Classes as values" note (Semantic Web Best Practices and Deployment Working Group 2005a), as "Approach 1". The note addresses the situation where one wishes to restrict the values of a property to a particular class and its subclasses. In this situation almost the same pattern can be applied. The main difference is that the second restriction is defined on the hierarchical property between classes (rdfs:subClassOf) instead of on a hierarchical property between instances (e.g. skos:broader). When defined on a skos:broader hierarchy (as in our case), this pattern remains within the bounds of OWL DL. When the pattern is defined over an rdfs:subClassOf hierarchy (as in the W3C case) it does not.[10]

[10]The W3C note presents alternatives that do remain within OWL DL.

The solution we describe above was used in practice. The E-Culture project used it in the annotation user interface of the E-Culture search system. An extended example of how the project specialized VRA Core for describing the Artchive collection metadata is given by Hollink (2006, Ch.6).

7.6 Relationship between VRA Core and Dublin Core

For the purpose of interoperability with other datasets, the E-Culture project is interested in a clear and simple relationship between VRA Core Categories and Dublin Core. We investigate the relationship between Dublin Core and VRA Core on three points: (1) can VRA Core metadata in RDF be interpreted as Dublin Core metadata in RDF; (2) can VRA Core be seen as a specialization of Dublin Core; (3) is our proposed solution for collection-specific value ranges compatible with Dublin Core guidelines on value ranges.

The VRA Core specification provides for each element a mapping to Dublin Core element(s). For example, Material is mapped to Format and ID Number to Identifier. It appears that mapped VRA properties can be interpreted as equivalents or specializations of Dublin Core. As the Categories should be a specialization of Dublin Core in the domain of visual art we declared each VRA property an rdfs:subPropertyOf the properties of the Dublin Core RDF schema. This solution indeed allows us to see VRA metadata in RDF as Dublin Core metadata. Notice that we use "specialization" here in the sense that for each VRA property, a more general property can be found in Dublin Core. We do not intend that VRA Core provides a more specific property for each Dublin Core property (for example, VRA Core does not specialize dc:audience).

The second question concerns whether VRA Core can be seen as a specialization of Dublin Core. The Dublin Core community has provided a framework for specifying specializations of Dublin Core in order to improve interoperability and reusability. A specialization for a type of application is called an *Application Profile* (AP). If VRA Core is to be seen as a specialization of Dublin Core it should conform to this framework. Below we investigate whether the VRA specification can be seen as such as an AP. One of the mandatory components of an AP is a Description Set Profile, which should describe (Nilsson et al. 2008a):

- the types of resources the metadata describes in this context;
- their intended usage;
- how many times the element may occur for one resource; and
- what values are allowed for an element.

The cardinality and value constraints on the elements are described in a structured table format, see e.g. the Collection AP (Dublin Core Collection Description Task Group 2007). The other mandatory components of an AP are: descriptions of the AP's *Objectives and Scope*, *Functional requirements* (what type of application functions and scenarios must the profile support) and *Domain model*.

We can compare these requirements to the VRA Core specification. Firstly, the specification describes the types of resources: Works and Images. Secondly, it describes their intended usage in the form of examples. Thirdly, it specifies that each element may be used zero or more times on each resource. Fourthly, it specifies recommended vocabularies for elements. We conclude that the VRA specification has an accurate Description Set Profile. It does not explicitly articulate the other mandatory components of an AP, but implicitly much is already present (e.g. the domain model will largely consist of the existing specification's description of the meaning of Works and Images). We conclude that VRA Core Categories can be seen as a Dublin Core AP.

The third question is whether our proposed solution for collection-specific value ranges is compatible with Dublin Core guidelines (DCMI Usage Board 2009). We look specifically at the machine-readable formats to express value ranges. DCMI has proposed two formats to express constraints, based on XML and RDF (Nilsson et al. 2007, Nilsson 2008). The RDF format is still in draft status; e.g. the namespace used in the proposal does not exist. We compare our approach with DCMI's RDF-based proposal. A dsp:NonLiteralConstraint can be asserted that specifies the class of values allowed (dsp:valueClass) for a particular property. This is similar to OWL's RDF syntax for specifying restrictions. However, the approach taken by DCMI is semantically different from the approach that OWL takes. The former specifies *constraints* to the allowed data, while the latter infers the consequences of the data that are still logically consistent. Given a value restriction on e.g. creator with range vocab:Artist, the former approach allows to detect values of that property that do not belong to vocab:Artist, but e.g. vocab2:Publisher. The latter approach will instead infer any values from other classes such as vocab2:Publisher to also be an instance of class vocab:Artist. Depending on the purpose one of the approaches may be more suitable. For the purpose of generating an annotation interface this makes no difference, for the purpose of data checking the DCMI solution makes more sense. Therefore this solution may be preferable over ours as soon as it is standardized. We conclude that our solution is not entirely compatible but close in spirit to the DCMI proposals.

7.7 Analysis of VRA Core Hierarchy

Our VRA Core schema is based on a relatively straightforward interpretation of the VRA specification. In this section we analyze the subproperty hierarchy of our representation. This is necessary because we assume that instances on the lower levels of the subproperty hierarchy can be safely interpreted as instances of the higher levels (especially the Dublin Core level). This is the main mechanism used to reach interoperability on each level. Figure 7.6 shows the part of the hierarchy with issues. An overview of which subproperty relationships are wrong is provided in Figure 7.7 at the end of this section.

We mainly use OntoClean's notion of *identity criteria* for analysis of the hierarchy. An identity criterion (IC) allows comparison of two instances of a class to determine whether they are the same instance or not. For example, the class Person can have the criterion "social security number". The ICs of subclasses need to be compatible with the IC of the superclass: the same criteria should allow us to identify instances of the subclass. For example, Student is a valid subclass of

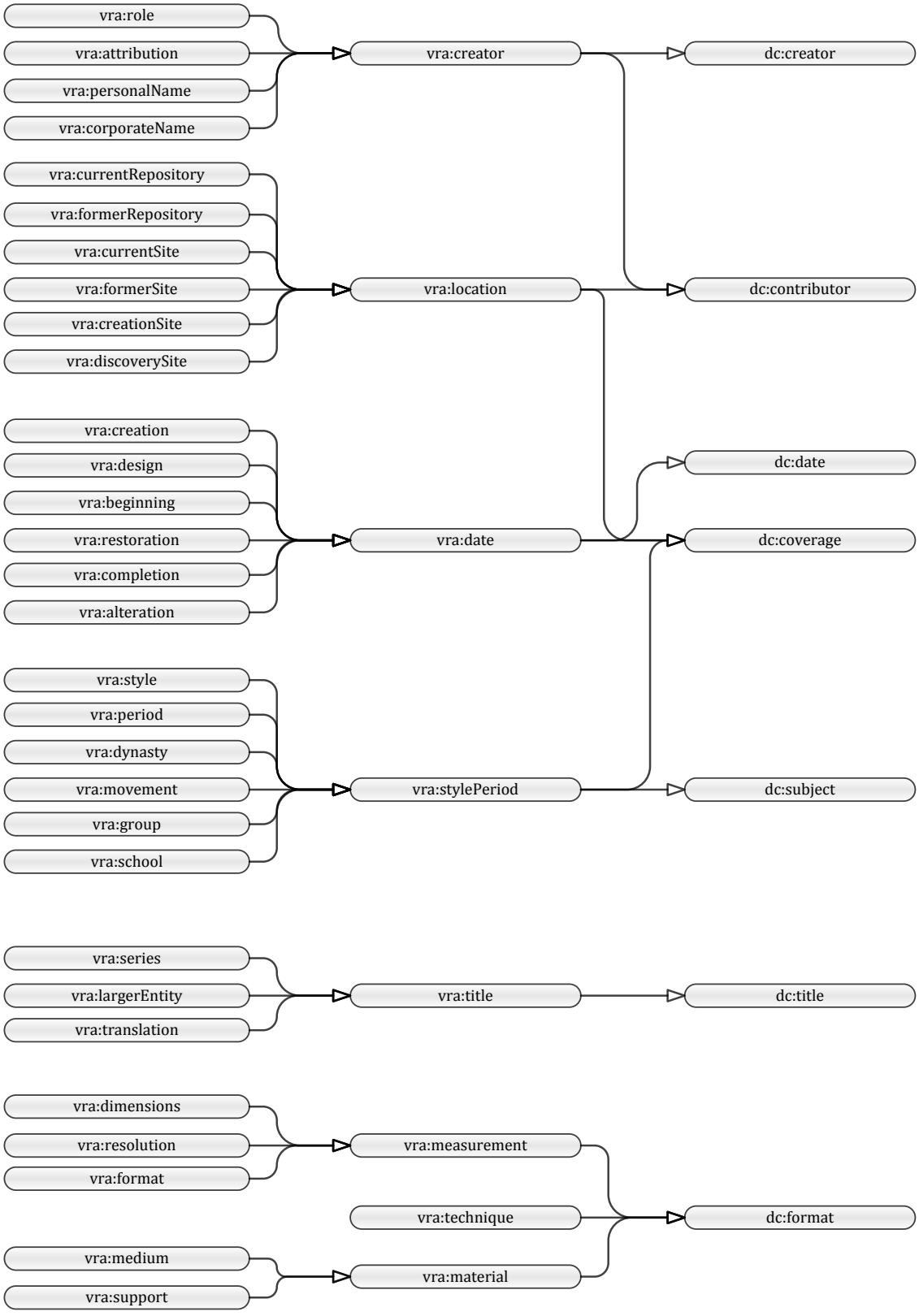

Figure 7.6 Part of the VRA/DC property hierarchy that contains problematic subproperty relationships. Arrows represent rdfs:subPropertyOf relations. The names of properties on the left side have been abbreviated.

Person because students have social security numbers (Welty and Guarino 2001). The designers of OntoClean acknowledge that it is not always possible to exactly pin down proper ICs. In these cases we will argue against a subproperty relation based on examples. The examples specify an inference (dumb down) that we feel is unwanted in applications such as the E-Culture system.

Location

The properties vra:location.formerSite, vra:location.currentSite, vra:location.creationSite and vra:location.discoverySite appear to have "city" or "named geographic area" as range (evidence comes from the examples in the VRA Core specification). However, the properties vra:location.currentRepository and vra:location.formerRepository contain both a city name and the name of an institution (e.g. "New Delhi (IND), National Museum of India"). If we assume that the intention of the last two properties is to indicate an institution (an organization), then we have an incompatibility between Current/FormerRepository and its superproperty vra:location. The identity criterion of a named geographic region (coordinates, name) cannot be applied to an organization (members, charter, governing body).

Under this assumption the subproperty relationships between vra:location.currentRepository, vra:location.formerRepository and vra:location are wrong and should be removed.

Measurements

The property vra:measurements has subproperties vra:measurements.dimensions, vra:measurement.resolution and vra:measurement.format (the range of vra:measurements and its subproperties is undefined). The property vra:measurements.dimensions is used to indicate "volume, weight, area or running time". The property vra:measurement.resolution is used to indicate image resolution, and vra:measurement.format is an enumeration of image formats[11] such as JPEG.

The ranges of these properties have different identity criteria (resolution is a quantity expressed in e.g. pixels, digital file formats are specified by a set of syntactic constraints, dimensions are one or more length quantities expressed in e.g. centimeter). It is impossible to come up with a criterium for measurement that is compatible with dimensions, resolutions and formats. The subproperty relations between vra:measurement and all of its subproperties should be removed.

Style/Period

The property vra:stylePeriod has subproperties such as vra:stylePeriod.movement, vra:stylePeriod.group and vra:stylePeriod.dynasty. The properties vra:stylePeriod.movement and vra:stylePeriod.group both denote a group of artists who share a set of artistic principles, but a dynasty indicates a royal family that offers patronage to artists. The property vra:stylePeriod.style denotes visual characteristics that the works of such groups of people share, while vra:stylePeriod.period describes a group of people and their style from the perspective of the time in which they lived. All the notions modeled by the subproperties are somehow related, but we think they are not the same.

[11] Besides image formats, the examples in the specification also give values such as Panel, which makes the semantics unclear.

We cannot think of an identity criterion that they share. The subproperty relationships should be removed.

Title

The property vra:title.series is used to refer to the series the object belongs to. It appears to be a mereological relationship (the object is a part of a whole called a series). Because it is a subproperty of vra:title this allows the inference that the title of an object is the same as the title of its series. However, a part does not always inherit the attributes of its whole (see e.g. discussion on propagation in Odell 1994), and certainly not in this case.

We can also use the notion of identity criterion to argue that the subproperty relationship is wrong. We could arguably use the name as (part of) an IC for the object. (We might even claim it is an owl:InverseFunctionalProperty although this is too strong because sometimes two works of art have the same title.) Any subproperty should also allow us to identify the object. For example, vra:title.translation is a valid subproperty. Names in foreign languages can also be used to identify an object, they are simply a subset of all names used to identify the object. Then vra:title.series is not a valid subproperty as it does not allow us to identify the object, because a series by definition refers to multiple objects (so it is certainly not inverse functional).

The property vra:title.largerEntity is also some kind of mereological relation: it refers to the object that this work is physically part of or located in, by denoting its name. The same objections can be made against the subproperty relation between vra:title.largerEntity and vra:title can be made as with vra:title.series.

These properties should be replaced with new properties vra:series and vra:largerEntity as subproperties of vra:relation.

Creator

The property vra:creator is meant for specifying who is (mainly) responsible for creating the Work or Image. It has subproperties vra:creator.personalName and vra:creator.corporateName which allow one to record whether the creator is a person or company. It appears that these are attributes of the creator, not of the object itself (just as in the case of vra:title.series). The properties should be removed from the subproperty hierarchy. An alternative is to create a class vra:Creator as range of vra:creator. Two subclasses can then be created (vra:Person and vra:Corporation) which can be used as domains for vra:personalName and vra:corporateName. These properties can be made a subproperty of a new property vra:name which has domain vra:Creator.

The subproperty vra:creator.role indicates the role or profession that was fulfilled by that creator (e.g. architect, photographer). We cannot think of an IC for a role that is compatible with an IC for a person or company. The subproperty relation should be removed.

There is another issue with vra:creator.role. It appears that vra:creator.role is present to allow a role to be assigned to creators. However, this is not possible with binary properties. If more than one creator is assigned to an object, it becomes impossible to distinguish who played which role.

Analysis of VRA - Dublin Core hierarchy

The mapping of VRA Core elements to Dublin Core elements is realized by making the former subproperties of the latter. Again there are some cases in which the identity criteria are not compatible.

Firstly, vra:technique is specified as a subproperty of dc:format.[12] While the former refers to a process (method of manufacturing), the latter refers to an object or characteristic of an object. For example, we consider the inference that the technique of "painting" (method of manufacturing) is a type of format to be a wrong inference. Of course, the process of painting usually has a painting (object) as end product (which we might consider to be a format), but these are not the same thing. For example, upper level ontologies such as DOLCE (Gangemi et al. 2003a) usually include objects and processes as fundamental concepts that cannot subsume one another.

Secondly, vra:stylePeriod has dc:subject and dc:coverage[13] as superproperty. These are both wrong, as a style or period is not the *topic* of a work of art. For example, the "Sunflowers" by Van Gogh has sunflowers as topic, not "expressionism". The style/period and topics in that style/period can have a strong correlation, but this is not enough to warrant a subproperty relationship. It is also not clear why vra:stylePeriod has two superproperties. It seems that the meaning of dc:subject and dc:coverage overlap: dc:coverage is specifically intended for the "spatial or temporal" subject of the resource.

Thirdly, vra:date has superproperties dc:coverage and dc:date. The former superproperty is wrong, as vra:date is not intended to convey the *topic* of an object. The latter superproperty is correct.

Fourthly, vra:location has dc:coverage and dc:contributer[14] as superproperties. The latter is wrong because a location is not an active entity (agent). The former is also wrong because the vra:location properties are intended to indicate the location of the object, not the *topic*. An alternative is to map vra:location to dc:relation. Technically this is a solution, but the meaning of the relationship is then lost on the Dublin Core level. It is remarkable that Dublin Core does not have a dc:location, as it appears that Dublin Core aims at covering all fundamental aspects of objects in its fifty-five elements. Geographic location appears a fundamental notion just as time is, which Dublin Core does include in the form of dc:date.

An alternative mapping for vra:location.currentRepository and vra:location.formerRepository is dc:publisher[15], as a museum (as an organization) is responsible for making an object available.

Fifthly, vra:creator has dc:creator and dc:contributor as superproperties. This is not appropriate as dc:creator is a subproperty of dc:contributor. The mapping to dc:contributor is superfluous and should be removed. However, it may be that the VRA specification intended that values of VRA's creator-element (persons) should be interpreted as either a DC creator or a DC contributor depending on whether the person is a the main contributor or a secondary contributor. This is somewhat likely because dc:creator and dc:contributor used to be disjoint (Baker 2008); i.e. cre-

[12] "The file format, physical medium, or dimensions of the resource."

[13] "The spatial or temporal topic of the resource, the spatial applicability of the resource, or the jurisdiction under which the resource is relevant."

[14] "An entity responsible for making contributions to the resource."

[15] "An entity responsible for making the resource available."

ators were not interpreted to be a specific kind of contributor. The remaining question is why VRA has not included its own contributor element side by side with its own creator element.

An additional problem is that vra:creator.role is not a proper subproperty of both VRA's and DC's creator. Moreover, a binary property does not allow to express which role is played by which creator. One solution is to introduce a 3-aried relation creation(Resource, Creator, Role) (e.g. through W3C's n-ary relation pattern), but has the drawback that instances of this relation require a rule to allow the creator-values to be dumbed-down to dc:creator. An alternative is to use the list of "Relators" defined by the Library of Congress as subproperties of dc:contributor (DCMI Usage Board 2005) (e.g. loc:etcher). The role of the creator of a resource then has to be indicated by using both the appropriate Relator property as well as dc:creator. This is actually an example of the pattern described in guideline 14 ("factor the third argument of a relation into subproperty names").

Sixthly, dc:format is superproperty of vra:material and vra:measurements. It is hard to imagine an IC that covers both physical substances and also image formats, dimensions and resolutions. The subproperty relations between dc:format and its subproperties should be deleted.

7.8 Consequences of Hierarchy Analysis

The analysis in the previous section showed why some subproperty relationships of the schema proposed in in Section 7.4 have to be removed. The result is that properties such as vra:measurements, vra:stylePeriod and its subproperties have no mapping to any Dublin Core element. The consequence is that the VRA Core schema cannot be seen as a specialization of Dublin Core. However, we feel this would be the approach of an "ontological purist". The goal of the Dublin Core element set is to provide a small, sensible set of primitives to represent object metadata. It does so at a very generic level (e.g. group all aspects related to an object's appearance). Although ontological theory says it is "wrong", Dublin Core elements represent a grouping designed to be useful and intuitive to users. If a user browses a painting's metadata and reads "Format: 100 x 200 cm" (generalized from vra:measurements.dimensions to dc:format – or in DC terminology "dumbed down") a user will probably correctly interpret this as the painting's dimensions. If a user later browses a statue's metadata that states "Format: oil on canvas" (dumbed down from vra:material.medium to dc:format) again the user can probably make the correct interpretation. Another example: a painting which has metadata "Subject: Impressionist" (dumbed down from vra:stylePeriod.style to dc:subject) can be understood as "this painting is impressionist and will probably contain typical impressionistic subject matter". For such display tasks these dumb downs (generalizations) are useful.

The limitations of Dublin Core show in other tasks, such as querying and facet-based browsing. For example, if a user uses dc:date to query for all objects *restored* on a date, s/he will not expect artworks that were *begun* on that date in the query result. Another example is a query where vra:location is constrained to a concept that represents the city of Amsterdam. This query can can retrieve artworks that were there in the past but are not there anymore (through vra:location.formerRepository).

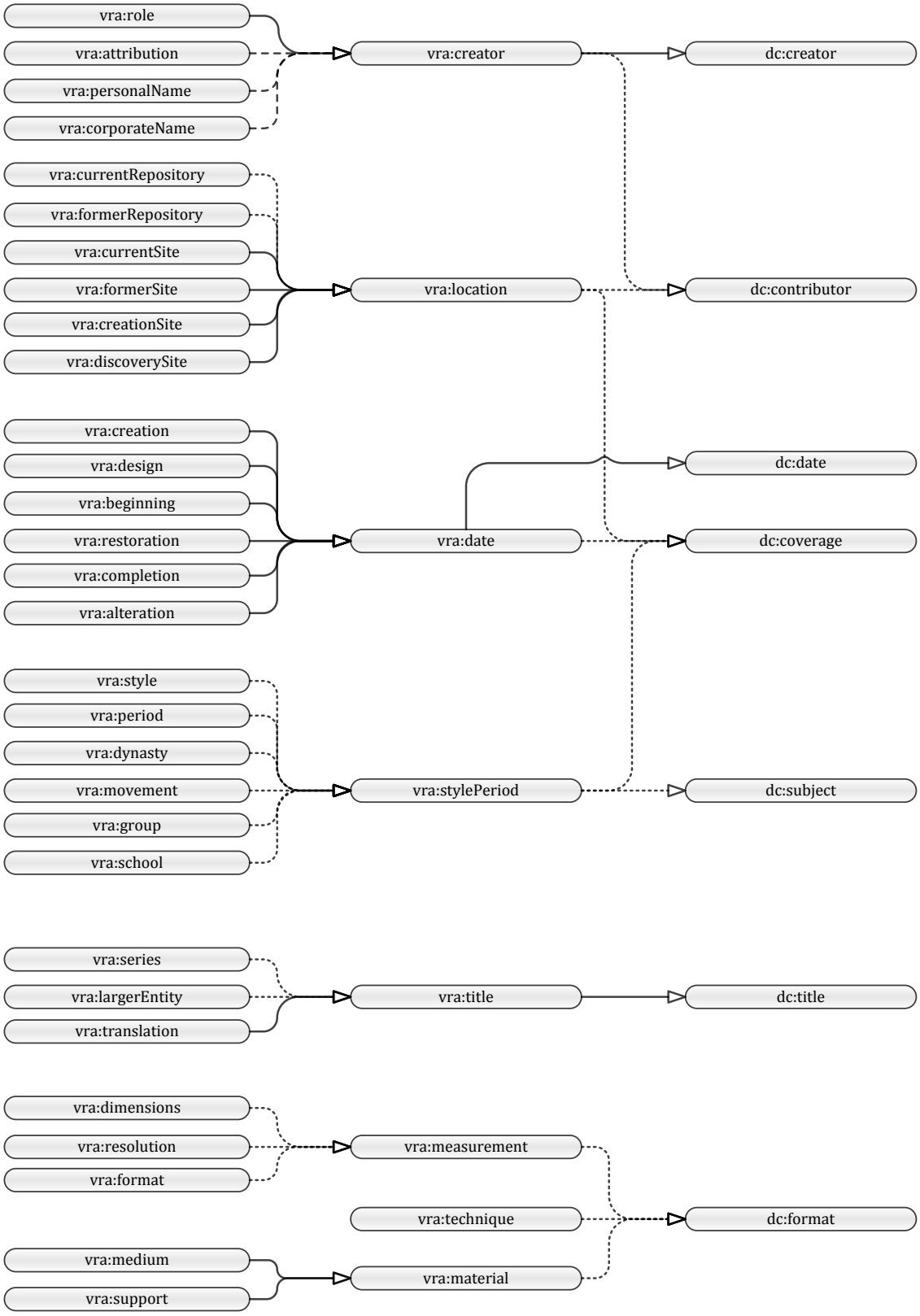

Figure 7.7 Part of the VRA/DC property hierarchy that contains problematic subproperty relationships. A dashed line indicates a relation that needs to be removed according to our initial analysis.

Concluding, we feel that the generalizations (superproperties) provided by VRA and Dublin Core are useful for some display and exploratory tasks, but have their limitations with respect to queries and facet displays. A consequence of this observation is that most of the subproperty relationships that we found problematic in Section 7.7 will not need to be removed, provided they are used for display tasks only. We again summarize the problematic properties combined with our final decision in Table 7.2.

There are three cases that remain problematic because the above observation does not apply to them. Firstly, vra:location has no alternative in Dublin Core. We find this surprising, as we feel that geographic aspects of a resource is as fundamental a notion as temporal aspects of a resource (which *is* present in the form of dc:date). Secondly, VRA Core makes no distinction between creators and contributors, which is present in Dublin Core. Without addition of a dc:location and a vra:contributor we cannot say that VRA Core is a complete specialization of Dublin Core. Thirdly, the meaning of dc:coverage is not clear to us, which makes it hard to decide whether it should be a superproperty of vra:stylePeriod or not. Part of its definition seems to indicate that it is concerned with geographic and temporal aspects of subjects. This would make it dc:coverage a subproperty of dc:subject. Under this interpretation it would be better to remove the mapping of vra:stylePeriod to dc:coverage, as dc:subject already covers all aspects related to the work's subject. It is necessary to contact DCMI to be certain.

Source	Target	Solution
vra:creator.Role	vra:creator	replace with MARC Relators
vra:creator.PersonalName	vra:creator	attach to new class vra:Person
vra:creator.CorporateName	vra:creator	attach to new class vra:Corporation
vra:title.Series	vra:title	rename to series, map to dc:isPartOf
vra:title.LargerEntity	vra:title	rename to largerEntity, map to dc:isPartOf
all vra:measurement subproperties	vra:measurements	useful; keep
all vra:stylePeriod subproperties	vra:stylePeriod	useful; keep
vra:technique	dc:format	useful; keep
vra:measurements	dc:format	useful; keep
vra:material	dc:format	useful; keep
vra:stylePeriod	dc:coverage, dc:subject	useful; keep
vra:date	dc:coverage	remove
vra:location	dc:coverage, dc:contributor	remove; new dc:location should be created as target
vra:location.CurrentRepository, vra:location.FormerRepository	vra:location	map to dc:publisher
vra:creator	dc:contributor	remove; new vra:contributor should be created as source

Table 7.2 Problematic subproperty relations and proposed solution. The upper part of the table lists relations between VRA Core properties, the lower part between VRA Core and Dublin Core properties.

Impact on E-Culture project

The E-Culture project used the schema presented in Section 7.4 since 2005 to convert several collections. The issues discussed above have no consequences for the converted collections, because

the data provided did not use (datafields equivalent to) vra:title.series and vra:title.largerEntity. Literal values found for vra:creator were mostly converted to concepts from ULAN, so vra:creator.personalName and vra:creator.corporateName were not used. For a subset of works the vra:stylePeriod.style property was manually filled with values from AAT. The project used vra:location.currentRepository and vra:location.formerRepository to indicate museums just as our analysis suggests. It did however use vra:measurements as a facet in a facet browser, thus mixing image formats and dimensions. Our expectation is that users will find this facet dissatisfactory when the facet lists many different types of values. It is better to use subproperties such as vra:measurements.resolution as facet. The current version of the application conforms to this recommendation.

Our analysis shows that the value of Dublin Core as a means to integrate collection metadata is limited. We recommend the E-Culture project team to investigate other options for data integration.

7.9 Summary

This chapter has addressed research question 3: "How can metadata schemas be represented in a way that allows for integration of collections that use different vocabularies?" This question is important for cross-collection search and annotation applications, such as the E-Culture project's system. Such an application requires a domain-specific metadata element set upon which it can build its search and browse infrastructure. The schema of each individual collection should ideally be a specialization of the domain-specific schema. The domain-specific schema should ideally be a specialization of a domain-independent schema.

The E-Culture team chose VRA Core Categories as domain-specific metadata element set. We investigated whether VRA Core Categories could be represented as a specialization of the domain-independent Dublin Core schema. We conclude that most of the properties can be related using the rdfs:subPropertyOf mechanism. Exceptions include vra:location and vra:creator.role. VRA Core also lacks a property to match dc:contributor.

The subproperty hierarchy we created based on the VRA Core Categories specification violates ontology design principles (especially constraints on identity criteria). Grouping properties into e.g. dc:format only makes sense on an intuitive level. Consequently, Dublin Core is only useful for generating displays for humans who can correctly interpret dumbed down property-value pairs. We conclude that it is useful to keep most of the subproperty relations for this goal, even though they violate ontology design principles. We do note that Dublin Core is not very useful for integration of collection metadata. Although it does allow integration, it does so at a too high level of abstraction. The properties that Dublin Core offers do not allow precise queries to be made. The E-Culture team should look for other ways to achieve data integration.

We investigated how the VRA Core schema can be specialized with respect to the vocabularies used (collection-specific value ranges). This allows checking data and generating tailored annotation interfaces. We described a pattern with which collection-specific value ranges can be represented using OWL restrictions.

In our efforts to link VRA with Dublin Core we found that Dublin Core prescribes ranges for properties such as creator and that OWL DL enforces that each property is either an owl:DatatypeProperty or owl:ObjectProperty. This creates a dilemma, because it cannot be known beforehand whether a particular collection uses a vocabulary or literals as values for metadata properties. This overcommitment in Dublin Core and OWL DL obstructs integration of metadata collections that make different decisions. We feel that this makes OWL DL unsuitable for data integration in the cultural heritage domain.

Acknowledgements

We thank the members of the E-Culture project team for useful discussions on VRA Core Categories. We specifically thank Jan Wielemaker for the idea for the OWL representation of collection-specific value ranges.

8

Conclusions and Discussion

This thesis has focused mainly on how vocabularies can be made available on the Semantic Web, which is a prerequisite for any Semantic Web application that wishes to use them. To that end, the problem statement was formulated as follows: *"How can existing vocabularies be made available to Semantic Web applications?"* The contributions of this thesis towards solving this problem are:

Methodological contributions

- a generic method for converting vocabularies to RDF/OWL, comprised of several steps and guidelines (Chapter 2);
- a specific method for converting vocabularies to SKOS (Chapter 3;
- a method for converting vocabularies focused on applications (Chapter 5);

Contributions related to the case studies

- illustrations of the methods (through case study descriptions)
- concrete conversions that can be used by querying them online and/or download them (i.e. RDF/OWL representations);
- actual (re)use of the WordNet and MeSH vocabularies by other parties (details follow);
- an analysis of the requirements placed on vocabulary representations by alignment tools, and identification of vocabulary features ignored by a state-of-the-art alignment tool (Chapter 6);
- an analysis of how the VRA Core Categories can be represented in RDF/OWL such that it may be used together with vocabularies in RDF/OWL (Chapter 7).

This chapter is structured as follows. Firstly, we revisit each separate Research Question. We summarize the approach and results, and provide conclusions. Secondly, we provide a discussion of the results and point out future research.

8.1 Research Questions Revisited

In this section we revisit the research questions raised in Chapter 1. Each question is considered in turn. We discuss how the research question was approached, concrete results and conclusions.

How can vocabularies be converted to an interoperable representation in an application-neutral way?

This question was addressed in Chapters 2 and 3. To answer this question we developed two methods that allow one to convert vocabularies. The approach to application neutrality is different in the two chapters. In the former, applications are not considered explicitly at all. The basis for conversion is the intentions of the original vocabulary developers, and how best to represent these intentions in RDF(S) and OWL. The underlying assumption is that converting the vocabulary by focusing on its content and semantics, independent of any application, produces a representation that is suitable for a wide variety of applications.

We first developed an initial method. Then we validated it with two case studies (MeSH and WordNet) of relatively complex vocabularies (i.e. they display a large number of vocabulary features). The cases showed that the method had to be extended with guidelines for dealing with n-ary relationships, references to external sources, creating URIs and avoiding redundancy. The case studies also showed that knowledge about the intentions of the vocabulary authors and conventions used is essential in understanding the meaning of the original digital format. For example, in the case of MeSH the XML format contained no explicit representation of the concept hierarchy. The hierarchy was encoded in the concept identifiers (see page 21).

This method has the benefit that all vocabulary features can be represented. However, it has as drawback that conversions produced by it might not be interoperable. Possible interoperation problems include: (1) a similar feature present in two vocabularies was represented in different ways; (2) the structure of the vocabularies are inherently incompatible. An example of the former problem is representing a term as either an instance of a class or as a literal attached to the concept through a datatype property. An example of the latter problem is reflected in the structures of MeSH and GTAA. The former has a three-layered structure: instances of classes Descriptor, Concept and Term (see page 39). The latter has a one-layered structure: instances of Concept and terms attached to them through a datatype property (see page 38). The vocabulary's metamodels are incompatible in the sense that one cannot be seen as equivalent or strict specialization of the other metamodel. Interoperation of these vocabularies would require rule-based mappings that collapse the Descriptor-Concept-Term structure in MeSH onto the Concept-term structure of GTAA.

The SKOS standard was developed in response to especially the first issue; it aims to promote interoperability and reuse of vocabularies by providing a standard way to express vocabularies and their features. Because SKOS helps promote interoperability (and thus is helpful for using vocabularies in Semantic Web applications) we developed a method specifically aimed at the SKOS metamodel. SKOS is not entirely application-neutral; it seems to aim mostly at subject indexing and retrieval of documents and images. For example, the requirements of lexically-oriented vocabularies such as WordNet were not addressed (until the SKOS XL extension was introduced).

Our analysis shows that SKOS is able to support vocabularies which have a simple structure such as GTAA. GTAA defines two types of associative relationships which could be accommodated with specializations of skos:related. The only non-standard feature "concept categories" could be interpreted as SKOS Collections (see page 38). Specializations were also necessary in

the IPSV case (specializations of skos:related, see page 36) and MeSH case (support for compound concepts, see page 41). The IPSV and MeSH cases showed that some features require an extension outside SKOS (e.g. to record the year in which a MeSH Descriptor was active, page 41). IPSV and MeSH store information about terms, which cannot be supported with specialization or a simple extension. At the time we proposed to the SKOS community to change the range of skos:prefLabel/skos:altLabel to a class Term[1] , but some thought this would complicate the standard too much for too little gain. A compromise was found in 2009: terms can be represented both as literals (through skos:prefLabel or skos:altLabel) and as instance of a class (skosxl:Label[2]). Instances of this class are attached to skos:Concept through skosxl:altLabel or skosxl:prefLabel. With the development of this extension it becomes possible to represent IPSV Terms and MeSH Concepts as instances, so that properties can be attached to them (see also the representation of WordNet in Section 4.9).

In conclusion, we feel that the usage of a standard — in particular SKOS— is to be preferred over development of a separate vocabulary metamodel. Usage of SKOS promotes interoperable conversions without sacrificing the semantics of most vocabularies. Some vocabularies may contain information that does not fit into SKOS, but this may be solved with an extension. Only when this is not possible should a tailored vocabulary metamodel be developed.

How can vocabularies be converted to an interoperable representation with given application constraints?

This research question was addressed in Chapters 4, 5 and 6. We approached this question by investigating if we can adapt our generic method introduced in Chapter 2 so that it also produces application-centric conversions. During the application-centric case studies we did, we found that we had to add new steps and guidelines. For example, a step was added to assess use cases and application requirements (pages 46, 48), and a step was added to handle publication of vocabularies on the Web (page 67). Guidelines were added to handle n-ary relations (page 89), and an existing guideline advocating the use of blank nodes for vocabulary concepts was changed to recommend the use of URIs instead (page 55).

All these additions and changes are useful for both generic as well as application-centric conversions. For example, it is useful to consider which use cases need to be supported, even if the conversion does not target a particular application. It is not the step or guideline which changes with the type of conversion, but how it is applied. We did identify one category of decisions which require different guidelines in application-oriented conversions. These are decisions about which content to include and in what form. For example, a generic conversion would convert the n-ary relations in the Getty vocabularies by using guideline 25 (a lossless conversion using an instance to represent the relation, see page 89). A complete conversion results in too many triples for usage in the E-Culture application, so n-ary relations need to be simplified by dropping arguments (guideline 26, page 89). We conclude that the generic method can also be used for application-centric conversion, but that it needs to be supplemented with more guidelines to guide the trade-off

[1] http://lists.w3.org/Archives/Public/public-esw-thes/2005Nov/0000.html
[2] http://www.w3.org/TR/2009/CR-skos-reference-20090317/#xl

between completeness and size. Defining these guidelines is future work.

During our discussions with the E-Culture project team we found that the vocabulary conversions had to be redone several times. Usually the reason was that new application requirements necessitated the inclusion of information that was originally left out, or because a different vocabulary structure was needed. Each time it was necessary to go back to the original digital format and redo the conceptual analysis and conversion. This can be prevented by doing a complete conversion to RDF/OWL the first time the vocabulary is converted. Any number of application-specific vocabulary representations can then be generated from the complete RDF/OWL.

In Chapter 6 we focused on a particular type of application, namely alignment tools. Alignments between vocabularies allow them to be used together. For example, the E-Culture application's graph selection algorithm can only relate artists that painted in the expressionist style if concepts denoting expressionism in two vocabularies are mapped to each other. Alignment applications such as Falcon-AO only accept OWL ontologies as input. This problem cannot be solved by an interpretation advocated in Step 2b of our method, because Falcon-AO cannot understand rdfs:subClassOf and rdfs:subPropertyOf statements. A "pure OWL file" needs to be created (page 105). Our method's step 2b needs to be extended with this guideline.

We found that Falcon-AO does not recognize vocabulary features such as qualifiers and guide terms, although this can be advantageous in alignment (page 116). The tool should be able to either identify them in the vocabulary representation, or understand an explicit representation of guide terms/qualifiers. We discussed different options for representing them explicitly (e.g. aat:GuideTerm, page 117). Each solution has consequences for other applications such as hierarchical browsers and autocomplete fields. We recommend to follow the SKOS standard, and if necessary create another representation as an add-on in Step 2b.

The context of Chapter 6 is a study that proposes new vocabulary alignment evaluation techniques. We found that in some cases a frequency-based evaluation gives a better impression of the quality of an alignment than an unweighted strategy. Estimating frequency of usage of a mapping is also useful in prioritizing manual checking of correspondences. Evaluation based on semantic distance for the task of document retrieval is useful when not only highly relevant results are desired.

How can metadata schemas be represented such that they can be combined with vocabularies?

This research question was addressed in Chapter 7. The motivation is that to represent collections on the Semantic Web a representation of the collection's metadata schema is also necessary. The metadata schema determines which aspects of the objects can be described. We answered this question for a specific case study: the E-Culture project.

The E-Culture project indicated three main requirements. Firstly, a representation in RDF/OWL of the VRA Core Categories was necessary. This is the domain-specific schema that the E-Culture project intended to use as its basic schema for collection metadata. All individual collection's schemas should ideally be a specialization of VRA Core Categories. In response to this requirement we created a basic VRA Core RDF/OWL representation by interpreting each element

as a property (page 124). The second requirement was that VRA Core should be a specialization of Dublin Core. This should be the case for their schemas, but the VRA Core specification should also fit into the framework that Dublin Core has developed for describing Dublin Core specializations. We were able to specify the VRA Core schema as a specialization of the Dublin Core schema, and establish that the VRA Core specification indeed fits into the Dublin Core framework (page 131).

However, an analysis of the subproperty hierarchy with the OntoClean methodology reveals problems. Some Dublin Core properties cannot have an identity criterium that is compatible with its VRA Core subproperties (page 132). If all the problematic subproperty relations are removed, then many VRA Core properties do not have a counterpart in Dublin Core. In that case we cannot consider VRA Core a specialization of Dublin Core. We feel that this conclusion is too drastic. Although the subproperty relations are not strictly correct, they represent a grouping of metadata elements that is useful for display purposes (page 137). We recommend to keep the subproperty relations. The use of Dublin Core is limited to these types of display tasks. Dublin Core does not help in more complex tasks. For example, Dublin Core cannot support precise queries such as "show me all paintings restored (not completed or begun) in 1917".

A last requirement imposed by the case study is that it should be possible to indicate for each individual collection which vocabularies are used. One museum may use vocabulary A to index the art style metadata element, while another uses vocabulary B. We term this "collection-specific value ranges" of metadata elements. It should also be possible to indicate for each metadata element which (part of a) vocabulary is used. We developed a pattern that enables such a representation through the use of OWL restrictions (page 130). The pattern can be used in conjunction with vocabulary representations as produced by our methods. The use of OWL restrictions allows to indicate value ranges without forcing other collections to use the same vocabularies for the metadata elements.

A problem in specifying a metadata schema that is compatible with OWL DL is that OWL DL enforces that each property is either an owl:DatatypeProperty or owl:ObjectProperty. It is not possible to make that choice (for either literal values or object values) when designing a metadata schema, because particular collections may have made the opposite choice (page 126). This is an overcommitment in the language that obstructs integration of metadata collections. We feel that this makes OWL DL unsuitable for data integration in the cultural heritage domain.

8.2 Discussion and Future Research

We structure this section along the following themes: (1) scope and validity of our methods; (2) versioning; (3) the tension between standardized vocabulary representations and expressivity; (4) vocabularies vs. ontologies; (5) the role of RDF and OWL in expressing vocabularies.

Scope and validity

This thesis has proposed three methods for converting vocabularies to an interoperable representation. Now that the methods have been developed, the question of their scope and validity –

i.e. the degree to which they are suitable and adequate for their purpose – becomes relevant. Methodological research is often qualitative in nature, and therefore hard to validate with the use of statistics or formal logic. In this thesis we have taken the approach of iterative case studies instead. Our generic method was validated by case studies concerning WordNet, MeSH (Chapter 2), and again WordNet (Chapter 4). It is also partially validated by the case studies for the other two methods (the three Getty vocabularies and GTAA and IPSV), as the analysis part of the methods largely overlap. The application-specific method was validated with WordNet in Chapter 4), and the Getty vocabularies in combination with the MultimediaN E-Culture portal. The SKOS method was validated with GTAA, IPSV and MeSH.

We argue that the set of vocabularies we chose is reasonably representative, and therefore our methods have a wide scope. Firstly, we chose vocabularies that together cover almost all vocabulary features mentioned in the ISO and ANSI/NISO standards for thesauri. Secondly, we chose a few complex vocabularies so that particular advanced features are also covered (WordNet, MeSH). Thirdly, vocabularies such as AAT, MeSH and WordNet are widely known and used. A method that covers those vocabularies thus also covers vocabulary features that are considered useful in practice. Fourthly, we chose vocabularies from various domains, of various specificity (e.g. WordNet is very general, AAT specific), of size, and of structural complexity (e.g. GTAA is simple, WordNet is complex). Fifthly, our method for SKOS covers a wide range of vocabularies because the SKOS community designed SKOS according to a set of representative use cases it gathered (Semantic Web Deployment Working Group 2007). Sixthly, as we performed new cases, the core of the method stayed stable. Adaptations only had to be made to incorporate more advanced features and application requirements.

Evidence for validity of the methods comes from uptake, in particular reuse of our WordNet and MeSH conversions. The Linked Open Data (LOD) project is aimed at making available online as many publicly available vocabularies and datasets as possible. The goal is to serve them in RDF in accordance with the Recipes, and align them. DBPedia[3] (an RDF conversion of Wikipedia) has a central role as a "hub" because of its wide subject scope. The LOD project is rapidly becoming one of the most important showcases of Semantic Web technology.[4] The creators of DBPedia have taken steps to align DBPedia with WordNet: WordNet is heavily used to type instances in DBPedia. For example, the synset Airline is linked to all DBPedia resources that represent an airline. The 338,061 links to DBPedia places it only second to the Flickr dataset in terms of interlinks with DBPedia (Hausenblas et al. 2008). OpenCyc is another vocabulary in the "LOD cloud". The OpenCyc project has added owl:sameAs links between its concepts and Wordnet synsets, e.g. for Airline.[5] The OpenGUID project maintains a list of identifiers which it links to several other datasets such as WordNet and OpenCyc (see e.g. Airline[6]). In summary, Linked Open Data plays a central role in the Semantic Web, and WordNet is a central node in the LOD

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

[4] For example, Tim Berners-Lee has termed it "The Semantic Web done right". See http://www.w3.org/2008/Talks/0617-lod-tbl/. Another example is the DBPedia Mobile application, which showcases how linked open data from several sources can be used together. It won the 2nd place in 2008's Semantic Web Challenge.

[5] http://sw.opencyc.org/2008/06/10/concept/en/AirlineCompany

[6] http://openguid.net/e694074f-da25-102b-9a03-2db401e887ec

network. An example of reuse of WordNet that is unrelated to LOD is the Topic Map conversion[7] of W3C's WordNet.

Reuse of MeSH occurs in the Health Care and Life Sciences (HCLS) Knowledge Base. It contains a conversion of MeSH in SKOS which is a minor modification of the one presented in this thesis.[8] The W3C's HCLS Interest Group[9] brings together a variety of Semantic Web scientists and specialists in the HCLS field (Ruttenberg et al. 2007). The motivation is the need to link data from different sources: (a) from drug laboratories to clinics; (b) from biological phenomena such as cells and organs to research results about these phenomena; and (c) from electronic patient records to clinical research. The HCLS Knowledge Base brings together vocabularies that can bridge gaps between these sources. Thus the incorporation of MeSH in the HCLS KB potentially has a high impact on the integration of HCLS data. The MeSH-SKOS conversion was also used as the basic vocabulary for the HealthFinland portal.[10]

Our methods' guidelines do not cover all aspects of conversion and interpretation of vocabularies. In order to perform Step 0 correctly, knowledge on requirement engineering is necessary. Step 1a requires experience in analysing source files that cannot easily be captured in guidelines. Ambiguities can sometimes only be solved by contacting the creators of the vocabulary. Similar skills are necessary in Step 1b and Step 2a: the conceptual model should be well understood in order to correctly interpret the digital model and make its semantics more explicit. Step 2b focuses on reinterpreting a vocabulary. Depending on the goal this may entail complete reengineering of a vocabulary, which requires a background in ontology engineering. Step 3 requires knowledge about the relationship between the conceptual model of the vocabulary and the schema of the standard, while Step 4 requires knowledge on how to configure a web server.

Versioning

Our methods help address the more general problem of how to publish data on the Semantic Web. One of the issues that we did not cover completely is that of versioning. In our WordNet conversion we included the version in the URI as we expect that multiple versions will be published. This allows distinguishing older from newer versions, which we feel is a minimal requirement in versioning. On the other hand, publishing separate vocabulary versions with version-specific URIs has its own drawbacks. It requires more resources to keep each version online, and application developers need to change all URIs in their code as soon as a new version is published. If an old version becomes unavailable, the application may crash, and the developer will have to find out where the new version is published. Leading projects in the Semantic Web-Cultural Heritage community such as MultimediaN E-Culture and MuseumFinland have not yet described their practices concerning versioning (they even do not publish their vocabularies as described in the Recipes). The LOD project also does not describe its version approach. As far as we can tell, new versions of e.g. DBPedia concepts are served at the same URIs as concepts of the old version (i.e. the old

[7] http://www.wandora.org/wandora/wiki/index.php?title=Topic_map_conversion_of_WordNet

[8] http://neurocommons.org/page/Bundles/mesh/mesh-skos

[9] http://www.w3.org/2001/sw/hcls/

[10] http://demo.seco.tkk.fi/tervesuomi/home

version is deleted).[11] We feel that the mentioned projects should be more proactive on the issue of versioning, as they fulfill central roles in the development of their communities.

Standardization vs. expressivity

The first two methods developed in this thesis have different aims. The first aims at conversion to a vocabulary representation that is as expressive as possible: there are no limitations except those imposed by RDF/OWL. The goal is that intended semantics should be preserved as much as possible. The second method aims at integration through the use of the SKOS standard. As we have seen in Chapter 3, a representation in SKOS may sacrifice some of the original vocabulary features. So on the one hand, integration is fostered by use of a standard, but at the same time limits the expressivity of vocabularies in RDF/OWL. Similar issues apply for standards such as Dublin Core (e.g. loss of specificity when converting object relations to DC's Relation element).

It appears that one has to choose between two alternatives. The first option is to represent a vocabulary according to a standard and accept information loss (the SKOS method). The second option is to develop an expressive, complete representation and accept integration problems (the generic method). However, this is a false dichotomy. RDF/OWL allows conversion of vocabularies as a specialization of a standard, and also the use of extensions to represent information that does not fit into the standard. Application developers can choose to invest in infrastructure that understands the specializations and extensions if necessary. For example, mesh:dateCreated is a specialization of skos:historyNote (page 41). Extensions can be created by linking new properties or classes to existing classes in the standard, e.g. skos:Concept. For example, mesh:descriptorClass and other attributes of MeSH Descriptors (page 41) can be attached to skos:Concept, so that no information is lost during conversion. In some cases extensions can even be standardized, such as SKOS XL. This allows representation of vocabulary terms, to which vocabulary-specific attributes such as AddedInVersion in IPSV (page 36) and `<LexicalTag>` in MeSH (page 40) can be attached. The representation of a term is now standardized so that it does not cause incompatibilities between vocabulary metamodels.

Vocabularies vs. ontologies

The amount of semantics found in schemas on the Semantic Web is diverse. While some researchers build rich, rigorously formal ontologies (e.g. OpenCyc, SUMO) others create lightweight class hierarchies. According to surveys such as by d'Aquin et al. (2007), the latter camp is dominant in terms of size.[12] It is also dominant in terms of applications: the vast majority of the winners of the Semantic Web Challenge use little more than a simple concept hierarchy. Vocabularies are mostly perceived as belonging in the "lightweight" camp, but we feel this is a wrong conclusion.

Firstly, this viewpoint ignores the fact that there exist rich vocabularies such as WordNet,

[11] RDF file dumps of old versions are available, see `http://wiki.dbpedia.org/Downloads32`

[12] Ontologies on the Web contain a small set of classes (43 on average). For each class in each ontology the P-density and H-density was measured. The former is the number of properties attached to a class, the latter the number of classes above a class in the subclass hierarchy. The maximum of these measures was calculated for each ontology, and then averaged over all ontologies. The average maximum P-density was 1.1, the average maximum H-density was 1.2.

IconClass[13] , FrameNet[14] and UMLS.[15] Although their classification hierarchy may not always conform to ontological theory, their concepts contain many links to other concepts, using a large set of property types. In fact, we contend that these vocabularies should actually be classified as ontologies.

Secondly, this viewpoint is based on the opinion that *knowledge-rich* is equivalent to *formally specified*. We think the real problem is not that vocabularies are knowledge-poor, but that the knowledge is implicit. For example, the WordNet gloss "*A driver is the operator of a motor vehicle*" describes a relationship between the synset Driver and Motor vehicle that is not made explicit. Gangemi et al. (2003b) describes how this relationship ("co-participation in the event of driving") might be extracted. A combination of natural language processing and background knowledge can be used in the extraction process. Another example is Soergel et al. (2004) who show how to specialize generic "Related" relations between AGROVOC concepts into e.g. growsIn and treatedWith. The used background knowledge is a specification of relationships that may exist between types of concepts.

There is also knowledge contained in vocabularies which is hard to formalize. For example, aat:adirondack chairs are "*Angular armchairs with slatted seats and backs*", and their aat:meaning / usage overlaps with that of aat:Westport chairs (because they are "*similar in form [...] but constructed of boards [...] and heavier and more amply proportioned*"). Because this type of knowledge is hard to formalize, it is not always considered when assessing whether a resource is a mere vocabulary or a full-fledged ontology. We repeat that this is a wrong conclusion. The actual problem is in expressing and leveraging this knowledge.

To summarize, vocabularies contain "human semantics" which we must learn to make explicit, instead of casting aside these vocabularies as "lightweight" or "informal". Especially for domains where vocabularies have been in development for decades it would be an enormous waste of resources to create ontologies from scratch. Rather, we hope that the vocabulary and Semantic Web communities will be able learn from each other. The goal should be to work together to make vocabulary semantics more explicit. Good examples of vocabularies incorporating new insights exist (cf. WordNet 3.0 which now separates instances from concepts[16] ; the improvement of AGROVOC[17] using semantic technologies; the development of SNOMED[18] in Description Logics), but need to grow in number. On the other hand, in my own experience, vocabulary developers can be reluctant to change the structure of their vocabulary according to insights from the ontology community (e.g. replace generic RT relations with more specific relations such as partOf or producedBy). This reluctance is usually based on healthy *investment vs. gain* considerations. The Semantic Web and ontology communities should realize that for various theories and practices they promote (e.g. OntoClean, Linked Open Data) it is not always clear how they improve performance and user support. For example, there is no solid evidence that an OntoClean-ed the-

[13] http://www.iconclass.nl

[14] http://framenet.icsi.berkeley.edu/

[15] http://www.nlm.nih.gov/pubs/factsheets/umlssemn.html

[16] http://wordnet.princeton.edu/wordnet/man/wngloss.7WN.html

[17] http://www.fao.org/agrovoc/

[18] http://www.ihtsdo.org/snomed-ct/

saurus will improve retrieval in a library system. Therefore, the vocabulary developer's adagium *no innovations without clear applications* should be at the heart of developments in the field for the coming years.

The role of RDF and OWL in expressing vocabularies

A primary role of RDF/OWL in representing vocabularies is to make explicit as much of a vocabulary's semantics as possible. The representation of vocabularies such as IPSV, MeSH, and the Getty vocabularies require no OWL constructs. Also, most applications in the past Semantic Web Challenges[19] use vocabularies in RDFS, and limited forms of reasoning (a popular approach is to support owl:sameAs and owl:SymmetricProperty in addition to RDFS). This begs the question whether either just not many constructs are necessary to represent those vocabularies, or that language features are missing.

Some authors are of the former opinion and have proposed that small extensions of RDFS are enough to support most application needs. For example, Allemang and Hendler (2008) propose RDFS-Plus. The results of this thesis support this viewpoint. The representation of vocabularies such as IPSV, MeSH, and the Getty vocabularies require no OWL constructs. However, this thesis is biased towards the cultural heritage domain. In e.g. the biomedical field there is a clear need for OWL DL. It remains to be seen whether OWL and OWL DL will be used frequently or infrequently in the Semantic Web as a whole.

In the meantime, it appears likely that different languages (with differences in expressiveness) will be used, even within one application. At least we can observe an improvement over the situation of one or two decades ago, when representation languages such as KL-ONE, KIF and OCML were in use that differed in their syntax, paradigm (frames, description logic, predicate logic) and expressiveness (Corcho and Gómez-Pérez 2000). The rise of the Web and adoption of RDF (and with it the adoption of URIs and the triple model) offers a basis for representing knowledge. This successful combination has lowered some of the barriers to the sharing of knowledge.

[19] http://challenge.semanticweb.org

A

Overview of Methods

Table A.1 Overview of Application-Neutral Method.

Step 0: preparation Analyze conceptual model, digital model and relation between them (page 15).		
Example	Each <Subject> record in AAT's digital model represents a vocabulary concept in the conceptual model.	page 81

Step 1a: structure-preserving translation Translate digital model to RDF, closely reflecting original structure (page 16).		
GL 1, p. 16	Use a basic set of RDF(S) constructs for the structure-preserving translation.	
GL 2, p. 16	Use XML support for datatyping.	
GL 3, p. 16	Preserve original naming as much as possible.	replaced by GL 18, p. 55
GL 4, p. 16	Translate relations of arity three or more into structures with blank nodes.	replaced by GL 25, p. 89
GL 5, p. 16	Do not translate semantically irrelevant ordering information.	
GL 6, p. 16	Avoid redundant information.	replaced by GL 15, p. 54
GL 7, p. 17	Avoid interpretation.	
GL 9, p. 20	Give preference to the relation-as-arc approach over the relation-as-node approach.	
GL 10, p. 20	Create proxy classes for references to external resources if they are not available in RDF.	
GL 11, p. 20	Only create rdf:IDs based on identifiers in the original source.	replaced by GL 16, p. 54 and GL 17, p. 55
GL 12, p. 20	Use the simplest solution that preserves the intended semantics.	
GL 13, p. 52	Create separate "union classes" for RDFS processing.	
GL 14, p. 53	Factor the third argument of a relation into subproperty names.	
GL 15, p. 54	Avoid redundant information (Modified version of Guideline 6, page 16)	Example modified
GL 16, p. 54	Consider storing the original IDs.	Replaces part of Guideline 11, page 20
GL 17, p. 55	Define URIs for all major entities	Replaces part of Guideline 11, page 20
GL 18, p. 55	Create URIs that are descriptive and persistent.	
GL 19, p. 57	Consider including a version-specific component in the URI.	

Step 1b: explication of syntax Explicate information that is implicit in the source format, using the same RDF(S) constructs as in step 1a (page 17).		
Example	Guide terms in AAT format only indicated with <> around term itself, not separately modelled. Make them instance of new class.	page 17

Table A.1 Overview of Application-Neutral Method.

GL 20, p. 58	Introduce superproperties to group properties with similar semantics.	
Example	Make the implicit hierarchy encoded in concept IDs explicit.	page 21

Step 2a: explication of semantics Use expressive RDFS and OWL constructs to further define the semantics (page 17).		
Example	Make descriptorName a subproperty of rdfs:label	page 21
Example	Use owl:unionOf to explicate the relationship between wn:NounSynset, wn:VerbSynset and NounOrVerbSynset	

Step 2b: interpretations Introduce interpretations not sanctioned by original model, but useful for an application (page 18).		
GL 8, p. 18	Consider treating the thesaurus schema as a metamodel.	
Example	Restore implicit links between ULAN birthplaces and TGN places.	page 95

Step 3: standardization Map vocabulary schema to a standardized vocabulary schema (page 18.		
Example	Make mesh:subTreeOf a subproperty of skos:broader	page 22
GL 21, p. 65	Create a new namespace for the SKOS representation if additional triples might be added that only hold for the SKOS representation.	

Step 4: Publishing on the Web Publish the vocabulary in a way that makes its URIs dereferencable. Added in Chapter 4		
GL 22, p. 68	Use slash URIs and 303 redirects when publishing large vocabularies on the Web.	
GL 23, p. 69	Implement Recipe four, five or six for publishing vocabularies.	
GL 24, p. 72	Consider creating a Full and Basic version.	

Table A.2 Overview of SKOS Method.

Step 1a: analyze thesaurus Compare thesaurus against a set of common thesaurus features (page 34).		
Example	GTAA is a term-based thesaurus with one non-standard feature called term “Categories”.	page 38
Example	IPSV is a concept-based thesaurus with pref/non-pref terms. It has non-standard features including a “broader default” relation.	page 35
Activity	Find identifiers in source that can be used to create URIs for concepts.	page 34

Step 1b: map data items to SKOS Create a table with three columns: (1) each type of data item in the input; (2) its feature/function; (3) mapping to properties/classes (page 34)		
Example	Table 3.2 provides an example for IPSV vocabulary.	page 36

Step 1c: create conversion program Develop a program that applies the mapping from the previous step to each input item (page 34)		
Example	GTAA conversion program had to take care of erroneous references, e.g. due to spelling mistakes or obsolete terms. Points out problems in GTAA maintenance.	page 39

Step 2: error checking and validation Not studied in this thesis; some observations are made in Section 4.7 and page 39

Table A.2 Overview of SKOS Method.

Step 3: publish encoding on the web Not treated in Chapter 3, but studied in detail in Section 4.10. The step described there can be incorporated here.

Table A.3 Overview of Application-Specific Method.

Step 0a: Case Study Description Make a description of the application, the use cases, application requirements, vocabulary requirements.		
Example	Use case: time-based search. Application should support structured queries such as "Picasso in late period of life"	page 79
Example	Application requirement: user can switch language of interface and data.	page 80
Example	Vocabulary requirement: all language information should be converted.	page 80

Step 0b: Digital and Conceptual Model Analyze conceptual model, digital model and relation between them.		
Example	The digital model makes no distinction between relations in AAT and TGN, but the former is a subclass relation, the latter a partitive relation.	page 94

Step 1a: Structural translation Similar to generic method, but some information/structure may be removed/adapted. Same guidelines may be used.		
GL 25, p. 89	Use the relation instance pattern (RI) for lossless conversion of n-ary relations.	replaces GL 4, p. 16
GL 26, p. 89	Use the reduction pattern (RED) to simplify n-ary relations.	

Step 1b: explication of syntax Similar to generic method, but some information/structure may be removed/adapted. Same guidelines may be used.

Step 2a: explication of semantics Similar to generic method, but some information/structure may be removed/adapted. Same guidelines may be used.

Step 2b: interpretation Similar to generic method. Same guidelines may be used.		
Example	Restore implicit links between ULAN birthplaces and TGN places.	page 95

Appendix B

Original Getty XML Records

The original XML for the example records presented in Chapter 5. Some line feeds and spaces have been added to improve readability. Of the ULAN record for "Rembrandt" only a small but representative portion is included for reasons of space.

B.1 AAT XML Record for "farms"

```
<Subject Subject_ID="300000206">
 <Merged_Status>Not Merged</Merged_Status>
 <Facet_Code>V.RG</Facet_Code>
 <Sort_Order>1</Sort_Order>
 <Record_Type>Concept</Record_Type>
 <Legacy_ID>206</Legacy_ID>
 <Descriptive_Note>
 <Note_Text>Complexes where plants or animals are raised
      for livelihood or commerce.</Note_Text>
 <Note_Contributors>
 <Note_Contributor>
 <Contributor_id>10000000/VP</Contributor_id>
 </Note_Contributor>
 </Note_Contributors>
 </Descriptive_Note>
 <Associative_Relationships>
 <Associative_Relationship>
 <Historic_Flag>Current</Historic_Flag>
 <Relationship_Type>2000/related to</Relationship_Type>
 <Related_Subject_ID>
 <VP_Subject_ID>300192802</VP_Subject_ID>
 </Related_Subject_ID>
 </Associative_Relationship>
 </Associative_Relationships>
 <Hierarchy><agricultural complexes> | <complexes by
      function> | complexes |
            Built Complexes and Districts | Built
                 Environment | Objects Facet</Hierarchy>
 <Parent_Relationships>
 <Preferred_Parent>
 <Parent_Subject_ID>300125766</Parent_Subject_ID>
 <Relationship_Type>Parent/Child</Relationship_Type>
 <Historic_Flag>Current</Historic_Flag>
 </Preferred_Parent>
 </Parent_Relationships>
 <Revision_History>
 <Revision>
 <Revision_ID>5000043934</Revision_ID>
 <Aspect>subject</Aspect>
 <Action>created</Action>
 <User_Name>AATLOAD</User_Name>
 <Date>1988-01-01 00:00:00</Date>
 </Revision>
 <Revision>
 <Revision_ID>5000043937</Revision_ID>
 <Aspect>Associative Relationships</Aspect>
 <Action>added</Action>
 <User_Name>ACHIPMAN</User_Name>
 <Date>1991-05-29 00:00:00</Date>
 </Revision>
 <Revision>
 <Revision_ID>5000043936</Revision_ID>
 <Aspect>scope note</Aspect>
 <Action>added</Action>
 <User_Name>DSANDERS</User_Name>
 <Date>1991-09-25 00:00:00</Date>
 </Revision>
 <Revision>
 <Revision_ID>5000043935</Revision_ID>
 <Aspect>Term</Aspect>
 <Action>added</Action>
 <User_Name>AAT</User_Name>
 <Date>1993-03-22 00:00:00</Date>
 </Revision>
 <Revision>
 <Revision_ID>5001113330</Revision_ID>
 <Aspect>Term</Aspect>
 <Action>added</Action>
 <User_Name>SYSADM</User_Name>
 <Date>2001-07-26 22:02:08</Date>
 <Note>complexes, farm (1000242074);</Note>
 </Revision>
 <Revision>
 <Revision_ID>5001113329</Revision_ID>
 <Aspect>Term</Aspect>
 <Action>added</Action>
 <User_Name>SYSADM</User_Name>
 <Date>2001-07-26 22:02:08</Date>
 <Note>farm (1000289951);</Note>
 </Revision>
 <Revision>
 <Revision_ID>5001113331</Revision_ID>
 <Aspect>Term</Aspect>
 <Action>added</Action>
 <User_Name>SYSADM</User_Name>
 <Date>2001-07-26 22:02:08</Date>
 <Note>farm complexes (1000078490);</Note>
 </Revision>
 <Revision>
 <Revision_ID>5001113328</Revision_ID>
 <Aspect>Term</Aspect>
 <Action>added</Action>
 <User_Name>SYSADM</User_Name>
 <Date>2001-07-26 22:02:08</Date>
 <Note>farms (1000000206);</Note>
 </Revision>
 <Revision>
```

```
<Revision_ID>5001113332</Revision_ID>
<Aspect>Term</Aspect>
<Action>added</Action>
<User_Name>SYSADM</User_Name>
<Date>2001-07-26 22:02:08</Date>
<Note>farmsteads (1000159944);</Note>
</Revision>
<Revision>
<Revision_ID>5001169921</Revision_ID>
<Aspect>Associative Relationships</Aspect>
<Action>added</Action>
<User_Name>SYSADM</User_Name>
<Date>2001-07-26 22:15:01</Date>
<Note>farms (300000206) 'related to' farming (300192802)
    ;</Note>
</Revision>
<Revision>
<Revision_ID>5001241970</Revision_ID>
<Aspect>Term</Aspect>
<Action>updated</Action>
<User_Name>SYSADM</User_Name>
<Date>2001-07-26 22:26:45</Date>
<Note>complexes, farm (1000242074);</Note>
</Revision>
<Revision>
<Revision_ID>5001241969</Revision_ID>
<Aspect>Term</Aspect>
<Action>updated</Action>
<User_Name>SYSADM</User_Name>
<Date>2001-07-26 22:26:45</Date>
<Note>farm (1000289951);</Note>
</Revision>
<Revision>
<Revision_ID>5001241971</Revision_ID>
<Aspect>Term</Aspect>
<Action>updated</Action>
<User_Name>SYSADM</User_Name>
<Date>2001-07-26 22:26:45</Date>
<Note>farm complexes (1000078490);</Note>
</Revision>
<Revision>
<Revision_ID>5001241968</Revision_ID>
<Aspect>Term</Aspect>
<Action>updated</Action>
<User_Name>SYSADM</User_Name>
<Date>2001-07-26 22:26:45</Date>
<Note>farms (1000000206);</Note>
</Revision>
<Revision>
<Revision_ID>5001241972</Revision_ID>
<Aspect>Term</Aspect>
<Action>updated</Action>
<User_Name>SYSADM</User_Name>
<Date>2001-07-26 22:26:45</Date>
<Note>farmsteads (1000159944);</Note>
</Revision>
<Revision>
<Revision_ID>5001375543</Revision_ID>
<Aspect>Term</Aspect>
<Action>updated</Action>
<User_Name>SYSADM</User_Name>
<Date>2001-08-10 11:39:06</Date>
<Note>complexes, farm (1000242074);</Note>
</Revision>
<Revision>
<Revision_ID>5001375542</Revision_ID>
<Aspect>Term</Aspect>
<Action>updated</Action>
<User_Name>SYSADM</User_Name>
<Date>2001-08-10 11:39:06</Date>
<Note>farm (1000289951);</Note>
</Revision>
<Revision>
<Revision_ID>5001375544</Revision_ID>
<Aspect>Term</Aspect>
<Action>updated</Action>
<User_Name>SYSADM</User_Name>
<Date>2001-08-10 11:39:06</Date>
<Note>farm complexes (1000078490);</Note>
</Revision>
<Revision>
<Revision_ID>5001375541</Revision_ID>
<Aspect>Term</Aspect>
<Action>updated</Action>
<User_Name>SYSADM</User_Name>
<Date>2001-08-10 11:39:06</Date>
<Note>farms (1000000206);</Note>
</Revision>
<Revision>
<Revision_ID>5001375545</Revision_ID>
<Aspect>Term</Aspect>
<Action>updated</Action>
<User_Name>SYSADM</User_Name>
<Date>2001-08-10 11:39:06</Date>
<Note>farmsteads (1000159944);</Note>
</Revision>
</Revision_History>
<Subject_Contributors>
<Subject_Contributor>
<Contributor_id>10000000/VP</Contributor_id>
</Subject_Contributor>
</Subject_Contributors>
<Terms>
<Preferred_Term>
<Term_Type>Descriptor</Term_Type>
<Term_Text>farms</Term_Text>
<Term_ID>1000000206</Term_ID>
<AACR2_Flag>N/A</AACR2_Flag>
<Display_Name>N/A</Display_Name>
<Display_Order>1</Display_Order>
<Historic_Flag>Current</Historic_Flag>
<Other_Flags>N/A</Other_Flags>
<Vernacular>Undetermined</Vernacular>
<Languages>
<Preferred_Language>70052/American English</
    Preferred_Language>
</Languages>
<Term_Contributors>
<Term_Contributor>
<Contributor_id>10000000/VP</Contributor_id>
<Preferred>Preferred</Preferred>
</Term_Contributor>
</Term_Contributors>
<Term_Sources>
<Term_Source>
<Source>
<Source_ID>2000000771/Adkins, Thesaurus of British
    Archaeology (1982)</Source_ID>
</Source>
<Page>IND</Page>
<Preferred>Preferred</Preferred>
</Term_Source>
<Term_Source>
<Source>
<Source_ID>2000046657/Avery Index (1963-)</Source_ID>
</Source>
<Page>(source AAT)</Page>
<Preferred>Preferred</Preferred>
</Term_Source>
<Term_Source>
<Source>
<Source_ID>2000024811/CDMARC Subjects: LCSH (1988-)</
    Source_ID>
</Source>
<Preferred>Preferred</Preferred>
</Term_Source>
<Term_Source>
<Source>
<Source_ID>2000039651/Canadian Thesaurus of Construction
    Science and Technology (1978)</Source_ID>
</Source>
<Preferred>Non Preferred</Preferred>
</Term_Source>
<Term_Source>
<Source>
<Source_ID>2000010621/Canadian Urban Thesaurus (1979)</
    Source_ID>
</Source>
<Preferred>Non Preferred</Preferred>
</Term_Source>
```

```
<Term_Source>
<Source>
<Source_ID>2000035251/RILA, Subject headings, unpub.
      (1975-1990)</Source_ID>
</Source>
<Preferred>Preferred</Preferred>
</Term_Source>
<Term_Source>
<Source>
<Source_ID>2000035571/ROOT Thesaurus (1981)</Source_ID>
</Source>
<Preferred>Preferred</Preferred>
</Term_Source>
<Term_Source>
<Source>
<Source_ID>2000008771/Roberts, Construction Industry
      Thesaurus , 2d ed. (1976)</Source_ID>
</Source>
<Preferred>Preferred</Preferred>
</Term_Source>
</Term_Sources>
</Preferred_Term>
<Non-Preferred_Term>
<Term_Type>Alternate Descriptor</Term_Type>
<Term_Text>farm</Term_Text>
<Term_ID>1000289951</Term_ID>
<AACR2_Flag>N/A</AACR2_Flag>
<Display_Name>N/A</Display_Name>
<Display_Order>2</Display_Order>
<Historic_Flag>Current</Historic_Flag>
<Other_Flags>N/A</Other_Flags>
<Vernacular>Undetermined</Vernacular>
<Languages>
<Non-Preferred_Language>70052/American English</Non-
      Preferred_Language>
</Languages>
<Term_Contributors>
<Term_Contributor>
<Contributor_id>10000000/VP</Contributor_id>
<Preferred>Non Preferred</Preferred>
</Term_Contributor>
</Term_Contributors>
<Term_Sources>
<Term_Source>
<Source>
<Source_ID>2000046717/Getty Vocabulary Program</
      Source_ID>
</Source>
<Preferred>Preferred</Preferred>
</Term_Source>
</Term_Sources>
</Non-Preferred_Term>
<Non-Preferred_Term>
<Term_Type>Used For Term</Term_Type>
<Term_Text>complexes, farm</Term_Text>
<Term_ID>1000242074</Term_ID>
<AACR2_Flag>N/A</AACR2_Flag>
<Display_Name>N/A</Display_Name>
<Display_Order>3</Display_Order>
<Historic_Flag>Current</Historic_Flag>
<Other_Flags>N/A</Other_Flags>
<Vernacular>Undetermined</Vernacular>
<Languages>
<Non-Preferred_Language>70052/American English</Non-
      Preferred_Language>
</Languages>
<Term_Contributors>
<Term_Contributor>
<Contributor_id>10000000/VP</Contributor_id>
<Preferred>Non Preferred</Preferred>
</Term_Contributor>
</Term_Contributors>
<Term_Sources>
<Term_Source>
<Source>
<Source_ID>2000046717/Getty Vocabulary Program</
      Source_ID>
</Source>
<Preferred>Non Preferred</Preferred>
</Term_Source>
</Term_Sources>
</Non-Preferred_Term>
<Non-Preferred_Term>
<Term_Type>Used For Term</Term_Type>
<Term_Text>farm complexes</Term_Text>
<Term_ID>1000078490</Term_ID>
<AACR2_Flag>N/A</AACR2_Flag>
<Display_Name>N/A</Display_Name>
<Display_Order>4</Display_Order>
<Historic_Flag>Current</Historic_Flag>
<Other_Flags>N/A</Other_Flags>
<Vernacular>Undetermined</Vernacular>
<Languages>
<Non-Preferred_Language>70052/American English</Non-
      Preferred_Language>
</Languages>
<Term_Contributors>
<Term_Contributor>
<Contributor_id>10000000/VP</Contributor_id>
<Preferred>Non Preferred</Preferred>
</Term_Contributor>
</Term_Contributors>
<Term_Sources>
<Term_Source>
<Source>
<Source_ID>2000046717/Getty Vocabulary Program</
      Source_ID>
</Source>
<Preferred>Non Preferred</Preferred>
</Term_Source>
</Term_Sources>
</Non-Preferred_Term>
<Non-Preferred_Term>
<Term_Type>Used For Term</Term_Type>
<Term_Text>farmsteads</Term_Text>
<Term_ID>1000159944</Term_ID>
<AACR2_Flag>N/A</AACR2_Flag>
<Display_Name>N/A</Display_Name>
<Display_Order>5</Display_Order>
<Historic_Flag>Current</Historic_Flag>
<Other_Flags>N/A</Other_Flags>
<Vernacular>Undetermined</Vernacular>
<Languages>
<Non-Preferred_Language>70052/American English</Non-
      Preferred_Language>
</Languages>
<Term_Contributors>
<Term_Contributor>
<Contributor_id>10000000/VP</Contributor_id>
<Preferred>Non Preferred</Preferred>
</Term_Contributor>
</Term_Contributors>
<Term_Sources>
<Term_Source>
<Source>
<Source_ID>2000046657/Avery Index (1963-)</Source_ID>
</Source>
<Page>(source AAT)</Page>
<Preferred>Non Preferred</Preferred>
</Term_Source>
<Term_Source>
<Source>
<Source_ID>2000010761/Harris, Dictionary of Architecture
      and Construction (1975)</Source_ID>
</Source>
<Preferred>Preferred</Preferred>
</Term_Source>
<Term_Source>
<Source>
<Source_ID>2000039001/Stilgoe, Common Landscape (1982)</
      Source_ID>
</Source>
<Page>149</Page>
<Preferred>Preferred</Preferred>
</Term_Source>
</Term_Sources>
</Non-Preferred_Term>
</Terms>
</Subject>
```

B.2 TGN XML Record for “Marakech”

```
<Subject Subject_ID="7000354">
<Merged_Status>Merged</Merged_Status>
<Sort_Order>1</Sort_Order>
<Record_Type>Administrative</Record_Type>
<Legacy_ID>7000354</Legacy_ID>
<Descriptive_Note>
<Note_Text>Located on fertile plain; one of Morocco&#39;s
      four imperial cities;
          founded in the 11th century as African capital
               of Almoravid dynasty;
     went to Almohads in 1147, then to Marinids; taken by
          French 1912;
     today noted for palace of sultan and several
          historic mosques.
</Note_Text>
<Note_Contributors>
<Note_Contributor>
<Contributor_id>10000000/VP</Contributor_id></
     Note_Contributor></Note_Contributors></
     Descriptive_Note>
<Coordinates>
<Standard>
<Latitude>
<Degrees>31</Degrees>
<Minutes>49</Minutes>
<Seconds>00</Seconds>
<Direction>North</Direction>
<Decimal>31.8167</Decimal></Latitude>
<Longitude>
<Degrees>008</Degrees>
<Minutes>00</Minutes>
<Seconds>00</Seconds>
<Direction>West</Direction>
<Decimal>-8</Decimal></Longitude></Standard>
</Coordinates>
<Hierarchy>Marrakech province | Al-Magreb | Africa |
     World</Hierarchy>
<Parent_Relationships>
<Preferred_Parent>
<Parent_Subject_ID>1001417</Parent_Subject_ID>
<Relationship_Type>Parent/Child</Relationship_Type>
<Historic_Flag>Current</Historic_Flag></Preferred_Parent>
</Parent_Relationships>
<Place_Types>
<Preferred_Place_Type>
<Place_Type_ID>83002/inhabited place</Place_Type_ID>
<Display_Order>1</Display_Order>
<Historic_Flag>Current</Historic_Flag>
<PT_Date>
<Display_Date>founded by Yusuf ibn-Tashfin in 1062</
     Display_Date>
<Start_Date>800</Start_Date>
<End_Date>9999</End_Date></PT_Date></Preferred_Place_Type
     >
<Non-Preferred_Place_Type>
<Place_Type_ID>83040/city</Place_Type_ID>
<Display_Order>2</Display_Order>
<Historic_Flag>Current</Historic_Flag>
<PT_Date>
<Display_Date>established 1912</Display_Date>
<Start_Date>1912</Start_Date>
<End_Date>9999</End_Date></PT_Date></Non-
     Preferred_Place_Type>
<Non-Preferred_Place_Type>
<Place_Type_ID>83115/provincial capital</Place_Type_ID>
<Display_Order>3</Display_Order>
<Historic_Flag>Current</Historic_Flag></Non-
     Preferred_Place_Type>
<Non-Preferred_Place_Type>
<Place_Type_ID>83324/commercial center</Place_Type_ID>
<Display_Order>4</Display_Order>
<Historic_Flag>Current</Historic_Flag></Non-
     Preferred_Place_Type>
<Non-Preferred_Place_Type>
<Place_Type_ID>83325/trade center</Place_Type_ID>
<Display_Order>5</Display_Order>
<Historic_Flag>Current</Historic_Flag></Non-
     Preferred_Place_Type>
<Non-Preferred_Place_Type>
<Place_Type_ID>83131/transportation center</Place_Type_ID
     >
<Display_Order>6</Display_Order>
<Historic_Flag>Current</Historic_Flag></Non-
     Preferred_Place_Type>
<Non-Preferred_Place_Type>
<Place_Type_ID>83150/religious center</Place_Type_ID>
<Display_Order>7</Display_Order>
<Historic_Flag>Current</Historic_Flag></Non-
     Preferred_Place_Type>
<Non-Preferred_Place_Type>
<Place_Type_ID>83351/tourist center</Place_Type_ID>
<Display_Order>8</Display_Order>
<Historic_Flag>Current</Historic_Flag></Non-
     Preferred_Place_Type>
<Non-Preferred_Place_Type>
<Place_Type_ID>83181/royal residence</Place_Type_ID>
<Display_Order>9</Display_Order>
<Historic_Flag>Historical</Historic_Flag></Non-
     Preferred_Place_Type>
<Non-Preferred_Place_Type>
<Place_Type_ID>83110/capital</Place_Type_ID>
<Display_Order>10</Display_Order>
<Historic_Flag>Historical</Historic_Flag>
<PT_Date>
<Display_Date>of Almoravid dynasty, until 1147; of
     Morocco, 1550-1660</Display_Date>
<Start_Date>1062</Start_Date>
<End_Date>1660</End_Date></PT_Date></Non-
     Preferred_Place_Type>
</Place_Types>
<Revision_History>
<Revision>
<Revision_ID>5000023848</Revision_ID>
<Aspect>subject</Aspect>
<Action>created</Action>
<User_Name>sysadm</User_Name>
<Date>1991-09-13 07:00:00</Date>
</Revision>
<Revision>
<Revision_ID>5000933358</Revision_ID>
<Aspect>subject</Aspect>
<Action>modified</Action>
<User_Name>laila</User_Name>
<Date>1997-04-09 11:43:00</Date>
</Revision>
<Revision>
<Revision_ID>5002319772</Revision_ID>
<Aspect>Term</Aspect>
<Action>updated</Action>
<User_Name>SYSADM</User_Name>
<Date>2001-10-31 00:30:57</Date>
<Note>Marakesh (181626);</Note>
</Revision>
<Revision>
<Revision_ID>5002319769</Revision_ID>
<Aspect>Term</Aspect>
<Action>updated</Action>
<User_Name>SYSADM</User_Name>
<Date>2001-10-31 00:30:57</Date>
<Note>Marrakech (92316);</Note>
</Revision>
<Revision>
<Revision_ID>5002319770</Revision_ID>
<Aspect>Term</Aspect>
<Action>updated</Action>
<User_Name>SYSADM</User_Name>
<Date>2001-10-31 00:30:57</Date>
<Note>Marrakesh (169061);</Note>
</Revision>
<Revision>
<Revision_ID>5002319771</Revision_ID>
<Aspect>Term</Aspect>
<Action>updated</Action>
```

```
<User_Name>SYSADM</User_Name>
<Date>2001-10-31 00:30:57</Date>
<Note>Morocco (181492);</Note>
</Revision>
<Revision>
<Revision_ID>5003415149</Revision_ID>
<Aspect>Term</Aspect>
<Action>updated</Action>
<User_Name>SYSADM</User_Name>
<Date>2001-10-31 01:04:45</Date>
<Note>Marakesh (181626);</Note>
</Revision>
<Revision>
<Revision_ID>5003415146</Revision_ID>
<Aspect>Term</Aspect>
<Action>updated</Action>
<User_Name>SYSADM</User_Name>
<Date>2001-10-31 01:04:45</Date>
<Note>Marrakech (92316);</Note>
</Revision>
<Revision>
<Revision_ID>5003415147</Revision_ID>
<Aspect>Term</Aspect>
<Action>updated</Action>
<User_Name>SYSADM</User_Name>
<Date>2001-10-31 01:04:45</Date>
<Note>Marrakesh (169061);</Note>
</Revision>
<Revision>
<Revision_ID>5003415148</Revision_ID>
<Aspect>Term</Aspect>
<Action>updated</Action>
<User_Name>SYSADM</User_Name>
<Date>2001-10-31 01:04:45</Date>
<Note>Morocco (181492);</Note>
</Revision>
<Revision>
<Revision_ID>5003481451</Revision_ID>
<Aspect>Term</Aspect>
<Action>updated</Action>
<User_Name>SYSADM</User_Name>
<Date>2002-04-23 20:34:20</Date>
<Note>Morocco (181492);</Note>
</Revision>
<Revision>
<Revision_ID>5004530480</Revision_ID>
<Aspect>Term</Aspect>
<Action>updated</Action>
<User_Name>SYSADM</User_Name>
<Date>2002-07-11 18:17:54</Date>
<Note>Marakesh (181626);</Note>
</Revision>
<Revision>
<Revision_ID>5004530477</Revision_ID>
<Aspect>Term</Aspect>
<Action>updated</Action>
<User_Name>SYSADM</User_Name>
<Date>2002-07-11 18:17:54</Date>
<Note>Marrakech (92316);</Note>
</Revision>
<Revision>
<Revision_ID>5004530478</Revision_ID>
<Aspect>Term</Aspect>
<Action>updated</Action>
<User_Name>SYSADM</User_Name>
<Date>2002-07-11 18:17:54</Date>
<Note>Marrakesh (169061);</Note>
</Revision>
<Revision>
<Revision_ID>5004530479</Revision_ID>
<Aspect>Term</Aspect>
<Action>updated</Action>
<User_Name>SYSADM</User_Name>
<Date>2002-07-11 18:17:54</Date>
<Note>Morocco (181492);</Note>
</Revision>
<Revision>
<Revision_ID>5004583922</Revision_ID>
<Aspect>subject</Aspect>
<Action>updated</Action>
<User_Name>JGOODELL</User_Name>
<Date>2002-10-24 14:16:08</Date>
</Revision>
</Revision_History>
<Subject_Contributors>
<Subject_Contributor>
<Contributor_id>10000000/VP</Contributor_id></
    Subject_Contributor>
</Subject_Contributors>
<Subject_Sources>
<Subject_Source>
<Source>
<Source_ID>9006447/Canby, Historic Places (1984)</
    Source_ID></Source>
<Page>I, 572</Page></Subject_Source>
<Subject_Source>
<Source>
<Source_ID>9006382/Encyclop$70aedia Britannica (1985)</
    Source_ID></Source>
<Page>VII, 870-871</Page></Subject_Source>
<Subject_Source>
<Source>
<Source_ID>9005014/Encyclop$70aedia Britannica (1988)</
    Source_ID></Source>
<Page>VII, 870-871</Page></Subject_Source>
<Subject_Source>
<Source>
<Source_ID>9006548/Times Atlas of World History (1993)</
    Source_ID></Source>
<Page>349</Page></Subject_Source>
<Subject_Source>
<Source>
<Source_ID>9006549/Times Atlas of the World (1994)</
    Source_ID></Source>
<Page>120</Page></Subject_Source>
<Subject_Source>
<Source>
<Source_ID>9006267/Webster&#39;s Geographical Dictionary
    (1988)</Source_ID></Source>
<Page>733</Page></Subject_Source>
</Subject_Sources>
<Terms>
<Preferred_Term>
<Term_Type>Noun</Term_Type>
<Term_Text>Marrakech</Term_Text>
<Term_ID>92316</Term_ID>
<AACR2_Flag>N/A</AACR2_Flag>
<Display_Name>N/A</Display_Name>
<Display_Order>1</Display_Order>
<Historic_Flag>Current</Historic_Flag>
<Other_Flags>N/A</Other_Flags>
<Vernacular>Vernacular</Vernacular>
<Term_Contributors>
<Term_Contributor>
<Contributor_id>10000000/VP</Contributor_id>
<Preferred>Non Preferred</Preferred></Term_Contributor></
    Term_Contributors>
<Term_Sources>
<Term_Source>
<Source>
<Source_ID>9006447/Canby, Historic Places (1984)</
    Source_ID></Source>
<Page>I, 572</Page>
<Preferred>Unknown</Preferred></Term_Source>
<Term_Source>
<Source>
<Source_ID>9006382/Encyclop$70aedia Britannica (1985)</
    Source_ID></Source>
<Page>VII, 870-871</Page>
<Preferred>Unknown</Preferred></Term_Source>
<Term_Source>
<Source>
<Source_ID>9006549/Times Atlas of the World (1994)</
    Source_ID></Source>
<Page>120</Page>
<Preferred>Unknown</Preferred></Term_Source></
    Term_Sources>
</Preferred_Term>
<Non-Preferred_Term>
<Term_Type>Noun</Term_Type>
```

```
<Term_Text>Marrakesh</Term_Text>
<Term_ID>169061</Term_ID>
<AACR2_Flag>N/A</AACR2_Flag>
<Display_Name>N/A</Display_Name>
<Display_Order>2</Display_Order>
<Historic_Flag>Current</Historic_Flag>
<Other_Flags>N/A</Other_Flags>
<Vernacular>Vernacular</Vernacular>
<Term_Contributors>
<Term_Contributor>
<Contributor_id>10000000/VP</Contributor_id>
<Preferred>Non Preferred</Preferred></Term_Contributor></
    Term_Contributors>
<Term_Sources>
<Term_Source>
<Source>
<Source_ID>9006548/Times Atlas of World History (1993)</
    Source_ID></Source>
<Page>349</Page>
<Preferred>Unknown</Preferred></Term_Source></
    Term_Sources>
</Non-Preferred_Term>
<Non-Preferred_Term>
<Term_Type>Noun</Term_Type>
<Term_Text>Marakesh</Term_Text>
<Term_ID>181626</Term_ID>
<AACR2_Flag>N/A</AACR2_Flag>
<Display_Name>N/A</Display_Name>
<Display_Order>3</Display_Order>
<Historic_Flag>Current</Historic_Flag>
<Other_Flags>N/A</Other_Flags>
<Vernacular>Vernacular</Vernacular>
<Languages>
<Preferred_Language>70051/English</Preferred_Language></
    Languages>
<Term_Contributors>
<Term_Contributor>
<Contributor_id>10000000/VP</Contributor_id>
<Preferred>Non Preferred</Preferred></Term_Contributor></
    Term_Contributors>
<Term_Sources>
<Term_Source>
<Source>
<Source_ID>9006267/Webster&#39;s Geographical Dictionary
    (1988)</Source_ID></Source>
<Page>733</Page>
<Preferred>Unknown</Preferred></Term_Source></
    Term_Sources>
</Non-Preferred_Term>
<Non-Preferred_Term>
<Term_Type>Noun</Term_Type>
<Term_Text>Morocco</Term_Text>
<Term_ID>181492</Term_ID>
<AACR2_Flag>N/A</AACR2_Flag>
<Display_Name>N/A</Display_Name>
<Display_Order>4</Display_Order>
<Historic_Flag>Historical</Historic_Flag>
<Other_Flags>N/A</Other_Flags>
<Vernacular>Vernacular</Vernacular>
<Term_Date>
<Display_Date>misnamed by Europeans, this name was also
    used for entire nation</Display_Date>
<Start_Date>1500</Start_Date>
<End_Date>9999</End_Date></Term_Date>
<Term_Contributors>
<Term_Contributor>
<Contributor_id>10000000/VP</Contributor_id>
<Preferred>Non Preferred</Preferred></Term_Contributor></
    Term_Contributors>
<Term_Sources>
<Term_Source>
<Source>
<Source_ID>9005014/Encyclop$70aedia Britannica (1988)</
    Source_ID></Source>
<Page>VII, 870-871</Page>
<Preferred>Unknown</Preferred></Term_Source>
<Term_Source>
<Source>
<Source_ID>9006267/Webster&#39;s Geographical Dictionary
    (1988)</Source_ID></Source>
<Page>733</Page>
<Preferred>Unknown</Preferred></Term_Source></
    Term_Sources>
</Non-Preferred_Term>
</Terms>
</Subject>
```

B.3 ULAN XML Record for "Rembrandt"

```
<Subject Subject_ID="500011051">
<Merged_Status>Merged</Merged_Status>
<Sort_Order>1</Sort_Order>
<Record_Type>Person</Record_Type>
<Legacy_ID>16023</Legacy_ID>
<Descriptive_Note>
<Note_Text>Rembrandt was one of the most popular and
    influential artists of his period. His work is
    characterized by the Baroque interest in dramatic
    scenes and
strong contrasts of light on a dark stage. The subjects
    of his works include
portraits, landscapes, figures, animals, an scenes of
    biblical and secular
history and mythology. He was very prolific, producing
    paintings, etchings,
and drawings: Rembrandt executed about 400 paintings,
    over 1000 drawings,
and around 300 etchings.
</Note_Text>
<Note_Contributors>
<Note_Contributor>
<Contributor_id>2500000013/VP</Contributor_id></
    Note_Contributor></Note_Contributors>
<Note_Sources>
<Note_Source>
<Source>
<Source_ID>2100042519/Grove Dictionary of Art online
    (1999-2002)</Source_ID></Source>
<Page>Accessed 07/18/2002.</Page></Note_Source></
    Note_Sources></Descriptive_Note>

<Associative_Relationships>

<Associative_Relationship>
<Historic_Flag>NA</Historic_Flag>
<Relationship_Type>2602/influenced</Relationship_Type>
<Related_Subject_ID>
<VP_Subject_ID>500027532</VP_Subject_ID></
    Related_Subject_ID></Associative_Relationship>
<Associative_Relationship>
<Historic_Flag>NA</Historic_Flag>
<Relationship_Type>1202/patron was</Relationship_Type>
<Related_Subject_ID>
<VP_Subject_ID>500010860</VP_Subject_ID></
    Related_Subject_ID></Associative_Relationship>
<Associative_Relationship>
<Historic_Flag>NA</Historic_Flag>
<Relationship_Type>1102/student of</Relationship_Type>
<Related_Subject_ID>
<VP_Subject_ID>500032894</VP_Subject_ID></
    Related_Subject_ID></Associative_Relationship>
<Associative_Relationship>
<Historic_Flag>NA</Historic_Flag>
<Relationship_Type>1101/teacher of</Relationship_Type>
<Related_Subject_ID>
```

```
<VP_Subject_ID>500031967</VP_Subject_ID></
      Related_Subject_ID></Associative_Relationship>
</Associative_Relationships>

<Biographies>

<Preferred_Biography>
<Biography_ID>4000028251</Biography_ID>
<Biography_Text>Dutch painter, draftsman and printmaker,
      1606-1669</Biography_Text>
<Birth_Place>4390330011/Leyden (South Holland,
      Netherlands)</Birth_Place>
<Birth_Date>1606</Birth_Date>
<Death_Place>4390000029/Amsterdam (North Holland,
      Netherlands)</Death_Place>
<Death_Date>1669</Death_Date>
<Sex>Male</Sex>
<Contributor>VP</Contributor></Preferred_Biography>
<Non-Preferred_Biography>
<Biography_ID>4000028256</Biography_ID>
<Biography_Text>Dutch artist, 1606-1669</Biography_Text>
<Contributor>AVERY</Contributor></Non-Preferred_Biography
      >
<Non-Preferred_Biography>
<Biography_ID>4000028254</Biography_ID>
<Biography_Text>Dutch painter, 1606-1669</Biography_Text>
<Contributor>PROV</Contributor></Non-Preferred_Biography>
</Biographies>

<Events>

<Preferred_Event>
<Event_ID>12002/active</Event_ID>
<Display_Order>1</Display_Order>
<Place>4390000029/Amsterdam (North Holland, Netherlands)<
      /Place>
<Event_Date>
<Display_Date>1631-1669</Display_Date>
<Start_Date>1631</Start_Date>
<End_Date>1669</End_Date></Event_Date></Preferred_Event>
</Events>

<Hierarchy>Person</Hierarchy>

<Nationalities>

<Preferred_Nationality>
<Nationality_Code>905020/Dutch</Nationality_Code>
<Display_Order>1</Display_Order></Preferred_Nationality>
</Nationalities>

<Parent_Relationships>

<Preferred_Parent>
<Parent_Subject_ID>500000002</Parent_Subject_ID>
<Relationship_Type>Parent/Child</Relationship_Type>
<Historic_Flag>Current</Historic_Flag></Preferred_Parent>
</Parent_Relationships>

<Revision_History>

<Revision>
<Revision_ID>5500030574</Revision_ID>
<Aspect>subject</Aspect>
<Action>created</Action>
<Date>1989-12-22 00:00:00</Date>
</Revision>

<Revision>
<Revision_ID>5501605444</Revision_ID>
<Aspect>Term</Aspect>
<Action>updated</Action>
<User_Name>PHARPRING</User_Name>
<Date>2003-02-04 11:16:11</Date>
<Note>Van Ryn, Paul Rembrandt (1500030915);</Note>
</Revision>

<Revision>
<Revision_ID>5501666982</Revision_ID>
<Aspect>Associative Relationships</Aspect>
<Action>added</Action>
<User_Name>JGOODELL</User_Name>
<Date>2003-07-14 14:21:43</Date>
<Note>Rembrandt van Rijn (500011051) &#39;patron was&#39;
      Uylenburgh, Hendrick (500010860);</Note>
</Revision>
</Revision_History>

<Roles>

<Preferred_Role>
<Role_ID>31100/artist</Role_ID>
<Display_Order>1</Display_Order>
<Historic_Flag>NA</Historic_Flag></Preferred_Role>
<Non-Preferred_Role>
<Role_ID>31175/draftsman</Role_ID>
<Display_Order>2</Display_Order>
<Historic_Flag>NA</Historic_Flag></Non-Preferred_Role>
<Non-Preferred_Role>
<Role_ID>31442/etcher</Role_ID>
<Display_Order>3</Display_Order>
<Historic_Flag>NA</Historic_Flag></Non-Preferred_Role>
<Non-Preferred_Role>
<Role_ID>31261/painter</Role_ID>
<Display_Order>4</Display_Order>
<Historic_Flag>NA</Historic_Flag></Non-Preferred_Role>
<Non-Preferred_Role>
<Role_ID>31437/printmaker</Role_ID>
<Display_Order>5</Display_Order>
<Historic_Flag>NA</Historic_Flag></Non-Preferred_Role>
<Non-Preferred_Role>
<Role_ID>40792/teacher</Role_ID>
<Display_Order>6</Display_Order>
<Historic_Flag>NA</Historic_Flag></Non-Preferred_Role>
</Roles>

<Terms>

<Preferred_Term>
<Term_Type>N/A</Term_Type>
<Term_Text>Rembrandt van Rijn</Term_Text>
<Term_ID>1500030898</Term_ID>
<AACR2_Flag>N/A</AACR2_Flag>
<Display_Name>Yes</Display_Name>
<Display_Order>1</Display_Order>
<Historic_Flag>N/A</Historic_Flag>
<Other_Flags>N/A</Other_Flags>
<Vernacular>Vernacular</Vernacular>
<Term_Date>
<Display_Date>&#34;Rijn&#34; refers to a geographic place
      , the site of the mill
            owned by his father in Leyden</
                  Display_Date>
<Start_Date>1606</Start_Date>
<End_Date>1669</End_Date></Term_Date>
<Term_Contributors>
<Term_Contributor>
<Contributor_id>2500000003/GRLPSC</Contributor_id>
<Preferred>Non Preferred</Preferred></Term_Contributor>
<Term_Contributor>
<Contributor_id>2500000009/JPGM</Contributor_id>
<Preferred>Preferred</Preferred></Term_Contributor>
<Term_Contributor>
<Contributor_id>2500000011/PROV</Contributor_id>
<Preferred>Non Preferred</Preferred></Term_Contributor>
<Term_Contributor>
<Contributor_id>2500000013/VP</Contributor_id>
<Preferred>Non Preferred</Preferred></Term_Contributor></
      Term_Contributors>
<Term_Sources>
<Term_Source>
<Source>
<Source_ID>2100039762/Gardner&#39;s Art Through the Ages
      (1996)</Source_ID></Source>
<Preferred>Preferred</Preferred></Term_Source>
<Term_Source>
<Source>
<Source_ID>2100000144/George Goldner</Source_ID></Source>
<Preferred>Unknown</Preferred></Term_Source>
<Term_Source>
```

```
<Source>
<Source_ID>2100042519/Grove Dictionary of Art online
      (1999-2002)</Source_ID></Source>
<Page>accessed 24 September 2002</Page>
<Preferred>Non Preferred</Preferred></Term_Source></
      Term_Sources>
</Preferred_Term>

<Non-Preferred_Term>
<Term_Type>N/A</Term_Type>
<Term_Text>Rijn, Rembrandt van</Term_Text>
<Term_ID>1500030912</Term_ID>
<AACR2_Flag>N/A</AACR2_Flag>
<Display_Name>N/A</Display_Name>
<Display_Order>2</Display_Order>
<Historic_Flag>N/A</Historic_Flag>
<Other_Flags>N/A</Other_Flags>
<Vernacular>Vernacular</Vernacular>
<Term_Contributors>
<Term_Contributor>
<Contributor_id>2500000011/PROV</Contributor_id>
<Preferred>Non Preferred</Preferred></Term_Contributor>
<Term_Contributor>
<Contributor_id>2500000013/VP</Contributor_id>
<Preferred>Non Preferred</Preferred></Term_Contributor></
      Term_Contributors>
<Term_Sources>
<Term_Source>
<Source>
<Source_ID>2100042247/B$00en$00ezit, Dictionnaire des
      Peintres (1976)</Source_ID></Source>
<Preferred>Non Preferred</Preferred></Term_Source>
<Term_Source>
<Source>
<Source_ID>2100039762/Gardner&#39;s Art Through the Ages
      (1996)</Source_ID></Source>
<Preferred>Non Preferred</Preferred></Term_Source>
<Term_Source>
<Source>
<Source_ID>2100042266/Getty Provenance Index Databases [
      online] (1999)</Source_ID></Source>
<Preferred>Non Preferred</Preferred></Term_Source>
<Term_Source>
<Source>
<Source_ID>2100042519/Grove Dictionary of Art online
      (1999-2002)</Source_ID></Source>
<Page>Accessed 07/18/2002.</Page>
<Preferred>Non Preferred</Preferred></Term_Source>
<Term_Source>
<Source>
<Source_ID>2100039763/Janson, History of Art, 3rd edition
      (1986)</Source_ID></Source>
<Preferred>Non Preferred</Preferred></Term_Source>
<Term_Source>
<Source>
<Source_ID>2100042219/Thieme-Becker, Allgemeines Lexikon
      der Kunstler (1980-1986)</Source_ID></Source>
<Preferred>Non Preferred</Preferred></Term_Source></
      Term_Sources>
</Non-Preferred_Term>

<Non-Preferred_Term>

<Term_Type>N/A</Term_Type>
<Term_Text>Rembrandt Harmensz. van Rijn</Term_Text>
<Term_ID>1500213094</Term_ID>
<AACR2_Flag>N/A</AACR2_Flag>
<Display_Name>N/A</Display_Name>
<Display_Order>3</Display_Order>
<Historic_Flag>N/A</Historic_Flag>
<Other_Flags>N/A</Other_Flags>
<Vernacular>Vernacular</Vernacular>
<Term_Contributors>
<Term_Contributor>
<Contributor_id>2500000013/VP</Contributor_id>
<Preferred>Non Preferred</Preferred></Term_Contributor></
      Term_Contributors>
<Term_Sources>
<Term_Source>
<Source>
<Source_ID>2100042519/Grove Dictionary of Art online
      (1999-2002)</Source_ID></Source>
<Page>Accessed 07/18/2002.</Page>
<Preferred>Non Preferred</Preferred></Term_Source></
      Term_Sources>
</Non-Preferred_Term>


</Terms>
</Subject>
```

Appendix C

VRA

C.1 VRA specification summary

Table C.1 VRA fields and their meaning. Range and mapping to DC are those stated in VRA documentation. Examples in parenthesis derived from VRA Core Categories specification.

VRA field	Range	Dublin Core	Meaning/example
Record Type	{work,image}	Type	type of record
Type	AAT	Type	“print”, “sculpture”, “digital”
Title	formatted text	Title	title or identifying phrase, “This is how it happened”
Title.Variant			“As Sucedi”
Title.Translation			in other language
Title.Series			name of series it is part of
Title.Larger Entity			name of work it is part of
Measurements	formatted text	Format	
Measurements.Dimensions			“24.5 x 35 cm”, “72 dpi”
Measurements.Format			“jpeg”
Measurements.Resolution			
Material	AAT	Format	Material object is made of
Material.Medium	AAT		“ink”
Material.Support	AAT		“paper”
Technique	AAT	Format	“etching”, “cabinet making”, “scanning”
Creator	ULAN, AAAF	Creator, Contributor	names, appellations, or other identifiers assigned to creator or contributor
Creator.Role	Controlled list		‘sculptor”
Creator.Attribution			
Creator.Personal name			“Wright, Frank L. (1867-1959)”
Creator.Corporate name			
Date	formatted text	Date, Coverage	“1985”, “5th century”, “ca. 1990”
Date.Creation			
Date.Design			
Date.Beginning			
Date.Completion			
Date.Alteration			
Date.Restoration			
Location	BHA, AAAF	Contributor, Coverage	geographic location and/or name of the repository, building, or site-specific work
Location.Current Site			
Location.Former Site			
Location.Creation Site			“Madrid (ESP)”
Location.Discovery Site			

Table C.1 VRA fields and their meaning. Range and mapping to DC are those stated in VRA documentation. Examples in parenthesis derived from VRA Core Categories specification.

VRA field	Range	Dublin Core	Meaning/example
Location.Current Repository Location.Former Repository			"Ann Arbor (MI,USA), University of Michigan Museum of Art"
ID Number.Current Repository ID Number.Former Repository ID Number.Current Accession ID Number.Former Accession		Identifier	The unique identifiers assigned to a Work or an Image
Style/Period Style/Period.Style Style/Period.Period Style/Period.Group Style/Period.School Style/Period.Dynasty Style/Period.Movement	AAT	Coverage, Subject	style, historical period, group, school, dynasty, movement, etc. whose characteristics are represented "Renaissance" "Vakataka dynasty"
Culture	AAT, LCSH	Coverage	"Indian"
Subject	AAT, TGM, other	Subject	Terms/phrases that describe the object and what it depicts or expresses
Relation Relation.Identity Relation.Type		Relation	relationship between the Work being catalogued and the related work
Description	text	Description	free-text note about the Work or Image, including comments, description, or interpretation
Source		Source	reference to the source of the information recorded
Rights		Rights	Information about rights management

C.2 VRA Schema

This is an RDF/Turtle version of `http://www.w3.org/2001/sw/BestPractices/MM/vracore3.rdfs` with some minor adjustments for readability.

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#
    > .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix vra: <http://www.vraweb.org/vracore3.htm#> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .


@base <http://www.vraweb.org/vracore/vracore3#> .

<http://www.vraweb.org/vracore/vracore3> rdf:type
     owl:Ontology ;
  rdfs:isDefinedBy <http://www.vraweb.org/vracore3.htm> ;
  rdfs:comment "OWL representation of the VRA element
  set (Visual Resource Association).
  The work described here was partly supported by the
  CHIME project, part of the NWO ToKeN programme.

  Mark van Assem, October 2005, for the MultimediaN
  project. mark@cs.vu.nl http://www.cs.vu.nl/~mark/" .

# Begin schema

<http://www.w3.org/2001/XMLSchema#string> rdf:type
     rdfs:Datatype .

# Additional statements to interpret the RDFS classes as
     OWL classes

<Work> a owl:Class .
<Image> a owl:Class .
<VisualResource> a owl:Class .

# Actual schema starts here

# Classes

<Work> a rdfs:Class ;
  rdfs:label "Work" ;
  rdfs:isDefinedBy <http://www.vraweb.org/vracore3.htm> ;
  rdfs:subClassOf <VisualResource> .

<Image> a rdfs:Class ;
```

```
   rdfs:label "Image" ;
   rdfs:subClassOf <VisualResource> ;
   rdfs:isDefinedBy <http://www.vraweb.org/vracore3.htm> .

<VisualResource> a rdfs:Class ;
   rdfs:label "VisualResource" ;
   rdfs:isDefinedBy <http://www.vraweb.org/vracore3.htm> .

# Properties

<idNumber> rdf:type owl:DatatypeProperty ,
   rdf:Property ;
   rdfs:label "ID Number" ;
   rdfs:isDefinedBy <http://www.vraweb.org/vracore3.htm#id
        %20number> ;
   rdfs:subPropertyOf dc:identifier ;
   rdfs:domain <VisualResource> ;
   rdfs:range <http://www.w3.org/2001/XMLSchema#string> .

<description> rdf:type owl:DatatypeProperty ,
   rdf:Property ;
   rdfs:label "Description" ;
   rdfs:isDefinedBy <http://www.vraweb.org/vracore3.htm#
        description> ;
   rdfs:subPropertyOf dc:description ;
   rdfs:domain <VisualResource> ;
   rdfs:range <http://www.w3.org/2001/XMLSchema#string> .

<type> a rdf:Property ;
   rdfs:label "Type" ;
   rdfs:isDefinedBy vra:type ;
   rdfs:subPropertyOf dc:type ;
   rdfs:domain <VisualResource> .

<title> a rdf:Property ;
   rdfs:label "Title" ;
   rdfs:isDefinedBy vra:title ;
   rdfs:subPropertyOf dc:title ;
   rdfs:domain <VisualResource> .

<title.variant> a rdf:Property ;
   rdfs:label "Title.Variant" ;
   rdfs:isDefinedBy vra:title ;
   rdfs:subPropertyOf <title> ;
   rdfs:domain <VisualResource> .

<title.translation> a rdf:Property ;
   rdfs:label "Title.Translation" ;
   rdfs:isDefinedBy vra:title ;
   rdfs:subPropertyOf <title> ;
   rdfs:domain <VisualResource> .

<title.series> a rdf:Property ;
   rdfs:label "Title.Series" ;
   rdfs:isDefinedBy vra:title ;
   rdfs:subPropertyOf <title> ;
   rdfs:domain <VisualResource> .

<title.largerEntity> a rdf:Property ;
   rdfs:label "Title.LargerEntity" ;
   rdfs:isDefinedBy vra:title ;
   rdfs:subPropertyOf <title> ;
   rdfs:domain <VisualResource> .

<measurements> a rdf:Property ;
   rdfs:label "Measurements" ;
   rdfs:isDefinedBy vra:format ;
   rdfs:subPropertyOf dc:format ;
   rdfs:domain <VisualResource> .

<measurements.dimensions> a rdf:Property ;
    rdfs:label "Measurements.Dimensions" ;
    rdfs:isDefinedBy vra:format ;
    rdfs:subPropertyOf <measurements> ;
    rdfs:domain <VisualResource> .

<measurements.format> a rdf:Property ;
    rdfs:label "Measurements.Format" ;
    rdfs:isDefinedBy vra:format ;
    rdfs:subPropertyOf <measurements> ;
    rdfs:domain <VisualResource> .

<measurements.resolution> a rdf:Property ;
    rdfs:label "Measurements.Resolution" ;
    rdfs:isDefinedBy vra:format ;
    rdfs:subPropertyOf <measurements> ;
    rdfs:domain <VisualResource> .

<material> a rdf:Property ;
   rdfs:label "Material" ;
   rdfs:isDefinedBy vra:material ;
   rdfs:subPropertyOf dc:format ;
   rdfs:domain <VisualResource> .

<material.medium> a rdf:Property ;
    rdfs:label "Material.Medium" ;
    rdfs:isDefinedBy vra:material ;
    rdfs:subPropertyOf <material> ;
    rdfs:domain <VisualResource> .

<material.support> a rdf:Property ;
   rdfs:label "Material.Support" ;
   rdfs:isDefinedBy vra:material ;
   rdfs:subPropertyOf <material> ;
   rdfs:domain <VisualResource> .

<technique> a rdf:Property ;
   rdfs:label "Technique" ;
   rdfs:isDefinedBy vra:technique ;
   rdfs:subPropertyOf dc:format ;
   rdfs:domain <VisualResource> .

<creator> a rdf:Property ;
   rdfs:label "Creator" ;
   rdfs:isDefinedBy vra:creator ;
   rdfs:subPropertyOf dc:creator ,
   dc:contributor ;
   rdfs:domain <VisualResource> .

<creator.role> a rdf:Property ;
   rdfs:label "Creator.Role" ;
   rdfs:isDefinedBy vra:creator ;
   rdfs:subPropertyOf dc:creator ;
   rdfs:domain <VisualResource> .

<creator.attribution> a rdf:Property ;
    rdfs:label "Creator.Attribution" ;
    rdfs:isDefinedBy vra:creator ;
    rdfs:subPropertyOf dc:creator ;
    rdfs:domain <VisualResource> .

<creator.personalName> a rdf:Property ;
   rdfs:label "Creator.Personal name" ;
   rdfs:isDefinedBy vra:creator ;
   rdfs:subPropertyOf dc:creator ;
   rdfs:domain <VisualResource> .

<creator.corporateName> a rdf:Property ;
   rdfs:label "Creator.Corporate name" ;
   rdfs:isDefinedBy vra:creator ;
   rdfs:subPropertyOf dc:creator ;
   rdfs:domain <VisualResource> .

<date> a rdf:Property ;
   rdfs:label "Date" ;
   rdfs:isDefinedBy vra:date ;
   rdfs:subPropertyOf dc:date ,
    dc:coverage ;
   rdfs:domain <VisualResource> .

<date.creation> a rdf:Property ;
   rdfs:label "Date.Creation" ;
   rdfs:isDefinedBy vra:date ;
   rdfs:subPropertyOf <date> ;
   rdfs:domain <VisualResource> .

<date.design> a rdf:Property ;
   rdfs:label "Date.Design" ;
   rdfs:isDefinedBy vra:date ;
   rdfs:subPropertyOf <date> ;
```

```
   rdfs:domain <VisualResource> .

<date.beginning> a rdf:Property ;
   rdfs:label "Date.Beginning" ;
   rdfs:isDefinedBy vra:date ;
   rdfs:subPropertyOf <date> ;
   rdfs:domain <VisualResource> .

<date.completion> a rdf:Property ;
    rdfs:label "Date.Completion" ;
    rdfs:isDefinedBy vra:date ;
    rdfs:subPropertyOf <date> ;
    rdfs:domain <VisualResource> .

<date.alteration> a rdf:Property ;
    rdfs:label "Date.Alteration" ;
    rdfs:isDefinedBy vra:date ;
    rdfs:subPropertyOf <date> ;
    rdfs:domain <VisualResource> .

<date.restoration> a rdf:Property ;
   rdfs:label "Date.Restoration" ;
   rdfs:isDefinedBy vra:date ;
   rdfs:subPropertyOf <date> ;
   rdfs:domain <VisualResource> .

<location> a rdf:Property ;
   rdfs:label "Location" ;
   rdfs:isDefinedBy vra:geographic ;
   rdfs:subPropertyOf dc:contributor ,
   dc:coverage ;
   rdfs:domain <VisualResource> .

<location.currentSite> a rdf:Property ;
   rdfs:label "Location.Current Site" ;
   rdfs:isDefinedBy vra:geographic ;
   rdfs:subPropertyOf <location> ;
   rdfs:domain <VisualResource> .

<location.formerSite> a rdf:Property ;
    rdfs:label "Location.Former Site" ;
    rdfs:isDefinedBy vra:geographic ;
    rdfs:subPropertyOf <location> ;
    rdfs:domain <VisualResource> .

<location.creationSite> a rdf:Property ;
   rdfs:label "Location.Creation Site" ;
   rdfs:isDefinedBy vra:geographic ;
   rdfs:subPropertyOf <location> ;
   rdfs:domain <VisualResource> .

<location.discoverySite> a rdf:Property ;
   rdfs:label "Location.Discovery Site" ;
   rdfs:isDefinedBy <http://www.vraweb.org/vracore3.htm#
         geographic> ;
   rdfs:subPropertyOf <location> ;
   rdfs:domain <VisualResource> .

<location.currentRepository> a rdf:Property ;
   rdfs:label "Location.Current Repository" ;
   rdfs:isDefinedBy <http://www.vraweb.org/vracore3.htm#
         geographic> ;
   rdfs:subPropertyOf <location> ;
   rdfs:domain <VisualResource> .

<location.formerRepository> a rdf:Property ;
   rdfs:label "Location.Former Repository" ;
   rdfs:isDefinedBy <http://www.vraweb.org/vracore3.htm#
         geographic> ;
   rdfs:subPropertyOf <location> ;
   rdfs:domain <VisualResource> .

<idNumber.currentRepository> a rdf:Property ;
   rdfs:label "ID Number.Current Repository" ;
   rdfs:isDefinedBy <http://www.vraweb.org/vracore3.htm#id
         %20number> ;
   rdfs:subPropertyOf <idNumber> ;
   rdfs:domain <VisualResource> ;
   rdfs:range <http://www.w3.org/2001/XMLSchema#string> .

<idNumber.formerRepository> a rdf:Property ;
   rdfs:label "ID Number.Former Repository" ;
   rdfs:isDefinedBy <http://www.vraweb.org/vracore3.htm#id
         %20number> ;
   rdfs:subPropertyOf <idNumber> ;
   rdfs:domain <VisualResource> ;
   rdfs:range <http://www.w3.org/2001/XMLSchema#string> .

<idNumber.currentAccession> a rdf:Property ;
   rdfs:label "ID Number.Current Accession" ;
   rdfs:isDefinedBy <http://www.vraweb.org/vracore3.htm#id
         %20number> ;
   rdfs:subPropertyOf <idNumber> ;
   rdfs:domain <VisualResource> ;
   rdfs:range <http://www.w3.org/2001/XMLSchema#string> .

<idNumber.formerAccession> a rdf:Property ;
    rdfs:label "ID Number.Former Accession" ;
    rdfs:isDefinedBy <http://www.vraweb.org/vracore3.htm#
          id%20number> ;
    rdfs:subPropertyOf <idNumber> ;
    rdfs:domain <VisualResource> ;
    rdfs:range <http://www.w3.org/2001/XMLSchema#string> .

<stylePeriod> a rdf:Property ;
   rdfs:label "Style/Period" ;
   rdfs:isDefinedBy <http://www.vraweb.org/vracore3.htm#
         style> ;
   rdfs:subPropertyOf dc:coverage ,
   dc:subject ;
   rdfs:domain <VisualResource> .

<stylePeriod.style> a rdf:Property ;
   rdfs:label "Style/Period.Style" ;
   rdfs:isDefinedBy <http://www.vraweb.org/vracore3.htm#
         style> ;
   rdfs:subPropertyOf <stylePeriod> ;
   rdfs:domain <VisualResource> .

<stylePeriod.group> a rdf:Property ;
   rdfs:label "Style/Period.Group" ;
   rdfs:isDefinedBy <http://www.vraweb.org/vracore3.htm#
         style> ;
   rdfs:subPropertyOf <stylePeriod> ;
   rdfs:domain <VisualResource> .

<stylePeriod.school> a rdf:Property ;
   rdfs:label "Style/Period.School" ;
   rdfs:isDefinedBy <http://www.vraweb.org/vracore3.htm#
         style> ;
   rdfs:subPropertyOf <stylePeriod> ;
   rdfs:domain <VisualResource> .

<stylePeriod.dynasty> a rdf:Property ;
    rdfs:label "Style/Period.Dynasty" ;
    rdfs:isDefinedBy <http://www.vraweb.org/vracore3.htm#
          style> ;
    rdfs:subPropertyOf <stylePeriod> ;
    rdfs:domain <VisualResource> .

<stylePeriod.movement> a rdf:Property ;
   rdfs:label "Style/Period.Movement" ;
   rdfs:isDefinedBy <http://www.vraweb.org/vracore3.htm#
         style> ;
   rdfs:subPropertyOf <stylePeriod> ;
   rdfs:domain <VisualResource> .

<culture> a rdf:Property ;
   rdfs:label "Culture" ;
   rdfs:isDefinedBy <http://www.vraweb.org/vracore3.htm#
         culture> ;
   rdfs:subPropertyOf dc:coverage ;
   rdfs:domain <VisualResource> .

<subject> a rdf:Property ;
   rdfs:label "Subject" ;
   rdfs:isDefinedBy <http://www.vraweb.org/vracore3.htm#
         subject> ;
   rdfs:subPropertyOf dc:subject ;
   rdfs:domain <VisualResource> .
```

```
<relation> a rdf:Property ;
   rdfs:label "Relation" ;
   rdfs:isDefinedBy <http://www.vraweb.org/vracore3.htm#
         related%20work> ;
   rdfs:subPropertyOf dc:relation ;
   rdfs:domain <VisualResource> ;
   rdfs:range <VisualResource> .

<relation.identity> a rdf:Property ;
   rdfs:label "Relation.Identity" ;
   rdfs:isDefinedBy <http://www.vraweb.org/vracore3.htm#
         related%20work> ;
   rdfs:subPropertyOf <relation> ;
   rdfs:domain <VisualResource> ;
   rdfs:range <VisualResource> .

<relation.type> a rdf:Property ;
   rdfs:label "Relation.Type" ;
   rdfs:isDefinedBy <http://www.vraweb.org/vracore3.htm#
         related%20work> ;
   rdfs:subPropertyOf <relation> ;
   rdfs:domain <VisualResource> ;
   rdfs:range <VisualResource> .

<source> a rdf:Property ;
    rdfs:label "Source" ;
    rdfs:isDefinedBy <http://www.vraweb.org/vracore3.htm#
          source> ;
    rdfs:subPropertyOf dc:source ;
    rdfs:domain <VisualResource> ;
    rdfs:range <http://www.w3.org/2001/XMLSchema#string> .

<rights> a rdf:Property ;
    rdfs:label "Rights" ;
    rdfs:isDefinedBy <http://www.vraweb.org/vracore3.htm#
          rights> ;
    rdfs:subPropertyOf dc:rights ;
    rdfs:domain <VisualResource> .

# Additions to the pure VRA schema

<relation.depicts> a rdf:Property ;
   rdfs:label "Relation.Depicts" ;
   rdfs:subPropertyOf <relation> ;
   rdfs:domain <Image> ;
   rdfs:range <Work> .
```

Bibliography

Allemang, D. and Hendler, J. (2008). *Semantic Web for the Working Ontologist*. Morgan Kaufmann Publishers, San Francisco, CA.

ANSI/NISO (2003). Guidelines for the construction, format, and management of monolingual thesauri. Ansi/niso z39.19-2003 (revision of z39.19-1980), ANSI/NISO.

Ashpole, B., Ehrig, M., Euzenat, J., and Stuckenschmidt, H., editors (2005). *Proceedings of the K-CAP 2005 Workshop on Integrating Ontologies*.

van Assem, M., Gangemi, A., and Schreiber, G. (2006a). Conversion of WordNet to a standard RDF/OWL representation. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy.

van Assem, M., Malaisé, V., Miles, A., and Schreiber, G. (2006b). A Method to Convert Thesauri to SKOS. In Sure, Y. and Domingue, J., editors, *Proceedings of the Third European Semantic Web Conference (ESWC'06)*, number 4011 in Lecture Notes in Computer Science, pages 95–109, Budva, Montenegro. Springer-Verlag.

van Assem, M., Menken, M. R., Schreiber, G., Wielemaker, J., and Wielinga, B. (2004). A Method for Converting Thesauri to RDF/OWL. In McIlraith, S. A., Plexousakis, D., and van Harmelen, F., editors, *Proceedings of the Third International Semantic Web Conference (ISWC'04)*, number 3298 in Lecture Notes in Computer Science, pages 17–31, Hiroshima, Japan. Springer-Verlag.

Baader, F., Calvanese, D., McGuinness, D., Nardi, D., and Patel-Schneider, P. (2003). *The Description Logics Handbook: Theory, Implementations, and Applications*. Cambridge University Press.

Baker, T. (2008). Revisions to DCMI Metadata Terms. Technical report, Dublin Core Metadata Initiative.

Beckett, D., Miller, E., and Brickley, D. (2002). Expressing Simple Dublin Core in RDF/XML. DCMI Recommendation, Dublin Core Metadata Initiative.

Berners-Lee, T. (1999). *Weaving the Web*. Orion Business, London.

Berners-Lee, T., Chen, Y., Chilton, L., Connolly, D., Dhanaraj, R., Hollenbach, J., Lerer, A., and Sheets, D. (2006). Tabulator: Exploring and Analyzing linked data on the Semantic Web. In *Proceedings of the 3rd International Semantic Web User Interaction Workshop (SWUI06)*, Athens, Georgia.

Brickley, D. and Guha, R. V. (2000). Resource description framework (RDF) schema specification 1.0. Candidate recommendation, W3C Consortium. See: `http://www.w3.org/TR/2000/CR-rdf-schema-20000327/`.

Brickley, D. and Miller, L. (2005). FOAF Vocabulary Specification. Namespace Document. Latest version: http://xmlns.com/foaf/0.1/.

Budanitsky, A. and Hirst, G. (2001). Semantic distance in WordNet: an experimental application oriented evaluation of five measures. In *Proceedings of the NACCL 2001 Workshop on WordNet and other lexical resources: Applications, extensions, and customizations*, pages 29–34, Pittsburgh, PA, USA.

Corcho, O. and Gómez-Pérez, A. (2000). A Roadmap to Ontology Specification Languages. In Dieng, R. and Corby, O., editors, *Proceedings of the 12th International Conference on Knowledge Acquisition, Modeling and Management (EKAW 2000)*, volume 1937 of *Lecture Notes in Computer Science*, pages 80–96. Springer.

d'Aquin, M., Baldassarre, C., Gridinoc, L., Angeletou, S., Sabou, M., and Motta, E. (2007). Characterizing Knowledge on the Semantic Web with Watson. In Garcia-Castro, R., Vrandecic, D., Gómez-Pérez, A., Sure, Y., and Huang, Z., editors, *5th International Workshop on Evaluation of Ontologies and Ontology-based Tools*, volume 329 of *CEUR Workshop Proceedings*, pages 1–10. CEUR-WS.org.

DCMI Usage Board (2005). MARC Relator terms and Dublin Core. Technical report, Dublin Core Metadata Initiative.

DCMI Usage Board (2009). Criteria for the Review of Application Profiles. DCMI Recommended Resource, Dublin Core Metadata Initiative. Latest version: http://dublincore.org/documents/profile-review-criteria/.

Dean, M., Schreiber, A. T., Bechofer, S., van Harmelen, F., Hendler, J., Horrocks, I., MacGuinness, D., Patel-Schneider, P., and Stein, L. A. (2004). OWL Web Ontology Language Reference. W3C Recommendation, World Wide Web Consortium. Latest version: http://www.w3.org/TR/owl-ref/.

Decker, S., Melnik, S., van Harmelen, F., Fensel, D., Klein, M., Broekstra, J., Erdmann, M., and Horrocks, I. (2000). The Semantic Web: The Roles of XML and RDF. *IEEE Internet Computing*, 15(3):63–74.

den Brink, W. V. and Koele, P. (2002). *Statistiek*, volume 3. Boom.

Dublin Core Collection Description Task Group (2007). Dublin Core Collections Application Profile. DCMI Application Profile, Dublin Core Metadata Initiative. Latest version: http://dublincore.org/groups/collections/collection-application-profile/.

Dublin Core Metadata Initiative (2008). Dublin Core Metadata Element Set, Version 1.1. DCMI Recommendation, Dublin Core Metadata Initiative.

Ehrig, M. and Euzenat, J. (2005). Relaxed precision and recall for ontology matching. In Ashpole et al. (2005).

European Commission (2004). European Interoperability Framework For Pan-European eGovernment Services. Technical report, European Commission.

Euzenat, J. (2007). Semantic precision and recall for ontology alignment evaluation. In Veloso, M. M., editor, *Proceedings of the International Joint Conferences on Artificial Intelligence*,

pages 348–353.

Euzenat, J., Isaac, A., Meilicke, C., Shvaiko, P., Stuckenschmidt, H., Šváb, O., Svátek, V., van Hage, W., and Yatskevich, M. (2007). Results of the OAEI 2007. In Shvaiko, P., Euzenat, J., Giunchiglia, F., and He, B., editors, *Proceedings of the Second International Workshop on Ontology Matching*.

Euzenat, J., Mochol, M., Shvaiko, P., Stuckenschmidt, H., Šváb, O., Svátek, V., van Hage, W., and Yatskevich, M. (2006). Results of the OAEI 2006. In Shvaiko, P., Euzenat, J., N. Noy, H. S., Benjamins, V. R., and Uschold, M., editors, *Proceedings of the ISWC 2006 International Workshop on Ontology Matching*.

Euzenat, J. and Shvaiko, P. (2007). *Ontology matching*. Springer-Verlag, Heidelberg.

Euzenat, J., Stuckenschmidt, H., and Yatskevich, M. (2005). Introduction to the ontology alignment evaluation 2005. In Ashpole et al. (2005).

Fellbaum, C., editor (1998). *WordNet: An Electronic Lexical Database*. Bradford Books.

Gangemi, A., Guarino, N., Masolo, C., and Oltramari, A. (2003a). Sweetening WORDNET with DOLCE. *AI Magazine*, 24(3):13–24.

Gangemi, A., Navigli, R., and Velardi, P. (2003b). The OntoWordNet Project: extension and axiomatisation of conceptual relations in WordNet. In Meersman, R., Tari, Z., and Schmidt, D. C., editors, *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE*, volume 2888, pages 820–838.

Goldbeck, J., Fragoso, G., Hartel, F., Hendler, J., Parsia, B., and Oberthaler, J. (2003). The National Cancer Institute's Thesaurus and Ontology. *Journal of Web Semantics*, 1(1).

Graves, A. and Gutierrez, C. (2006). Data representations for WordNet: A case for RDF. In *Proceedings of the 3rd International WordNet Conference*, Jeju Island, Korea.

Gruber, T. R. (1994). Towards principles for the design of ontologies used for knowledge sharing. In Guarino, N. and Poli, R., editors, *Formal Ontology in Conceptual Analysis and Knowledge Representation*. Kluwer, Boston.

Guarino, N., Masolo, C., and Vetere, G. (1999). Ontoseek: Content-based access to the web. *IEEE Intelligent Systems*, 14(3):70–80.

Guarino, N. and Welty, C. (2004). An Overview of OntoClean. In Staab, S. and Studer, R., editors, *The Handbook on Ontologies*, pages 151–172. Springer-Verlag, Berlin.

van Hage, W., Isaac, A., and Aleksovski, Z. (2007). Sample evaluation of ontology-matching systems. In *Proceedings of the Fifth International Evaluation of Ontologies and Ontology-based Tools*, Busan, Korea.

Hausenblas, M., Halb, W., Raimond, Y., and Heath, T. (2008). What is the Size of the Semantic Web? In *Proceedings of the International Conference on Semantic Systems (I-SEMANTICS08)*, pages 9–16, Graz, Austria.

Hildebrand, M., van Ossenbruggen, J., and Hardman, L. (2006). /facet: A Browser for Heterogeneous Semantic Web Repositories. In *Proceedings of the Fifth International Semantic Web Conference (ISWC2006)*, pages 272–285, Athens, USA.

Hillman, D. (2005). Using Dublin Core. DCMI Recommended Resource, Dublin Core Metadata Initiative. Latest version: http://dublincore.org/documents/usageguide/.

Hollink, L. (2006). *Semantic annotation for retrieval of visual resources*. PhD thesis, VU University Amsterdam.

Hollink, L., Schreiber, A. T., Wielemaker, J., and Wielinga, B. J. (2003). Semantic annotation of image collections. In Handschuh, S., Koivunen, M., Dieng, R., and Staab, S., editors, *Knowledge Capture 2003 – Proceedings Knowledge Markup and Semantic Annotation Workshop*, pages 41–48.

Hollink, L., van Assem, M., Wang, S., Isaac, A., and Schreiber, G. (2008). Two Variations on Ontology Alignment Evaluation: Methodological Issues. In Bechhofer, S., Hauswirth, M., Hoffmann, J., and Koubarakis, M., editors, *Proceedings of the Fifth European Semantic Web Conference (ESWC'08)*, volume 5021 of *Lecture Notes in Computer Science*, pages 388–401, Canary Islands, Spain. Springer-Verlag.

Hu, W. and Qu, Y. (2007). Discovering Simple Mappings Between Relational Database Schemas and Ontologies. In *Proceedings of the Sixth International Semantic Web Conference*, pages 225 – 238.

Hyvönen, E., Mäkelä, E., Salminen, M., Valo, A., Viljanen, K., Saarela, S., Junnila, M., and Kettula, S. (2005). Museumfinland – finnish museums on the semantic web. *Journal of Web Semantics*, 3(2):25.

Ide, N. and Véronis, J. (1998). Introduction to the special issue on word sense disambiguation: the state of the art. *Computational Linguistics*, 24(1):2–40.

International Organization for Standardization (1986). Documentation - Guidelines for the establishment and development of monolingual thesauri. Ref. No. ISO 2788-1986, International Organization for Standardization.

Internet Engineering Task Force (2004). application/rdf+xml media type registration. Request for comments, The Internet Society.

Internet Engineering Task Force (2005). Uniform Resource Identifier (URI): Generic Syntax. Request for comments, The Internet Society.

Isaac, A., Zinn, C., Matthezing, H., van der Meij, L., Schlobach, S., and Wang, S. (2007). The value of usage scenarios for thesaurus alignment in cultural heritage context. In *Proceedings of International Workshop on Cultural Heritage on the Semantic Web, ISWC2007*, Korea.

Johnston, D., Nelson, S. J., Schulman, J.-L. A., Savage, A. G., and Powell, T. P. (1998). Redefining a thesaurus: Term-centric no more. In *Proceedings of the 1998 AMIA Annual Symposium*.

Kalfoglou, Y. and Schorlemmer, M. (2003). Ontology mapping: the state of the art. *The Knowledge Engineering Review*, 18(1):1–31.

Kekäläinen, J. and Järvelin, K. (2002). Using graded relevance assessments in IR evaluation. *Journal of the American Society for Information Science and Technology*, 53(13).

Leacock, C. and Chodorow, M. (1998). Combining local context and wordnet similarity for word sense identification. In *WordNet: An Electronic Lexical Database*, chapter 11, pages 265 – 285.

MIT Press. Edited by C. Fellbaum.

Miles, A. and Matthews, B. (2004). Review of RDF thesaurus work. Deliverable 8.2, version 0.1, SWAD-Europe.

Miles, A., Matthews, B., and Wilson, M. (2004a). RDF encoding of multilingual thesauri. Deliverable 8.3, version 0.1, SWAD-Europe.

Miles, A., Rogers, N., and Beckett, D. (2004b). Migrating Thesauri to the Semantic Web - Guidelines and case studies for generating RDF encodings of existing thesauri. Deliverable 8.8, SWAD-Europe.

Nilsson, M. (2008). Description Set Profiles: A constraint language for Dublin Core Application Profiles. DCMI Working Draft, Dublin Core Metadata Initiative. Latest version: `http://dublincore.org/documents/dc-dsp/`.

Nilsson, M., Baker, T., and Johnston, P. (2008a). The Singapore Framework for Dublin Core Application Profiles). DCMI Recommended Resource, Dublin Core Metadata Initiative. Latest version: `http://dublincore.org/documents/singapore-framework/`.

Nilsson, M., Miles, A. J., Johnston, P., and Enoksson, F. (2007). Formalizing Dublin Core Application Profiles: Description Set Profiles and Graph Constraints. In *Proceeding of the 2nd International Conference on Metadata and Semantics Research*, Corfu, Greece.

Nilsson, M., Powell, A., Johnston, P., and Naeve, A. (2008b). Expressing Dublin Core metadata using the Resource Description Framework (RDF). DCMI Recommendation, Dublin Core Metadata Initiative. Latest version: `http://dublincore.org/documents/dc-rdf/`.

Odell, J. J. (1994). Six different kinds of composition. *Journal Of Object-Oriented Programming*, 5(8).

Peterson, T. (1994). *Introduction to the Art and Architecture Thesaurus*. Oxford University Press.

Powell, A., Nilsson, M., Naeve, A., Johnston, P., and Baker, T. (2007). DCMI Abstract Model). DCMI Recommendation, Dublin Core Metadata Initiative.

RDF Core Working Group (2004a). RDF Primer. W3C Recommendation, World Wide Web Consortium. Edited by F. Manola and E. Miller. Latest version: `http://www.w3.org/TR/rdf-primer/`.

RDF Core Working Group (2004b). RDF Semantics. W3C Recommendation, World Wide Web Consortium. Edited by P. Hayes. Latest version: `http://www.w3.org/TR/rdf-mt/`.

RDF Core Working Group (2004c). Resource Description Framework (RDF): Concepts and Abstract Syntax. W3c recommendation, World Wide Web Consortium. Edited by G. Klyne and J. J. Carroll. Latest version: `http://www.w3.org/TR/rdf-concepts/`.

RDF Data Access Working Group (2008). SPARQL Query Language for RDF. W3c recommendation, World Wide Web Consortium. Edited by E. Prud'hommeaux and A. Seaborne. Latest version: `http://www.w3.org/TR/rdf-sparql-query/`.

Ruttenberg, A., Clark, T., Bug, W., Samwald, M., Bodenreider, O., Chen, H., Doherty, D., Forsberg, K., Gao, Y., Kashyap, V., Kinoshita, J., Luciano, J., Marshall, M. S., Ogbuji, C., Rees, J., Stephens, S., Wong, G., Wu, E., Zaccagnini, D., Hongsermeier, T., Neumann, E., Herman, I., and Cheung, K.-H. (2007). Advancing translational research with the semantic web. *BMC*

Bioinformatics, 8(Suppl 3):S2.

Schreiber, G., Akkermans, H., Anjewierden, A., de Hoog, R., Shadbolt, N., Van de Velde, W., and Wielinga, B. (2000). *Knowledge Engineering and Management: The CommonKADS Methodology*. MIT Press, Cambridge, Massachusetts.

Schreiber, G., Amin, A., van Assem, M., de Boer, V., Hardman, L., Hildebrand, M., Hollink, L., Huang, Z., van Kersen, J., de Niet, M., Omelayenko, B., van Ossenbruggen, J., Siebes, R., Taekema, J., Wielemaker, J., and Wielinga, B. (2006). MultimediaN E-Culture demonstrator. In Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., and Aroyo, L., editors, *Proceedings of the Fifth International Semantic Web Conference (ISWC'06)*, number 4273 in Lecture Notes in Computer Science, pages 951–958, Athens, Georgia, USA. Springer-Verlag.

Semantic Web Best Practices and Deployment Working Group (2005a). Representing Classes As Property Values on the Semantic Web. W3C Working Group Note, World Wide Web Consortium. Edited by N. Noy. Latest version: `http://www.w3.org/TR/swbp-classes-as-values/`.

Semantic Web Best Practices and Deployment Working Group (2005b). SKOS Core Guide. W3C Public Working Draft, World Wide Web Consortium. Edited by A. Miles and D. Brickley. Latest version: `http://www.w3.org/TR/swbp-skos-core-guide`.

Semantic Web Best Practices and Deployment Working Group (2005c). SKOS Core Vocabulary Specification. W3C Public Working Draft, World Wide Web Consortium. Edited by A. Miles and D. Brickley. Latest version: `http://www.w3.org/TR/swbp-skos-core-spec/`.

Semantic Web Best Practices and Deployment Working Group (2006a). Best Practice Recipes for Publishing RDF Vocabularies. W3C Working Draft, World Wide Web Consortium. Edited by A. Miles, T. Baker and R. Swick. Latest version: `http://www.w3.org/TR/swbp-vocab-pub/`.

Semantic Web Best Practices and Deployment Working Group (2006b). Defining N-ary Relations on the Semantic Web. W3C Working Group Note, World Wide Web Consortium. Edited by N. Noy and A. Rector. Latest version: `http://www.w3.org/TR/swbp-n-aryRelations/`.

Semantic Web Best Practices and Deployment Working Group (2006c). RDF/OWL Representation of WordNet. W3C Working Draft, World Wide Web Consortium. Edited by M. van Assem, A. Gangemi and G. Schreiber. Latest version: `http://www.w3.org/TR/wordnet-rdf/`.

Semantic Web Deployment Working Group (2007). SKOS Use Cases and Requirements. W3C Working Draft, World Wide Web Consortium. Edited by A. Isaac, J. Phipps and D. Rubin. Latest version: `http://www.w3.org/TR/skos-ucr/`.

Semantic Web Deployment Working Group (2008a). SKOS Simple Knowledge Organization System Primer. W3C Working Draft, World Wide Web Consortium. Edited by A. Isaac and E. Summers. Latest version: `http://www.w3.org/TR/skos-primer/`.

Semantic Web Deployment Working Group (2008b). SKOS Simple Knowledge Organization System Reference. W3C Working Draft, World Wide Web Consortium. Edited by A. Miles and S. Bechofer. Latest version: `http://www.w3.org/TR/skos-reference/`.

Shvaiko, P. and Euzenat, J. (2005). A survey of schema-based matching approaches. *Journal on*

Data Semantics, 3730:146–171.

Smith, B. and Welty, C. (2001). Ontology: Towards a new synthesis. In Welty, C. and Smith, B., editors, *Formal Ontology in Information Systems*, pages iii–x.

Soergel, D., Lauser, B., Liang, A., Fisseha, F., Keizer, J., and Katz, S. (2004). Reengineering Thesauri for New Applications: the AGROVOC Example. *Journal of Digital Information*, 4(4).

Soualmia, L., Goldbreich, C., and Darmoni, S. (2004). Representing the MeSH in OWL: Towards a Semi-Automatic Migration. In *Proceedings of the First International Workshop on Formal Biomedical Knowledge Representation (KR-MED 2004)*, pages 81–87, Whistler, Canada. ISSN 1613-0073, Vol.102.

Stickler, P. (2005). CBD - Concise Bounded Description. W3C Member Submission, World Wide Web Consortium. Latest version: `http://www.w3.org/Submission/CBD/`.

U.S. National Library of Medicine (2001). Introduction to MeSH in XML format.

U.S. National Library of Medicine (2004). MeSH tree structures.

W3C Technical Architecture Group (2004). Architecture of the World Wide Web, Volume One. W3C Recommendation, World Wide Web Consortium. Edited by I. Jacobs and N. Walsh. Latest version: `http://www.w3.org/TR/webarch/`.

Web Ontology Working Group (2004). OWL Web Ontology Language Guide. W3C Recommendation, World Wide Web Consortium. Edited by M. K. Smith, C. Welty and D. L. McGuinness. Latest version: `http://www.w3.org/TR/owl-guide/`.

Web Ontology Working Group (2009). OWL 2 Web Ontology Language Direct Semantics. W3C Recommendation, World Wide Web Consortium. Edited by B. Motik, P. Patel-Schneider and B. C. Grau. Latest version: `http://www.w3.org/TR/owl2-direct-semantics/`.

Welty, C. and Guarino, N. (2001). Supporting ontological analysis of taxonomic relationships. *Data & Knowledge Engineering*, 39(1):51–74.

Wielemaker, J., Schreiber, A. T., and Wielinga, B. J. (2003). Prolog-based infrastructure for RDF: performance and scalability. In Fensel, D., Sycara, K., and Mylopoulos, J., editors, *Proceedings of the Second International Semantic Web Conference (ISWC03)*, LNCS, pages 644–658, Sanibel Island, Florida. Springer Verlag.

Wielinga, B., Wielemaker, J., Schreiber, G., and van Assem, M. (2004). Methods for Porting Resources to the Semantic Web. In Bussler, C., Davies, J., Fensel, D., and Studer, R., editors, *Proceedings of the First European Semantic Web Symposium (ESWS2004)*, number 3053 in Lecture Notes in Computer Science, pages 299–311, Heraklion, Greece. Springer-Verlag.

Wroe, C., Stevens, R., Goble, C., and Ashburner, M. (2003). A methodology to migrate the Gene ontology to a description logic environment using DAML+OIL. In *Proceedings of the 8th Pacific Symposium on Biocomputing (PSB 2003)*, pages 624–635, Lihue, Hawaii, USA.

XML Core Working Group (2006). Namespaces in XML 1.0 (Second Edition). W3C Recommendation, World Wide Web Consortium. Edited by T. Bray, D. Hollander, A. Laynan and R. Tobin. Latest version: `http://www.w3.org/TR/REC-xml-names/`.

Summary: Converting and Integrating Vocabularies for the Semantic Web

Institutions such as libraries, museums and other archives have been collecting books, paintings, statues and other objects for centuries. To manage these collections, cataloguers have described each object with respect to its title, author, subjects, materials and other attributes. This process is called "indexing", and simplifies the process of searching through the collections. An object description created during indexing is essentially a set of attribute-value pairs. Such a description might consist e.g. of pairs *author=Rembrandt, title=Anatomy Lesson, date=1632, type=painting, subject=group portrait*. Such descriptions are also called *metadata* (data about the actual object).

The values for the pairs are usually taken from *vocabularies* . Vocabularies are lists of concepts with definitions and play a key role in indexing and search. Firstly, they offer a set of agreed upon concepts that cataloguers can pick from. Secondly, concepts provide a convenient place to group synonymous terms (e.g. "clair-obscure" and "chiaroscuro" which both refer to Rembrandt's painting style). Thirdly, the concepts usually have a unique identifier, which allows the cataloguer to indicate the correct concept even though the concept has an ambiguous term (e.g. "painting" as the process of applying a protective coating to an object vs. "painting" as the process of creating an expressive or communicative image). Fourthly, the concepts are often placed into a hierarchy (e.g. "origami" below "Japanese art") which simplifies search (a search for books on Japanese art will also return books on origami).

To aid the indexing process, each institution not only prescribes a set of vocabularies to be used, but also the attributes and their names. Attributes are called *elements*, and the set of allowed attributes is together called the *metadata element set*.

With the advent of the Web people and institutions have started sharing their data. This has the potential benefit that all information on a particular topic, say the paintings of Vincent van Gogh, can be queried as if they were stored in one system. This requires that the data is *integrated*: it must conform to a particular format and structure that the search system understands. Two obstacles to integration are the different data formats in use (the *syntactic integration problem*) and the different terms in use to denote similar concepts (the *semantic integration problem*). One example of the latter problem is when one institution has a concept called "clair-obscure" and another has a concept called "chiaroscuro". Another example is when one institution uses a metadata element called "creator" while another has an element called "author". These concepts and elements are highly similar and a search for clair-obscure paintings by Rembrandt needs to query through both concepts and elements.

The Semantic Web is a research area that proposes particular solutions to these problems. Firstly, it proposes to use a family of web-based knowledge representation languages that have

RDF as underlying model (RDFS and the several flavours of OWL). Conversion of data to this family of languages solves a substantial part of the syntactic integration problem. Secondly, the languages have a few simple mechanisms to relate similar concepts to each other, solving part of the semantic integration problem. For example, it is possible to state that "creator" is equivalent to "author", so that a query on either element will automatically include the results obtained from querying with the other.

In this thesis we assume that the approach and languages proposed by the Semantic Web community are useful for achieving integration, and aim to apply these in the context of the cultural heritage domain. The problem of converting the original data sets to RDF/OWL has not been investigated much. In this thesis we focus mostly on conversion of vocabularies. Our problem statement is as follows: *How can existing vocabularies be made available to Semantic Web applications?* Problems that need to be solved include understanding the original syntactical format in which the vocabulary is expressed, understanding the conceptual model that lies behind it, linking this conceptual model to that of RDF/OWL, and finding an appropriate way to convert the former model into the latter. These tasks are far from being automated, but the demand for proper conversions will increase in the coming years. Therefore, this thesis has focused on developing *methods* for conversion of vocabularies. Methods are step-wise processes with guidelines that can be followed by people performing the conversion task. Several choices have to be made during the process that affect the resulting representation. Conversion can be performed for the benefit of one particular application and tuned to its specific needs, but conversions can also aim at a representation that is as complete and reusable as possible for any application. The main contribution of this thesis is the development of two separate methods to cater to both situations.

In Chapter 2 we develop a first version of a generic method for conversion. The assumption is that a faithful and complete conversion of the vocabularies results in a representation that is useful for most applications. A method consisting of several steps and guidelines was drafted, and then applied to two case studies: conversions of the MeSH and WordNet vocabularies. These helped to improve the method; they showed which additional guidelines were needed to adequately handle these cases. We deliberately chose two complex vocabularies so that a broad range of vocabulary features were covered. Two tailor-made schemas for each vocabulary and a complete conversion of their content to these schemas are the outcome.

Another way to convert vocabularies suitable for a wide range of applications is to use a standard, widely supported vocabulary schema. In Chapter 3 we developed a method aimed at the emerging SKOS standard. We chose three vocabularies as use cases: a simple one (GTAA), an intermediately complex one (IPSV) and a complex case (MeSH). We found that SKOS was suitable for converting GTAA and IPSV (making use of RDF/OWL abilities to specialize a schema), but MeSH could not be covered completely because SKOS did not allow a concept's terms to be represented as instances themselves.

In Chapter 4 we returned to the problem of generic conversion as approached in Chapter 2. The assumption that our method results in vocabularies useful for many applications was tested by comparing our generic conversion of WordNet to a conversion developed with application use cases in mind (in the W3C Semantic Web Best Practices Working Group). The comparison of the two WordNets showed how our generic method should be changed to cater for a wider range

of applications. However, another outcome of the chapter is that a generic method cannot cater to all requirements an application might have, because it may require that content is left out or structured differently than in the original source. We also improved the WordNet conversion to SKOS by applying a newly developed extension that allows terms to be represented as instances themselves. This solves most of the problems noted in the MeSH conversion to SKOS.

Given the results from Chapter 4, we developed a new method that can be used to cater to specific applications in Chapter 5. We adapted the generic method by introducing specific steps to determine the requirements and use cases that need to be covered. The principle of complete and faithful conversion of the original source was dropped. The case study is the MultimediaN E-Culture search and browsing application, for which the AAT, TGN and ULAN were converted. This case study pointed out that the conversion made by the E-Culture team without our method missed several pieces of information needed by the application use cases.

In Chapter 6 we continued to investigate conversions targeted at specific applications. In this chapter we concentrated on alignment applications, which take two or more vocabularies in RDF/OWL as input and produce mapping relations between concepts of the vocabularies. Our analysis showed that these applications cannot handle representations as produced by our methods. We provided a conversion technique to mitigate this problem. The analysis is part of a study on new evaluation techniques for alignments of vocabularies. Alignment is a central ingredient of integration as promoted by the Semantic Web community. The outcome of the study is that our proposed techniques are better tuned to evaluating the quality of an alignment for a particular application than existing techniques.

Integration of collections relies on integration of both vocabularies and metadata element sets. In Chapter 7 we study how an existing metadata element set can be represented in RDF/OWL in a way that is interoperable with vocabulary representations as advocated in this thesis. The metadata element set is called VRA and caters specifically to cultural heritage. We show how VRA can be implemented as a specialization of the more generic Dublin Core element set. Linking VRA with Dublin Core allows integration of collections from different domains (television archives, libraries, etcetera) into one search system. We also show how VRA can be specialized to reflect that e.g. the Rijksmuseum uses e.g. ULAN as range of the “creator” element (we term this feature *collection-specific value ranges*).

In summary, this thesis contributes to the integration of metadata collections in three ways. Firstly and chiefly through the development of conversion methods for vocabularies and through contributing actual conversions made with the methods. Secondly, through investigating how metadata schemas can be represented in a way that allows using them together with vocabulary representations produced by the methods. Thirdly, by contributing a study on how aligmnents can be evaluated on their usefulness for particular applications.

Samenvatting: Conversie en Integratie van Vocabulaires voor het Semantisch Web

Al eeuwen verzamelen bibliotheken, musea en andere archieven objecten als boeken, schilderijen, standbeelden enzovoorts. Om zulke collecties te ontsluiten worden de objecten door archivarissen omschreven middels hun titel, auteur, onderwerpen, materialen en andere attributen. Dit proces wordt "indexeren" genoemd en vereenvoudigt het proces van zoeken door collecties. Een objectomschrijving bestaat uit een set van attribuut-waarde paren. Zo'n beschrijving kan bijvoorbeeld bestaan uit *auteur=Rembrandt, titel=Anatomieles, datum=1632, type=schilderij, onderwerp = groepsportret*. Zulke beschrijvingen worden ook wel *metadata* genoemd (gegevens over het eigenlijke object).

De waarden in de paren komen meestal uit *vocabulaires*. Vocabulaires zijn lijsten concepten met definities en spelen een sleutelrol in indexeren en zoeken. Ten eerste vormen ze een set van concepten waarvan de archivarissen het over eens zijn dat ze nuttig zijn voor het beschrijven van de objecten. Ten tweede geven concepten de mogelijkheid om synoniemen aan elkaar te koppelen (bv. "clair-obscure" en "chiaruscuro" welke beide refereren aan de schilderstijl van Rembrandt). Ten derde wordt aan concepten een unieke code toegekend, die het mogelijk maakt om het juiste concept aan een object toe te kennen zelfs als het concept een ambigue term heeft (bv. "schilderen" als het aanbrengen van een beschermlaag op een object tegenover "schilderen" als het weergeven van een expressief of communicatief beeld). Ten vierde worden concepten vaak in een hierarchie onder elkaar geplaatst (bv. "origami" onder "Japanse kunst") waardoor zoeken naar Japanse kunst ook boeken over origami zal opleveren.

Instituten geven gewoonlijk een aantal vocabulaires aan dat gebruikt kan worden, maar ook de attributen die toegepast kunnen worden. Attributen worden ook wel *elementen* genoemd, en de set van toegestane attributen heet het *metadata element set*.

Met de opkomst van het Web zijn mensen en instituten hun informatie gaan delen. Dit heeft het potentiële voordeel dat alle informatie over een bepaald onderwerp, zeg de schilderijen van Vincent van Gogh, kunnen worden opgezocht alsof ze in één systeem waren opgeslagen. Hiervoor is het noodzakelijk dat de data *geïntegreerd* is: het moet voldoen aan een bepaald formaat en structuur dat het zoeksysteem kan verwerken. Twee obstakels voor integratie zijn de verschillende formaten die gebruikt worden (het *syntactische integratieprobleem*) en verschillende termen voor gelijkende concepten (het *semantische integratie probleem*). Een voorbeeld van het laatste is wanneer het ene instituut de term "clair-obscure" gebruikt en het andere de term "chiaroscuro". Een ander voorbeeld is wanneer het ene instituut een metadata element gebruikt met de naam "maker" en het andere een element met de naam "auteur". Deze concepten en elementen zijn zeer vergelijkbaar en een zoekopdracht naar clair-obscure schilderijen van Rembrandt zal beide concepten en elementen moeten gebruiken in de opdrachtformulering.

Het Semantisch Web is een onderzoeksgebied dat bepaalde oplossingen voor deze problemen voorstelt. Ten eerste stelt het voor een familie van representatietalen in te zetten die RDF als onderliggend model gebruiken (RDFS en de verschillende versies van OWL). Conversie van data naar deze talen lost een substantieel deel van het syntactische integratieprobleem op. Ten tweede hebben de talen een paar simpele mechanismen om vergelijkbare concepten aan elkaar te relateren, wat een deel van het semantische integratieprobleem oplost. Het is bijvoorbeeld mogelijk om uit te drukken dat "maker" en "auteur" equivalent aan elkaar zijn, zodat een zoekopdracht op één van beide elementen automatisch ook resultaten opvraagt via het andere element.

In dit proefschrift nemen we aan dat de aanpak en talen voorgesteld door de Semantisch Web onderzoekers bruikbaar zijn om integratie mogelijk te maken, en trachten ze toe te passen op het terrein van cultureel erfgoed. Het probleem van het converteren van originele data sets naar RDF/OWL is nog niet veel onderzocht. In dit proefschrift richten we ons vooral op de conversie van vocabulaires. De probleemomschrijving is als volgt: *Hoe kunnen bestaande vocabulaires beschikbaar worden gemaakt voor Semantisch Web applicaties?* Obstakels daarbij zijn wat de betekenis is van het originele syntactische formaat waarin het vocabulaire is gerepresenteerd, het conceptuele model dat achter het ontwerp van het vocabulaire schuil gaat begrijpen, het conceptuele model verbinden met het conceptuele model van RDF/OWL, en het vinden van een afdoende manier om de ene in de andere om te zetten. Deze taken zijn moeilijk te automatiseren, terwijl de vraag naar goede conversies de komende jaren zal gaan stijgen. Daarom richt dit proefschrift zich op het ontwikkelen van *methoden* voor de conversie van vocabulaires. Methodes zijn stapsgewijze processen met richtlijnen die door mensen uitgevoerd kunnen worden om een conversie tot stand te brengen. Er moeten tijdens dit proces verschillende beslissingen genomen worden die het eindresultaat beïnvloeden. Conversie kan uitgevoerd worden voor gebruik in een bepaalde applicatie, maar het proces kan er ook op gericht zijn om een vocabulaire zo compleet en herbruikbaar mogelijk te converteren voor welke applicatie dan ook. De voornaamste bijdrage van dit proefschrift is het ontwikkelen van twee methoden die zich apart op deze situaties richten.

In hoofdstuk 2 ontwikkelen we een eerste versie van een generieke methode voor conversie. De aanname is dat een waarheidsgetrouwe en complete conversie van een vocabulaire resulteert in een representatie die voor de meeste applicaties herbruikbaar is. Een methode bestaande uit verschillende stappen en richtlijnen werd opgesteld, en daarna getest in twee studies: conversies van de MeSH en WordNet vocabulaires. Deze hielpen de methode te verbeteren; ze toonden welke extra richtlijnen nodig waren om deze gevallen adequaat te verwerken. We kozen met opzet twee complexe vocabulaires zodat een breed spectrum aan vocabulaire eigenschappen kon worden behandeld. Twee op maat gesneden schema's en complete conversies van de inhoud van de vocabulaires naar deze schema's was het resultaat.

Een andere manier om vocabulaires te converteren op een manier die bruikbaar is voor een breed spectrum aan applicaties is om een standaard, breed gedragen schema te gebruiken. In hoofdstuk 3 ontwikkelden we een methode gericht op de nieuwe SKOS standaard. We kozen drie vocabulaires als toepassingsstudies: een simpele (GTAA), matig ingewikkelde (IPSV) en complex geval (MeSH). De bevindingen waren dat SKOS bruikbaar was voor het converteren van GTAA en IPSV (gebruik makend van de mogelijkheid in RDF/OWL om een schema te specialiseren), maar MeSH kon niet compleet vertaald worden omdat SKOS niet toestaat om een de termen van

een concept als zelfstandige instanties te representeren.

In hoofdstuk 4 keerden we terug naar het probleem van generieke conversies zoals we dat benaderden in hoofdstuk 2. De aanname dat gebruik van onze methode resulteert in vocabulaires die geschikt zijn voor veel applicaties werd getest door het vergelijken van onze generieke conversie van WordNet met een conversie ontwikkeld met specifieke toepassingen binnen een applicatie in gedachten (binnen de W3C Semantic Web Best Practices Working Group). De vergelijking van de twee Wordnets liet zien hoe onze generieke methode moest worden aangepast om te kunnen voorzien in de behoeften van een breder spectrum aan applicaties. Een andere uitkomst was echter dat een generieke methode niet altijd in alle behoeften kan voorzien, omdat het nodig kan zijn om informatie uit het originele vocabulaire weg te laten of anders te structureren dan in de originele bron. We verbeterden ook de WordNet conversie naar SKOS, door het toepassen van een nieuwe SKOS extensie die wel toestaat dat concept termen als instanties worden gerepresenteerd.

Met behulp van de resultaten van hoofdstuk 4 ontwikkelden we een nieuwe methode gericht op conversies voor specifieke applicaties in hoofdstuk 5. De generieke methode werd aangepast door nieuwe stappen te introduceren die ontwerpeisen en toepassingsscenarios vastleggen. Het principe van volledige en waarheidsgetrouwe conversie werd losgelaten. De studie bestond dit maal uit de Multimedian E-Culture zoek- en weergavesysteem, waarvoor de AAT, TGN en ULAN vocabulaires werden geconverteerd. Deze studie wees uit dat de conversies die het E-Culture team zelf had gemaakt enkele stukken informatie misten die nodig waren in de toepassingsscenarios.

In hoofdstuk 6 zetten we het onderzoek naar conversies voor specifieke applicaties voort. We concentreerden ons op zogenaamde alignment applicaties, die twee of meer vocabulaires in RDF/OWL als invoer nemen en “mappings” tussen vergelijkbare concepten in de vocabulaires produceren (de mappings heten tezamen een “alignment”). Onze analyse liet zien dat deze applicaties niet om kunnen gaan met vocabulaires zoals geproduceerd door onze methodes. We presenteerden een conversie techniek om dit probleem te verhelpen. De analyse is onderdeel van een studie naar nieuwe technieken om de kwaliteit van een alignment te evalueren. Alignment is een primair ingredient van integratie zoals gepropageerd door de Semantisch Web onderzoekswereld. De uitkomst van de studie is dat de door ons voorgestelde technieken een betere weergave van de kwaliteit van een alignment geven wanneer de alignment gebruikt moet worden in een bepaalde toepassen (ten opzichte van bestaande evaluatie technieken).

Integratie van collecties is gebaseeerd op integratie van zowel vocabulaires als metadata element sets. In hoofdstuk 7 bestudeerden we hoe we bestaande metadata element sets gerepresenteerd kunnen worden in RDF/OWL op een manier die interoperabel is met vocabulaires zoals geproduceerd door onze methoden. De bestudeerde metadata element set heet VRA en is gericht op cultureel erfgoed. We lieten zien hoe VRA kan worden geïmplementeerd als een specialisatie van de meer generieke Dublin Core element set. Het verbinden van VRA met Dublin Core maakt integratie van collecties uit verschillende domeinen mogelijk (televisie archieven, bibliotheken, etcetera) voor gebruik in één zoeksysteem. We lieten ook zien VRA gespecialiseerd kan worden om weer te geven dat bv. het Rijksmuseum ULAN gebruikt als waardebereik voor het “auteur” element (we noemen dit *collectie-specifieke waardebereiken*).

Resumerend, dit proefschrift draagt bij aan de integratie van metadata collecties op drie manieren. Ten eerste en primair door de ontwikkeling van methodes voor conversie van vocabulaires

en een aantal daadwerkelijke conversies uitgevoerd met behulp van de methoden. Ten tweede, door te onderzoeken hoe metadata schema's gerepresenteerd kunnen worden op een manier die het toelaat ze samen te gebruiken met vocabulaire representaties zoals geproduceerd met onze methoden. Ten derde, door de studie naar hoe de kwaliteit van alignments beter kan worden ingeschat met betrekking tot specifieke applicaties.

SIKS Dissertation Series

Abbreviations: SIKS - Dutch Research School for Information and Knowledge Systems; CWI - Centrum voor Wiskunde en Informatica, Amsterdam; EUR - Erasmus Universiteit, Rotterdam; KUB - Katholieke Universiteit Brabant, Tilburg; KUN - Katholieke Universiteit Nijmegen; RUG - Rijksuniversiteit Groningen; RUL - Rijksuniversiteit Leiden; FONS - Ferrologisch Onderzoeksinstituut Nederland/Sweden; RUN - Radboud Universiteit Nijmegen; TUD - Technische Universiteit Delft; TU/e - Technische Universiteit Eindhoven; UL - Universiteit Leiden; UM - Universiteit Maastricht; UT - Universiteit Twente, Enschede; UU - Universiteit Utrecht; UvA - Universiteit van Amsterdam; UvT - Universiteit van Tilburg; VU - Vrije Universiteit, Amsterdam.

1998

1998-1 Johan van den Akker (CWI) DEGAS - An Active, Temporal Database of Autonomous Objects

1998-2 Floris Wiesman (UM) Information Retrieval by Graphically Browsing Meta-Information

1998-3 Ans Steuten (TUD) A Contribution to the Linguistic Analysis of Business Conversations within the Language/Action Perspective

1998-4 Dennis Breuker (UM) Memory versus Search in Games

1998-5 Eduard Oskamp (RUL) Computerondersteuning bij Straftoemeting

1999

1999-1 Mark Sloof (VU) Physiology of Quality Change Modelling; Automated Modelling of Quality Change of Agricultural Products

1999-2 Rob Potharst (EUR) Classication using Decision Trees and Neural Nets

1999-3 Don Beal (UM) The Nature of Minimax Search

1999-4 Jacques Penders (UM) The Practical Art of Moving Physical Objects

1999-5 Aldo de Moor (KUB) Empowering Communities: A Method for the Legitimate User-Driven Specication of Network Information Systems

1999-6 Niek Wijngaards (VU) Re-Design of Compositional Systems

1999-7 David Spelt (UT) Verication Support for Object Database Design

1999-8 Jacques Lenting (UM) Informed Gambling: Conception and Analysis of a Multi-Agent Mechanism for Discrete Reallocation

2000

2000-1 Frank Niessink (VU) Perspectives on Improving Software Maintenance

2000-2 Koen Holtman (TU/e) Prototyping of CMS Storage Management

2000-3 Carolien Metselaar (UvA) Sociaal-organisatorische Gevolgen van Kennistechnologie; een Procesbenadering en Actorperspectief

2000-4 Geert de Haan (VU) ETAG, A Formal Model of Competence Knowledge for User Interface Design

2000-5 Ruud van der Pol (UM) Knowledge-Based Query Formulation in Information Retrieval

2000-6 Rogier van Eijk (UU) Programming Languages for Agent Communication

2000-7 Niels Peek (UU) Decision-Theoretic Planning of Clinical Patient Management

2000-8 Veerle Coupé (EUR) Sensitivity Analyis of Decision-Theoretic Networks

2000-9 Florian Waas (CWI) Principles of Probabilistic Query Optimization

2000-10 Niels Nes (CWI) Image Database Management System Design Considerations, Algorithms and Architecture

2000-11 Jonas Karlsson (CWI) Scalable Distributed Data Structures for Database Management

2001

2001-1 Silja Renooij (UU) Qualitative Approaches to Quantifying Probabilistic Networks

2001-2 Koen Hindriks (UU) Agent Programming Languages: Programming with Mental Models

2001-3 Maarten van Someren (UvA) Learning as Problem Solving

2001-4 Evgueni Smirnov (UM) Conjunctive and Disjunctive Version Spaces with Instance-Based Boundary Sets

2001-5 Jacco van Ossenbruggen (VU) Processing Structured Hypermedia: A Matter of Style

2001-6 Martijn van Welie (VU) Task-Based User Interface Design

2001-7 Bastiaan Schonhage (VU) Diva: Architectural Perspectives on Information Visualization

2001-8 Pascal van Eck (VU) A Compositional Semantic Structure for Multi-Agent Systems Dynamics

2001-9 Pieter Jan t Hoen (RUL) Towards Distributed Development of Large Object-Oriented Models, Views of Packages as Classes

2001-10 Maarten Sierhuis (UvA) Modeling and Simulating Work Practice BRAHMS: a Multiagent Modeling and Simulation Language for Work Practice Analysis and Design

2001-11 Tom van Engers (VU) Knowledge Management: The Role of Mental Models in Business Systems Design

2002

2002-1 Nico Lassing (VU) Architecture-Level Modiability Analysis

2002-2 Roelof van Zwol (UT) Modelling and Searching Web-based Document Collections

2002-3 Henk Ernst Blok (UT) Database Optimization Aspects for Information Retrieval

2002-4 Juan Roberto Castelo Valdueza (UU) The Discrete Acyclic Digraph Markov Model in Data Mining

2002-5 Radu Serban (VU) The Private Cyberspace Modeling Electronic Environments Inhabited by Privacy-Concerned Agents

2002-6 Laurens Mommers (UL) Applied Legal Epistemology; Building a Knowledge-based Ontology of the Legal Domain

2002-7 Peter Boncz (CWI) Monet: A Next-Generation DBMS Kernel For Query-Intensive Applications

2002-8 Jaap Gordijn (VU) Value Based Requirements Engineering: Exploring Innovative E-Commerce Ideas

2002-9 Willem-Jan van den Heuvel (KUB) Integrating Modern Business Applications with Objectied Legacy Systems

2002-10 Brian Sheppard (UM) Towards Perfect Play of Scrabble

2002-11 Wouter Wijngaards (VU) Agent Based Modelling of Dynamics: Biological and Organisational Applications

2002-12 Albrecht Schmidt (UvA) Processing XML in Database Systems

2002-13 Hongjing Wu (TU/e) A Reference Architecture for Adaptive Hypermedia Applications

2002-14 Wieke de Vries (UU) Agent Interaction: Abstract Approaches to Modelling, Programming and Verifying Multi-Agent Systems

2002-15 Rik Eshuis (UT) Semantics and Verication of UML Activity Diagrams for Workow Modelling

2002-16 Pieter van Langen (VU) The Anatomy of Design: Foundations, Models and Applications

2002-17 Stefan Manegold (UvA) Understanding, Modeling, and Improving Main-Memory Database Performance

2003

2003-1 Heiner Stuckenschmidt (VU) Ontology-Based Information Sharing in Weakly Structured Environments

2003-2 Jan Broersen (VU) Modal Action Logics for Reasoning About Reactive Systems

2003-3 Martijn Schuemie (TUD) Human-Computer Interaction and Presence in Virtual Reality Exposure Therapy

2003-4 Milan Petkovìc (UT) Content-Based Video Retrieval Supported by Database Technology

2003-5 Jos Lehmann (UvA) Causation in Articial Intelligence and Law - A Modelling Approach

2003-6 Boris van Schooten (UT) Development and Specication of Virtual Environments

2003-7 Machiel Jansen (UvA) Formal Explorations of Knowledge Intensive Tasks

2003-8 Yong-Ping Ran (UM) Repair-Based Scheduling

2003-9 Rens Kortmann (UM) The Resolution of Visually Guided Behaviour

2003-10 Andreas Lincke (UT) Electronic Business Negotiation: Some Experimental Studies on the Interaction between Medium, Innovation Context and Cult

2003-11 Simon Keizer (UT) Reasoning under Uncertainty in Natural Language Dialogue using Bayesian Networks

2003-12 Roeland Ordelman (UT) Dutch Speech Recognition in Multimedia Information Retrieval

2003-13 Jeroen Donkers (UM) Nosce Hostem - Searching with Opponent Models

2003-14 Stijn Hoppenbrouwers (KUN) Freezing Language: Conceptualisation Processes across ICT-Supported Organisations

2003-15 Mathijs de Weerdt (TUD) Plan Merging in Multi-Agent Systems

2003-16 Menzo Windhouwer (CWI) Feature Grammar Systems - Incremental Maintenance of Indexes to Digital Media Warehouse

2003-17 David Jansen (UT) Extensions of Statecharts with Probability, Time, and Stochastic Timing

2003-18 Levente Kocsis (UM) Learning Search Decisions

2004

2004-1 Virginia Dignum (UU) A Model for Organizational Interaction: Based on Agents, Founded in Logic

2004-2 Lai Xu (UvT) Monitoring Multi-party Contracts for E-business

2004-3 Perry Groot (VU) A Theoretical and Empirical Analysis of Approximation in Symbolic Problem Solving

2004-4 Chris van Aart (UvA) Organizational Principles for Multi-Agent Architectures

2004-5 Viara Popova (EUR) Knowledge Discovery and Monotonicity

2004-6 Bart-Jan Hommes (TUD) The Evaluation of Business Process Modeling Techniques

2004-7 Elise Boltjes (UM) VoorbeeldI G Onderwijs; Voorbeeldgestuurd Onderwijs, een
Opstap naar Abstract Denken, vooral voor Meisjes

2004-8 Joop Verbeek (UM) Politie en de Nieuwe Internationale Informatiemarkt, Grensregionale Politiële Gegevensuitwisseling en Digitale Expertise

2004-9 Martin Caminada (VU) For the Sake of the Argument; Explorations into Argument-based Reasoning

2004-10 Suzanne Kabel (UvA) Knowledge-rich Indexing of Learning-objects

2004-11 Michel Klein (VU) Change Management for Distributed Ontologies

2004-12 The Duy Bui (UT) Creating Emotions and Facial Expressions for Embodied Agents

2004-13 Wojciech Jamroga (UT) Using Multiple
Models of Reality: On Agents who Know how to Play

2004-14 Paul Harrenstein (UU) Logic in Conict. Logical Explorations in Strategic Equilibrium

2004-15 Arno Knobbe (UU) Multi-Relational Data Mining

2004-16 Federico Divina (VU) Hybrid Genetic Relational Search for Inductive Learning

2004-17 Mark Winands (UM) Informed Search in Complex Games

2004-18 Vania Bessa Machado (UvA) Supporting the Construction of Qualitative Knowledge Models

2004-19 Thijs Westerveld (UT) Using generative probabilistic models for multimedia
retrieval

2004-20 Madelon Evers (Nyenrode) Learning from Design: facilitating multidisciplinary
design teams

2005

2005-1 Floor Verdenius (UvA) Methodological Aspects of Designing Induction-Based Applications

2005-2 Erik van der Werf (UM) AI techniques for the game of Go

2005-3 Franc Grootjen (RUN) A Pragmatic Approach to the Conceptualisation of Language

2005-4 Nirvana Meratnia (UT) Towards Database Support for Moving Object data

2005-5 Gabriel Infante-Lopez (UvA) Two-Level
Probabilistic Grammars for Natural Language Parsing

2005-6 Pieter Spronck (UM) Adaptive Game AI

2005-7 Flavius Frasincar (TU/e) Hypermedia Presentation Generation for Semantic Web Information Systems

2005-8 Richard Vdovjak (TU/e) A Model-driven Approach for Building Distributed Ontology-based Web Applications

2005-9 Jeen Broekstra (VU) Storage, Querying
and Inferencing for Semantic Web Languages

2005-10 Anders Bouwer (UvA) Explaining Behaviour: Using Qualitative Simulation in Interactive Learning Environments

2005-11 Elth Ogston (VU) Agent Based Matchmaking and Clustering - A Decentralized Approach to Search

2005-12 Csaba Boer (EUR) Distributed Simulation in Industry

2005-13 Fred Hamburg (UL) Een Computermodel voor het Ondersteunen van Euthanasiebeslissingen

2005-14 Borys Omelayenko (VU) Web-Service configuration on the Semantic Web; Exploring how semantics meets pragmatics

2005-15 Tibor Bosse (VU) Analysis of the Dynamics of Cognitive Processes

2005-16 Joris Graaumans (UU) Usability of XML Query Languages

2005-17 Boris Shishkov (TUD) Software Specication Based on Re-usable Business Components

2005-18 Danielle Sent (UU) Test-selection strategies for probabilistic networks

2005-19 Michel van Dartel (UM) Situated Representation

2005-20 Cristina Coteanu (UL) Cyber Consumer Law, State of the Art and Perspectives

2005-21 Wijnand Derks (UT) Improving Concurrency and Recovery in Database Systems by Exploiting Application Semantics

2006

2006-1 Samuil Angelov (TU/e) Foundations of B2B Electronic Contracting

2006-2 Cristina Chisalita (VU) Contextual issues in the design and use of information technology in organizations

2006-3 Noor Christoph (UvA) The role of metacognitive skills in learning to solve problems

2006-4 Marta Sabou (VU) Building Web Service Ontologies

2006-5 Cees Pierik (UU) Validation Techniques for Object-Oriented Proof Outlines

2006-6 Ziv Baida (VU) Software-aided Service Bundling - Intelligent Methods & Tools for
Graphical Service Modeling

2006-7 Marko Smiljanic (UT) XML schema matching - balancing efciency and effectiveness by means of clustering

2006-8 Eelco Herder (UT) Forward, Back and
Home Again - Analyzing User Behavior on the Web

2006-9 Mohamed Wahdan (UM) Automatic Formulation of the Auditors Opinion

2006-10 Ronny Siebes (VU) Semantic Routing in Peer-to-Peer Systems

2006-11 Joeri van Ruth (UT) Flattening Queries over Nested Data Types

2006-12 Bert Bongers (VU) Interactivation - Towards an ecology of people, our technological environment, and the arts

2006-13 Henk-Jan Lebbink (UU) Dialogue and Decision Games for Information Exchanging Agents

2006-14 Johan Hoorn (VU) Software Requirements: Update, Upgrade, Redesign - towards a Theory of Requirements Change

2006-15 Rainer Malik (UU) CONAN: Text Mining in the Biomedical Domain

2006-16 Carsten Riggelsen (UU) Approximation Methods for Efcient Learning of Bayesian Networks

2006-17 Stacey Nagata (UU) User Assistance for Multitasking with Interruptions on a Mobile Device

2006-18 Valentin Zhizhkun (UvA) Graph transformation for Natural Language Processing

2006-19 Birna van Riemsdijk (UU) Cognitive Agent Programming: A Semantic Approach

2006-20 Marina Velikova (UvT) Monotone models for prediction in data mining

2006-21 Bas van Gils (RUN) Aptness on the Web

2006-22 Paul de Vrieze (RUN) Fundaments of Adaptive Personalisation

2006-23 Ion Juvina (UU) Development of a Cognitive Model for Navigating on the Web

2006-24 Laura Hollink (VU) Semantic Annotation for Retrieval of Visual Resources

2006-25 Madalina Drugan (UU) Conditional log-likelihood MDL and Evolutionary MCMC

2006-26 Vojkan Mihajlovic (UT) Score Region Algebra: A Flexible Framework for Structured Information Retrieval

2006-27 Stefano Bocconi (CWI) Vox Populi: generating video documentaries from semantically annotated media repositories

2006-28 Borkur Sigurbjornsson (UvA) Focused Information Access using XML Element Retrieval

2007

2007-1 Kees Leune (UvT) Access Control and Service-Oriented Architectures

2007-2 Wouter Teepe (RUG) Reconciling Information Exchange and Condentiality: A Formal Approach

2007-3 Peter Mika (VU) Social Networks and the Semantic Web

2007-4 Jurriaan van Diggelen (UU) Achieving Semantic Interoperability in Multi-agent Systems: a dialogue-based approach

2007-5 Bart Schermer (UL) Software Agents, Surveillance, and the Right to Privacy: a Legislative Framework for Agent-enabled Surveillance

2007-6 Gilad Mishne (UvA) Applied Text Analytics for Blogs

2007-7 Natasa Jovanovic (UT) To Whom It May Concern - Addressee Identication in Face-to-Face Meetings

2007-8 Mark Hoogendoorn (VU) Modeling of Change in Multi-Agent Organizations

2007-9 David Mobach (VU) Agent-Based Mediated Service Negotiation

2007-10 Huib Aldewereld (UU) Autonomy vs. Conformity: an Institutional Perspective on Norms and Protocols

2007-11 Natalia Stash (TU/e) Incorporating Cognitive/Learning Styles in a General-Purpose Adaptive Hypermedia System

2007-12 Marcel van Gerven (RUN) Bayesian Networks for Clinical Decision Support: A Rational Approach to Dynamic Decision-Making under Uncertainty

2007-13 Rutger Rienks (UT) Meetings in Smart Environments; Implications of Progressing Technology

2007-14 Niek Bergboer (UM) Context-Based Image Analysis

2007-15 Joyca Lacroix (UM) NIM: a Situated Computational Memory Model

2007-16 Davide Grossi (UU) Designing Invisible Handcuffs. Formal investigations in Institutions and Organizations for Multi-agent Systems

2007-17 Theodore Charitos (UU) Reasoning with Dynamic Networks in Practice

2007-18 Bart Orriens (UvT) On the development and management of adaptive business collaborations

2007-19 David Levy (UM) Intimate relationships with articial partners

2007-20 Slinger Jansen (UU) Customer Conguration Updating in a Software Supply Network

2007-21 Karianne Vermaas (UU) Fast diffusion and broadening use: A research on residential adoption and usage of broadband internet in the Netherlands between 2001 and 2005

2007-22 Zlatko Zlatev (UT) Goal-oriented design of value and process models from patterns

2007-23 Peter Barna (TU/e) Specication of Application Logic in Web Information Systems

2007-24 Georgina Ramírez Camps (CWI) Structural Features in XML Retrieval

2007-25 Joost Schalken (VU) Empirical Investigations in Software Process Improvement

2008

2008-1 Katalin Boer-Sorbán (EUR) Agent-Based Simulation of Financial Markets: A modular, continuous- time approach

2008-2 Alexei Sharpanskykh (VU) On Computer-Aided Methods for Modeling and Analysis of Organizations

2008-3 Vera Hollink (UvA) Optimizing hierarchical menus: a usage-based approach

2008-4 Ander de Keijzer (UT) Management of Uncertain Data - towards unattended integration

2008-5 Bela Mutschler (UT) Modeling and simulating causal dependencies on process-aware information systems from a cost perspective

2008-6 Arjen Hommersom (RUN) On the Application of Formal Methods to Clinical Guidelines, an Articial Intelligence Perspective

2008-7 Peter van Rosmalen (OU) Supporting the tutor in the design and support of adaptive e-learning

2008-8 Janneke Bolt (UU) Bayesian Networks: Aspects of Approximate Inference

2008-9 Christof van Nimwegen (UU) The paradox of the guided user: assistance can be counter-effective

2008-10 Wauter Bosma (UT) Discourse oriented Summarization

2008-11 Vera Kartseva (VU) Designing Controls for Network Organizations: a Value-Based Approach

2008-12 Jozsef Farkas (RUN) A Semiotically oriented Cognitive Model of Knowledge Representation

2008-13 Caterina Carraciolo (UvA) Topic Driven Access to Scientic Handbooks

2008-14 Arthur van Bunningen (UT) Context-Aware Querying; Better Answers with Less Effort

2008-15 Martijn van Otterlo (UT) The Logic of Adaptive Behavior: Knowledge Representation and Algorithms for the Markov Decision Process Framework in First-Order Domains

2008-16 Henriette van Vugt (VU) Embodied Agents from a Users Perspective

2008-17 Martin Opt Land (TUD) Applying Architecture and Ontology to the Splitting and Allying of Enterprises

2008-18 Guido de Croon (UM) Adaptive Active Vision

2008-19 Henning Rode (UT) From document to entity retrieval: improving precision and performance of focused text search

2008-20 Rex Arendsen (UvA) Geen bericht, goed bericht. Een onderzoek naar de effecten van de introductie van elektronisch berichtenverkeer met een overheid op de administratieve lasten van bedrijven

2008-21 Krisztian Balog (UvA) People search in the enterprise

2008-22 Henk Koning (UU) Communication of IT-architecture

2008-23 Stefan Visscher (UU) Bayesian network models for the management of ventilator-associated pneumonia

2008-24 Zharko Aleksovski (VU) Using background knowledge in ontology matching

2008-25 Geert Jonker (UU) Efcient and Equitable exchange in air trafc management plan repair using spender-signed currency

2008-26 Marijn Huijbregts (UT) Segmentation, diarization and speech transcription: surprise data unraveled

2008-27 Hubert Vogten (OU) Design and implementation strategies for IMS learning design

2008-28 Ildikó Flesh (RUN) On the use of independence relations in Bayesian networks

2008-29 Dennis Reidsma (UT) Annotations and subjective machines - Of annotators, embodied agents, users, and other humans

2008-30 Wouter van Atteveldt (VU) Semantic network analysis: techniques for extracting, representing and querying media content

2008-31 Loes Braun (UM) Pro-active medical information retrieval

2008-32 Trung Hui (UT) Toward affective dialogue management using partially observable Markov decision processes

2008-33 Frank Terpstra (UvA) Scientic worflow design; theoretical and practical issues

2008-34 Jeroen De Knijf (UU) Studies in Frequent Tree Mining

2008-35 Benjamin Torben-Nielsen (UvT) Dendritic morphology: function shapes structure

2009

2009-1 Rasa Jurgenelaite (RUN) Symmetric Causal Independence Models

2009-2 Willem Robert van Hage (VU) Evaluating Ontology-Alignment Techniques

2009-3 Hans Stol (UvT) A Framework for Evidence-based Policy Making Using IT

2009-4 Josephine Nabukenya (RUN) Improving the Quality of Organisational Policy Making using Collaboration Engineering

2009-5 Sietse Overbeek (RUN) Bridging Supply and Demand for Knowledge Intensive Tasks - Based on Knowledge, Cognition, and Quality

2009-6 Muhammad Subianto (UU) Understanding Classication

2009-7 Ronald Poppe (UT) Discriminative Vision-Based Recovery and Recognition of Human Motion

2009-8 Volker Nannen (VU) Evolutionary Agent-Based Policy Analysis in Dynamic Environments

2009-9 Benjamin Kanagwa (RUN) Design, Discovery and Construction of Service-oriented Systems

2009-10 Jan Wielemaker (UvA) Logic programming for knowledge-intensive interactive applications

2009-11 Alexander Boer (UvA) Legal Theory, Sources of Law and the Semantic Web

2009-12 Peter Massuthe (TU/e) Operating Guidelines for Services

2009-13 Steven de Jong (UM) Fairness in Multi-Agent Systems

2009-14 Maksym Korotkiy (VU) From ontology-enabled services to service-enabled ontologies (making ontologies work in e-science with ONTO-SOA)

2009-15 Rinke Hoekstra (UvA) Ontology Representation - Design Patterns and Ontologies that Make Sense

2009-16 Fritz Reul (UvT) New Architectures in Computer Chess

2009-17 Laurens van der Maaten (UvT) Feature Extraction from Visual Data

2009-18 Fabian Groffen (CWI) Armada, An Evolving Database System

2009-19 Valentin Robu (CWI) Modeling Preferences, Strategic Reasoning and Collaboration in Agent-Mediated Electronic Markets

2009-20 Bob van der Vecht (UU) Adjustable Autonomy: Controling Influences on Decision Making

2009-21 Stijn Vanderlooy (UM) Ranking and Reliable Classification

2009-22 Pavel Serdyukov (UT) Search For Expertise: Going beyond direct evidence

2009-23 Peter Hofgesang (VU) Modelling Web Usage in a Changing Environment

2009-24 Annerieke Heuvelink (VU) Cognitive Models for Training Simulations

2009-25 Alex van Ballegooij (CWI) RAM: Array Database Management through Relational Mapping

2009-26 Fernando Koch (UU) An Agent-Based Model for the Development of Intelligent Mobile Services

2009-27 Christian Glahn (OU) Contextual Support of social Engagement and Reflection on the Web

2009-28 Sander Evers (UT) Sensor Data Management with Probabilistic Models

2009-29 Stanislav Pokraev (UT) Model-Driven Semantic Integration of Service-Oriented Applications

2009-30 Marcin Zukowski (CWI) Balancing vectorized query execution with bandwidth-optimized storage

2009-31 Sofiya Katrenko (UvA) A Closer Look at Learning Relations from Text

2009-32 Rik Farenhorst and Remco de Boer (VU) Architectural Knowledge Management:
Supporting Architects and Auditors

2009-33 Khiet Truong (UT) How Does Real Affect Affect Affect Recognition In Speech?

2009-34 Inge van de Weerd (UU) Advancing in Software Product Management: An Incremental Method Engineering Approach

2009-35 Wouter Koelewijn (UL) Privacy en Politiegegevens; Over geautomatiseerde normatieve informatie-uitwisseling

2009-36 Marco Kalz (OUN) Placement Support for Learners in Learning Networks

2009-37 Hendrik Drachsler (OUN) Navigation Support for Learners in Informal Learning Networks

2009-38 Riina Vuorikari (OU) Tags and self-organisation: a metadata ecology for learning resources in a multilingual context

2009-39 Christian Stahl (TUE, Humboldt-Universität zu Berlin) Service Substitution - A Behavioral Approach Based on Petri Nets

2009-40 Stephan Raaijmakers (UvT) Multinomial Language Learning: Investigations into the Geometry of Language

2009-41 Igor Berezhnyy (UvT) Digital Analysis of Paintings

2009-42 Toine Bogers (UvT) Recommender Systems for Social Bookmarking

2010

2010-1 Matthijs van Leeuwen (UU) Patterns that Matter

2010-2 Ingo Wassink (UT) Work flows in Life Science

2010-3 Joost Geurts (CWI) A Document Engineering Model and Processing Framework for Multimedia documents

2010-4 Olga Kulyk (UT) Do You Know What I Know? Situational Awareness of Co-located Teams in Multidisplay Environments

2010-5 Claudia Hauff (UT) Predicting the Effectiveness of Queries and Retrieval Systems

2010-6 Sander Bakkes (UvT) Rapid Adaptation of Video Game AI

2010-7 Wim Fikkert (UT) A Gesture interaction at a Distance

2010-8 Krzysztof Siewicz (UL) Towards an Improved Regulatory Framework of Free Software. Protecting user freedoms in a world of software communities and eGovernments

2010-9 Hugo Kielman (UL) A Politiële gegevensverwerking en Privacy, Naar een effectieve waarborging

2010-10 Rebecca Ong (UL) Mobile Communication and Protection of Children

2010-11 Adriaan Ter Mors (TUD) The world according to MARP: Multi-Agent Route
Planning

2010-12 Susan van den Braak (UU) Sensemaking software for crime analysis

2010-13 Gianluigi Folino (RUN) High Performance Data Mining using Bio-inspired techniques

2010-14 Sander van Splunter (VU) Automated Web Service Reconfiguration

2010-15 Lianne Bodenstaff (UT) Managing Dependency Relations in Inter-Organizational Models

2010-16 Sicco Verwer (TUD) Efficient Identification of Timed Automata, theory and practice

2010-17 Spyros Kotoulas (VU) Scalable Discovery of Networked Resources: Algorithms, Infrastructure, Applications

2010-18 Charlotte Gerritsen (VU) Caught in the Act: Investigating Crime by Agent-Based Simulation

2010-19 Henriette Cramer (UvA) People's Responses to Autonomous and Adaptive Systems

2010-20 Ivo Swartjes (UT) Whose Story Is It Anyway? How Improv Informs Agency and Authorship of Emergent Narrative

2010-21 Harold van Heerde (UT) Privacy-aware
data management by means of data degradation

2010-22 Michiel Hildebrand (CWI) End-user Support for Access to Heterogeneous Linked Data

2010-23 Bas Steunebrink (UU) The Logical Structure of Emotions

2010-24 DmytroTykhonov Designing Generic and Efficient Negotiation Strategies

2010-25 Zulfiqar Ali Memon (VU) Modelling Human-Awareness for Ambient Agents: A Human Mindreading Perspective

2010-26 Ying Zhang (CWI) XRPC: Efficient Distributed Query Processing on Heterogeneous XQuery Engines

2010-27 Marten Voulon (UL) Automatisch contracteren

2010-28 Arne Koopman (UU) Characteristic Relational Patterns

2010-29 Stratos Idreos(CWI) Database Cracking: Towards Auto-tuning Database Kernels

2010-30 Marieke van Erp (UvT) Accessing Natural History: Discoveries in Data Cleaning, Structuring, and Retrieval

2010-31 Viktor de Boer (UvA) Ontology Enrichment from Heterogeneous Sources on the Web

2010-32 Marcel Hiel (UvT) An Adaptive Service Oriented Architecture: Automatically solving Interoperability Problems

2010-33 Robin Aly (UT) Modeling Representation Uncertainty in Concept-Based Multimedia Retrieval

2010-34 Teduh Dirgahayu (UT) Interaction Design in Service Compositions

2010-35 Dolf Trieschnigg (UT) Proof of Concept: Concept-based Biomedical Information Retrieval

2010-36 Jose Janssen (OU) Paving the Way for Lifelong Learning; Facilitating competence development through a learning path specification

2010-37 Niels Lohmann (TUE) Correctness of services and their composition

2010-38 Dirk Fahland (TUE) From Scenarios to components

2010-39 Ghazanfar Farooq Siddiqui (VU) Integrative modeling of emotions in virtual agents

2010-40 Mark van Assem (VU) Converting and Integrating Vocabularies for the Semantic Web