



universität  
wien

# DISSERTATION

Titel der Dissertation

**Supporting Web Vocabulary Development by  
Automated Quality Checks**

Verfasser

Dipl.- Ing. Christian Mader

angestrebter akademischer Grad

Doktor der technischen Wissenschaften (Dr. techn.)

Wien, im Juli 2015

Studienkennzahl lt. Studienblatt: A 786 880

Dissertationsgebiet lt. Studienblatt: Informatik

Betreuer: Prof. Dr. Wolfgang Klas



# Abstract

On the Web, controlled vocabularies have proved as a useful tool for knowledge organization and search and retrieval tasks. They are used, e.g., to index documents, support navigation, or enable queries that span multiple datasets as they help to achieve a common understanding on the semantics of resources. The Simple Knowledge Organization System (SKOS) introduces a data schema that provides a standard set of classes and relations which can be used to model controlled vocabularies. SKOS is based on RDF, a standard way for publishing datasets on the Web, and therefore allows to express controlled vocabularies as *Web vocabularies*, utilizing the Linked Data paradigm.

Despite the existence of automated solutions, Web vocabulary development in most cases remains an intellectual process performed by human contributors. As a consequence, errors and shortcomings can slip in, causing quality problems. Especially in collaborative development environments, overseeing all changes for the purpose of quality assurance can become difficult for human users. Another aspect is that the value of datasets on the Web increases if linked to other online resources which provide additional information. Given the vast amount of Web vocabularies of various sizes and complexity available on the Web, quality is a crucial factor for deciding whether to select a particular vocabulary on the Web for linking or reuse.

The impact of quality issues in Web vocabularies can be manifold. They can impair search precision and recall, guide users to irrelevant information, break automated processing applications like information retrieval, or decrease understandability of the vocabulary content for human users. In addition, Web vocabulary developers want to link their datasets to vocabularies of good quality that fit and support their requirements.

Numerous guidelines on development and evaluation of controlled vocabularies currently exist, covering both “traditional” controlled vocabularies and Web vocabularies. However, many of these publications suggest intellectual checks that require further domain knowledge. Existing Linked Data publication guidelines mostly focus on syntactic and formal constraint violations using reasoning techniques.

In this thesis, we reviewed existing work on controlled vocabulary development and adopted quality-related guidelines for application to Web vocabularies expressed using SKOS. We focused on generally applicable, less intellectually-loaded checks that can be automatically computed and go beyond formal data-level constraints. As one of the contributions of this thesis we provide a catalog of potential quality issues, which is the result of a literature review and expert feedback through a survey we conducted. In a case study we show to what extent currently available Web vocabularies are affected by these quality issues and provide best practices for expressing and publishing Web vocabularies. We furthermore contribute tools that can process Web vocabularies and automatically report occurrences of quality issues from the catalog. As the notion of quality is also to a large degree usage-scenario dependent and subjective, the tools can be integrated into vocabulary development processes in order to leave the final judgment of appropriateness up to human Web vocabulary developers.

Our studies showed that Web vocabularies that are in development as well as already publicly available Web vocabularies are affected by the quality issues we defined in our catalog. Communicating these findings to the vocabulary developers led to improvements in some cases. The tools developed in the context of this work are actively used, adopted, and extended by Web vocabulary developers and the Linked Data community. In another case study we also show that integrating automatic quality checks in a Web vocabulary development process helps in reducing the number of observed quality issues.

# Zusammenfassung

Kontrollierte Vokabulare haben sich als hilfreiche Werkzeuge zur Organisation von Wissen sowie zum Suchen und Abrufen von Informationen im Web bewährt. Sie werden beispielsweise dazu verwendet um Dokumente zu indizieren, als Navigationshilfe, oder um systemübergreifende Abfragen von Datensätzen zu realisieren. Letzteres wird durch ihre Eigenschaft, ein gemeinsames Verständnis der semantischen Bedeutung einer Ressource herzustellen, ermöglicht. Mit der Einführung des Simple Knowledge Organization System (SKOS), ist ein Datenschema verfügbar, das einen standardisierten Grundstock an Klassen und Beziehungen bereitstellt die dazu verwendet werden können, kontrollierte Vokabulare auszudrücken. SKOS basiert auf RDF, einem ebenfalls standardisierten Format zum Austausch von Datensätzen im Web, und erlaubt es daher, Webvokabulare gemäß der Linked Data Prinzipien auszudrücken.

Obwohl automatisierte Lösungen existieren, ist die Erstellung von Webvokabularen nach wie vor in den meisten Fällen ein intellektueller Prozess, der manuell erfolgt. Dementsprechend können sich Fehler und Unzulänglichkeiten in die Webvokabulare einschleichen, die Qualitätsprobleme verursachen. Besonders in kollaborativen Umgebungen ist es schwierig, alle eingepflegten Änderungen an einem Webvokabular zwecks Qualitätssicherung im Auge zu behalten. Ein zusätzlicher Aspekt ist, dass der Informationsgehalt von Datensätzen durch Hinzufügen von Links zu anderen Ressourcen im Web steigt, da letztere zusätzliche Informationen einbringen. Durch die große Anzahl an verfügbaren Webvokabularen verschiedener Größe und Komplexität ist die Qualität dieser Vokabulare auch ein wichtiger Faktor der die Entscheidung, ob das Vokabular verlinkt oder wiederverwendet werden soll, beeinflusst.

Die Auswirkungen von Qualitätsproblemen in Webvokabularen können vielfältig sein. Sie beeinträchtigen beispielsweise die Genauigkeit und Trefferquote von Suchanfragen, leiten Benutzer zu irrelevanter Information, behindern das automatisierte Abschöpfen von Daten oder verringern die Verständlichkeit des Inhalts von Webvokabularen für menschliche Benutzer. Zusätzlich streben Vokabularentwickler danach, ihre Datensätze

auch zu möglichst hochqualitativen “externen” Webvokabularen zu verlinken, die ihre Erfordernisse hinsichtlich Qualität erfüllen.

Es existieren zahllose Richtlinien zur Entwicklung und Evaluierung kontrollierter Vokabulare, die sowohl “traditionelle” als auch Webvokabulare behandeln. Viele dieser Publikationen schlagen Qualitätskriterien vor, deren Evaluierung zusätzliches Domänenwissen benötigt. Verfügbare Vorgaben zum Publizieren von Linked Data beschränken sich hingegen meistens auf syntaktische und formale Korrektheit der Datensätze.

Wir haben in dieser Dissertation existierende Publikationen hinsichtlich Richtlinien betreffend Vokabularqualität untersucht und diese an die Erfordernisse und Möglichkeiten von Webvokabularen adaptiert. Dabei konzentrieren wir uns auf allgemein anwendbare, automatisch auswertbare Kriterien die über existierende formale Kriterien hinausgehen und kein zusätzliches Domänenwissen erfordern. Ein zentraler Beitrag unserer Arbeit stellt ein Katalog von potenziellen Qualitätsproblemen dar, die in Webvokabularen auftreten können. Der Katalog ist einerseits Ergebnis unserer Literaturrecherche und beruht andererseits auf Erfahrungen die wir mittels einer Expertenumfrage sammeln konnten. Mittels einer Fallstudie untersuchen wir zu welchem Grad aktuell verfügbare Webvokabulare von den Qualitätsproblemen in unserem Katalog betroffen sind und entwickeln Empfehlungen zu Formulierung und Publikation von Webvokabularen. Einen weiteren Beitrag unserer Arbeit stellen die entwickelten Werkzeuge dar, die Webvokabulare auf das Auftreten von Qualitätsproblemen aus unserem Katalog untersuchen und eine Auswertung beziehungsweise Benachrichtigung generieren. Da die Auffassung von Qualität auch in hohem Maße vom Einsatzzweck und subjektiven Entscheidungen abhängt, können unsere Werkzeuge in Entwicklungsprozesse von Webvokabularen eingebunden werden, um die weitere Behandlung der gefundenen Qualitätsprobleme den Vokabularentwicklern zu überlassen.

Unsere Fallstudien haben gezeigt, dass sowohl Webvokabulare die sich in Entwicklung befinden, als auch jene die bereits öffentlich verfügbar sind, von den Qualitätsproblemen die wir in unserem Katalog definieren, betroffen sind. Unsere Rückmeldungen dieser Probleme an die Entwickler haben in manchen Fällen zu Verbesserungen der Vokabulare geführt. Die Werkzeuge die im Kontext dieser Arbeit entwickelt wurden, werden von Webvokabularentwicklern und der Linked Data Gemeinschaft verwendet und auch teilweise erweitert. In einer weiteren Fallstudie konnten wir zeigen, dass die Integration automatisierter Prüfung auf Qualitätsprobleme in Webvokabular-Entwicklungsprozessen helfen kann, die Zahl der gefundenen Qualitätsprobleme im fertigen Webvokabular zu reduzieren.

# Acknowledgements

First of all, I would like to thank my supervisor, Prof. Dr. Wolfgang Klas, for guiding me through the whole process of creating this thesis and always providing time for discussion whenever I requested it. Furthermore, my special thanks goes to Dr. Bernhard Haslhofer with whom I authored a number of papers and who, as an experienced researcher, provided valuable input and feedback on my work. The countless fruitful discussions with him were an important factor for helping me to complete this thesis. I also want to express my gratitude to my colleagues at Semantic Web Company, especially to Andreas Blumauer and Martin Kaltenböck, for giving me the opportunity to not only incorporate parts of my research in a commercial product, but also to participate in various research projects that allowed me to further pursue my research. Special thanks goes to all my colleagues I had the honor to work together in the course of authoring research papers or cooperating in scientific projects, especially Osma Suominen, Christian Wartena, Jürgen Jakobitsch, and Helmut Nagy. Last but not least I want to thank Ioanna Lytra for doing an excellent job in proof-reading this thesis and my family and friends, for their support in various ways.





# Table of Contents

<b>List of Figures</b>	<b>xi</b>
<b>List of Tables</b>	<b>xiii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivating Examples . . . . .	3
1.1.1 The MEKETRE Project . . . . .	3
1.1.2 Existing Controlled Vocabularies . . . . .	5
1.2 Problem Description . . . . .	6
1.3 Contributions . . . . .	10
1.4 Methodology . . . . .	12
1.4.1 Existing Literature . . . . .	12
1.4.2 Public Web Vocabularies . . . . .	13
1.4.3 Expert Consultation . . . . .	15
1.5 Organization . . . . .	16
<b>2 Background and Related Work</b>	<b>19</b>
2.1 Controlled Vocabularies . . . . .	19
2.1.1 Historical Outline . . . . .	19
2.1.2 The Need for Vocabulary Control . . . . .	21
2.1.3 Types of Controlled Vocabularies . . . . .	22
2.1.4 Web vocabularies - Vocabulary Control in the Context of Linked Data . . . . .	24
2.1.5 Examples of Web Vocabularies . . . . .	27
2.1.6 The Notion of Quality . . . . .	28
2.2 Controlled Vocabulary Evaluation and Quality Assurance . . . . .	28
2.3 Quality of Linked Data . . . . .	30
2.3.1 Ontology Engineering and Evaluation . . . . .	30
2.3.2 Automated Validation Approaches . . . . .	31
2.3.3 Linked Data Notifications . . . . .	33
2.3.4 Approaches for Evaluating Quality Assurance Methods . . . . .	34
2.4 Quality of Web Vocabularies . . . . .	35
2.4.1 Proprietary Approaches . . . . .	35
2.4.2 Data Quality and SKOS . . . . .	36
2.4.3 Tools Related to Web Vocabulary Checking . . . . .	37
2.5 Connecting to Related Work . . . . .	38
2.5.1 Relation to Ontology Evaluation Approaches . . . . .	38

2.5.2	Notification Approaches . . . . .	39
2.5.3	Controlled Vocabulary Evaluation and Quality Assurance . . . . .	40
2.5.4	Quality of Web Vocabularies . . . . .	41
2.6	Summary . . . . .	42
<b>3</b>	<b>Formal Definition of Quality Issues</b>	<b>43</b>
3.1	Catalog of Quality Issues . . . . .	43
3.1.1	Labeling and Documentation Issues . . . . .	46
3.1.2	Structural Issues . . . . .	54
3.1.3	Linked Data Specific Issues . . . . .	62
3.1.4	SKOS Consistency Issues . . . . .	66
3.2	Summary . . . . .	68
<b>4</b>	<b>Quality Checking Techniques</b>	<b>71</b>
4.1	General Design Considerations . . . . .	72
4.2	qSKOS - On-demand Vocabulary Quality Checking . . . . .	73
4.2.1	Implementation . . . . .	74
4.2.2	Quality Issue Evaluation . . . . .	74
4.2.3	PoolParty Product Integration . . . . .	83
4.2.4	Online SKOS Vocabulary Quality Checker . . . . .	85
4.3	rsine - On-change Vocabulary Quality Checking . . . . .	86
4.3.1	Requirements and Design Considerations . . . . .	88
4.3.2	Approach . . . . .	88
4.3.3	Subscribing for Notifications . . . . .	90
4.3.4	Implementation . . . . .	92
4.3.5	Management-related Notifications . . . . .	94
4.3.6	Quality Notifications . . . . .	95
4.4	Summary . . . . .	97
<b>5</b>	<b>Case Studies and Findings</b>	<b>99</b>
5.1	Expert Perception of Quality Issues . . . . .	100
5.1.1	Survey Structure and Question Design . . . . .	100
5.1.2	Survey Response Analysis Methodology . . . . .	101
5.1.3	Usage Scenarios . . . . .	102
5.1.4	Labeling and Documentation Issues . . . . .	103
5.1.5	Structural Issues . . . . .	106
5.1.6	Linked Data Specific Issues . . . . .	107
5.1.7	Responses . . . . .	108
5.1.8	Summarized Findings . . . . .	109
5.1.9	Recommendations for Best Practices . . . . .	109
5.2	Quality Analysis of Existing Vocabularies . . . . .	113
5.2.1	Vocabulary Statistics . . . . .	115
5.2.2	Labeling and Documentation Issues . . . . .	116
5.2.3	Structural Issues . . . . .	123
5.2.4	Linked Data Specific Issues . . . . .	128
5.2.5	Adherence to SKOS Integrity Conditions . . . . .	131
5.3	Online Vocabulary Checker . . . . .	133

5.3.1	Methodology . . . . .	133
5.3.2	Overall Issue Occurrences . . . . .	134
5.3.3	Changes in Quality Issue Occurrences . . . . .	135
5.3.4	Degradations and Unchanged Vocabularies . . . . .	138
5.3.5	Service Usage . . . . .	140
5.4	Automated Quality Checking in a Teaching Context . . . . .	140
5.4.1	Methodology . . . . .	141
5.4.2	qSKOS Quality Analysis Results . . . . .	144
5.5	Summary . . . . .	148
<b>6</b>	<b>Conclusions and Future Work</b>	<b>151</b>
6.1	Summary . . . . .	151
6.2	Discussion . . . . .	153
6.2.1	Expert Perception of Quality Issues . . . . .	153
6.2.2	Quality Analysis of Existing Vocabularies . . . . .	153
6.2.3	Online Vocabulary Checker . . . . .	154
6.2.4	Automated Quality Checking in a Teaching Context . . . . .	155
6.3	Limitations of Our Approach . . . . .	156
6.3.1	Catalog of Quality Issues and On-demand Vocabulary Quality Check- ing . . . . .	156
6.3.2	rsine - On-change Vocabulary Quality Checking . . . . .	157
6.3.3	Usability Issues . . . . .	158
6.4	Future Work . . . . .	159
	<b>Bibliography</b>	<b>161</b>
	<b>Appendices</b>	<b>167</b>
	<b>A Implemented Tools Output and Configuration</b>	<b>169</b>



# List of Figures

1.1	Auto-completion at the STW Web page (Version 8.06).	5
1.2	Improved auto-completion functionality for STW (Version 8.12).	6
1.3	Unconnected components in a vocabulary. The arrows indicate the hierarchically broader concept.	6
1.4	Quality issue catalog and tool development approach.	13
4.1	Overview of the seven integrated quality issues (three of them are detected in this case).	85
4.2	Conceptual overview on rsine architecture.	89
5.1	Number of vocabularies affected by quality issues in the first and last uploaded version.	137
5.2	Quality changes by issue.	138
5.3	Quality changes by Web vocabulary.	139
5.4	Number of vocabularies affected by quality issues before and after QA.	144
5.5	Quality changes by issue.	146
A.1	Start screen of the <i>online SKOS Quality Checker</i> Web application.	174
A.2	Vocabulary upload and overview interface of the <i>online SKOS Quality Checker</i> .	175
A.3	Vocabulary analysis in progress using <i>online SKOS Quality Checker</i> .	176
A.4	Vocabulary analysis finished in <i>online SKOS Quality Checker</i> .	176
A.5	Overview of all detected quality issues in <i>PoolParty Thesaurus Server</i> .	177
A.6	List of overlapping labels found by <i>PoolParty Thesaurus Server</i> .	177
A.7	List of relation clashes with one focused issue shown in <i>PoolParty Thesaurus Server</i> .	178



# List of Tables

1.1	Vocabularies selected for further analysis. The Concepts column shows the number of authoritative SKOS concepts in the vocabulary, i.e., concepts whose URI is within the URI namespace of the vocabulary. . . . .	15
3.1	Linked Data schemas used in quality issue definitions . . . . .	46
4.1	Equivalence in quality issue naming between the PPTS user interface and the quality issue catalog. . . . .	84
4.2	Namespaces and their abbreviations used in the subscription document examples. . . . .	96
5.1	Vocabulary usage scenarios. . . . .	101
5.2	Importance of quality issues for usage scenarios. . . . .	103
5.3	Vocabulary statistics as determined by <i>qSKOS</i> , ordered by the number of authoritative concepts . . . . .	115
5.4	Validation results using <i>qSKOS</i> , Part 1: <i>Labeling and Documentation Issues</i> . The X-marks indicate that no common language (cf. Section 5.2.2.3) could be detected in the corresponding vocabularies. . . . .	117
5.5	Validation results using <i>qSKOS</i> , Part 2: <i>Structural Issues</i> . . . . .	123
5.6	Validation results using <i>qSKOS</i> , Part 3: <i>Linked Data Specific Issues</i> . Values marked with an asterisk (*) have been extrapolated from a randomly sampled subset of the concepts. . . . .	129
5.7	Validation results using <i>qSKOS</i> , Part 4: <i>SKOS Consistency Issues</i> . . . . .	132
5.8	Detected quality issues and their presence in the total number of uploaded vocabularies. . . . .	135
5.9	Detected quality issues and their presence in the number of vocabularies of both first and latest uploaded version. . . . .	136
5.10	Upload count of each vocabulary. . . . .	141
5.11	Thesaurus domains with German references. . . . .	142
5.12	Occurrences of quality issues before and after QA. . . . .	145





# Chapter 1

## Introduction

Controlled vocabularies are a means for organizing and connecting knowledge. They are used to capture relevant terms of a certain domain and establish relations between them in a meaningful way. Harpring [Har10] defines controlled vocabularies as “an organized arrangement of words and phrases used to index content and/or to retrieve content through browsing or searching” and the ANSI/NISO Z39.19 standard [NIS05] identifies their main purpose in improving “the effectiveness of information storage and retrieval systems, Web navigation systems, and other environments that seek to both identify and locate desired content via some sort of description using language”.

Well-known examples of controlled vocabularies are Classification Schemes like the *Binomial nomenclature* that is used in zoology and botany for naming species in a standardized way<sup>1</sup>. Another example of a classification scheme is the *Dewey Decimal Classification* (DDC) that is used by libraries to assign each book a unique number based on the book’s topic, allowing to easily locate it on the shelves. A slightly different, but also widely used kind of controlled vocabulary are Subject Headings. Most often they are used in libraries to describe the topic and content of books and articles and provide cross-references that enables the library users to locate similar material. Examples of subject headings are the *Library of Congress Subject Headings* (LCSH), which is mostly used in North America but also internationally, the *Schlagwortnormdatei* published by the German National Library, and the *Répertoire d’autorité-matière encyclopédique et alphabétique unifié* used in French speaking countries. There are many more flavors of controlled vocabularies, mainly distinguished by the level of their structural complexity, i.e., the number and kind of relations among the managed terms.

---

<sup>1</sup>Each species is given a name consisting of the genus (e.g., “homo”) and the species within this genus (e.g., “sapiens”).

Organizing knowledge and locating information is also a task virtually every contributor and user of the Web is faced with. Within the last few years, we witness the transition from a Web of documents to a Web of data: Governments and institutions worldwide publish their data as *Open Data*, i.e., they do not impose any restriction on usage or redistribution. Some data providers even publish their data sets as Linked Data [HB11] in a standardized, machine-readable form. This allows them to connect their data sets to other resources on the Web by using standard Web techniques. This way, the Web evolves into a global knowledge graph that enables completely new ways for retrieving and combining information. Automatic agents, for example, can collect and combine data tailored to a user's needs. However, in order to combine information from various sources in a meaningful way, a common understanding of the meaning of this data is needed and controlled vocabularies are a means to provide this.

As a consequence, controlled vocabularies have been widely adopted in Web applications. In their report on the library Linked Data domain, Isaac et al. [IWYZ11] list numerous “Published value vocabularies”, i.e., controlled vocabularies that provide terms “with which metadata records can be populated”. Among them are the resources mentioned above (e.g., DDC, LCSH) that exist since long before the Web has been established. Isaac et al. complement their vocabulary listing by referencing online services that make use of these vocabularies. Many of the “pre-Web” vocabularies have been converted into machine-readable formats adhering to the Linked Data design issues<sup>2</sup>. The *Simple Knowledge Organization System* (SKOS) [MB09] is a data model for expressing controlled vocabularies used for this kind of applications. It has become a de-facto standard for expressing controlled vocabularies and publishing them as Linked Data on the Web. We refer to these vocabularies as *Web vocabularies* throughout this thesis.

The benefits of Web vocabularies are manifold. With the evolution of SKOS and its acceptance as a W3C recommendation in 2009, a method has been established that enables vocabulary developers to formulate their vocabularies in an agreed-upon, extensible format. It establishes compatibility among various vocabularies about different topics authored by multiple contributors, made available by multiple publishers. This compatibility allows developers to, e.g.,

- extend their vocabularies by adopting or reusing existing vocabularies,
- implement reusable search algorithms and presentation interfaces that exploit the standardized data format and set of relations of Web vocabularies,

---

<sup>2</sup>Linked Data design issues: <http://www.w3.org/DesignIssues/LinkedData.html>. Retrieved 2015-06-23.

- use the same vocabulary for indexing documents from various origins, making it possible to find related documents across system boundaries.

To take full advantage of the potential of Web vocabularies, developers must make sure their vocabularies are capable of covering the scenarios exemplified above. During the vocabulary creation process, errors can potentially be introduced that hamper applicability of the Web vocabulary for a specific usage scenario, such as declined search recall and precision or understandability by human users. We consider these shortcomings to be quality deficiencies of Web vocabularies that should be avoided. Web vocabularies of poor quality are less likely to be reused and linked and therefore defeat the advantages of distributed organization of knowledge. However, quality issues also impact vocabulary usage on a local scale, i.e., when they are used in systems that do not make use of “external” Web resources like, e.g., in company-wide intranets.

The goal of this thesis is to provide guidance on what kinds of properties of a controlled vocabulary can indicate its quality. It provides methods and tools for automatic assessment of the quality of existing Web vocabularies as well as for supporting the Web vocabulary development process. We study the effectiveness of our techniques and how they can help to improve existing tools and processes.

## 1.1 Motivating Examples

In the following section, we give examples on the practical usage of Web vocabularies. We show how they can help improving the usability of digital collections and how shortcomings in Web vocabulary quality influence exemplary real-world usage scenarios.

### 1.1.1 The MEKETRE Project

The author of this thesis was responsible for developing the computer-science part of the MEKETRE project<sup>3</sup> [MHP11], an interdisciplinary project conducted by the University of Vienna’s Institute for Egyptology and Multimedia Information Systems research group. The goal of the project was to collect and study digital representations of two-dimensional artworks (reliefs and paintings) stemming from tombs built during the Middle Kingdom (MK) in ancient Egypt. The gathered data was made available by a

---

<sup>3</sup>The project was funded by the Austrian Science Fund (FWF) and scheduled for three years from early 2010 until late 2012. Further information is available at <http://www.meketre.org>. Retrieved 2015-06-23.

Web application (the MEKETREpository) and as Linked Open Data on the Web, enabling adoption and reuse for other scholars in the field or future (information retrieval) applications.

As the main task in the project (from an information organizing point of view) was to structure the available items of MK art and publish them on the Web, we chose an approach that allows scholars to upload images and describe them using both free text and controlled vocabularies. The novelty of the approach was provided by adoption of the Linked Open Data guidelines, i.e., we used SKOS and standard ontologies to publish all data collected and developed during the project. Data in the Egyptology domain so far is used to be available in proprietary formats and under very restrictive licensing conditions.

However, especially in Egyptology controlled vocabularies are highly relevant because often multiple terms exist within one language that denote the same object. For example, some kind of funerary figurine is referred to as “ushabti”, “shabti”, “shawabti” and even more variant spellings. This constitutes a problem because users that search for images showing this figurine want the system to return all relevant art items regardless of the used spelling. This feature was vital for the MEKETRE project because the data should become available for the interested public. It improves usability in terms of browsing and search result quality for users that are unaware of the different term variants.

Due to lack of available reusable controlled vocabularies dealing with MK art items and unclear licensing issues, Egyptologists created a custom classification scheme and term lists for categorizing and describing art items relevant for the MEKETRE project. They cover, e.g., necropolis names, dating information, or general terms and are used for both (i) describing art items and (ii) assisting information retrieval functions like faceted search or synonym resolution. Furthermore, the open publication format enabled us to integrate MEKETRE with the PELAGIOS project<sup>4</sup> which aims to collect data about historical sites, and make them accessible using Linked Open Data techniques in order to answer research questions using a combined dataset.

By examining the controlled vocabularies developed collaboratively by Egyptologists in the course of the MEKETRE project, we found some issues that degrade the user experience of the MEKETREpository in various ways like:

- The majority of concepts is lacking labels in all four languages that should be supported. French and Arabic labels are most often omitted, so users should best enter search terms in English or German.

---

<sup>4</sup>MEKETRE contributions to the PELAGIOS project: <http://pelagios-project.blogspot.co.at/2012/07/meketre-new-project-partner-introduction.html>. Retrieved 2015-06-23.

- Concept URIs were not resolvable anymore, which was caused by data conversion, relocation, or server misconfigurations (e.g., redirections were not configured properly).

From our experiences with the MEKETRE project we can therefore summarize that finding quality issues in controlled vocabularies that operate search and retrieval systems helps in finding inconsistencies, errors, and other deficiencies in the overlying application(s). Furthermore, we noticed that the identified quality issues are not always consequences of limited resources in terms of financial and personnel means, but also indicate that vocabulary quality assessment must accompany the development and maintenance process of a software solution.

### 1.1.2 Existing Controlled Vocabularies

We were able to spot another practical implication on the importance of keeping the application logic in line with the underlying vocabulary. When we analyzed the *Thesaurus for Economics*<sup>5</sup> (STW) in version 8.06, we found that seven concepts had identical terms assigned. For example, the term “Forage crops” was assigned as descriptor (preferred label) to two different concepts. Another example is the term “Tax evasion” that was assigned to one concept as a preferred label and to another concept as an alternative label. These overlapping labels directly influence the search functionality on the homepage of the thesaurus which is implemented as a textbox that automatically suggests terms as the user types. Figure 1.1 shows two identical suggestions that are caused by the label overlaps, without any way for the user to disambiguate the terms.

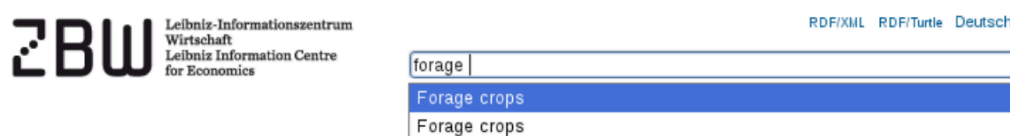


FIGURE 1.1: Auto-completion at the STW Web page (Version 8.06).

We contacted the developers of the STW thesaurus about this issues and also presented it in our talk<sup>6</sup> at the NKOS 2011 workshop [MH11]. In later revisions of the STW vocabulary we could no longer reproduce the observed behavior. In the current version (8.12) of the vocabulary (depicted in Figure 1.2) the affected concepts were renamed and can now easily be distinguished.

<sup>5</sup>Thesaurus for Economics: <http://zbw.eu/stw/versions/latest/about>. Retrieved 2015-06-23.

<sup>6</sup>Presentation slides: [https://at-web1.comp.glam.ac.uk/pages/research/hypermedia/nkos/nkos2011/presentations/mader\\_nkos.pdf](https://at-web1.comp.glam.ac.uk/pages/research/hypermedia/nkos/nkos2011/presentations/mader_nkos.pdf). Retrieved 2015-06-23.

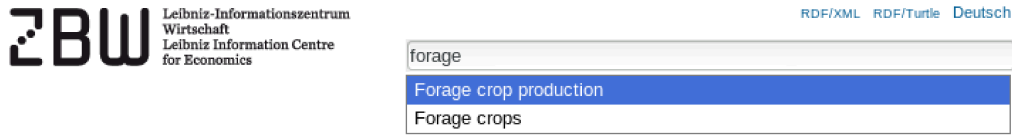


FIGURE 1.2: Improved auto-completion functionality for STW (Version 8.12).

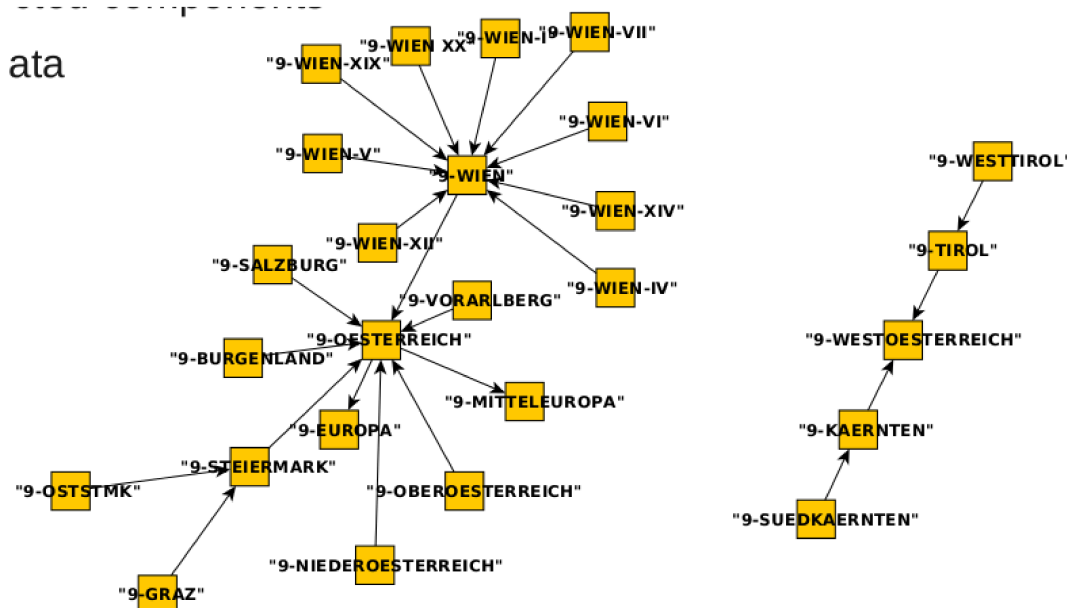


FIGURE 1.3: Unconnected components in a vocabulary. The arrows indicate the hierarchically broader concept.

During our research we were faced with the task of converting an existing “legacy” vocabulary that was available in a proprietary, CSV<sup>7</sup>-based format into a Web vocabulary. In the original dataset we found unconnected components as shown in Figure 1.3. The vocabulary contains, among other concepts, regions of Austria and it seems illogical why the regions in the right part of the figure should not be connected to the structure on the right hand side. After talking to the vocabulary creators, they confirmed that the structure(s) we identified were obsolete “testing data”.

## 1.2 Problem Description

Controlled vocabularies are, in the majority of all cases, created by domain experts for a specific usage scenario. Although methods exist that allow extraction of indexing thesauri from text corpora, creation of controlled vocabularies still largely remains an intellectual process and, as such, is prone to errors that “slip in”. To avoid these errors, vocabulary

<sup>7</sup>Comma-separated values.

creators follow guidelines and procedures published in standardization documents or apply their own custom checks. However, most of these checks rely on human judgment, i.e., they cannot be automatically employed and thus do not scale for large vocabularies or short vocabulary publication cycles. Especially in collaborative environments, where multiple domain experts are allowed to work on a controlled vocabulary simultaneously, or in very complex controlled vocabularies, keeping an overview on all terms and relations becomes increasingly difficult.

Before and during the vocabulary development process, existing guidelines suggest to review other resources to reuse and adopt relevant terms. Due to the increasing awareness of the advantages of publishing datasets according to the Linked Data principles, the probability increases that a concept which should become a member of the newly created thesaurus is already published on the Web. To find these concepts, portals like datahub<sup>8</sup> or Web Data Commons<sup>9</sup> are available. They allow downloading existing ontologies or controlled vocabularies, which are either provided directly by the creators (datahub) or are retrieved by crawling the Web (Web Data Commons). Once similar controlled vocabularies have been found, they should be *mapped* to the newly developed vocabulary by using, e.g., the SKOS mapping relations that can express exact, close, related, broader, or narrower matches. The intention of, e.g., `skos:exactMatch` is to state that two concepts have equivalent meaning [IS09], i.e., they share enough properties to substitute for each other [HH10] in certain contexts. Vocabularies can also be *aligned* with one another, i.e., finding corresponding concepts and connecting them by using the property `owl:sameAs`. While `owl:sameAs` is intended to state that two URI references actually refer to the same thing [BvHH<sup>+</sup>04], Halpin et al. [HH10] point out that this property is often used to mean other things than specified, violating the principles of transitivity and substitutivity, which are inherent in the notion identity.

Mapping and alignment are important because they, for example,

- can increase the efficiency of the development process. Properties that are expensive to add, such as translations of labels to multiple languages, do not need to be crafted from scratch.
- improve both manual and automated exploration and inference of knowledge by navigating through the Web of Data.

However, in case potential Web vocabularies for linking (henceforth also called “remote vocabularies”) can be found, the decision if the vocabulary actually should be linked is

---

<sup>8</sup>Datahub: <http://datahub.io/>. Retrieved 2015-06-23.

<sup>9</sup>Web Data Commons: <http://webdatacommons.org/>. Retrieved 2015-06-23.

To summarize, a scalable, automated solution for determining the quality of a vocabulary is needed during multiple phases of a controlled vocabulary’s lifecycle:

- On the Web, the AAA Slogan [AH11] holds: “Anyone can say Anything about Any topic”. This means that anyone can publish Web vocabularies that assert facts (e.g., hierarchy definitions) about concepts defined in another vocabulary. These assertions are beyond the scope of influence of the developers that originally coined these concepts. However, agents that collect information on the Web are free to choose their sources, exploiting the democratic nature of the Web. On the other hand, combination of resources from different origins (and potentially different views of the world) might cause inconsistencies that do not match with the information collector’s expectations and intended use of the data. Also for these cases an automated tool is needed to efficiently decide on the applicability of the information.



As each single stage in the controlled vocabulary development lifecycle is prone to the introduction of quality problems, effective integration of quality assessment methods for Web vocabularies becomes necessary. During the development process, contributors to a controlled vocabulary require immediate feedback on the changes they perform. Moreover, a developer also needs information about changes that are introduced by other contributors affecting resources he or she authored. With growing size, complexity, and change frequency of the controlled vocabulary, it becomes increasingly difficult for developers to manually track changes to these resources. For example, contributors who develop controlled vocabularies, typically want to know whenever, e.g., the meaning of a concept is fundamentally changed. This is because the concept might have been used for indexing documents and the changed meaning impairs search precision. Reliable detection whether (and to what extent) a resource is used and modified by others provides valuable input for vocabulary developers to further refine and maintain their vocabularies. Thus, it would be possible to, e.g., delete unused concepts without introducing broken links or provide more accurate definitions of concepts that have been referenced inadequately by others.

On the other hand, a notification-based approach as outlined in the previous paragraph can also fall short in a number of situations. For example, if quality assessment algorithms are very costly in terms of processing power and runtime requirements, it is often not feasible to evaluate them on each user interaction. Therefore, quality assessment strategies that check a controlled vocabulary “as a whole” can be used at certain periods of time. A similar approach, *Continuous Integration* [FF06], has become a standard practice in software development: all recent changes to the source code of a software system are combined to a common codebase and automatic tests (up to hundreds or thousands in large systems) are run. They either indicate success or failure and with the resulting test reports, developers can spot bugs that need to be fixed in the final version of the system. We believe that such an approach can also work for developing controlled vocabularies, provided that automated tests based on an agreed-upon understanding of vocabulary quality are in place and set up in a testing environment.

It is currently unclear, how or to what extent existing work such as guidelines for creation, maintenance, and evaluation of controlled vocabularies as well as data quality notions and Linked Data evaluation approaches can be used to assess the quality of Web vocabularies. However, such a quality assessment technique is needed to tackle the problems identified above. Therefore we see our work as a contribution towards (i) a comprehensive catalog of automated quality assessment algorithms applicable to Web vocabularies and (ii) getting an understanding on acceptance of these algorithms by human users as well as (iii) integration in Web vocabulary development processes.

### 1.3 Contributions

In this thesis we present an approach for performing automated quality checks on Web vocabularies which assist vocabulary developers in finding inconsistencies, missing data or other errors. In contrast to existing approaches we focus on finding errors in the formalization and modeling of data against a specific syntax or language; our approach specifically targets the problems and requirements of development and utilization of Web vocabularies. The central part of our approach is the definition of a catalog of functions for assessing the quality of Web vocabularies. We apply these functions to existing well-known Web vocabularies, contributing an overview about the state of these resources from a data quality perspective. We furthermore report on the perception of our defined quality functions by experts in the field of controlled vocabulary development, introduce implementations and present how Web vocabulary quality is affected by deploying our approach to productive environments.

The target audience of our approach are developers of Web vocabularies who, despite their experience and care they put into their development efforts, can profit from assistive quality checks. The approach is also especially useful to support developers in converting existing controlled vocabularies into Web vocabularies by e.g., helping during the review and publication process.

In the following we list the research questions we target in this thesis and which are directly related to our main contributions. For each research question we provide an outline on how we addressed it. We furthermore reference contributions to these research questions that have already been published in our earlier work.

#### Research Question 1

*What properties of a Web vocabulary have an impact on its quality as perceived by human users?*

One of the main contributions of this thesis is a catalog that identifies potential quality issues of Web vocabularies. It is based on literature review, analysis of existing Web vocabularies and informal face-to-face discussions with experts in the field of controlled (Web) vocabulary development. From this catalog it is possible to infer formalized, automatically computable quality functions. We published a preliminary version of the catalog alongside with exemplary results from evaluating it against existing Web vocabularies in [MH11]. In our consecutive work [MH12] we provided informal definitions of the quality functions inferred from the catalog. We again applied these function to a corpus of existing vocabularies on the Web and added a detailed coverage of our findings. Later we further extended this work by identifying more quality issues and investigating

possibilities for automated repair, the latter part of which was contributed by the author of the *Skosify* tool (cf. Suominen’s earlier work [SH12, SM13]). However, automated repair strategies for the identified quality issues are not within the scope of this thesis. In order to refine and verify our catalog, we performed a survey on the perception and relevance of the identified quality issues [MH13]. Due to the wide area of applications that make use of Web vocabularies and their different requirements, this catalog can not be considered complete and is expected to be extended and updated.

### Research Question 2

*How can the quality of a Web vocabulary be automatically assessed?*

To address Research Question 2 we contribute two tools that implement the quality functions inferred from the catalog: *qSKOS* and *rsine*. The tools follow a different approach for invoking the quality functions on linked data sets. While *qSKOS* is designed to evaluate the quality functions against snapshots of vocabularies provided as files, *rsine* is capable of evaluating them immediately when the Web vocabulary is updated. The source code of both tools is available online<sup>10</sup> under open-source licenses. As an additional contribution to the Linked Data community, we provide a version of *qSKOS* online<sup>11</sup> that checks uploaded Web vocabularies and provides the quality report for download or per email.

We used *qSKOS* for evaluating existing Web vocabularies in [MHI12, SM13] and since then continually improved it by, e.g., adding new quality functions and improving report output and user interface. *rsine* was developed as part of the LOD2 project<sup>12</sup> and described in project deliverables<sup>13</sup> as well as in the LOD2 book [MMS14].

### Research Question 3

*How can automated quality assessment of a Web vocabulary be integrated into collaborative controlled vocabulary development processes and what is its impact?*

In [Mad12] we outlined an approach for integrating Web vocabulary quality assessment into collaborative development processes as a contribution to this research question. Following the proposed approach, we integrated (an adapted version of) *qSKOS* into the *PoolParty Thesaurus Server*<sup>14</sup> (PPT) which since then is shipped as a premium feature

<sup>10</sup>*qSKOS*: <https://github.com/cmader/qSKOS>, *rsine*: <https://github.com/rsine/rsine>. Both retrieved 2015-06-23.

<sup>11</sup>PoolParty SKOS Quality Checker: <http://qskos.poolparty.biz>. Retrieved 2015-06-23.

<sup>12</sup>LOD2 project: <http://lod2.eu/>.

<sup>13</sup>D5.3.1: <http://svn.aksw.org/lod2/WP5/D5.3.1/Deliverable-5.3.1-Final.pdf>, D5.3.2: <http://svn.aksw.org/lod2/WP5/D5.3.2/d5.3.2-revised.docx>. Both retrieved 2015-06-23.

<sup>14</sup>PoolParty Thesaurus Server: <http://www.poolparty.biz/portfolio-item/poolparty-thesaurus-server/>. Retrieved 2015-06-23.

of this commercial application. We first presented it at the ISKO UK biennial conference 2013<sup>15</sup>. We also integrated *rsine* into PPT and studied the applicability of the approach in a production-like setting as part of the LOD2 project<sup>16</sup> in cooperation with Wolters Kluwer Germany<sup>17</sup>. To study the effects of quality assessment during the vocabulary development process, we performed a case study [MW14] among students educated in development of controlled vocabularies.

## 1.4 Methodology

In order to address the Research Questions 1 (*What properties of a Web vocabulary have an impact on its quality as perceived by human users?*) and 2 (*How can the quality of a Web vocabulary be automatically assessed?*), it is essential to define what the notion of quality means for Web vocabularies, i.e., what distinguishes a “good” vocabulary from a “bad” one. To accomplish this, we utilized three main sources for our research: (i) existing literature, (ii) publicly available Web vocabularies, and (iii) consulted experts in the field. We describe them in detail in the following sections.

Figure 1.4 illustrates our approach for developing a catalog of potential Web vocabulary quality issues and tools that are based on this catalog. They are indicated with a green box and constitute parts of the main scientific contributions of this thesis (as described in Section 1.3). Based on the three sources described above, we first created a preliminary catalog of potential quality issues, formalized computable quality functions based on this catalog and implemented tools capable of evaluating Web vocabularies using these functions. Based on the evaluation results and expert feedback we incorporated refinements to our contributions in an iterative process, i.e., we improved and extended the catalog and updated the quality functions and tools.

### 1.4.1 Existing Literature

We reviewed existing standards, guidelines and tutorials on design, construction, and evaluation of controlled vocabularies. In particular we focused on the identification of linguistic and structural patterns that can be considered bad practice and are suitable for automated assessment. We, of course, also considered approaches from related research topics like, e.g., ontology evaluation and data quality and data validation techniques and evaluated in what way they can be applied on Web vocabularies.

<sup>15</sup>ISKO UK 2013 slides: <http://www.iskouk.org/sites/default/files/MaderSlides.pdf>. Retrieved 2015-06-23.

<sup>16</sup>D7.3: <http://svn.aksw.org/lod2/WP7/D7.3/D7.3.pdf>. Retrieved 2015-06-23.

<sup>17</sup>Wolters Kluwer Germany: <http://www.wolterskluwer.de/>. Retrieved 2015-06-23.

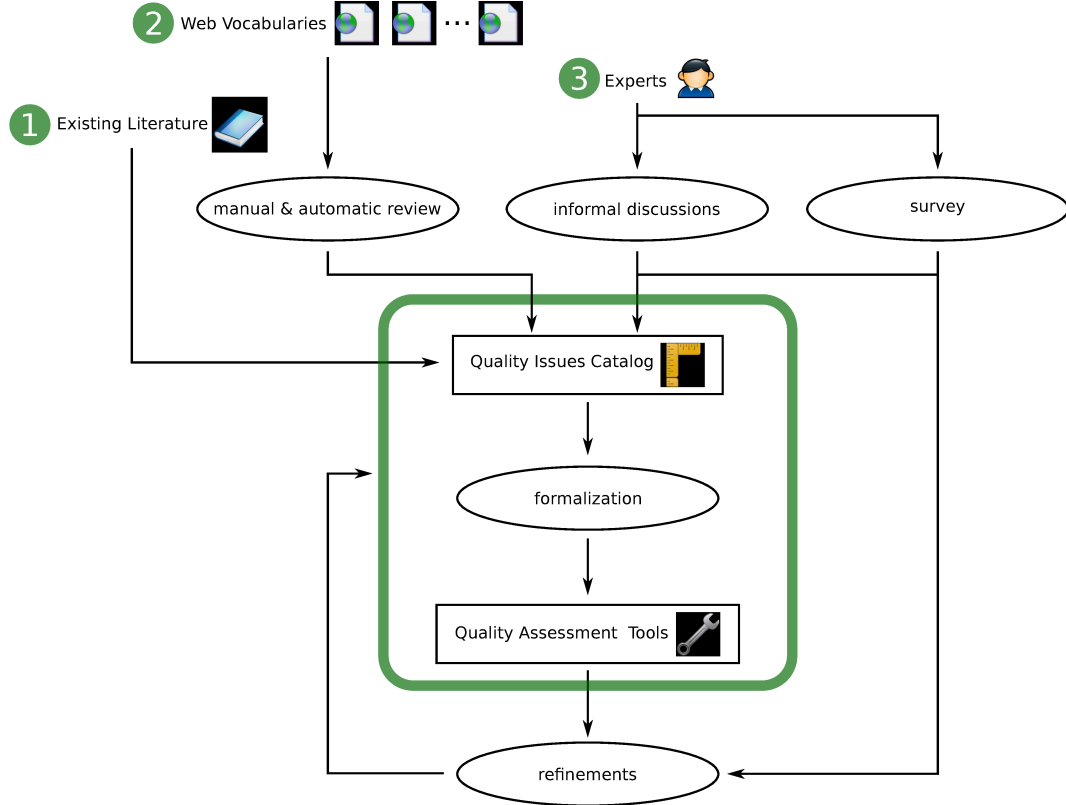


FIGURE 1.4: Quality issue catalog and tool development approach.

### 1.4.2 Public Web Vocabularies

We analyzed currently published vocabularies on the Web in two different ways. First, we manually reviewed them against a preliminary catalog of quality issues developed from the literature review. We performed an ad-hoc review of “well-known” vocabularies such as the STW or the *New York Times People directory*<sup>18</sup> (NYTP) in [MH11]. This helped us in refining and extending the catalog. After the implementation of an initial version of the tools, we performed automated reviews of a higher number of larger vocabularies. The generated quality reports again served us for refining the catalog but also for requesting expert feedback. In our subsequent work, we broadened the selection of vocabularies [MHI12] and improved the methodology of selecting the studied vocabularies [SM13].

#### 1.4.2.1 Vocabulary Selection

Because the number of vocabularies published on the Web is growing continually, we cannot analyze all of them. Therefore, we collected a representative sample of Web

<sup>18</sup>NYTP: <http://data.nytimes.com/people.rdf>. Retrieved 2015-06-23.

vocabularies from multiple domains and a broad range in terms of size and complexity. To collect this data set, we used the following procedure:

First, in order to ensure a wide coverage of domains, we looked for vocabularies in each of the seven categories of the Linked Open Data cloud domain classification<sup>19</sup>. For each domain, we then selected one small (up to 3 000 concepts), one medium-size (3 001 to 10 000 concepts) and one large (more than 10 000 concepts) Web vocabulary. This two-dimensional matrix gave us 21 slots to fill with a vocabulary. For each slot, we used three data sources to select a prominent, recently updated (not older than 2009) Web vocabularies that were available for download or SPARQL access from (i) the Datasets page<sup>20</sup> of the SKOS wiki, which mentions approximately 40 sources, some of which contain several SKOS vocabularies; (ii) the Web vocabularies listed in the *datahub* data catalog, containing approximately 150 datasets tagged `format-skos` or `skos`; and (iii) the survey of Web vocabularies by Abdul Manaf et al. [AMBS12b], containing 478 vocabularies. We also included vocabularies that are not available for public access, e.g., the *LVAk thesaurus* used by the Austrian army and the *Peroxisome Knowledge Base* (PXV) that was provided to us as a data dump. As the slot for a medium size vocabulary in the Geographical domain was still unfilled, we chose to use the *New York Times Locations* vocabulary instead, which has 1 920 concepts and is thus relatively large, although not large enough for the medium-size category. Finally, we chose to include all the very large vocabularies, having more than 100 000 concepts, regardless of their domain: *DBpedia Categories*, the *DDC*, *Gemeenschappelijke Thesaurus Audiovisuele Archieven* (GTAA), *LCSH*, *RAMEAU* and *SNOMED clinical terms*. The final set of 24 vocabularies is shown in Table 1.1. Detailed statistics about each vocabulary are summarized in Table 5.3 and discussed in Section 5.2.

#### 1.4.2.2 Analysis of Vocabularies

To gain an understanding of the current quality of Web vocabularies published online, we analyzed the 24 vocabularies in Table 1.1 using the *qSKOS* quality analysis tool. For performance reasons, we performed checks for *Missing Incoming Links* and *Broken Links* on the largest vocabularies only on randomly sampled subsets of the concepts. The reported values were extrapolated from the measurements on the subset.

<sup>19</sup>Linked Data by Domain: <http://wifo5-03.informatik.uni-mannheim.de/lodcloud/state/#domains>. Retrieved 2015-06-23.

<sup>20</sup>SKOS datasets wiki: <http://www.w3.org/2001/sw/wiki/SKOS/Datasets>. Retrieved 2015-06-23.

Abbrev	Vocabulary Name	Concepts	Version	Domain	Size
ODT	Open Data Thesaurus	107	2012-09-11	Cross-domain	small
GeoNames	GeoNames Ontology	680	3.01	Geographic	small
Reegle	Clean Energy and Climate Change Thesaurus	1 447	2012-09-28	Government	small
PXV	Peroxisome Knowledge Base	1 686	1.6	Life sciences	small
NYTL	New York Times Locations	1 920	2012-09-11	Geographic	(medium)
SSW	Social Semantic Web Thesaurus	1 943	2012-09-11	User-generated content	small
IPTC	IPTC NewsCodes / Media Topic	2 061	2012-09-12	Media	small
UNESCO	UNESCO nomenclature for fields of science and technology	2 509	2012-12-20	Publications	small
Plant	Plant Building Vocabulary	3 246	2012-09-11	User-generated content	medium
IPSV	Integrated Public Sector Vocabulary	4 732	2.00	Government	medium
NYTP	New York Times People	4 979	2012-09-10	Media	medium
GEMET	The GEneral Multilingual Environmental Thesaurus	5 209	2012-09-11	Life sciences	medium
STW	STW Thesaurus for Economics	6 789	8.10	Publications	medium
Eurovoc	The EU's multilingual thesaurus	6 797	4.3	Cross-domain	medium
LVAk	Austrian Armed Forces Thesaurus	13 411	0.9	Government	large
EARTH	The Environmental Applications Reference Thesaurus	14 351	2012-08-30	Geographic	large
UMBEL	UMBEL Vocabulary and Reference Concept Ontology	26 389	1.05	Cross-domain	large
AGROVOC	United Nations Agricultural Thesaurus	32 291	2012-07-26	Publications	large
SNOMED	SNOMED clinical terms (French)	102 614	3.5-VF-20091001	Life sciences	large
GTAA	Gemeenschappelijke Thesaurus Audiovisuele Archieven	171 991	2010-08-25	Media	large
RAMEAU	French National Library subject headings	207 272	2009-04-23	Publications	large
DDC	Dewey Decimal Classification	251 977	2012-09-28	Publications	large
LCSH	Library of Congress Subject Headings	408 923	2012-03-01	Publications	large
DBpedia	DBpedia Categories	865 902	3.8	User-generated content	large

TABLE 1.1: Vocabularies selected for further analysis. The Concepts column shows the number of authoritative SKOS concepts in the vocabulary, i.e., concepts whose URI is within the URI namespace of the vocabulary.

### 1.4.3 Expert Consultation

During the whole process of creating the catalog of quality issues we continually requested feedback from experts in the Linked Data and Knowledge Organization domain by discussions on public mailing lists, workshop publications and a structured survey. Based on this feedback we were able to refine our findings.

#### 1.4.3.1 Informal Discussions

The means for getting into contact with experts were diverse. We published our findings from the manual literature review in the qSKOS wiki<sup>21</sup> and requested feedback from experts via public mailing lists<sup>22</sup> related to Web vocabulary development. Based on this feedback we published a preliminary catalog of quality issues at the NKOS 2011 workshop [MH11]. In [Mad12] we elaborate on the catalog in more detail and also provide a conceptual model on how to integrate controlled vocabulary quality checks into a continuous quality assessment process.

<sup>21</sup>qSKOS quality issues wiki: <https://github.com/cmader/qSKOS/wiki/Quality-Issues>. Retrieved 2015-06-23.

<sup>22</sup>e.g., [DC-VOCABULARY@JISCMAIL.AC.UK](mailto:DC-VOCABULARY@JISCMAIL.AC.UK), [NKOS-L@OCLC.ORG](mailto:NKOS-L@OCLC.ORG), [public-esw-thes@w3.org](mailto:public-esw-thes@w3.org), [public-lod@w3.org](mailto:public-lod@w3.org).

### 1.4.3.2 Survey on Vocabulary Quality

To learn about the perception and relevance of quality issues from a taxonomist's point of view, we conducted an online survey between September 20th and December 6<sup>th</sup> 2012 (see [MH13]). Our goal was (i) to get quantitative feedback on the usefulness of the identified quality issues and (ii) to improve the existing quality issues by collecting and analyzing qualitative feedback from open-ended questions.

Our survey targeted practitioners working with Web vocabularies: *vocabulary managers* who curate vocabularies, *contributors* who propose terms to be changed or included, and *users* who have no rights or intentions to change a vocabulary. We announced the survey on the same mailing lists we also used for publishing the quality issues catalog. We also contacted the Semantic Web Company's customer network and posted an invitation on its blog<sup>23</sup>. In the middle of the scheduled survey period on October 29<sup>th</sup> 2012, we sent reminders via the same communication channels.

## 1.5 Organization

This thesis is structured as follows:

Chapter 1 provides general information about the problem domain by briefly introducing controlled vocabularies, Web vocabularies, and vocabulary quality. We motivate the need for automated quality assessment by examples from practical usage scenarios of Web vocabularies. Based on these examples, we describe the problems developers of Web vocabularies are currently facing from a quality assurance perspective and describe our contributions and methodology to tackle these problems.

Chapter 2 gives a more in-depth introduction to controlled vocabularies, their design, and usage on the Web. We provide an overview on existing publications on quality evaluation and assessment and outline their relevance on the problem field covered by this thesis. We review existing approaches and guidelines towards enhancing data quality in both pre-Web information systems as well as Linked Data applications. We finally introduce existing tools for (automated) quality control of Web vocabularies and discuss their commonalities and differences to the approach proposed in this thesis.

In Chapter 3, we provide a formal definition of Web vocabulary quality issues. We express the often subjective notion of Web vocabulary quality with the help of RDF(S) semantics [LS99], establishing a basis for the implementation of automated quality assessment tools.

---

<sup>23</sup>Survey announcement blog entry: <http://tinyurl.com/d8wyntj>. Retrieved 2015-06-23.



In Chapter 4, we describe two different techniques for implementing the quality issues introduced in Chapter 3. For each of them we elaborate on design considerations, the intended use cases, and implementation details.

In Chapter 5, we report on our findings of applying the tools developed in Chapter 4 to currently published controlled vocabularies and provide a detailed coverage on the quality issues we could observe. We also report on the results of a survey intended to find out how the relevance of the identified quality issues is perceived by experts and users of controlled vocabularies and how important they are in relation to various usage scenarios. Furthermore, we provide insights on the effects of integrating quality assessment in the controlled vocabulary process. We provide the data and feedback collected from users of productive installations of our tools and give recommendations on how existing tools and Web vocabulary development processes can be improved.

Chapter 6 provides conclusions and lessons learned from developing our approach and the supporting tools. We reflect on our findings from Chapter 5, discuss limitations of our approach, discuss usability issues, and provide directions for future work.



## Chapter 2

# Background and Related Work

In this chapter, we provide a short introduction to controlled vocabularies by describing different types and areas of usage as well as giving an historical outline and discussing the role of controlled vocabularies in the context of current Web applications and the Linked Data paradigm. As in this thesis we are focusing on measuring Web vocabulary quality and integrating quality assessment methods into development and publication processes, we provide an overview on publications in similar areas of research and how they relate to the contributions of our work.

### 2.1 Controlled Vocabularies

The need for organizing knowledge dates back to antique times (e.g., the development of mnemonic principles by the Pythagoreans) and since then a vast number of approaches have been developed to make knowledge “accessible”, i.e., to find a structure that ensures efficient location of needed information. Controlled vocabularies play a central role in many of these structures and therefore have a long history. They occur in various formats under multiple names and can be used for a multitude of usage scenarios. In the following sections we give a short outline on the historical development and usage of controlled vocabularies. We cover the different kinds of controlled vocabularies mentioned in the literature and discuss their significance for knowledge organization on the Web.

#### 2.1.1 Historical Outline

The development of the *Ars Memoria* [Yat66] can be seen as one of the very early manifestations of the need of individuals to structure and organize their knowledge. It leverages the visual sense and spatial orientation to keep a larger number of specific facts

or objects in memory. This is done, for example, by assigning them to rooms of a fictional building. To reconstruct them in memory, these rooms are revisited in mind again. From this individual level of knowledge organization, more general approaches were developed that aimed for capturing the whole knowledge at the time and make it accessible for all people. One of these approaches was the idea of a Memory Theatre of Giulio Camillo (1480 - 1544), an amphitheater holding the knowledge of that time, organized in cases and boxes in a systematic order. From the approaches that have actually been realized, the *Encyclopédie ou Dictionnaire raisonné des sciences, des arts et des métiers* (*Encyclopedia, or a Systematic Dictionary of the Sciences, Arts, and Crafts*) by Denis Diderot and Jean le Rond d'Alembert is among the most important. It was published in France in 1751 and uses the “Figurative system of human knowledge” for organizing the content. This system is in fact a controlled vocabulary, a tree-like hierarchical structure with three main branches, “Memory”/History, “Reason”/Philosophy, and “Imagination”/Poetry.

Hierarchical classification systems were traditionally also used for organizing libraries to find the exact location of a book on the shelf. However, as stated by Hedden [Hed10], only by the end of the 19<sup>th</sup> century richer taxonomies were developed that allowed for assigning multiple terms and supplemental descriptions for each book in the catalog. Examples are the “American Library Association Subject Headings” (1895) or the “Library of Congress Subject Headings” (1898). These controlled vocabularies grew structurally more complex over time (by, e.g., added hierarchical and associative relations) and some of them are highly relevant even today. These vocabularies have been adopted in the 20<sup>th</sup> century by publishers and organizations that index periodical literature such as newspapers, magazines, or journals. The vocabularies were adapted to the specific information needs by taxonomists and thus diverged over time. In the 1950s thesauri were started to be used for controlling the vocabularies of information retrieval systems ([Shi12]). Standards for the naming and semantics of relations among terms in a thesaurus emerged, most notably ISO 2788 (1986) and ISO 5964 (1985) which were also adopted by national standardization institutions like the British Standards Institution (BSI) or the American National Standards Institute (ANSI). The standards were continually revised and influenced each other. The perhaps most prominent current issues are ANSI/NISO Z39.19 [NIS05] which was published in 2005 and revised in 2010, and ISO 25964 that was published in 2011 and consists of two parts, focusing on (i) creation and management of thesauri and (ii) interoperability with other vocabularies.

Usage of controlled vocabularies in online systems started to become popular in the 1970s. Commercial vendors such as Dialog used thesauri to improve the quality of database search ([Shi12]), at first limited to bibliographic material and within very specific subject areas ([Hed10]). With the technological advancements in the 1980s, taxonomy management software became available, supporting the creation and utilization of

enterprise-wide taxonomies (i.e., taxonomies customized to an enterprise's content and users such as those developed by WAND<sup>1</sup>). The public availability of the Web led to a tremendous increase of interest in controlled vocabularies. Many new small publishers offered online information services and companies established intranets that required effective means for search and navigation.

Despite advancements in full-text search engines, controlled vocabularies for expressing and organizing knowledge will not cease to be essential for finding information on the Web. Publishers are increasingly aware of the importance to provide unrestricted access to their content using standardized, machine-readable formats. This helps in utilizing the overwhelming amount of information available on today's Web. It is necessary to leverage the ability of machines to locate and filter *relevant* information and controlled vocabularies can serve here as the bridge between machine reasoning and human understanding.

### 2.1.2 The Need for Vocabulary Control

Vocabulary control means to define a terminology that is used in a specific context for expressing and organizing knowledge. This terminology (i.e., the controlled vocabulary) is typically developed by one or more domain experts and comprises of standard terms that should be used in a specific domain. In case a vocabulary is edited by multiple contributors, vocabulary control also means to control the process *how* it is changed. An example of such a process is, e.g., that contributors suggest new terms that are collected and later, at editorial meetings, discussed if and how they should be incorporated. Obligatory rules for the editing process must be established because otherwise the vocabulary would not be in control anymore.

Therefore, the main reason for "traditional" controlled vocabularies is to help humans to get a common understanding of terms. The ANSI/NISO Z39.19-2005 standard [NIS05] mentions eliminating ambiguity of natural language as the main purpose of vocabulary control, as it can occur with synonyms or homonyms. It suggests to overcome this ambiguity by providing additional scope and meaning to the terms and link synonym terms accordingly. The standard provides a list of five purposes served by controlled vocabularies: translation, consistency, indication of relationships, label and browse, and retrieval. However these purposes cannot be clearly distinguished as, for example, browsing can also be seen as a retrieval task and translation is also certainly useful when searching a knowledge base with documents in different or unknown languages.

---

<sup>1</sup>WAND Taxonomies: <http://www.wandinc.com/taxonomies.aspx>. Retrieved 2015-06-23.

While pointing out consistency, Hedden [Hed10] also states the objectives of controlled vocabularies to “ensure consistency in the application of index terms, tags, or labels to avoid ambiguity and the overlooking of information if the wrong search term is used”. In addition, she provides a more concise list of three primary functions of vocabulary control:

- *Indexing support*: Controlled vocabularies ensure consistency in cataloging multiple documents by multiple indexers<sup>2</sup>. They serve both people doing indexing *and* end users who also should have access to the used vocabulary.
- *Retrieval support*: Using controlled vocabularies can improve search results because users can avoid terms that are not used for indexing (“non-preferred terms”) or the system can suggest terms with a more general or specific meaning to broaden or narrow the search result set.
- *Organization and navigation support*: The structure of the controlled vocabulary helps the user to orientate herself in the provided content like, e.g., in the form of a table of contents or vocabulary-driven navigational menus.

Numerous other purposes of controlled vocabularies have been listed in literature but are related to one or more of these items. For example, usage for data-integration like to “facilitate the combination of multiple databases” [Shi12] can be seen as contribution to retrieval support, as well as the issues highlighted by Soergel [Soe97]:

- Mapping the users’ query terms to the descriptors used in each of the databases
- Mapping the query descriptors from one database to another
- Providing a common search language from which to map to multiple databases

### 2.1.3 Types of Controlled Vocabularies

A vast number of publications are available that cover a wide variety of aspects of controlled vocabularies such as creation, evaluation, maintenance, and publication. However, these documents do not always use a uniform nomenclature on the different types of controlled vocabularies, their purpose, and structure. In the following, we therefore give a short overview on the types of controlled vocabularies identified in literature, and how they are referred to.

---

<sup>2</sup>Indexers are taxonomists, information specialists, librarians or subject area specialists who identify concepts and relations among them in a corpus of items, e.g., text documents

Attempto Controlled English (ACE) [FS96] is a specification language with restricted grammar and vocabulary that resembles natural English but can be automatically translated into first-order-logic. This makes it useful as a language for writing unambiguous specifications, queries or creating a knowledge base. In this thesis we do not build on or extend work related to ACE. Instead, the subject we focus on are controlled vocabularies as knowledge organization systems for collecting and structuring the terms of a knowledge domain, using standardized practices and constructs.

ANSI/NISO Z39.19-2005 lists controlled vocabularies by their structural complexity that is also adopted by Hedden [Hed10] and Harpring [Har10] who provide a more fine-grained distinction and extend the list. In their most basic form, controlled vocabularies can be expressed as *lists*, i.e., terms (also “flat [term] lists”, “controlled lists”, or “pick lists”) that describe objects sharing certain commonalities. They can be ordered alphabetically or logically and hold unique terms that do not overlap in meaning and are equally specific. A *synonym ring* is a special kind of list as it holds terms identical in meaning and is only used in retrieval, not in indexing. Harpring and Hedden also separately mention *Subject Heading Lists* and *Authority Files* which, in addition to lists, also feature cross-references, preferred, non-preferred, and alternative or related term forms.

A term that is very frequently used in literature to refer to controlled vocabularies is *taxonomy*. As Hedden [Hed10] states, this term can be used in a narrow or broad sense. The narrow sense is that of a hierarchical classification vocabulary, such as the Binomial nomenclature mentioned above. This is also in line with ANSI/NISO Z39.19-2005 that describes a taxonomy as “preferred terms, all of which are connected in a hierarchy or polyhierarchy”. However, according to Hedden, in the broad sense a taxonomy can be seen as a general means for “organizing concepts of knowledge” and stand for any kind of Knowledge Organization System (KOS). Harpring additionally mentions *Alphanumeric Classification Schemes* which are basically taxonomies that do not order preferred terms but identifiers consisting of letters and numbers as it is the case with the DDC system.

There is a high level on consensus in literature about the term *thesaurus*. A thesaurus is an ordered structure of terms or concepts that are interlinked with a standardized set of relations. These relations state (i) equivalence (synonym), (ii) hierarchy (broader/narrower), or (iii) association (related), all of which are described in recent standards [NIS05, Iso11a]. Thesauri can be multilingual, i.e., contain terms in multiple languages and, optionally, contain additional information for a term, such as historical or scope notes.

In recent years, the term *ontology* is frequently mentioned in the context of utilizing controlled vocabularies to support Web-based search and retrieval systems. According

to ISO 25964-1, “ontologies usually provide more specific and closely defined relationships” than thesauri, like specialized broader or narrower relations. However, ontologies may even omit the “traditional” relation types used in thesauri and compile a custom conceptual model for representing knowledge such as custom relationships, constraints and datatypes. Ontologies do not only serve as a means for expressing knowledge but also for inferencing new knowledge based on asserted facts. The Web Ontology Language (OWL) [71, BvHH<sup>+</sup>04] has emerged as a standard way to represent ontologies in a machine-readable format on the Web and many data models, including SKOS, build upon OWL.

Summarizing, we can see that among the various kinds of controlled vocabularies, there is often no common agreement on their names and structural richness. In this thesis, we regard any type of controlled vocabulary as being a KOS. To avoid confusion, we refrain from using the term “taxonomy” in favor of “classification scheme”. Moreover, we focus on analyzing content and structural properties typically found in thesauri and therefore use the terms “thesaurus” and “controlled vocabulary” interchangeably. However, if not otherwise noted, in this thesis the term “controlled vocabulary” is to be understood in its most general meaning, i.e., it denotes any of the kinds of controlled vocabularies described above not including ontologies. We consider the latter specialized datasets on the Web that model knowledge by using properties that go beyond (or extend) those defined by the SKOS data model. Therefore, ontologies fall outside the scope of this thesis. The type of controlled vocabulary this thesis focuses on is *Web vocabularies* which we elaborate on in the following section.

#### 2.1.4 Web vocabularies - Vocabulary Control in the Context of Linked Data

Just as “traditional” controlled vocabularies help human users in getting a common understanding of the meaning and usage of terms, controlled vocabularies published on the Web can transfer this understanding, to some extent, to machines that are processing these vocabularies (sometimes also referred to as “machine users”). Machines, of course, do not really understand the meaning of, e.g., a term or the relation of two concepts. However, by knowing the location of the data, its format, and after application of a specified set of rules, they are able to retrieve and infer additional knowledge that is relevant in a certain information retrieval context such as, combining information of different sources.



To work towards this goal, controlled vocabularies must meet some prerequisites that are described by the *Linked Data* [HB11] paradigm. In his often-cited article<sup>3</sup> Tim Berners-Lee outlined four key issues that datasets must fulfill in order to integrate within a “Web of Data”. They were later complemented by a “5 star deployment scheme for Open Data”<sup>4</sup> which introduces five criteria that must be fulfilled by a dataset to count as a Linked Open Dataset:

1. Available on the Web under an open license.
2. Available as machine-readable structured data (e.g., Excel instead of image scan of a table).
3. As (2) plus using a non-proprietary format (e.g., comma-separated values instead of Excel).
4. All the above plus using open standards provided by the Word Wide Web Consortium<sup>5</sup> (W3C) like RDF [LS99] and SPARQL [PS08].
5. All of the above plus linking the data to other people’s data to provide context.

Based on these criteria we define a Web vocabulary as follows:

**Definition 2.1** (Web vocabulary). A Web vocabulary is a controlled vocabulary that is available online, adhering to the 5 Star Open Data principles. All elements, i.e., terms, concepts, and relations of a Web vocabulary can be expressed as (sub)classes and (sub)properties of resources defined in the SKOS data model [MB09].

The Simple Knowledge Organization System (SKOS) [MB09] is a “a standard way to represent knowledge organization systems using the Resource Description Framework (RDF)”. A detailed coverage can be found in the SKOS reference documentation [MB09] and therefore we only provide a brief introduction here. The goal of SKOS is to express controlled vocabularies as those mentioned in Section 2.1.3 in a common way as machine-readable data. The basic unit of a vocabulary expressed in SKOS (sometimes referred to as “SKOS vocabulary” and, in this thesis, as “Web vocabulary”) is a *concept*. It denotes the object of discourse, a thing, an abstract construct, or even immaterial things like a feeling. Concepts can be optionally organized into *concept schemes*, e.g., for thematic aggregation. SKOS provides ways to assign labels to concepts, covering the standard relations of traditional term-based vocabularies like, e.g., descriptors, non-descriptors, and synonyms. Furthermore, the SKOS data model provides relations for establishing

---

<sup>3</sup>Linked Data: <http://www.w3.org/DesignIssues/LinkedData.html>. Retrieved 2015-06-23.

<sup>4</sup>Five Star Open Data: <http://5stardata.info/>. Retrieved 2015-06-23.

<sup>5</sup>Word Wide Web Consortium: <http://www.w3.org/>. Retrieved 2015-06-23.

hierarchical or associative connections between concepts. Concepts can also be grouped into collections, mapped to other vocabularies on the Web or equipped with additional documentation such as scope or history notes.

Listing 2.1 shows an exemplary SKOS vocabulary in the Turtle RDF serialization format<sup>6</sup>, collected from the examples given in the SKOS Primer [IS09]. RDF is used to express facts in the form of statements, which are triples consisting of subject, predicate and object. The first triple, for example, in the snippet below consists of the subject (`ex:animals`), the predicate (`rdf:type`) and the object `skos:Concept`. The example describes four concepts (animals, mammals, birds and ornithology), their descriptors (preferred labels indicated using `skos:prefLabel`), synonyms (alternative labels, indicated using `skos:altLabel`) and their relations among each other (some animals are mammals, which is reflected by a `skos:broader` connection).

---

```
@prefix skos: < http://www.w3.org/2004/02/skos/core#> .
ex:animals rdf:type skos:Concept;
  skos:prefLabel "animals"@en;
  skos:altLabel "creatures"@en;
  skos:narrower ex:mammals.
ex:mammals rdf:type skos:Concept;
  skos:prefLabel "mammals"@en;
  skos:broader ex:animals.
ex:birds rdf:type skos:Concept;
  skos:prefLabel "birds"@en;
  skos:related ex:ornithology.
ex:ornithology rdf:type skos:Concept;
  skos:prefLabel "ornithology"@en.
```

---

LISTING 2.1: An exemplary SKOS vocabulary.

From a technical perspective, SKOS uses OWL [71, BvHH<sup>+</sup>04] as a data model to express controlled vocabularies and OWL itself relies on RDF(S) [LS99] to express axioms and facts. Therefore SKOS can be used in combination with ontologies formulated in OWL to model and express a richer semantics if this is needed for some use case (at the cost of decreased interoperability). However, SKOS does not aim to be a means for formalizing a KOS with exact semantics. Instead its intended usage is to express controlled vocabularies as Linked Data and to help in converting existing controlled vocabularies to a Web-enabled format, avoiding costly re-engineering.

In most “traditional” controlled vocabularies, terms form the central entities that are “expressed in some identified natural language”<sup>7</sup> and set into relation with one another. Terms denote concepts and are often qualified (e.g., “Bank (financial)”) to make clear the

<sup>6</sup>Terse RDF Triple Language (Turtle): <http://www.w3.org/TR/turtle/>. Retrieved 2015-06-23.

<sup>7</sup>As outlined in Bernard Vatant’s talk at ISKO UK 2010, [http://www.iskouk.org/sites/default/files/Vatant\\_isko-2010-09-14-BVT.ppt](http://www.iskouk.org/sites/default/files/Vatant_isko-2010-09-14-BVT.ppt). Retrieved 2015-06-23.

abstract idea behind them. Recently, with the introduction of SKOS and ISO 25964, we experienced a “switching from a term-centric to a concept-centric view”. The concept-centric approach as, e.g., implemented by SKOS, identifies concepts as (globally) unique items, identified by an abstract id, e.g., a URI or database key. Attached to these concepts are terms like descriptors, non-descriptors, or hierarchy relations that add meaning and context to the concept. The concept-centric approach adds flexibility in maintenance and publication processes of the controlled vocabulary. For example, if a concept is identified by an identifier that never changes, it is easier to link it to other concepts or change the terms assigned to it.

### 2.1.5 Examples of Web Vocabularies

An increasing number of Web vocabularies are available online, created and maintained by experts for specific usage scenarios. Here we provide an exemplary list of some well-known vocabularies that are also part of a case study (Section 5.2) carried out in the context of this thesis:

- The *AGROVOC* thesaurus<sup>8</sup> developed and maintained by the United Nations Food and Agriculture Organization that contains over 32 000 concepts in up to 20 languages and is used, e.g., for automatic document indexing.
- The *New York Times authoritative news vocabulary*<sup>9</sup> has been maintained for more than 160 years and drives so-called topic pages, which provide access to all relevant articles the New York Times has ever written about a certain subject.
- *EuroVoc*<sup>10</sup> is a multilingual thesaurus that supports 23 languages. It is used for indexing the documentation generated by the activities of the European Union (European legislation and other legal texts) as well as indexing and translation purposes. Development is carried out by a “team of documentalists and librarians from the European Parliament, the European Commission, and the Publications Office<sup>11</sup>”. Users are, e.g., the European Parliament, national and regional parliaments in Europe and private users.
- The *STW Thesaurus for Economics*<sup>12</sup> contains “6 000 standardized subject headings and about 19 000 entry terms” on economic subjects but also other topics such

---

<sup>8</sup>AGROVOC: <http://aims.fao.org/standards/agrovoc/about>. Retrieved 2015-06-23.

<sup>9</sup>New York Times Linked Open Data: <http://data.nytimes.com/>. Retrieved 2015-06-23.

<sup>10</sup>EuroVoc: <http://eurovoc.europa.eu/>. Retrieved 2015-06-23.

<sup>11</sup>Publications Office of the European Union: [http://publications.europa.eu/index\\_en.htm](http://publications.europa.eu/index_en.htm). Retrieved 2015-06-23.

<sup>12</sup>STW Thesaurus for Economics: <http://zbw.eu/stw/versions/latest/about.en.html>. Retrieved 2015-06-23.

as technology, sociology, geographic names. It drives the search capabilities of the EconBiz<sup>13</sup> economics portal of the Leibniz Information Centre for Economics and its online catalog ECONIS<sup>14</sup> that holds “more than 5,02 million title records for business studies, economics, and practice-oriented economic literature”.

### 2.1.6 The Notion of Quality

Maintainers typically want to achieve high quality of their vocabularies because this has a direct impact on the usage scenarios the vocabulary is designed to support. Furthermore, the notion of quality of Web vocabularies is to a great extent domain-specific and depends on the perception of the person(s) curating the vocabularies, as they know their target audience and need to craft the vocabulary to best fulfill the expectations of their user base. Developers have to, e.g., decide on the set of terms that are relevant for inclusion into the vocabulary, the term’s lexical form (e.g., singular or plural), or the meaning of hierarchy (e.g., part-of or instance-of relations). Therefore quality cannot be seen as an “isolated” property but goes hand in hand with the intended functionality it should support and the perception of both developers and user base.

However, we do believe that, apart from the intellectual effort that goes into controlled vocabulary development, it is possible to identify properties of Web vocabularies that in the majority of usage scenarios and for many target audiences are seen as quality problems. It is one of the goals of this thesis to provide a catalog of (some of) these properties (Section 3.1), which we refer to as “quality issues”.

## 2.2 Controlled Vocabulary Evaluation and Quality Assurance

Quality aspects of controlled vocabularies have already been discussed in standardized guidelines [NIS05, Iso11a], manuals [ABG03, Har10, Hed10, Sve03], and tutorials [Soe02]. These most often rely on manual, precise analysis of individual statements in the data, as, e.g., done by Spero [Spe08]. The author analyzes pairs of hierarchically related terms in the Library of Congress Subject Headings (LCSH) by analyzing pairs of related terms and considers them as not valid if the relation’s semantics is other than “*is a kind of*” or “*part of*”. He also points out other errors related to term forms (inverted headings, plural vs. singular form) or redundant links. Unfortunately, the author does not explain

<sup>13</sup>EconBiz: <http://www.econbiz.de/en/search/search/search-all/>. Retrieved 2015-06-23.

<sup>14</sup>Online Catalogue of the ZBW - German National Library of Economics: <http://www.econis.eu/DB=1/LNG=EN/>. Retrieved 2015-06-23.

the detected quality problems in detail nor does he provide a methodology for reviewing LCSH in a structured, maybe automated way for finding and studying more occurrences.

Kless and Milton [KM10] go a step further and provide an overview of intrinsic abstract measurement constructs for thesaurus evaluation that are presumably useful as thesaurus quality measures. They are classified into five areas, namely “Concept-related”, “Term-related”, “Structure-related”, “Documentation”, and “Overall” but are, to a large extent, subjective in nature and no formal description is provided.

In [Soe02] Soergel proposes “Characteristics for describing and evaluating KOS” that are mainly intended to be judged by humans such as specificity of concepts, appropriate breadth and depth of coverage, completeness of terms and relationships, inclusion of “all necessary facets”, or appropriateness of terms.

In a similar way, Hedden [Hed10] suggests to ensure indexing quality with establishing an editorial policy for human users, that requires, e.g., subjects or names to be “sufficiently relevant”, specifies the level of detail in the vocabulary (e.g., minimum number of terms per indexed document), or governs the permissibility of term combinations. In order to improve taxonomies, Hedden advocates for removing or merging infrequently used terms, split overused terms or reword terms in case of misuse. In this thesis, we partly cover these issues by analyzing and reporting links to other controlled vocabularies on the Web, but leave the decision on whether a term is misused or overused to human users. However, providing automated support for making this decision is subject of our future research. Hedden furthermore proposes guidelines that target the form of terms in a controlled vocabulary, which requires an understanding of natural language. According to the author, terms in the vocabulary should, e.g., make use of the same wording as looked up by users, be consistent in style, or avoid term inversions (e.g., “commercial loans” instead of “loans, commercial”). For some structural properties like orphan concepts or associative relations within the same hierarchy, Hedden gives no clear recommendation but considers these structures “unusual” or “needless information”.

Summarizing, an extensive corpus of literature exists that provides guidelines for developing controlled vocabularies. Our work builds on this literature, but focuses on those guidelines that can be checked (semi-)automatically. We formalize and adjust these guidelines (most of which have been established before SKOS or the Web have gained popularity) for automated application on Web vocabularies in order to assist vocabulary users, developers, or publishers.

## 2.3 Quality of Linked Data

The topic of data quality is also extensively discussed in Semantic Web and Linked Data research. Berners-Lee’s article<sup>15</sup> and the five star Linked Data deployment scheme are the most fundamental guidelines that cover availability on the Web (using URIs), the data format (structured, non-proprietary, RDF), and linkage to other datasets. From these baseline requirements, more detailed and fine-grained quality aspects have been derived.

Hogan et. al [HHP<sup>+</sup>10], identify common errors and shortcomings, focusing on publication issues and RDF as the used data format or linked datasets. They are divided into four categories of symptoms, “incomplete”, “incoherent”, “hijack”, and “inconsistent”. Concrete cases of errors are, e.g., undereferencable URIs, OWL reasoning inconsistencies, bogus inverse-functional properties, or literals incompatible with datatype range. Heath and Bizer [HB11] focus on the data publication aspect and describe and summarize best practices encompassing, e.g., the syntactical form of URIs, access issues (HTTP redirects and content negotiation) and options for providing dataset metadata. Most of the errors Hogan et al. [HHP<sup>+</sup>10] describe are related to RDF parsing and RDFS/OWL reasoning that are already covered by existing tools and are not specific for Web vocabularies, therefore being not within the scope of this work. Also Heath and Bizer [HB11], in contrast to our work, do not cover guidelines specially considering the quality of Web vocabularies or KOS.

### 2.3.1 Ontology Engineering and Evaluation

Related work in the area of ontology engineering exists (see [DA09, SPL04, TAM<sup>+</sup>05]), but hardly focuses on instance-level quality criteria, as it would be interesting for assessing thesauri or controlled vocabularies. Metrics have been developed to evaluate and validate ontologies [GCCL05, TA07]. Common to these metrics is the fact that they are designed to be applied to general ontologies and instance data not restricted to be expressed in RDF(S) or using OWL constructs. As a consequence, they either do not deal with specific requirements in development of controlled vocabularies and applicability of the metrics for measuring Web vocabulary quality is still unclear.

Ontology evaluation, i.e., measuring the quality of an ontology, has also been discussed extensively [PVdCSFGP12, TDM09, Vra10]. However, the authors focus on RDF datasets and ontologies in general. Most approaches propose catalogs of patterns that can degrade ontology quality in terms of, e.g., understandability, validity, or consistency.

<sup>15</sup>Linked Data: <http://www.w3.org/DesignIssues/LinkedData.html>. Retrieved 2015-06-23.

However, most of these patterns and validation checks cannot be applied to Web vocabularies, because the SKOS schema imposes very few formal constraints.

Recently, Kontokostas et al. [KWA<sup>+</sup>14] have proposed a framework for test-driven Linked Data quality assessment. They provide quality test patterns which are based on SPARQL query templates. As they focus on establishing the framework, they do not define quality metrics themselves and adopt existing measures, among them the integrity constraints from the SKOS reference document and some of the quality issues we already identified and implemented in our earlier work<sup>16</sup>.

The authors also provide a measurement tool for determining the test-case coverage and support automatic test instantiations by exploiting RDFS/OWL axioms as integrity conditions (e.g., for the properties `rdfs:domain`, `rdfs:range`, `owl:cardinality`, `owl:disjointClass`). However, as we were able to observe in our work, expressivity of SPARQL is not sufficient for all kinds of checks or, depending on the used implementation, does not scale as size and complexity of Web vocabularies increase.

### 2.3.2 Automated Validation Approaches

In order to check the conformance of a dataset against the W3C standards RDF(S) or OWL(2), online validation services have been developed. The W3C RDF Validation Service<sup>17</sup> takes RDF data in the RDF/XML serialization format or fetches it from a provided URI. The version of the RDF specification against which the service validates this input data is not completely clear, however, it seems to focus on syntax and datatype validation as specified in the RDF Primer [IS09].

The VAPOUR Linked Data validator [BFF08] and RDF:Alerts are online validation tools<sup>18</sup> that check RDF data against guidelines described in [BFF08], the Linked Data principles<sup>19</sup>, best practice recipes<sup>20</sup>, and “cool” URIs<sup>21</sup> but do not specifically cover Knowledge Organization Systems.

SPARQL Inferencing Notation (SPIN)<sup>22</sup> is a SPARQL-based language which can be used to specify integrity constraints for RDF data. The TopBraid Composer<sup>23</sup> suite is one tool

<sup>16</sup>That is, in an earlier version of our Web vocabulary quality assessment tool *qSKOS* which is described in detail in Section 4.2.

<sup>17</sup>W3C RDF validator: <http://www.w3.org/RDF/Validator/>. Retrieved 2015-06-23.

<sup>18</sup>VAPOUR Linked Data validator: <http://validator.linkeddata.org/vapour>, RDF:Alerts: <http://swse.deri.org/RDFAAlerts/>. Both retrieved 2015-06-23.

<sup>19</sup><http://www.w3.org/DesignIssues/LinkedData.html>. Retrieved 2015-06-23.

<sup>20</sup>Best Practice Recipes for Publishing RDF Vocabularies: <http://www.w3.org/TR/swbp-vocab-pub/>. Retrieved 2015-06-23.

<sup>21</sup>Cool URIs for the Semantic Web: <http://www.w3.org/TR/cooluris/>. Retrieved 2015-06-23.

<sup>22</sup>SPIN: <http://spinrdf.org>. Retrieved 2015-06-23.

<sup>23</sup>TopBraid Composer: <http://www.topquadrant.com/tools/IDE-topbraid-composer-maestro-edition/>, Retrieved 2015-06-23.



supporting SPIN-based validation, and it includes a SPIN ruleset that implements testing of some of the SKOS integrity conditions. However, in our implementation of assessing occurrences of the quality issues from our catalog, we do not make use of SPIN because our approach of combining SPARQL queries with custom processing logic implemented in Java proved to be adequate in performance and maintainability. Furber et al. [FH10a, FH10b] show that SPARQL and SPIN can be used for data quality management and provide exemplary queries for literal values. However, they do not specifically discuss Web vocabularies and if and how they are affected by data quality issues.

One issue when assessing the quality of datasets on the Web is the so-called “Open World Assumption”, which underlies the Web of Data itself. Established quality notions from closed-world systems, such as referential integrity or schema validation, do not hold anymore, because available information may be incomplete and non-explicitly stated facts cannot be determined as true or false. On the Web, anyone can publish information about anything by asserting facts in the form of RDF triples. The strategy of making use of this data is to infer new knowledge based on the facts known (or retrieved) so far. However, most datasets are created, published and maintained by a single developing organization on a defined namespace they control. On the Web nothing prevents individuals to publish additional facts about resources of this dataset; it is, after all, the idea of building a Semantic Web. The drawback is that such facts can introduce inconsistencies on the macroscopic level, encompassing all existing triples in the universe.

For developing purposes it is therefore useful to look at datasets without taking assertions into account that are created by other contributors on the Web. In order to employ validation algorithms, a common methodology is to analyze only the information asserted locally (i.e., in isolation of other datasets on the Web, pretending a “closed world”) and to specify rules the data must comply with. One example is the Pellet ICV reasoner [SPG<sup>+</sup>07] which re-interprets OWL axioms with integrity constraint semantics: instead of inferring new knowledge from the asserted facts in a dataset, they are used to find inconsistencies and missing information.

In version 4.2, the *PoolParty Thesaurus Server* has been extended to support additional relational semantics defined in RDFS such as, e.g., `rdfs:domain`, `rdfs:range` and subclass relations. The application uses a similar approach as Pellet insofar that it restricts creation of relations between resources of types that do not match the (optionally) defined domain and range values.

Summarizing, numerous tools for evaluating Linked Datasets exist but none of them provide checks that specifically target quality assurance of Web vocabularies or go into detail about if and how identified metrics can be relevant for that purpose. Our work aims to fill this gap.



### 2.3.3 Linked Data Notifications

The work mentioned above covers the evaluation and assessment of metrics concerning the quality of Linked Data but does not make any statement on the methodology for checking against defined rules and guidelines. As already stated, the most common approach is to employ a closed-world view and check the dataset as a whole. Another method is to hook into the development process and observe changes to the dataset at the most basic level, the RDF triples. In the context of this work we propose a solution (*rsine*, see Section 4.3) that is designed to notify registered users on introduction of changes to an observed dataset. Several solutions for receiving notifications on changes performed in RDF datasets exist and will be discussed in the following paragraphs.

SparqlPuSH [PM10] is a subscription/notification framework that allows for “proactive notification of data updates in RDF stores”. Users express the resources they are interested in as SPARQL queries, which are used by the service to create RSS or Atom feeds. These feeds are published on “hubs” using the PubSubHubbub protocol<sup>24</sup> which handles the dissemination of notifications.

SDShare<sup>25</sup> is a protocol for the distribution of changes to resources that are represented in RDF. A server that exposes data provides four different Atom feeds that provide information about the state of the data and update information. The protocol is designed to support replications of linked data sources and relies on clients actively monitoring the provided feeds. Furthermore, clients only get information about the updated resource URIs and are expected to fetch the actual changes of resources themselves.

In the course of the REWERSE project, a “general framework for evolution and reactivity in the Semantic Web” has been proposed [PPW06] that is based on Event-Condition-Action (ECA) rules. The framework is designed to be independent from the languages used to define events, conditions, and actions.

ResourceSync [VdSSK<sup>+</sup>12, KSVdS<sup>+</sup>13] is an upcoming NISO standard for synchronizing large resource collections. The approach is designed to satisfy various requirements arising from the need of supporting different resource types, change types, coverage, and performance issues. The approach is able to handle synchronization tasks involving textual metadata as well as large images or video files. It supports creation, update, and deletion changes and can be used for both, baseline or incremental synchronization and an audit use case to check if a resource is up-to-date. ResourceSync is a pull-based approach where “targets” that want to fetch or update a local copy of one or more resources can

---

<sup>24</sup>PubSubHubbub Core 0.4 Working Draft: <http://pubsubhubbub.github.io/PubSubHubbub/pubsubhubbub-core-0.4.html>. Retrieved 2015-06-23.

<sup>25</sup>SDShare <http://www.sdshare.org/>. Retrieved 2015-06-23.

request the needed information from the providing service (the data “source”). This is done by data dumps (snapshots of the data at certain points in time) and change description lists (information about what changes occurred to what resources) published by the data sources.

### 2.3.4 Approaches for Evaluating Quality Assurance Methods

Literature reporting on practical application and effectiveness of quality assurance methods in the creation process is still underrepresented. Coronado et al. [dCWF<sup>+</sup>09] describe automated and manual quality assurance techniques applied on the editing and publication phase of the National Cancer Institute Thesaurus (NCIt). However, no figures on the actual number of found issues are provided. Goncalves et al. [GPS11] provide a structural analysis of consecutive versions of the NCIt but do not focus on specific quality measures.

Concerning ontology evaluation, studies have been performed that use experiments to investigate feasibility and effectiveness of measures such as complexity or correctness. While we cannot directly apply these measures to controlled vocabularies, similar evaluation methodologies can be employed. Orme et al. [OYE07] define ontology complexity and cohesion metrics like “Number of Properties” and “Average Fanout of Root Class” and perform an empirical analysis to evaluate them. Two experiments were performed to measure the correlation of the defined metrics with human perception of cohesion and complexity. In one experiment, 12 separate ontology instances were modified three times and reviewed by 18 evaluators.

Strasunskas et al. [ST08] define measures for syntactic correctness and fitness of an ontology in a search task and evaluate their findings in an experiment among 21 students working with four different ontologies in two different versions. The students were divided into two groups and required to perform search tasks with subsequent judgment of the relevance of the results.

In the case study we performed in a teaching context and which we elaborate on in Section 5.4, we follow a similar experimental approach as Orme et al. [OYE07] and Strasunskas et al. [ST08]. However, we employ a within-subjects design, i.e., we let the vocabulary creators themselves decide about the feasibility of the quality report findings and incorporate changes respectively.

## 2.4 Quality of Web Vocabularies

Automated quality analysis procedures are usually defined as part of existing quality checking tools and bound to the formalism or model a vocabulary is expressed in. SKOS, for instance, defines in total six integrity conditions [MB09], each of which is a statement that defines under which circumstances data is consistent with the SKOS data model. For example, “a resource has no more than one value of `skos:prefLabel` per language tag”. Tools that can check whether these conditions are met are already available. Two of the six conditions are defined formally in the OWL representation of SKOS, using the `owl:disjointWith` and `owl:unionOf` properties for assertions. Therefore, OWL reasoners can be used to find contradictions in the model caused by these integrity conditions.

To the best of our knowledge, one of the first tools that implemented checks for the other integrity conditions was the *PoolParty SKOS Thesaurus Consistency Checker* originally developed by Semantic Web Company<sup>26</sup>.

As outlined earlier, typical application scenarios of Web vocabularies are, e.g., classification, indexing, or auto-completion. In our earlier work [NPM11] we proposed assumptions on how structural properties of Web vocabularies (e.g., number of concepts and labels, equivalence relations, presence of polyhierarchies) affect these scenarios. In our follow-up work [MH13] we continued these studies by investigating how the quality issues defined in our catalog affect vocabulary usage scenarios from an expert’s point of view (Section 5.1).

### 2.4.1 Proprietary Approaches

Assuring the quality of their developed vocabularies is practiced by providers of some widely known datasets on the Web. They usually apply their own procedures for evaluating and improving the content and structure of their vocabularies.

For the STW Thesaurus of Economics, the developers of the SKOS version describe the use of SPARQL queries to find inconsistencies in the vocabulary [Neu09]. They introduce two subclasses of `skos:Concept` to reflect the fact that STW essentially consists of two vocabularies: the set of descriptors which are used for indexing and an hierarchical classification scheme which is not used for indexing. In order to check if these structures are hierarchically disjoint, they provide one exemplary SPARQL query. However, they do not describe the other checks they used in detail.

---

<sup>26</sup>Semantic Web Company: <http://www.semantic-web.at/>. Retrieved 2015-06-23.

Kawtrakul et al. [KIT<sup>+</sup>05] describe an approach to automatically improve the quality of a Web vocabulary. They aim to increase the semantic precision between hierarchically related terms in AGROVOC. Therefore, they utilize WordNet<sup>27</sup> to detect related terms that lack precision and suggest replacement based on rules specified by experts or acquired through machine learning. The authors outline a number of problematic relations such as incorrect synonyms or inconsistent interpretations of hierarchical and associative properties. In their approach, rules can be automatically evaluated which serve to introduce additional relational semantics (e.g., “subclassOf” or “madeFrom” relations) with the goal to convert AGROVOC into an ontology. The rules rely on additional information about the terms which is either available directly from AGROVOC (terms are classified as, e.g., “Geographic term” or “Taxonomic term: Animal”) or provided by experts that manually tag term senses and specify appropriate relationships. In the latter case, additional rules are inferred by using machine learning techniques based on WordNet information.

Coronado et al. [dCWF<sup>+</sup>09] report on the quality assurance life cycle used in development of the *NCI Thesaurus*, a biomedical ontology curated by the National Cancer Institute<sup>28</sup>. It is developed in a collaborative and multi-step process accompanied with quality assurance measures, involving manual merging of changes, review against defined content guidelines, automated edit checks, end-user feedback, and creation of a “QA report” prior to publication. While the content guidelines are mostly informal, they specify literal forms and the kind of properties required for describing a concept and advocate for, e.g., “complete and accurate” concept definitions. On the other hand, some of the proposed checks such as detection of duplicates or missing definitions can be adapted and formulated for general Web vocabularies expressed in SKOS, as we did in this work. However, many of the mentioned checks are too specific to be used for general vocabularies (e.g., concept names “must begin with [a] letter or underscore”) or specify access policies which we do not cover in our work.

### 2.4.2 Data Quality and SKOS

Allemang et al. [AH11] cover SKOS in a dedicated chapter. They provide a basic introduction and some guidelines like propagating concept scheme membership on concepts in an hierarchy or stating that concept schemes should have a small number (less than six)

---

<sup>27</sup>WordNet (<http://wordnet.princeton.edu/>, retrieved 2015-06-23) is an online database of synonym sets for distinct concepts.

<sup>28</sup>National Cancer Institute, Center for Biomedical Informatics and Information Technology: <http://cbiit.nci.nih.gov/>. Retrieved 2015-06-23.

of top concepts. However, no rationale is provided for these recommendations. The authors also provide two exemplary SPARQL queries for checking two integrity constraints mentioned in the SKOS reference.

Abdul Manaf et al. [AMBS12a] identified three types of common problems (“slips”) in SKOS vocabularies as well as possible ways to correct them. They can be found by OWL reasoning and are partly based on the axioms defined in the SKOS reference ontology. Among them are, e.g., missing type declarations of used SKOS properties, class disjointness violations (concepts that are also of type `skos:ConceptScheme` or `skos:Collection`), or invalid datatypes not recognized by the used OWL reasoner. However, although the authors focus on Web vocabularies expressed in SKOS, the nature of the proposed slips is not tightly bound to the SKOS data schema, but can be used also with other datasets using OWL constructs.

Based on their earlier work, Abdul Manaf et al. have also surveyed the landscape of SKOS vocabularies available on the Web and analyzed some structural properties, such as the number of concepts, maximum hierarchy depth, or SKOS property usage distribution among different vocabularies [AMBS12b]. However, the authors do not draw any conclusion about the implications of these properties on usability or quality of the analyzed Web vocabularies.

Suominen et al. [SH12] present an approach to automatically repair validation and quality issues in Web vocabularies. The authors describe and compare the quality measures implemented by three tools, *PoolParty SKOS Thesaurus Consistency Checker*, *qSKOS* (as contributed in our earlier work [MH11, Mad12, MHI12] and described in detail in Section 4.2) and *Skosify*, that implements their repair strategy approach. Similar to our approach [MH11, MHI12] they identify a set of Web vocabularies alongside the number of found (potential) quality issues. The authors show that the introduced repair methods help to decrease the number of found quality issues.

### 2.4.3 Tools Related to Web Vocabulary Checking

Recently, a number of quality assessment tools have been published. The *PoolParty SKOS Thesaurus Consistency Checker*<sup>29</sup> implements tests for the six integrity conditions that are defined as part of the SKOS model and introduces custom checks such as URI syntax validation or missing labels. It was originally motivated by the need of a tool that checks whether a vocabulary can be imported into the *PoolParty Thesaurus Server* as it imposes some restrictions on Web vocabularies in terms of structural and labeling properties. The *PoolParty SKOS Thesaurus Consistency Checker* is now discontinued

---

<sup>29</sup>Offline since January 14<sup>th</sup> 2014.

and superseded by the *online SKOS Quality Checker*<sup>30</sup>, a tool based on *qSKOS* which has been developed in the course of this thesis, capable for evaluating the catalog of quality issues we propose. With this catalog we particularly focused on going beyond application-specific requirements and specifying a comprehensive suite of quality issues intended for use by developers that craft Web vocabularies for any purpose and usage scenario.

The *Skosify*<sup>31</sup> tool focuses on automatic repair of quality issues as it was originally developed for converting controlled vocabularies to RDF datasets using SKOS. The developers included checks against the integrity conditions defined in the SKOS reference as well as conversion-related issues such as label whitespace removals and detection of hierarchical cycles [SH12]. As presented in our earlier work [SM13] an improved version of the tool that has been “refined to better address issues detected by *qSKOS*” can help in reducing the quality issues found by *qSKOS*. However, these automated repair strategies are the contribution of the authors of *Skosify* and are therefore not within the scope of this thesis.

## 2.5 Connecting to Related Work

Concerning publication of linked datasets, in this work we partly build on the suggestions provided by Hogan et al. [HHP<sup>+</sup>10] and Heath and Bizer [HB11]. We adapted them to meet the requirements in a Web vocabulary context (checking for, e.g., undefined SKOS resources) and developed metrics that are both inexpensive to compute in an automated assessment process and provide value for human users when reviewing or comparing Web vocabularies (e.g., incoming and outgoing links). In particular we found link “dereferencability issues”, “undefined classes and properties” and “members of deprecated classes/properties” being also highly relevant in the context of Web vocabularies. We also adopted basic URI validity and dereferencability checks as they are vital for every Web vocabulary.

### 2.5.1 Relation to Ontology Evaluation Approaches

While we could adapt some criteria already suggested in existing work targeting ontology evaluation [PVdCSFGP12, TDM09, Vra10], such as consistent tagging of literals, these need to be completed by considering SKOS-specific properties. Like most existing work, in this thesis we also adopt a “closed world” view in our approach when defining and

<sup>30</sup>PoolParty online SKOS Quality Checker: <http://qskos.poolparty.biz>. Retrieved 2015-06-23.

<sup>31</sup>Skosify: <https://code.google.com/p/skosify/>. Retrieved 2015-06-23.

checking against quality issues of Web vocabularies. However, we do not use OWL built-in semantics as integrity constraints, as this is already done by existing tools and is only of minor relevance for assessing the quality of Web vocabularies, because in most cases they do not make use of OWL axioms. Where necessary, in order to evaluate the quality issues of our catalog, we perform reasoning in the RDFS domain and infer additional knowledge from the Web.

Abdul Manaf et al. [AMBS12a], for example, use OWL reasoning for finding common problems in SKOS vocabularies, focusing on patterns similar to those identified in existing work in the field of ontology evaluation. This is in contrast to our work, as we focus on Web vocabularies that pass such validation and reasoning checks but are troubled with potential problems on a higher level concerning the usefulness of the Web vocabulary from a knowledge representation point of view, concerning practical usage scenarios of Web vocabularies.

Once a suitable tool for automatic quality assessment of datasets is available (as, e.g., proposed by Kontokostas et al. [KWA<sup>+</sup>14]), integrating it into the development workflow is an obvious next step. Therefore, a test-driven approach has already been suggested in our earlier work [Mad12]. Kontokostas' approach is also similar to our contribution to Linked Data notifications (*rsine*, see Section 4.3) which we developed within the LOD2 project that is also based on specifying patterns using SPARQL. A difference is that the approach presented by Kontokostas et al. needs to be triggered externally whereas our notification approach is intended to instantaneously check on every triple change. Furthermore, in contrast to *rsine*, the generation of detailed easily readable reports is not within the scope of Kontokostas' work. From the perspective of quality assessment, both approaches are suitable for checking Web vocabularies against some of the issues from our catalog.

### 2.5.2 Notification Approaches

Our approach on Linked Data notifications is closely related to SparqlPuSH [PM10] but is designed to operate on a more general level. In particular, creation and subscription to feeds as proposed in SparqlPuSH is only one possible option of notifying subscribers. Furthermore, SparqlPuSH only relies on the extensiveness of the data contained in the underlying RDF store. Thus, it is not possible to make use of common change metadata in order to, e.g. find out about all resources deleted by a specific user in a certain period of time. Compared to SparqlPuSH our approach has the following advantages:

- Detection of changes to the dataset is done on the lowest possible level (e.g., database triggers) in the used triple store. This way we can assure that also

changes performed by applications using various connector libraries are detected. SparqlPuSH only detects data loaded into the triple store via an HTTP interface. However, as stated by the authors, it is planned to integrate the detection and update process more tightly with the triple store.

- Notification queries can make use of a common ontology for changeset metadata.
- It provides an extensible system for notification dissemination (semantic pingback, RSS or Atom feeds, email, twitter,...).
- Notification query results are directly delivered to the subscriber, making them also usable for machine users.

With *rsine* we stick to the approach of ECA rules as proposed by the REWERSE project [PPW06], but utilize a custom RDF ontology (Section 4.3.3) to express these rules. We furthermore decided to use SPARQL for definitions of both events and conditions because of its wide acceptance and our focus on RDF data. This results in a light-weight approach, eliminating the need for custom event matchers and detection engines in favor of SPARQL endpoints and incremental RDF changesets. Actions are represented in our Rsine ECA rules by specifying one or multiple notifiers (using the `rsine:notifier` property).

Although *rsine* is intended for notification rather than synchronization usecases, in comparison to the approach taken by the ResourceSync [VdSSK<sup>+</sup>12, KSVdS<sup>+</sup>13] framework, both approaches have in common that they both rely on persisting all changes that affect a resource managed by the data source. However, as our approach is push-based, we provide a means of selecting subgraphs of the dataset that is of interest and disseminate it to the subscriber.

### 2.5.3 Controlled Vocabulary Evaluation and Quality Assurance

In this thesis we utilize a similar classification of the identified quality issues as Kless and Milton [KM10] do, but, in contrast to their work, pursue a more formal approach in defining quality issues. Some constructs given by Kless are designed for intellectual evaluation (e.g., “Conceptual clarity” or “Complexity”), whereas some can serve as starting point for defining formalized measurements (e.g., “Documentation completeness” or “Structural correctness”). Thus, for our catalog of potential quality issues we adopt and refine some of these measurements in a formal way.

We found that we can adopt some of Soergel’s recommendations [Soe02] and (re-)formulate them to be used for automated Web vocabulary evaluation. Among them are, e.g., the



recommendations to provide context and definitions for concepts, support for multiple languages, inclusion of relationships to other KOS, or “Completeness of coverage of the terminology from a given language”.

From Hedden’s work [Hed10] we do not pick up any recommendations that require evaluation against a text corpus such as the number of terms per document used for indexing or frequency of term usage. In the context of this work we focus on analyzing a Web vocabulary as “self-contained” entity, i.e., without making use of corpora of items indexed with terms from the vocabulary. Hence, we adopted some structures Hedden considers as “unusual” or “needless information” like orphan concepts that have also been mentioned in other publications (e.g., Aitchison et al. [ABG03]) for usage with Web vocabularies and added them to our catalog of quality issues. We leave her other recommendations that require language processing and intellectual understanding of term meanings for our future work.

#### 2.5.4 Quality of Web Vocabularies

In this work we mainly focus on quality issues that go beyond the integrity constraints that are defined in the SKOS reference document [MB09]. However, we also include them in our catalog of potential quality issues and contribute an implementation. The reason is that they can be seen as a starting point for SKOS validation and since they have not been specified as OWL axioms, we consider it necessary to provide our interpretation of them (Section 3.1.4). Apart from that they are furthermore important in our research methodology when reviewing the effects of integrating our quality assessment approach into the vocabulary development process.

In contrast to the approach taken by Kawtrakul et al. [KIT<sup>+</sup>05] who use third-party datasets (WordNet) and an expert-defined set of rules to enrich a thesaurus, our approach focuses on finding potential problems. It is similar to Kawtrakul et al. in the way we define a ruleset (the catalog of quality issues) against which a vocabulary is checked. Another commonality with our approach is that we also require human experts to judge the results of the algorithms, i.e., the suggested relations in case of Kawtrakul and the generated quality report in our approach. However, our approach is designed to be applicable on any kind of SKOS vocabulary, so we cannot rely on additional type information of the concepts or WordNet information.

Coronado et al. [dCWF<sup>+</sup>09], underline the importance of manually reviewing changes in the editing phase of the *NCI Thesaurus*, which is a process we addressed when developing a notification framework for dataset changes (Section 4.3) and apply it for assessing

potential quality problems. We also make use of automated report generation but focus on rules (quality issues) applicable on a more general level for Web vocabularies.

Finally, automatic repair strategies are explicitly out of the scope of this thesis. Although it has been shown [SH12, SM13] that such mechanisms can be used to reduce the number of occurring quality issues, verification of these repair strategies remains an open question. Therefore, we believe that judgment of validity and severity of identified quality issues must be performed by human developers. However, we consider automatic repair strategies useful for improving the usability of future tools when, e.g. resolving a large number of quality issues is needed.

## 2.6 Summary

In this chapter we provided an introduction to the field of controlled vocabularies, their usage, and relevance for applications that are making use of Linked Data. We discussed publications of solutions adjacent to the approaches that we propose in this work and outline similarities, differences, and the relation to our contributions.

Due to the large number of publications paying attention to controlled vocabulary quality we found the topic to be of importance for both “traditional” vocabularies as well as Web vocabularies. Numerous proprietary approaches and guidelines for quality assurance exist. However, they have in common that they are not generally applicable, not automatically assessable, or do not take Web vocabularies into account. Regarding quality of Linked Data, most existing approaches find formal errors in data representation syntax or focus on inconsistencies caused by missing data or information inferred using reasoning approaches. As work on formal quality constraints that focuses on the SKOS semantic model is currently underrepresented, this thesis contributes towards filling this gap. We propose a catalog of quality issues that are founded in the covered related work but have been adapted to meet the requirements of developing and using Web vocabularies.

From the existing work covering Linked Data notification approaches, we were able to adopt certain approaches that led to the implementation of a tool capable for reporting Web vocabulary quality violations as soon as changes to the vocabulary are performed. When evaluating the integration of our quality assessment approach into Web vocabulary development processes, we also adopted methodologies of existing work, as we perform a study following a within-subject design.

## Chapter 3

# Formal Definition of Quality Issues

In this chapter we describe our approach for covering Research Question 1 (*What properties of a Web vocabulary have an impact on its quality as perceived by human users?*). We provide a catalog of *quality issues*, i.e., patterns observed in Web vocabularies that can potentially degrade their quality. We formally describe each quality issue and show how a SKOS vocabulary can be checked for occurrences of the issue.

The catalog is one of the main scientific contributions of this thesis and parts of it have already been published [Mad12, MHI12, SM13]. Here we extend our earlier work by

- formal definitions of each quality issue, based on the RDF formal semantics [LS99] which is also described in [HKR10],
- providing additional quality issues that we identified in our subsequent research, and
- including a more detailed coverage of the design rationale of each quality issue.

### 3.1 Catalog of Quality Issues

The methodology described in Section 1.4 allowed us to identify 29 quality issues, and their respective quality functions. These functions identify subgraphs in the RDF representation of Web vocabularies that, to a high probability, indicate quality problems.

We divided the quality issues into four categories: *Labeling and Documentation Issues* focus on the (lack of) definition of literal resources that help human users in understanding and using the vocabulary. *Structural Issues* cover patterns of presence and kind of

specific relations that enable or restrict the vocabulary to be used for certain usage scenarios. As we cover these two categories, we look at the vocabulary as a self-contained entity, which means that we do not take links to other vocabularies into account. These links are considered by the issues in the category *Linked Data Specific Issues*. The last category, *SKOS Consistency Issues*, covers consistency checks that are mentioned in the SKOS reference documentation without a formal definition.

In the following sections, we explain the origin and design rationale for each quality issue and describe how occurrences can be detected in a SKOS vocabulary. We provide the formal definition of each function that is based on model-theoretic semantics of RDF(S) given by Hitzler et al. [HKR10], which is again based on Hayes [Hay04]. In order to keep the definitions of the quality functions as simple as possible, we introduce “intermediate” sets and functions where needed which are in some cases reused by subsequent definitions. For a better understanding of our definitions and to improve readability, we furthermore include parts of Hitzler’s work that we build upon.

All issues in each section are provided without assigning grades of severity to the issues, because such a judgment is highly dependent on the context and intended application of the Web vocabulary. However, in the course of the catalog development process, we received individual feedback from experts that serves as an indicator for defining levels of severity for a few issues of this catalog regarding the vocabulary usage-scenarios. We cover this in more detail in Section 5.1.

Hitzler et al. define a vocabulary  $V$  as an arbitrary set of URIs and literals describing a domain of interest by defining “individuals [...] and their relations” as well as, e.g., their “types or classes” like “person” or “institution”. They also define the notion of an *interpretation*  $\mathcal{I}$  of an RDF graph: an interpretation of an RDF graph constitutes *one* “possible world” or “reality” which is described by the graph. Depending on what semantic model or knowledge representation language (e.g., RDFS<sup>1</sup>) is used, different facts can be inferred from the graph.

The most basic interpretation defined by Hitzler et al. is the *simple interpretation* for handling resources, literals, and properties of an RDF graph. A *simple interpretation*  $\mathcal{I}$  of a vocabulary  $V$  contains

- $IR$ , a non-empty set of *resources*, alternatively called domain or universe of discourse of  $\mathcal{I}$ ,
- $IP$ , the set of *properties* of  $\mathcal{I}$  (which may overlap with  $IR$ ),

---

<sup>1</sup>RDF Schema: <http://www.w3.org/TR/rdf-schema/>. Retrieved 2015-06-23.

- the *extension function*  $I_{EXT}$  with  $I_{EXT} : IP \rightarrow 2^{IR \times IR}$  that assigns a set of pairs of resources from  $IR$  to each property in  $IP$ .

**Definition 3.1** (Interpretation Function). An interpretation function  $\cdot^{\mathcal{I}}$  maps all (typed and untyped) literals and URIs that are contained in a vocabulary  $V$  to resources and properties:

- Literals with language information are mapped to pairs that hold both the label and the language information, i.e.,  $(\text{"a"@t})^{\mathcal{I}} = \langle a, t \rangle$ .
- Every URI  $u$  is mapped to  $I_S(u)$ , i.e.,  $u^{\mathcal{I}} = I_S(u)$  with  $I_S$  being a function that maps URIs from a vocabulary to the union of the sets of  $IR$  and  $IP$ , i.e.,  $I_S : V \rightarrow IR \cup IP$ .

A more in-depth coverage can be found in Hitzler et al. [HKR10].

Interpretations hence map the elements of RDF graphs (i.e., nodes and edges defined as triples) which constitute of resources (URIs and blank nodes) and literals, to the sets  $IR$  and  $IP$ . Based on the *simple interpretation*, Hitzler et al. define the *RDF interpretation* and *RDFS interpretation* of a Web vocabulary  $V$  which introduce increasing levels of semantic complexity. The RDF interpretation adds, e.g., the possibility to assign types to resources (using the property `rdfs:type`) whereas the *RDFS interpretation*, among others, introduces the notion of classes, subclasses, or subproperties. Therefore they introduce a *class extension function*  $I_{CEXT} : IR \rightarrow 2^{IR}$  that maps resources to sets of resources.  $I_{CEXT}(y)$  contains only those elements  $x$  for which  $\langle x, y \rangle$  is contained in  $I_{EXT}(\text{rdfs:type}^{\mathcal{I}})$ . Based on  $I_{CEXT}$ , a valid RDFS interpretation of a vocabulary  $V$  must also satisfy the criteria that:

- $IR = I_{CEXT}(\text{rdfs:Resource}^{\mathcal{I}})$ , i.e., every resource has the type `rdfs:Resource`, and
- $LV = I_{CEXT}(\text{rdfs:Literal}^{\mathcal{I}})$ , i.e., every untyped or well-typed literal has the type `rdfs:Literal`.

**Definition 3.2** (Terminology for RDFS Interpretations of SKOS Vocabularies). For each RDFS interpretation that is a valid model of a SKOS vocabulary, in this thesis we refer to the following terminology:

- $C \subseteq IR$  with  $C = I_{CEXT}(\text{skos:Concept}^{\mathcal{I}})$  being the set of *SKOS concepts*,

- $AC \subseteq C$  being the set of *authoritative concepts*, i.e., all concepts that are identified by URIs in the vocabulary namespace(s), as opposed to concepts from other vocabularies that are referenced in the vocabulary,
- $SR = I_{EXT}(\text{skos:semanticRelation}^T)$  being the set of *semantic relations* associating concepts with one another, and
- $CS = I_{EXT}(\text{skos:ConceptScheme}^T)$  being the set of *SKOS concept schemes*.

We provided here only the conditions that interpretations must meet in order to be valid simple or RDF(S) interpretations of a SKOS vocabulary  $V$  and which are necessary to understand the definitions for the quality issues which we define in the following. The complete definitions can be found in [Hay04, HKR10].

The quality issues we introduce in the remainder of this section make use of various Linked Data schemas. An overview of them is provided in Table 3.1.

Name	Prefix	Namespace
Simple Knowledge Organization System	skos	<a href="http://www.w3.org/2004/02/skos/core#">http://www.w3.org/2004/02/skos/core#</a>
Resource Description Framework	rdf	<a href="http://www.w3.org/1999/02/22-rdf-syntax-ns#">http://www.w3.org/1999/02/22-rdf-syntax-ns#</a>
RDF Schema	rdfs	<a href="http://www.w3.org/2000/01/rdf-schema#">http://www.w3.org/2000/01/rdf-schema#</a>
Dublin Core Metadata Element Set	dc	<a href="http://purl.org/dc/elements/1.1/">http://purl.org/dc/elements/1.1/</a>
DCMI Metadata Terms	dcterms	<a href="http://purl.org/dc/terms/">http://purl.org/dc/terms/</a>
Web Ontology Language	owl	<a href="http://www.w3.org/2002/07/owl#">http://www.w3.org/2002/07/owl#</a>

TABLE 3.1: Linked Data schemas used in quality issue definitions

### 3.1.1 Labeling and Documentation Issues

The quality issues introduced in this section focus on presence and proper definition of certain literals in  $LV$ . Throughout this catalog we assume that the SKOS integrity condition S12 defined in the SKOS schema<sup>2</sup> is fulfilled, i.e., “The `rdfs:range` of each of `skos:prefLabel`, `skos:altLabel`, and `skos:hiddenLabel` is the class of RDF plain literals”.

#### 3.1.1.1 Omitted or Invalid Language Tags

SKOS defines a set of properties that link resources with RDF literals, which are plain text strings in natural language with an optional language tag. This includes the labeling properties `skos:prefLabel`, `skos:altLabel`, `skos:hiddenLabel`, all of which are subproperties of `rdfs:label`. In addition, the SKOS documentation properties, that are

<sup>2</sup>Integrity condition S12: <http://www.w3.org/TR/skos-reference/#S12>. Retrieved 2015-06-23.

subproperties of `skos:note` (such as `skos:definition` or `skos:scopeNote`) are also often used to assign textual information to a context, although the SKOS reference imposes “no restriction on the nature of this information, e.g., it could be plain text, hypertext, or an image”<sup>3</sup>.

This quality issue requires that the language of each literal which is intended to hold information for human users should be provided consistently in the form of a “language tag”. This has also been pointed out in [Vra10] on a more general level. Omitting language tags or using non-standardized, private language tags in a SKOS vocabulary could unintentionally limit the result set of language-dependent queries.

We can define the quality checking function for this issue by first defining:

- $DR$ , a set of pairs from  $IR \times LV$ , denoting all *documented resources* in  $V$ , i.e.,  $DR = \{\langle ir, lv \rangle : \langle ir, lv \rangle \in I_{EXT}(\text{rdfs:label}^I) \cup I_{EXT}(\text{skos:note}^I), lv \in LV\}$ ,
- $LANG$ , a set of all language tags in the vocabulary  $V$ , i.e.,  $LANG = \{\pi_2(lv) : lv \in LV \text{ with } lv \text{ carrying language information, i.e., } lv = \langle a, t \rangle = (\text{"a"@t})^I\}$  (where  $\pi_2$  denotes the projection of the second element in the pair  $lv = \langle a, t \rangle$ ),
- $lang$ , a function mapping each literal to its language information, i.e.,  $lang : LV \rightarrow LANG$  with

$$lang(lv) = \begin{cases} \pi_2(lv) & \text{if the literal carries language information} \\ \emptyset & \text{otherwise} \end{cases}$$

, and

- $tag$ , a function indicating validity of a literal’s language tag, i.e.,  $tag : LV \rightarrow \{0, 1\}$  with

$$tag(lv) = \begin{cases} 1 & \text{if } lang(lv) \text{ is a language tag that (i) complies with the syntactic} \\ & \text{rules of BCP47}^4 \text{ and (ii) contains language codes listed in the} \\ & \text{ISO 639}^5 \text{ standard} \\ 0 & \text{otherwise} \end{cases}$$

We define  $oilt$  to be a function that checks a resource in  $IR$  for omitted or invalid language tags ( $oilt$ ), i.e.,  $oilt : IR \rightarrow \{0, 1\}$  with

$$oilt(ir) = \begin{cases} 1 & \text{if } \langle ir, lv \rangle \in DR \text{ and } lang(lv) = \emptyset \vee tag(lv) = 0 \\ 0 & \text{otherwise} \end{cases}$$

<sup>3</sup>SKOS reference documentation properties description: <http://www.w3.org/TR/skos-reference/#L2860>. Retrieved 2015-06-23.

An RDF graph  $G$  that defines a SKOS vocabulary therefore contains *omitted or invalid language tags* if for all possible interpretations  $I$  that are a model of  $G$ ,  $oilt(r^{\mathcal{I}}) = 1$  for at least one resource node  $r$  in  $G$ .

### 3.1.1.2 Incomplete Language Coverage

The set of language tags used by the literal values linked with a concept should be the same for all concepts. This is, for example, suggested in [Iso11a]: “So that a thesaurus can function effectively in a multilingual context, the concepts included need to be represented in all of the languages present, enabling speakers of these languages to have access to them”. If this is not the case, appropriate actions like splitting concepts or introducing scope notes should be taken by the thesaurus developers. This is particularly important for applications that rely on internationalization and translation use cases.

In order to define the quality checking function we first let:

- $EXT$  be the set of all possible extensions in  $V$ , i.e.,  $EXT = \bigcup_{p \in IP} I_{EXT}(p)$ , and
- $lc$ , be a language coverage function that maps each authoritative concept in  $V$  to the set of language tags of its assigned literals, i.e.,  $lc : AC \rightarrow 2^{LANG}$  with  $lc : ac \mapsto \{lang(lv) : \langle ac, lv \rangle \in EXT, lv \in LV \text{ and } lv \text{ carrying language information, i.e., } lv = \langle a, t \rangle = (\text{"a"@t})^{\mathcal{I}}\}$ .

Based on these definitions we can formalize the quality checking function for incomplete language coverage ( $ilc$ ) as  $ilc : AC \rightarrow \{0, 1\}$  with

$$ilc(ac) = \begin{cases} 1 & \text{if } LANG \setminus lc(ac) \neq \emptyset \\ 0 & \text{otherwise} \end{cases}$$

An RDF graph  $G$  that defines a SKOS vocabulary therefore has *incomplete language coverage* if for all possible interpretations  $I$  that are a model of  $G$ ,  $ilc(r^{\mathcal{I}}) = 1$  for at least one resource node  $r$  in  $G$ .

### 3.1.1.3 No Common Language

As pointed out in the IFLA *Guidelines for Multilingual Thesauri*<sup>6</sup>, one practice for developing multilingual thesauri is to start with one language and add other languages

---

<sup>6</sup>An abstract can be found at <http://www.ifla.org/publications/ifla-professional-reports-115>. Retrieved 2015-06-23.



when necessary. Therefore, checking for a common language is useful to identify “gaps” in such an initial thesaurus.

However, it is not always possible to describe each concept in a vocabulary using the same set of languages. In such cases, the ISO 25964-1 standard [Iso11a] suggests “to treat the different language versions of the multilingual thesaurus as if they were two or more parallel monolingual thesauri and to establish mappings between the corresponding terms”. It is then necessary for each concept in the language-specific subthesauri to be documented in this common language. We therefore regard it as a quality issue if (a subset of) all authoritative concepts in a vocabulary are not documented in at least one common language.

We define the quality checking function for no common language (ncl)  $ncl : 2^{AC} \rightarrow \{0, 1\}$  that maps a set of authoritative concepts to a truth value as follows:

$$ncl(\{ac_1, \dots, ac_n\}) = \begin{cases} 1 & \bigcap_{i=0}^n lc(ac_i) = \emptyset \\ 0 & \text{otherwise} \end{cases}$$

An RDF graph  $G$  that defines a SKOS vocabulary therefore has *no common language* if for all possible interpretations  $I$  that are a model of  $G$ ,  $ncl(R^I) = 1$  for some set  $R$  of resource nodes in  $G$ .

#### 3.1.1.4 Undocumented Concepts

The SKOS Reference [MB09] defines a set of “documentation properties”, all of which are subproperties of `skos:note`. For example, `skos:scopeNotes` are, according to the SKOS schema specification, used to help to clarify the meaning and/or the use of a concept (in relation to other concepts) and the property `skos:historyNote` can serve as a means for documenting the evolution of a vocabulary. The requirements of being able to correctly interpret (*interpretability*) and understand (*understandability*) information are often mentioned as data quality dimensions (e.g., [BS06, SLW97]) and the use of SKOS documentation properties helps in fulfilling them.

In order to define the quality function we let  $DAC \subseteq AC$  be the set of all *documented authoritative concepts*, i.e.,  $DAC = \{dac : \langle dac, lv \rangle \in I_{EXT}(\text{skos:note}^I), lv \in LV\}$ . The quality checking function for undocumented concepts (uc) can be defined as  $uc : AC \rightarrow \{0, 1\}$  with

$$uc(ac) = \begin{cases} 1 & \text{if } ac \notin DAC \\ 0 & \text{otherwise} \end{cases}$$

An RDF graph  $G$  that defines a SKOS vocabulary therefore contains *undocumented concepts* if for all possible interpretations  $I$  that are a model of  $G$ ,  $uc(r^I) = 1$  for at least one resource node  $r$  in  $G$ .

### 3.1.1.5 Overlapping Labels

The SKOS Primer [IS09] recommends that “no two concepts have the same preferred lexical label in a given language when they belong to the same concept scheme” (see also Section 3.1.4.3). For this quality issue we generalize the above recommendation and search for all concept pairs with identical `skos:prefLabel`, `skos:altLabel` or `skos:hiddenLabel` property values.

We first define  $lab$ , a function that maps each concept to the set of literals it has asserted by one of the SKOS label properties, i.e.  $lab : C \rightarrow 2^{LV}$  with  $lab : c \mapsto \{lv : \langle c, lv \rangle \in I_{EXT}(\text{skos:prefLabel}^I) \cup I_{EXT}(\text{skos:altLabel}^I) \cup I_{EXT}(\text{skos:hiddenLabel}^I)\}$ .

We can then define the quality checking function for overlapping labels as  $ol : C \times C \rightarrow \{0, 1\}$  with

$$ol(c_1, c_2) = \begin{cases} 1 & \text{if } lab(c_1) \cap lab(c_2) \neq \emptyset \\ 0 & \text{otherwise} \end{cases}$$

An RDF graph  $G$  that defines a SKOS vocabulary therefore contains *overlapping labels* if for all possible interpretations  $I$  that are a model of  $G$ ,  $ol(rr^I) = 1$  for at least one pair  $rr$  of resource nodes in  $G$ .

For practical reasons, we extend the definition of this quality function with a certain similarity threshold that must be met in order to regard two labels as overlapping. The rationale is that, depending on the used string similarity function, this threshold can be adjusted for, e.g., performing case-tolerant comparisons or finding identical labels that contain accidentally swapped characters. Although issues of this kind are acceptable for some thesauri, they can affect some application scenarios such as auto-completion, which anticipates search terms based on user input.

For the extended version of  $ol'$  of this quality checking function, we define  $sim$ , a label similarity function that maps each pair of literal values in  $LV$  to a similarity value, i.e.,  $sim : LV \times LV \rightarrow [0, 1]$  where 1 means that the literals can be considered identical.

We can then rewrite the above definition of  $ol$  as  $ol' : C \times C \rightarrow \{0, 1\}$  with

$$ol'(c_1, c_2) = \begin{cases} 1 & \text{if } sim(lv_1, lv_2) \geq t \text{ with } lv_1 \in lab(c_1), lv_2 \in lab(c_2) \text{ and} \\ & lang(lv_1) = lang(lv_2) \\ 0 & \text{otherwise} \end{cases}$$

where  $t$  is a defined threshold value in the interval  $[0,1]$ .

### 3.1.1.6 Missing Labels

In order to improve readability and understandability of a controlled vocabulary by human users, labels should be assigned to each `skos:Concept` and `skos:ConceptScheme`. Labels are required for utilizing the vocabulary in search and retrieval use cases based on human input. For this issue we adopt a definition originally provided by the now defunct *PoolParty online vocabulary consistency checker*. The rationale behind this quality issue is that each `skos:Concept` should have at least one preferred label assigned whereas `skos:ConceptSchemes` should make use of label properties from commonly known schemas such as `rdfs:label`, `dc:title`, or `dcterms:title`<sup>7</sup>.

We base the definition of the quality checking function for this issue on the following three sets:

- $LAC \subseteq AC$ , the set of *authoritative concepts with preferred labels*, i.e.,  $LAC = \{ac : \langle ac, lv \rangle \in I_{EXT}(\text{skos:prefLabel}^{\mathcal{I}}), ac \in AC, lv \in LV\}$ ,
- $RCL$ , the set of pairs of resources  $\langle r, l \rangle$ , such that  $r$  has asserted a particular common-typed label, defined as follows:  $RCL = I_{EXT}(\text{rdfs:label}^{\mathcal{I}}) \cup I_{EXT}(\text{dc:title}^{\mathcal{I}}) \cup I_{EXT}(\text{dcterms:title}^{\mathcal{I}})$ , and
- $LCS$ , the set of *labeled concept schemes*, i.e.,  $LCS = CS \cap \bigcup_{rcl \in RCL} \pi_1(rcl)$ .

According to the definitions provided above, the quality checking function for missing labels (ml) checks each authoritative `skos:Concept` or `skos:ConceptScheme` for assigned labels, formally:  $ml : AC \cup CS \rightarrow \{0, 1\}$  with

$$ml(c) = \begin{cases} 1 & \text{if } c \notin LAC \cup LCS \\ 0 & \text{otherwise} \end{cases}$$

An RDF graph  $G$  that defines a SKOS vocabulary therefore contains *missing labels* if for all possible interpretations  $I$  that are a model of  $G$ ,  $ml(r^{\mathcal{I}}) = 1$  for at least one resource node  $r$  in  $G$ .

<sup>7</sup>See also the discussion at the `public-esw-thes@w3.org` mailing list: <http://lists.w3.org/Archives/Public/public-esw-thes/2011Mar/0010.html>. Retrieved 2015-06-23.

### 3.1.1.7 Unprintable Characters in Labels

In most cases, concept labels in Web vocabularies are used for indexing or search purposes, and thus they are intended to be read and used by human users. Also when performing dataset queries, literal values of concept labels are used as input for string comparison algorithms. For these purposes invisible control characters like tab stops or line breaks can cause search tasks to fail if the algorithm requires every single character to match. We therefore consider these invisible control characters problematic if they are contained in concept labels.

Let  $upc : LV \rightarrow \{0, 1\}$  be a function that maps a literal value to the value 1 if it contains unprintable control characters (i.e., Unicode characters assigned to category “C”) and to 0 else. We can then define the quality checking function for unprintable characters in labels ( $ucil$ ) for this issue as  $ucil : C \rightarrow \{0, 1\}$  with

$$ucil(ac) = \begin{cases} 1 & \text{if } upc(lv) = 1 \text{ with } lv \in lab(ac) \\ 0 & \text{otherwise} \end{cases}$$

An RDF graph  $G$  that defines a SKOS vocabulary therefore contains *unprintable characters in labels* if for all possible interpretations  $I$  that are a model of  $G$ ,  $ucil(r^I) = 1$  for at least one resource node  $r$  in  $G$ .

### 3.1.1.8 Empty Labels

It is not only sufficient to assign labels to the resources of a vocabulary, but these labels also have to carry useful textual information. Labels consisting of no text (empty strings) or only whitespaces do not provide additional information to the vocabulary users. This issue is related to the “Extra Whitespace” quality criterion defined in [SH12]. However, we do not focus on whitespace characters at the start or end of textual labels but instead target labels that consist *only* of whitespaces or have zero-length literals and thus do not contain any alphanumeric character.

We define the quality checking function for empty labels ( $el$ ) as  $el : IR \rightarrow \{0, 1\}$  with

$$el(r) = \begin{cases} 1 & \text{if } \exists \langle r, l \rangle \in RCL \text{ with } l \in LV \text{ and } l = \langle a, t \rangle \text{ (if } l \text{ carries language} \\ & \text{information } t \text{) or } l = a \text{ (if not carrying any language information)} \\ & \text{and } a \text{ contains no alphanumeric characters} \\ 0 & \text{otherwise} \end{cases}$$

An RDF graph  $G$  that defines a SKOS vocabulary therefore contains *empty labels* if for all possible interpretations  $I$  that are a model of  $G$ ,  $el(r^I) = 1$  for at least one resource node  $r$  in  $G$ .

### 3.1.1.9 Ambiguous Notation References

The ISO 25964-1 standard defines a *notation* as a “set of symbols representing a concept [...] in a structured vocabulary [...], especially a classification scheme [...]”. The document furthermore states that notations are used for sorting and locating concepts and to impose a structure for displaying the vocabulary. Notations are defined in, e.g., the Universal Decimal Classification or the Dewey Decimal Classification to identify each classification subject.

SKOS supports assigning multiple notations to a resource by means of the `skos:notation` property<sup>8</sup>. The SKOS reference also treats notations in conformance to controlled vocabulary standards by stating that “no two concepts in the same concept scheme are given the same notation” because in a SKOS vocabulary the value of the notation property may be used “to uniquely refer to a concept”. Additionally we stipulate that each authoritative concept should have assigned at most one unique notation.

To define the quality checking function for this issue, we let

- $ccs : AC \rightarrow 2^{CS}$  be a function that maps an authoritative concept to the set of `skos:ConceptSchemes` to which it is assigned (the “containing” concept schemes), i.e.,  $ccs(ac) = \{cs : \langle ac, cs \rangle \in I_{EXT}(\text{skos:inScheme}^I) \cup I_{EXT}(\text{skos:topConceptOf}^I) \text{ or } \langle cs, ac \rangle \in I_{EXT}(\text{skos:hasTopConcept}^I)\}$ , and
- $not : AC \rightarrow 2^{LV}$  be a function that maps an authoritative concept to the unique notations it has assigned, i.e.,  $not(ac) = \{lv : \langle ac, lv \rangle \in I_{EXT}(\text{skos:notation}^I)\}$

The quality checking function  $anr$  finds ambiguous notation references ( $anr$ ) for a pair of authoritative concepts. It can be defined as  $anr : AC \times AC \rightarrow \{0, 1\}$  with

$$anr(ac_1, ac_2) = \begin{cases} 1 & \text{if } \exists \langle ac_1, n \rangle, \langle ac_2, n \rangle \in I_{EXT}(\text{skos:notation}^I) \text{ with} \\ & ccs(ac_1) \cap ccs(ac_2) \neq \emptyset \text{ or } |not(ac_1)| > 1 \text{ or } |not(ac_2)| > 1 \\ 0 & \text{otherwise} \end{cases}$$

<sup>8</sup>SKOS reference notations description: <http://www.w3.org/TR/skos-reference/#L2064>. Retrieved 2015-06-23.

An RDF graph  $G$  that defines a SKOS vocabulary therefore contains *ambiguous notation references* if for all possible interpretations  $I$  that are a model of  $G$ ,  $anr(rr^I) = 1$  for at least one pair  $rr$  of resource nodes in  $G$ .

### 3.1.2 Structural Issues

The quality issues we introduce in the following sections cover the relations between resources of the types `skos:Concept` and `skos:ConceptScheme` within a Web vocabulary. Search and retrieval applications that use thesauri often exploit their hierarchical and associative structure to (automatically) broaden or narrow the set of search results (query expansion) or rely on these relations for visualizations. It is therefore important to have quality checks in place that can help to analyze if the structure of a vocabulary may impede these application functionality.

#### 3.1.2.1 Orphan Concepts

This issue is motivated by the notion of “orphan terms” in the literature [Hed10], i.e., terms without any associative or hierarchical relationships. Checking for such terms is common in thesaurus development and also suggested by the ANSI/NISO Z39.19 guidelines [NIS05]. Since SKOS follows a concept-centric approach, we define an orphan concept as being a concept that has no semantic relation to any other concept. Although it might have attached lexical labels, it lacks valuable context information, which can be essential for term disambiguation or retrieval tasks such as query expansion.

We define the set  $SRC$  to contain all *semantically related concepts*, i.e.,  $SRC = \{\pi_i(sr) : sr \in I_{EXT}(\text{skos:semanticRelation}^I), 0 \leq i \leq 1\}$ . The quality checking function  $oc : C \rightarrow \{0, 1\}$  for orphan concepts ( $oc$ ) can then be defined as

$$oc(c) = \begin{cases} 1 & \text{if } c \notin SRC \\ 0 & \text{otherwise} \end{cases}$$

An RDF graph  $G$  that defines a SKOS vocabulary therefore contains *orphan concepts* if for all possible interpretations  $I$  that are a model of  $G$ ,  $oc(r^I) = 1$  for at least one resource node  $r$  in  $G$ .

#### 3.1.2.2 Disconnected Concept Clusters

Concepts which are defined in Web vocabularies are sometimes split into separate disconnected “clusters”, i.e., sets of concepts that are semantically related among each other

but not related to concepts contained in another cluster. Reasons for this can be, e.g., incomplete data acquisition, deprecated terms, or accidental deletion of relations. Disconnected concept clusters can affect operations that rely on navigating a connected vocabulary structure, such as query expansion or suggestion of related terms. However, besides the intended application that should be supported by the vocabulary, also the type of the vocabulary itself may determine whether disconnected clusters of concepts are allowed or not. Svenonius [Sve], for example, states that “While the classificatory structure of a classification like the DDC is often likened to a gigantic upside down tree, that of a thesaurus might be said to resemble a collection of small shrubs”.

We define

- $G_{SRC} = \langle N, E \rangle$  as an undirected graph of all semantically related concepts in a SKOS vocabulary containing the nodes  $N = SRC$  and the edges  $E = I_{EXT}(\text{skos:semanticRelation}^T)$ , and
- $MCC \subset 2^C$  being a set that contains all *maximally connected components*<sup>9</sup> of  $G_{SRC}$ .

An RDF graph  $G$  that defines a SKOS vocabulary therefore contains *disconnected concept clusters* if for all possible interpretations  $I$  that are a model of  $G$ ,  $|MCC| > 1$ .

### 3.1.2.3 Cyclic Hierarchical Relations

This issue is motivated by Soergel et al. [Soe02] who suggest a “check for hierarchy cycles” because they can potentially “throw the program [into] a loop in the generation of a complete hierarchical structure”. Also Hedden [Hed10], Harpring [Har10], and Aitchison et al. [ABG03] argue that there exist common hierarchy types such as “generic-specific”, “instance-of”, or “whole-part” where cycles would be considered a logical contradiction.

We define two sets that formalize the notion of hierarchical relationships in a vocabulary  $V$ :

- $HR$ , being a set of all ordered pairs of *directly hierarchically related concepts*, i.e.,  $HR = \{ \langle c_1, c_2 \rangle : \langle c_1, c_2 \rangle \in I_{EXT}(\text{skos:broaderTransitive}^T) \vee \langle c_2, c_1 \rangle \in I_{EXT}(\text{skos:narrowerTransitive}^T) \}$  with  $c_1, c_2 \in C$ , and
- $HPATH$ , the set of all *hierarchical paths* in  $V$ , i.e.,  $HPATH = \{ \langle c_1, \dots, c_n \rangle : \langle c_i, c_{i+1} \rangle \in HR, 1 \leq i < n, c_i \in C, n \leq |C| \}$ .

<sup>9</sup>A definition is provided at <http://xlinux.nist.gov/dads/HTML/maximallyConnectedComponent.html>. Retrieved 2015-06-23.

We can then define the quality checking function  $chr : C \times C \rightarrow \{0, 1\}$  that checks if two concepts are part of a cyclic hierarchical relation ( $chr$ ) as

$$chr(c_1, c_2) = \begin{cases} 1 & \text{if } \langle c_1, \dots, c_2 \rangle \in HPATH \wedge \langle c_2, \dots, c_1 \rangle \in HPATH \\ 0 & \text{otherwise} \end{cases}$$

An RDF graph  $G$  that defines a SKOS vocabulary therefore contains *cyclic hierarchical relations* if for all possible interpretations  $I$  that are a model of  $G$ ,  $chr(rr^{\mathcal{I}}) = 1$  for at least one pair  $rr$  of resource nodes in  $G$ .

### 3.1.2.4 Valueless Associative Relations

The ISO 25964-1 standard [Iso11a] suggests that terms that share a common broader term should not be related associatively if this relation is only justified by the fact that they are siblings. This is also advocated by Hedden [Hed10] and Aitchison et al. [ABG03] who point out “the risk that thesaurus compilers may overload the thesaurus with valueless relationships”, having a negative effect on search precision.

The SKOS reference document defines the property `skos:related` as symmetric property, i.e.,  $\text{skos:related}^{\mathcal{I}} \in I_{EXT}(\text{owl:SymmetricProperty}^{\mathcal{I}})$ . Therefore, in the scope of this quality issue, we stipulate that a valid RDFS interpretation of  $V$  must also satisfy the criterion that if  $\langle c_1, c_2 \rangle \in I_{EXT}(\text{skos:related}^{\mathcal{I}})$  then  $\langle c_2, c_1 \rangle \in I_{EXT}(\text{skos:related}^{\mathcal{I}})$ .

To formalize the quality checking function, we let  $SIB$  be the set of all *sibling pairs*, i.e., pairs of concepts that share an immediate common broader concept. Formally,  $SIB = \{\langle c_1, c_2 \rangle : \langle c_1, c_p \rangle \in HR \wedge \langle c_2, c_p \rangle \in HR\}$ .

The quality checking function  $var : C \times C \rightarrow \{0, 1\}$  for finding valueless associative relations ( $var$ ) between two concepts can then be defined as

$$var(c_1, c_2) = \begin{cases} 1 & \text{if } \langle c_1, c_2 \rangle \in SIB \wedge \langle c_1, c_2 \rangle \in I_{EXT}(\text{skos:related}^{\mathcal{I}}) \\ 0 & \text{otherwise} \end{cases}$$

An RDF graph  $G$  that defines a SKOS vocabulary therefore contains *valueless associative relations* if for all possible interpretations  $I$  that are a model of  $G$ ,  $var(rr^{\mathcal{I}}) = 1$  for at least one pair  $rr$  of resource nodes in  $G$ .



### 3.1.2.5 Solely Transitively Related Concepts

Two concepts that are explicitly related by `skos:broaderTransitive` and/or `skos:narrowerTransitive` can be regarded a quality issue because, according to the SKOS Reference [MB09], these properties are “not used to make assertions”<sup>10</sup>. Transitive hierarchical relations in SKOS are meant to be inferred by the vocabulary consumer, which is reflected in the SKOS ontology by, for instance, `skos:broaderTransitive` being defined as a subproperty of `skos:broaderTransitive`.

The SKOS reference document defines the property `skos:broader` as inverse property of `skos:narrower` and vice versa. Therefore, in the scope of this quality issue, we stipulate that a valid RDFS interpretation of  $V$  must also satisfy the criterion that if  $\langle c_1, c_2 \rangle \in I_{EXT}(\text{skos:narrower}^{\mathcal{I}})$  then  $\langle c_2, c_1 \rangle \in I_{EXT}(\text{skos:broader}^{\mathcal{I}})$ . In other words, every `skos:narrower` relation can be written as a `skos:broader` relation with swapped resources. This allows us to define the quality checking function for this issue only for the case of `skos:broader` relations which improves readability.

To formalize the quality checking function we first define

- $SPBT$ , as the set of all subproperties of `skos:broaderTransitive`, i.e.,  $SPBT = \{sp : \langle sp, \text{skos:broaderTransitive}^{\mathcal{I}} \rangle \in I_{EXT}(\text{rdfs:subPropertyOf}^{\mathcal{I}})\}$ ,
- $BR$ , the set of all pairs of concepts related by a subproperty of `skos:broaderTransitive`, e.g., `skos:broader` or `skos:broadMatch`, as follows:  

$$BR = \bigcup_{spbt \in SPBT} I_{EXT}(spbt), \text{ and}$$
- $BT$ , the set of all pairs of concepts related by explicit assertion of `skos:broaderTransitive` and not by one of its subproperties, as follows:  

$$BT = I_{EXT}(\text{skos:broaderTransitive}^{\mathcal{I}}) \setminus BR.$$

The quality checking function  $strc : C \times C \rightarrow \{0, 1\}$  for solely transitively related concepts ( $strc$ ) can then be defined as

$$strc(c_1, c_2) = \begin{cases} 1 & \text{if } \langle c_1, c_2 \rangle \in BT \\ 0 & \text{otherwise} \end{cases}$$

An RDF graph  $G$  that defines a SKOS vocabulary therefore contains *solely transitively related concepts* if for all possible interpretations  $I$  that are a model of  $G$ ,  $strc(rr^{\mathcal{I}}) = 1$  for at least one pair  $rr$  of resource nodes in  $G$ .

<sup>10</sup>SKOS reference semantic relations preamble: <http://www.w3.org/TR/skos-reference/#L2810>. Retrieved 2015-06-23.

### 3.1.2.6 Unidirectionally Related Concepts

Inclusion of the complete set of reciprocal and symmetric relations can increase performance and recall of queries in systems where no inferencing is or can be used. The ANSI/NISO Z39.19 standard, for example, suggests that “relationship indicators should be employed reciprocally”. On the other side, explicit assertion of inferable facts can be seen as redundant. In practical settings, the use of each strategy can be observed. For consistency reasons or in order to meet specific application requirements, it is therefore important to be informed about what strategy is employed in a Web vocabulary at hand.

Before we formalize the quality checking function for this issue we define

- $SP$ , the set of all *symmetric properties* in  $V$  as  $SP = I_{EXT}(\text{owl:SymmetricProperty}^I)$ ,
- $RP$ , the set of all pairs of *reciprocal properties* in  $V$ , as  $RP = I_{EXT}(\text{owl:inverseOf}^I)$ ,
- $MSR$  as the set of pairs of *concepts with missing symmetric relation*, i.e.,  $MSR = \{\langle c_1, c_2 \rangle : \langle c_1, c_2 \rangle \in I_{EXT}(sp), \langle c_2, c_1 \rangle \notin I_{EXT}(sp), sp \in SP\}$ , and
- $MRR$  as the set of pairs of *concepts with missing reciprocal relation*, i.e.,  $MRR = \{\langle c_1, c_2 \rangle : \langle c_1, c_2 \rangle \in I_{EXT}(\pi_1(rp)), \langle c_2, c_1 \rangle \notin I_{EXT}(\pi_2(rp)), rp \in RP\}$ .

The quality checking function  $urc : C \times C \rightarrow \{0, 1\}$  for unidirectionally related concepts (urc) can then be defined as

$$urc(c_1, c_2) = \begin{cases} 1 & \text{if } \langle c_1, c_2 \rangle \in MSR \cup MRR \\ 0 & \text{otherwise} \end{cases}$$

An RDF graph  $G$  that defines a SKOS vocabulary therefore contains *unidirectionally related concepts* if for all possible interpretations  $I$  that are a model of  $G$ ,  $urc(rr^I) = 1$  for at least one pair  $rr$  of resource nodes in  $G$ .

### 3.1.2.7 Omitted Top Concepts

The SKOS data schema provides the class `skos:ConceptScheme`, which is intended as a means for grouping concepts, e.g. if multiple sub-thesauri need to be defined<sup>11</sup>. In order to provide entry points to such a group of concepts, one or more concepts can be marked as *top concepts*. This helps to provide “efficient access” [IS09] and simplifies orientation in the vocabulary.

<sup>11</sup>SKOS reference concept scheme description: <http://www.w3.org/TR/skos-reference/#L2430>. Retrieved 2015-06-23.

Let  $tcs$  be a function  $tcs : CS \rightarrow 2^C$  that maps a concept scheme to its asserted top concepts, i.e.,  $tcs(cs) = \{tc : \langle tc, cs \rangle \in I_{EXT}(\text{skos:topConceptOf}^{\mathcal{I}}) \vee \langle cs, tc \rangle \in I_{EXT}(\text{skos:hasTopConcept}^{\mathcal{I}})\}$ . We can then define the quality checking function for finding omitted top concepts ( $otc$ ) as  $otc : CS \rightarrow \{0, 1\}$  with

$$otc(cs) = \begin{cases} 1 & \text{if } tcs(cs) = \emptyset \\ 0 & \text{otherwise} \end{cases}$$

An RDF graph  $G$  that defines a SKOS vocabulary therefore contains *omitted top concepts* if for all possible interpretations  $I$  that are a model of  $G$ ,  $otc(r^{\mathcal{I}}) = 1$  for at least one resource node  $r$  in  $G$ .

### 3.1.2.8 Top Concepts Having Broader Concepts

Allemang et al. [AH11] propose to “not indicate any concepts internal to the tree as top concepts”, which we interpret in the way that top concepts should not have broader concepts.

The hierarchical relations of top concepts that are of interest for this issue slightly differ from our definition of  $HR$  provided in 3.1.2.3. This is required because for this issue we take into account only assertions of `skos:broader` and not of the mapping relation `skos:broadMatch`. Mappings are not part of a vocabulary’s “intrinsic” definition and a top concept in one vocabulary that has a broader concept in another vocabulary may be perfectly valid. Therefore we define  $HR' \subseteq HR$  as  $HR' = HR \setminus \{I_{EXT}(\text{skos:broadMatch}^{\mathcal{I}}) \cup I_{EXT}(\text{skos:narrowMatch}^{\mathcal{I}})\}$ . Furthermore, we let the set  $TC$  contain all top concepts in  $V$ , i.e.,  $TC = \bigcup_{cs \in CS} tcs(cs)$ .

We can then define the quality checking function for top concepts having broader concepts ( $tchbc$ ) as  $tchbc : C \rightarrow \{0, 1\}$  with

$$tchbc(c) = \begin{cases} 1 & \text{if } c \in TC \wedge \langle c, c' \rangle \in HR', c' \in C \\ 0 & \text{otherwise} \end{cases}$$

An RDF graph  $G$  that defines a SKOS vocabulary therefore contains *top concepts having broader concepts* if for all possible interpretations  $I$  that are a model of  $G$ ,  $tchbc(r^{\mathcal{I}}) = 1$  for at least one resource node  $r$  in  $G$ .

### 3.1.2.9 Hierarchical Redundancy

It is often intended by vocabulary developers that hierarchical relations are interpreted as being transitive, i.e., if a concept B is a broader concept of concept A and concept C is a broader concept of B, then it can be inferred concept C is also a broader concept of A. The SKOS reference document explicitly states that this entailment is not allowed using `skos:broader` and `skos:narrower` relations. Furthermore, it makes clear that the hierarchical relations `skos:broader` and `skos:narrower` should relate only concepts that are immediate neighbors in the hierarchy<sup>12</sup>. Hierarchical relations may be used to express various semantic connections, such as e.g., part-of, subclass-of, or instance-of. The concrete meaning of `skos:broader` and `skos:narrower` therefore depends on the intended usage scenario of the vocabulary.

We define the notion of *hierarchical redundancy* as two concepts being related by a path of `skos:broader` or `skos:narrower` relations but are also directly related by these properties. Hierarchical redundancy may indicate

- that hierarchical relations in the vocabulary should not be interpreted as being transitive,
- an illogical relation (modeling error) concerning the intended semantics of hierarchical relations, or
- superfluous information.

We define the quality checking function  $hr : C \times C \rightarrow \{0, 1\}$  for detecting hierarchical redundancy (hr) as

$$hr(c_1, c_2) = \begin{cases} 1 & \text{if } \langle c_1, c_2 \rangle \in HR \text{ and } \langle x_1, \dots, x_n \rangle \in HPATH, n > 2, c_1 = x_1, c_2 = x_n \\ 0 & \text{otherwise} \end{cases}$$

An RDF graph  $G$  that defines a SKOS vocabulary therefore has *hierarchical redundancy* if for all possible interpretations  $I$  that are a model of  $G$ ,  $hr(rr^I) = 1$  for at least one pair  $rr$  of resource nodes in  $G$ .

### 3.1.2.10 Reflexively Related Concepts

As already mentioned above, hierarchical relations can be interpreted in various ways. This is also outlined by Svenonius [Sve] who states that the “strictest interpretation” of

<sup>12</sup>SKOS reference transitivity treatment: <http://www.w3.org/TR/skos-reference/#L2413>. Retrieved 2015-06-23.

the hierarchical relation is the inclusion relation which “has the mathematical properties of reflexivity, transitivity, and antisymmetry”. We already covered transitivity and antisymmetry for hierarchical relations in the Sections 3.1.2.9 and 3.1.2.3. This issue covers reflexivity: if, e.g., the `skos:broader` relation is interpreted as the mathematical subset ( $\subseteq$ ) relation, using it in a reflexive way (i.e., asserting a concept to be a broader concept of itself) is feasible. On the other hand, if this relation is interpreted as the proper subset ( $\subset$ ) relation, it might constitute a quality issue.

For the related term (RT) relationship, Svenonius states that “the only mathematical property it always possesses is that of symmetry”. She furthermore argues that the RT relationship “serves to stimulate the verbal imagination of the user of a controlled vocabulary thereby leading him to terms more appropriate to his search topic than those originally coming to mind”. We therefore regard a concept which is connected to itself (i.e., reflexively) by a `skos:related` relation a quality issue.

In the SKOS schema hierarchical, associative (`skos:related`), and mapping relations (some of which also carry hierarchical and associative semantics) are subclasses of `skos:semanticRelation`. For this quality issue we hence identify concepts that are related to themselves by a `skos:semanticRelation` as a potential quality impairment.

The quality checking function for reflexively related concepts (`rrc`)  $rrc : C \rightarrow \{0, 1\}$  can be defined as

$$rrc(c) = \begin{cases} 1 & \text{if } \langle c, c \rangle \in I_{EXT}(\text{skos:semanticRelation}^I) \\ 0 & \text{otherwise} \end{cases}$$

An RDF graph  $G$  that defines a SKOS vocabulary therefore contains *reflexively related concepts* if for all possible interpretations  $I$  that are a model of  $G$ ,  $rrc(r^I) = 1$  for at least one resource node  $r$  in  $G$ .

### 3.1.2.11 Mapping Relations Misuse

According to the SKOS reference documentation, mapping relations like, e.g., `skos:exactMatch`, `skos:broadMatch` or `skos:narrowMatch` “are used to state mapping (alignment) links between SKOS concepts in different concept schemes”<sup>13</sup>. As a consequence, it can be considered a potential quality problem if concepts that are members of the same `skos:ConceptScheme` are linked by mapping relations.

<sup>13</sup>SKOS reference mapping properties preamble: <http://www.w3.org/TR/skos-reference/#L4307>. Retrieved 2015-06-23.

To define the quality checking function for finding mapping relations misuse (mrm), we let  $MR$  be the set of all *mapped authoritative concepts*, i.e.,  $MR = \{\langle ac_1, ac_2 \rangle : \langle ac_1, ac_2 \rangle \in I_{EXT}(\text{skos:mappingRelation}^I), ac_1, ac_2 \in AC\}$ . Based on this definition we define the quality checking function for mapping relations misuse (mrm) as  $mrm : AC \times AC \rightarrow \{0, 1\}$  with

$$mrm(ac_1, ac_2) = \begin{cases} 1 & \text{if } \exists \langle ac_1, ac_2 \rangle \in MR \vee \exists \langle ac_2, ac_1 \rangle \in MR \text{ and} \\ & ccs(ac_1) \cap ccs(ac_2) \neq \emptyset \\ 0 & \text{otherwise} \end{cases}$$

An RDF graph  $G$  that defines a SKOS vocabulary therefore contains *mapping relations misuse* if for all possible interpretations  $I$  that are a model of  $G$ ,  $mrm(rr^I) = 1$  for at least one pair  $rr$  of resource nodes in  $G$ .

### 3.1.3 Linked Data Specific Issues

This section covers quality issues related to the interconnection of the “local” vocabulary  $V$  with other vocabularies  $V'$  published on the Web, i.e., stored on a different host and therefore using their own namespace  $ns'$ . Analogous to the *simple interpretation* definition of  $V$ , interpretations of  $V'$  consist of the sets of resources  $IR'$ , properties  $IP'$  and an interpretation function  $I'_{EXT} : IP' \rightarrow 2^{IR' \times IR'}$ .

#### 3.1.3.1 Missing Incoming Links

When vocabularies are published on the Web, `skos:Concepts` that are identified by an HTTP URI are linkable (i.e., dereferencable) resources. We say that the vocabulary  $V$  has an incoming link if a resource in a vocabulary  $V'$  references a resource in  $V$ . The number of incoming links can indicate the prominence and trustworthiness of a vocabulary. Furthermore, vocabulary developers need to take additional care if modifying a concept that has numerous incoming links. Changing its meaning or even deleting it may change the behavior of third-party applications that use information provided by the concept.

We define *incoming links* to a concept  $ac \in AC$  as the set of resources in  $V'$  that reference  $ac$  using a property  $p' \in IP'$ . The function  $il : AC \rightarrow 2^{IR'}$  maps an authoritative concept to the resources in  $V'$  that are referencing it, i.e.,  $il(ac) = \{ir' : \langle ir', ac \rangle \in I'_{EXT}(p')\}$ . The quality checking function  $mil$  for missing incoming links (mil) can then be defined

as  $mil : AC \rightarrow \{0, 1\}$  with

$$mil(ac) = \begin{cases} 1 & \text{if } il(ac) = \emptyset \\ 0 & \text{otherwise} \end{cases}$$

An RDF graph  $G$  that defines a SKOS vocabulary therefore contains *missing incoming links* if for all possible interpretations  $I$  that are a model of  $G$ ,  $mil(r^I) = 1$  for at least one resource node  $r$  in  $G$ .

### 3.1.3.2 Missing Outgoing Links

`skos:Concepts` should also be linked with other related concepts on the Web, “enabling seamless connections between data sets” [HB11]. We say that the vocabulary  $V$  has an outgoing link if a resource in  $V$  references a resource in vocabulary  $V'$ . Outgoing links are essential for, e.g., queries over multiple datasets. This way, applications can collect data from different resources and combine it in previously unintended ways. This issue identifies the set of all authoritative concepts that do not have links to other resources on the Web.

We define *outgoing links* of a concept  $ac \in AC$  as the set of resources in  $V'$  that are referenced by this concept. The function  $ol : AC \rightarrow 2^{IR'}$  maps an authoritative concept to the resources it references, i.e.,  $ol(ac) = \{ir' : \langle ac, ir' \rangle \in I_{EXT}(p) \vee \langle ir', ac \rangle \in I_{EXT}(p), p \in IP\}$ . Analogous to the definition of  $mil$  we can then define the quality checking function for missing outgoing links ( $mol$ ) as  $mol : AC \rightarrow \{0, 1\}$  with

$$mol(ac) = \begin{cases} 1 & \text{if } ol(ac) = \emptyset \\ 0 & \text{otherwise} \end{cases}$$

An RDF graph  $G$  that defines a SKOS vocabulary therefore contains *missing outgoing links* if for all possible interpretations  $I$  that are a model of  $G$ ,  $mol(r^I) = 1$  for at least one resource node  $r$  in  $G$ .

### 3.1.3.3 Broken Links

Just as in the “traditional” Web of documents, broken links hinder navigability also in the Web of Data and should therefore be avoided. Popitsch et al. [PH10] recognize the problem of “structurally broken links” (i.e., a link whose “target resource had representations that are not retrievable anymore”) still being evident on the Web of Data and introduce various solution strategies. Among these strategies they mention the “Detect

and Correct” method which we adopt for this quality issue: links are reported to be broken and, if so, a correction attempt is expected to be made by human vocabulary curators.

We define broken links as RDF resources that return HTTP error responses or no response at all when being dereferenced. Our detection strategy is therefore to consider a link as broken if the HTTP response code after resolving the resource’s URL is other than 200 after following possible redirections. As with all other quality issues discussed here, we do not provide a correction strategy but instead provide the list of broken links to vocabulary developers for choosing an adequate resolution action.

We define

- $IR_{URI} \subseteq IR$  as the set of all resources in  $V$  that are identified by URIs<sup>14</sup>,
- $IR_{HTTP} \subseteq IR_{URI}$  as the set of all resources identified by an URI resolvable using the HTTP protocol, i.e., the scheme name of the URI is equal to “http” or “https”, and
- $deref : IR_{HTTP} \rightarrow \mathbb{N}$ , a function mapping a resource  $hir \in IR_{HTTP}$  to the HTTP status code<sup>15</sup> after dereferencing its URI and following possible redirects.

The quality checking function for broken links ( $bl$ ) can then be defined as  $bl : IR_{HTTP} \rightarrow \{0, 1\}$  with

$$bl(ir) = \begin{cases} 1 & \text{if } deref(ir) \neq 200 \\ 0 & \text{otherwise} \end{cases}$$

An RDF graph  $G$  that defines a SKOS vocabulary therefore contains *broken links* if for all possible interpretations  $I$  that are a model of  $G$ ,  $bl(r^I) = 1$  for at least one URI resource node  $r$  in  $G$ .

### 3.1.3.4 Undefined SKOS Resources

The SKOS schema is defined within its own namespace, “<http://www.w3.org/2004/02/skos/core#>”. However, some vocabularies use resources from within this namespace, which are unresolvable for two main reasons:

<sup>14</sup>As originally specified in RFC 1738: <http://www.rfc-editor.org/rfc/rfc1738.txt>. Retrieved 2015-06-23.

<sup>15</sup>As specified in RFC 2616: <http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>. Retrieved 2015-06-23.



1. Vocabulary creators mint new terms within the SKOS namespace instead of introducing them in a separate namespace. Reasons for this can be simple typographical mistakes, but also the vocabulary author's intention to introduce a resource to express a new semantic relationship that is not (yet) covered by SKOS.
2. Use of deprecated SKOS elements due to, e.g., lack of maintenance of the Web vocabulary or compatibility reasons with older versions of SKOS.

To define the quality checking function for this issue, we make use of two sets, namely

- $ILR \subseteq IR_{HTTP}$ , the set of *illegal resources*, i.e., resources in  $IR_{HTTP}$  as that have the same namespace as the SKOS schema but are not defined in the SKOS schema itself, and
- $DR$ , the set of *deprecated SKOS resources* as defined in the skos reference documentation, i.e.,  $DR = \{\text{skos:symbol}^{\mathcal{I}}, \text{skos:prefSymbol}^{\mathcal{I}}, \text{skos:altSymbol}^{\mathcal{I}}, \text{skos:CollectableProperty}^{\mathcal{I}}, \text{skos:subjectIndicator}^{\mathcal{I}}, \text{skos:isSubjectOf}^{\mathcal{I}}, \text{skos:isPrimarySubjectOf}^{\mathcal{I}}, \text{skos:primarySubject}^{\mathcal{I}}, \text{skos:subject}^{\mathcal{I}}\}$ .

We can then define the quality checking function for undefined SKOS resources ( $usr$ )  $usr : IR_{HTTP} \rightarrow \{0, 1\}$  as

$$usr(ir) = \begin{cases} 1 & \text{if } ir \in DR \cup ILR \\ 0 & \text{otherwise} \end{cases}$$

An RDF graph  $G$  that defines a SKOS vocabulary therefore contains *undefined SKOS resources* if for all possible interpretations  $I$  that are a model of  $G$ ,  $usr(r^{\mathcal{I}}) = 1$  for at least one resource node  $r$  in  $G$ .

### 3.1.3.5 HTTP URI Scheme Violation

The second principle of Tim Berners-Lee's article on Linked Data<sup>16</sup> encourages the use of (dereferencable) HTTP URIs as names for things described in the dataset. This way datasets can be interlinked which makes it possible to, e.g., execute queries that involve datasets which are distributed across multiple servers. According to our Definition 2.1, a vocabulary without HTTP URIs cannot be considered a Web vocabulary.

Let  $ULR$  be the set of *unlinkable resources*, which are all resources in  $V$  that are identified by URIs but not dereferencable using the HTTP(S) protocol, i.e.,  $ULR = IR_{URI} \setminus$

<sup>16</sup>Four rules for Linked Data publication: <http://www.w3.org/DesignIssues/LinkedData.html>. Retrieved 2015-06-23.

$IR_{HTTP}$ . We can then define the quality checking function for HTTP URI scheme violations (husv)  $husv : IR \rightarrow \{0, 1\}$  as

$$husv(ir) = \begin{cases} 1 & \text{if } ir \in ULR \\ 0 & \text{otherwise} \end{cases}$$

An RDF graph  $G$  that defines a SKOS vocabulary therefore contains *HTTP URI scheme violations* if for all possible interpretations  $I$  that are a model of  $G$ ,  $husv(r^I) = 1$  for at least one resource node  $r$  in  $G$ .

### 3.1.4 SKOS Consistency Issues

The SKOS RDF schema defines six “semantic conditions” that are not expressed formally. As a consequence, there is some room for interpretation of these conditions, so we provide here possible formal definitions of four of them, S13, S14, S27, and S46.

We already stated in Section 3.1.1 that we assume S12 to hold for all quality checking functions in this catalog, so we do not provide a quality checking function for this condition. Furthermore, we do not provide a quality checking function for condition S36 because it targets membership properties of `skos:Collections` which are not in the scope of this catalog.

#### 3.1.4.1 Relation Clashes

The SKOS integrity condition S27 states that the associative relationship “`skos:related`” is disjoint with the property `skos:broaderTransitive`. Two concepts that are in the same hierarchical transitive closure (as inferred by `skos:broaderTransitive` or `skos:narrowerTransitive` relations) must not be associatively related by the `skos:related` property.

We define the quality checking function for relation clashes (rc) as  $rc : C \times C \rightarrow \{0, 1\}$  with

$$rc(c_1, c_2) = \begin{cases} 1 & \text{if } \langle c_1, \dots, c_n \rangle \in HPATH \wedge \exists \langle c_i, c_j \rangle \in I_{EXT}(\text{skos:related}^I) \text{ with} \\ & 1 \leq i, j \leq n, i \neq j \\ 0 & \text{otherwise} \end{cases}$$

An RDF graph  $G$  that defines a SKOS vocabulary therefore contains *relation clashes* if for all possible interpretations  $I$  that are a model of  $G$ ,  $rc(rr^I) = 1$  for at least one pair  $rr$  of resource nodes in  $G$ .

### 3.1.4.2 Mapping Clashes

The SKOS integrity condition **S46** states that the mapping relationship “`skos:exactMatch`” is disjoint with each of the properties `skos:broadMatch` and `skos:relatedMatch`”.

In order to formalize the quality checking function  $mc$  for this issue we first define

- $EEM$ , a set of all pairs of concepts related by an *exact equivalent mapping* as follows:  $EEM = \{\langle c_1, c_2 \rangle : \langle c_1, c_2 \rangle \in I_{EXT}(\text{skos:exactMatch}^I) \vee \langle c_2, c_1 \rangle \in I_{EXT}(\text{skos:exactMatch}^I)\}$ ,
- $EEMPATH$  as the set of all *exact equivalent mapping paths*, i.e.,  $EEMPATH = \{\langle c_1, \dots, c_n \rangle : \langle c_i, c_{i+1} \rangle \in EEM, 1 \leq i < n, c_i \in C, n \leq |C|\}$ ,
- $P_{HAM}$  as the set of hierarchical and associative mapping properties, i.e.,  $P_{HAM} = \{\text{skos:broadMatch}^I, \text{skos:narrowMatch}^I, \text{skos:relatedMatch}^I\}$ , and
- $HAM$ , a set of all pairs of *hierarchically or associatively mapped concepts*, i.e.,  $HAM = \{\langle c_1, c_2 \rangle : \langle c_1, c_2 \rangle \in I_{EXT}(p) \vee \langle c_2, c_1 \rangle \in I_{EXT}(p), p \in P_{HAM}\}$ .

The quality checking function for mapping clashes ( $mc$ ) can then be defined as  $mc : C \times C \rightarrow \{0, 1\}$  with

$$mc(c_1, c_2) = \begin{cases} 1 & \text{if } \langle c_1, c_2 \rangle \in HAM \wedge \langle x_1, \dots, x_n \rangle \in EEMPATH, \text{ with} \\ & n \geq 2, c_1 = x_1, c_2 = x_n \\ 0 & \text{otherwise} \end{cases}$$

An RDF graph  $G$  that defines a SKOS vocabulary therefore contains *mapping clashes* if for all possible interpretations  $I$  that are a model of  $G$ ,  $mc(rr^I) = 1$  for at least one pair  $rr$  of resource nodes in  $G$ .

### 3.1.4.3 Inconsistent Preferred Labels

The integrity condition **S14** in the SKOS reference documentation states that “A resource has no more than one value of `skos:prefLabel` per language tag, *and no more than one value of `skos:prefLabel` without [a] language tag*”. The latter part of this definition is only present in the comments of `skos:prefLabel` in the SKOS RDF Schema.

Let  $pl$  be a function  $pl : AC \rightarrow 2^{LV}$  that maps an authoritative concept to the set of all its assigned preferred labels, i.e.,  $ac \mapsto \{lv : \langle ac, lv \rangle \in I_{EXT}(\text{skos:prefLabel}^I)\}$ .

We can define the quality checking function for inconsistent preferred labels (ipl) as  $ipl : AC \rightarrow \{0, 1\}$  with

$$ipl(ac) = \begin{cases} 1 & \text{if } \exists pl_1 \in pl(ac) \wedge \exists pl_2 \in pl(ac') : lang(pl_1) = lang(pl_2), ac' \in AC, ac \neq ac' \\ 0 & \text{otherwise} \end{cases}$$

An RDF graph  $G$  that defines a SKOS vocabulary therefore contains *inconsistent preferred labels* if for all possible interpretations  $I$  that are a model of  $G$ ,  $ipl(r^I) = 1$  for at least one resource node  $r$  in  $G$ .

#### 3.1.4.4 Disjoint Labels Violation

Integrity condition S13 is defined as “`skos:prefLabel`, `skos:altLabel` and `skos:hiddenLabel` are pairwise disjoint properties.”. In other words this means that no `skos:Concept` is allowed to have identical literals assigned using each two of the aforementioned labeling properties.

Similar to the definition of  $pl$  we define the functions  $al, hl : AC \rightarrow 2^{LV}$  that map an authoritative concept to its assigned alternative and hidden labels:

- $al : ac \mapsto \{lv : \langle ac, lv \rangle \in I_{EXT}(\text{skos:altLabel}^I)\},$
- $hl : ac \mapsto \{lv : \langle ac, lv \rangle \in I_{EXT}(\text{skos:hiddenLabel}^I)\}$

We furthermore define the function  $ndl : AC \rightarrow LV \times LV$  that maps an authoritative concept to its *non-disjoint labels*, i.e.,  $ac \mapsto \{lv : \langle lv, lv \rangle \in pl(ac) \times al(ac) \cup al(ac) \times hl(ac) \cup pl(ac) \times hl(ac)\}$ . We can then define the quality checking function for disjoint labels violations (dlv) as  $dlv : AC \rightarrow \{0, 1\}$  with

$$dlv(ac) = \begin{cases} 1 & \text{if } ndl(ac) \neq \emptyset \\ 0 & \text{otherwise} \end{cases}$$

An RDF graph  $G$  that defines a SKOS vocabulary therefore contains *disjoint labels violations* if for all possible interpretations  $I$  that are a model of  $G$ ,  $dlv(r^I) = 1$  for at least one resource node  $r$  in  $G$ .

## 3.2 Summary

In this chapter we brought the notion of “Web vocabulary quality” to a formal level by defining in total 29 quality issues. They are based on existing literature in the field

of controlled vocabulary development, Linked Data publishing (cf. Section 2) and the SKOS reference documentation, but also reflect our findings from expert discussion and practical experience in thesaurus development.

We divided the identified quality issues into four categories that constitute key properties of controlled Web vocabularies:

- *Labeling and Documentation Issues* target textual properties that support human users in exploring and understanding the vocabulary. They are also essential for presenting the vocabulary in a meaningful way and are required to drive applications that, e.g., use the vocabulary for searching and retrieving information from some document corpus.
- *Structural Issues* cover relations between vocabulary elements that characterize different kinds of controlled vocabularies. They determine the suitability of a vocabulary for specific usage scenarios to a large extent and provide valuable context information.
- *Linked Data Specific Issues* focus on interconnections (mappings, alignments) between Web vocabularies and other general resources on the Web. They allow for browsing knowledge that is stored as RDF datasets on remote systems and to combine it in previously unintended ways, leading to discovery of new knowledge.
- *SKOS Consistency Issues* target comments and informal consistency conditions described in the SKOS reference documentation.

Developing the catalog is important for a number of reasons. It functions as a tool that helps in demystifying the abstract notion of quality by formalizing criteria that are automatically assessable. Therefore it can supply the basis for applications that help human users in, e.g.,

- discovering vocabularies on the Web that suit their purposes,
- finding areas of improvement,
- receiving guidance in the vocabulary maintenance process,
- ensuring proper vocabulary presentation and publication.

However, development and maintenance of controlled vocabularies remains an intellectual process. Despite the support by automated tools, the final judgment whether to incorporate changes based on a detected quality issue should be up to the vocabulary

developer(s). We therefore envision assistive applications based on our catalog that integrate in the controlled vocabulary development process, comparable to, continuous integration practices in software development or even grammar and spell-checkers in word processors.

## Chapter 4

# Quality Checking Techniques

In this chapter we present our contributions to Research Question 2 (*How can the quality of a Web vocabulary be automatically assessed?*). We introduce two tools, *qSKOS* and *rsine*, that follow different approaches for checking quality issues. The first approach is designed to evaluate the quality of a Web vocabulary as a whole, similar to compilers or data validators that check for syntax errors in provided files. While this approach is suitable to give the vocabulary developer the possibility to decide when to check the vocabulary, our second approach uses a different strategy. It is designed to be integrated into the data persistence layer and can observe changes to Web vocabularies as soon as they are performed by the user. We call this “on-change checking” because each time the vocabulary changes, evaluation against potential quality issues is performed and vocabulary contributors get an immediate feedback on what consequences the change might have.

We provide details of the used technologies, architectures, and implemented algorithms of the two approaches. The *qSKOS* tool has already been covered to some extent in our earlier work [MHI12, SM13], however we cover it in more detail in this chapter. We developed *rsine* as a general-purpose notification utility for linked datasets in the course of the EU-funded LOD2<sup>1</sup> project. To allow its application for quality evaluation of Web vocabularies, we implemented extensions to use *rsine* with a subset of the quality issues defined in Chapter 3. We published our work regarding *rsine* architecture, implementation, integration, and usage in the LOD2 Project Deliverables 5.3.1, 5.3.2, and 7.3 and in [MMS14].

---

<sup>1</sup>LOD2 project website: <http://lod2.eu/Welcome.html>. Retrieved 2015-06-23.

## 4.1 General Design Considerations

Based on the quality issues we introduced in Chapter 3 we implemented two tools that evaluate them and report the findings in a human-readable form. Both tools are designed to be integrated into (collaborative) Web vocabulary development processes but follow different strategies for invoking computation of the quality issue occurrences:

- *On-demand Vocabulary Quality Checking*: The Web vocabulary is evaluated against each quality issue as a whole and the results are collected. The term “on-demand” refers to the fact that this strategy is usually employed on-demand, i.e., when the vocabulary is considered to be in some kind of consistent state, e.g., when a revision cycle is completed or the vocabulary is published online. During evaluation of the quality issue, the vocabulary is expected to remain as it is, i.e., it is not changed until the quality report is finished.
- *On-change Vocabulary Quality Checking*: Each time the Web vocabulary is changed, the responsible developer is notified immediately in case the change she introduced causes a potential quality issue. This strategy requires a pre-selection of the quality issues that must be evaluated. It is not required to evaluate the whole vocabulary against each quality issue because the type of change and the involved properties and resources restrict the set of potential quality issues to be introduced. For example, if a label is changed, quality issues that target the vocabulary’s structure do need to be computed.

*qSKOS* is our implementation of a tool following the “on-demand” approach. It takes a SKOS vocabulary file as input and produces a report on the identified quality issues together with detailed information on the affected vocabulary elements. The underlying strategy for most issues (except for *Missing In-links* and *Broken Links*) in the analysis process is to treat a vocabulary as a self-contained entity, resembling the closed world assumption, as is generally done when validating RDF data. An in-depth description is provided in Section 4.2.

*rsine* contributes to the on-change checking approach. We implemented the tool as general-purpose framework that detects custom specified change patterns in RDF datasets and disseminates notifications to a defined set of recipients. We give a detailed coverage of the framework in Section 4.3 and describe its application with some of the introduced quality issues. It is motivated by the following three use cases, commonly encountered by contributors in (collaborative) controlled Web vocabulary development environments:



- *Changes in Meaning*: A contributor changes the textual description (e.g., the property `skos:scopeNote`) of a concept in a way that it has an altered meaning. A thesaurus manager would be interested in all concepts where these properties have recently been updated.
- *Changes in Structure*: Contributors would be interested in changes to outgoing and incoming hierarchical relations to concepts they have created. These kinds of relations provide contextual information and scope to the concept, thus potential misunderstandings in the usage or meaning of the concept can be clarified by notifying the responsible contributors.
- *Remote Changes*: On the Web of Data, establishing outgoing links to other “third-party” vocabularies is crucial in order to leverage the full potential of queries across datasets of different origin. Concepts that are linked by other resources on the Web are more sensitive to updates because, e.g., changing their URIs would introduce broken links and degrade the information content of the referencing vocabulary. Thus, contributors want to get notified whenever someone on the Web links to any resource of the locally developed vocabulary.

## 4.2 qSKOS - On-demand Vocabulary Quality Checking

*qSKOS* is a library and command line application that implements evaluation algorithms for a SKOS vocabulary against occurrences of the quality issues introduced in Chapter 3. Its purpose is to analyze a given Web vocabulary in a serialized RDF format and provide a human-readable report of all occurring quality issues. The report contains both a summary and detailed overview and aims to provide vocabulary developers with the necessary information for tracking down the issues and, if required, correct them manually.

Development of the tool started on 17 April 2011 with the first code committed to the public source code repository<sup>2</sup>. We later reimplemented it using the Java programming language and made it available for download at a different repository<sup>3</sup> which superseded the original implementation. The tool is designed to work both as a library for integration into existing (Java) software projects, but can also be downloaded as a self-contained application that is controlled by command-line parameters.

<sup>2</sup>GitHub repository for the early *qSKOS* implementation in Ruby: <https://github.com/cmader/qSKOS4rb>. Retrieved 2015-06-23.

<sup>3</sup>*qSKOS* current development repository: <https://github.com/cmader/qSKOS>. Retrieved 2015-06-23.

### 4.2.1 Implementation

From the early stages of development we shared our intentions and results with the Linked Data and thesaurus development community. We announced the beginning of development of *qSKOS* and an initial version of the catalog of quality issues (see Section 3.1) as well as milestone releases at online mailing lists<sup>4</sup>.

The *qSKOS* application is available under an open-source license (GPLv3). Also, the tools required for building and operating it are publicly available for download at no additional cost. We use Java 1.7 as programming language and Maven 3 as build environment. Internally we rely on the libraries OpenRDF Sesame<sup>5</sup> for storing and querying the Web vocabulary, JGraphT<sup>6</sup> for executing graph algorithms, Apache HttpComponents<sup>7</sup> for looking up HTTP resources, and JCommander<sup>8</sup> for parsing command line parameters.

The Web vocabularies that can be passed to *qSKOS* as input can be represented by each of the RDF serialization formats RDF/XML, Turtle, Notation3 (N3), N-Triples, TriX, or TriG. It can either generate a summary of some vocabulary statistics properties (such as number of concepts, semantic relations, concept schemes, or collections) or analyze the vocabulary for occurrences of quality issues defined in our catalog. It is also possible to check only for a subset of the issues from this catalog by naming the checks that should be performed or excluded from analysis. The exact usage and a synopsis of the supported parameters can be found in the documentation<sup>9</sup> or when invoking the tool without any parameters.

### 4.2.2 Quality Issue Evaluation

In order to analyze a given Web vocabulary, *qSKOS* first performs the following steps before evaluation of quality issue occurrences can be performed:

1. Create and initialize an OpenRDF Sesame in-memory repository that supports forward-chaining RDFS inferencing.

---

<sup>4</sup>Announcement of initial version of quality issue catalog: <http://lists.w3.org/Archives/Public/public-esw-thes/2011Apr/0018.html>, first release announcement of *qSKOS*: <http://lists.w3.org/Archives/Public/public-esw-thes/2012Jun/0004.html>, both retrieved 2015-06-23.

<sup>5</sup>OpenRDF Sesame project: <http://rdf4j.org/>. Retrieved 2015-06-23.

<sup>6</sup>JGraphT graph library: <http://jgrapht.org/>. Retrieved 2015-06-23.

<sup>7</sup>Apache HttpComponents project: <http://hc.apache.org/>. Retrieved 2015-06-23.

<sup>8</sup>JCommander project: <http://jcommander.org/>. Retrieved 2015-06-23.

<sup>9</sup><https://github.com/cmader/qSKOS/blob/master/README.rdoc>. Retrieved 2015-06-23.

2. Fetch the SKOS data schema from the Web<sup>10</sup> in its RDF serialization and add it to the created repository as a named graph.
3. Add the Web vocabulary that is subject of analysis to the same repository.

In our implementation, we restricted ourselves to RDF and RDFS interpretations of SKOS vocabularies mainly for practical reasons: (i) RDFS inferencing is already implemented in the OpenRDF library we used and (ii) the majority of axioms on the Web use features from only RDF and RDFS [GHKP12]. However, in future versions of the tool we will also consider subsets of the OWL language, such as OWL LD, which was proposed by Glimm et al. [GHKP12].

Depending on the quality issue, we use different methodologies to evaluate their occurrences. In the most basic cases it is sufficient to evaluate an appropriate SPARQL query and provide the result in the quality report. However, evaluating more advanced quality issues needs to invoke multiple SPARQL queries and to combine their results programmatically. For some quality issues, the capabilities of SPARQL do not suffice and using additional libraries is necessary, e.g., with graph-theoretic algorithms. In order to create reports that can easily be interpreted by vocabulary developers, fetching additional information like concept labels from the vocabulary repository is needed which increases complexity of the evaluation queries and algorithms.

The current implementations of the quality issue evaluation is not optimal yet in terms of performance and efficiency, but they have been thoroughly tested and proved in practical use with large vocabularies of file sizes up to 650 Megabyte.

For some quality issues, *qSKOS* provides parameters that need to be adjusted to guarantee accurate analysis of the Web vocabularies:

- *SPARQL endpoints used for checking the number of incoming links*: *qSKOS* per default uses hardcoded settings<sup>11</sup> but they can be overwritten to test against custom endpoints.
- *SKOS-XL support*: If enabled, *qSKOS* runs SPARQL construct queries to add SKOS label relations and literals (`skos:prefLabel`, `skos:altLabel`, `skos:hiddenLabel`) if SKOS-XL labels are defined in the analyzed vocabulary.
- *Subset analysis*: Users can define a percentage of authoritative concepts or HTTP URIs that should be checked for missing incoming links or broken links. The results

<sup>10</sup>SKOS data schema used for *qSKOS* quality issue evaluation: <http://www.w3.org/2009/08/skos-reference/skos.rdf>. Retrieved 2015-06-23.

<sup>11</sup>Datahub (<http://semantic.ckan.net/sparql>). Retrieved 2015-06-23.

are then extrapolated to the whole vocabulary. This feature allows to improve analysis performance for large vocabularies at the cost of result accuracy.

- *Authoritative concept identifier*: Users can provide a substring of an URI that identifies authoritative concepts from the set of all concepts. If no such identifier is provided, *qSKOS* checks the host part of the URIs of all involved concepts and assumes that the host name that appears most often identifies authoritative concepts.

In the following sections, we describe the implementation of evaluating each quality issue in *qSKOS* and the resulting data structures. These structures hold the basic necessary information for creating a tabular report suitable for human users and are thus either maps, lists, or boolean values. The issue count computed in the summary section of the report is therefore the number of key-value pairs or total count of list entries.

#### 4.2.2.1 Omitted or Invalid Language Tags

We use a SPARQL query for getting all literals that are assigned to resources by predicates which are subclasses of `rdfs:label` or `skos:note`. We programmatically iterate over the results and check if the literal can be considered problematic, i.e., (i) it does not have a language tag assigned or (ii) the language tag is invalid. We detect invalid language tags by checking their syntax and conformity to ISO 639 by using methods from the standard Java 1.7 `Locale` class. The resulting data structure of this issue is a map assigning each resource URI the set of problematic literals.

#### 4.2.2.2 Incomplete Language Coverage

Evaluation of this issue is performed in a three-step process:

1. We first create a *language coverage map* by iterating over each concept and finding all assigned literals with defined language tags. This information is used to create a lookup table with the concept as key and the list of its assigned literals distinct language tags as values.
2. We determine the *list of used languages* in the vocabulary by collecting all distinct language tags in the language coverage map.
3. We iterate over the language coverage map and find those problematic concepts with their covered languages not identical to the list of all used languages.

The resulting data structure is a map assigning each URI of a problematic concept to their list of not covered languages.

#### 4.2.2.3 No Common Language

For evaluation of this issue we also generate a language coverage map and a list of all used languages like we did in Step one and two of *Incomplete Language Coverage*. By iterating over the language coverage map we keep only those languages in the list of all used languages that are covered by all concepts. The resulting data structure is the list of common languages of all concepts in the vocabulary. If this list is empty, the quality check can be considered to have failed.

#### 4.2.2.4 Undocumented Concepts

We check for each authoritative concept if it has defined a literal for one of the properties `skos:note`, `skos:changeNote`, `skos:definition`, `skos:editorialNote`, `skos:example`, `skos:historyNote`, or `skos:scopeNote`. If not, the concept is added to the result list.

#### 4.2.2.5 Overlapping Labels

We iterate over all concepts and group all assigned `skos:prefLabel`, `skos:altLabel` and `skos:hiddenLabel` literals by a similarity function. This similarity function returns true if a certain similarity threshold is met and false otherwise. It is currently implemented as a case insensitive string comparison: a concept with, e.g., the preferred label “label” is detected as similar to another concept labeled “LABEL”, causing a *label conflict*. The result of this quality issue evaluation is the set of all label conflicts, holding information about the affected concept URIs, label types and literal values.

#### 4.2.2.6 Missing Labels

Evaluation of this quality issue is implemented by issuing a SPARQL ASK query for each authoritative concept and concept scheme. The queries check (i) if concepts have `skos:prefLabels` assigned and (ii) if concept schemes have literals asserted using at least one of the properties `rdfs:label`, `dc:title`, or `dcterms:title`. The result of the evaluation is a list of resources that are failing these checks.

#### 4.2.2.7 Unprintable Characters in Labels

To find unprintable characters we iterate over all authoritative concepts and collect their assigned `skos:prefLabel`, `skos:altLabel` and `skos:hiddenLabel` literals. Using a regular expression, we check if one of these labels contains a Unicode character that belongs to general category C (“Other”<sup>12</sup>) and add them to the result set if they do. This result set is a list of label literals containing unprintable characters together with information about the affected concept URI and label type.

#### 4.2.2.8 Empty Labels

We evaluate this issue using a SPARQL query that finds all triples involving the predicates `rdfs:label`, `dc:title`, `dcterms:title`, `skos:prefLabel`, `skos:altLabel`, and `skos:hiddenLabel`. An empty label is found if these literals have a length of zero after removing whitespaces (using the Java `trim` method). The resulting data structure is a map, assigning to each resource the predicates which are used for defining empty labels.

#### 4.2.2.9 Ambiguous Notation References

Our implementation finds ambiguous notations (i) within one concept as well as (ii) between multiple concepts. We iterate over all authoritative concepts and find all literals assigned to the concept by the `skos:notation` predicate. If more than one of them are found (case (i)), we add the resource together with the notation literal to the result set. In order to find ambiguous notations between different concepts (case (ii)), we check if the notation literal of a concept appears in the triple store as notation of another concept that is either member of the same concept scheme or both concepts are members of no concept scheme. In this case, the resulting data structure is the list of all concepts with identical notation literals.

#### 4.2.2.10 Orphan Concepts

To evaluate this issue we use a SPARQL query for finding all resources with relations to other resources with the constraint that these relations are subproperties of `skos:semanticRelation`. Since this gives us all non-orphan concepts, we calculate the complementary set against all involved concepts. The result of this operation is the set of all orphan concepts and it is returned as the resulting data structure of the evaluation.

---

<sup>12</sup>Documentation for the Unicode Character Database: <http://www.unicode.org/reports/tr44/>. Retrieved 2015-06-23.

#### 4.2.2.11 Disconnected Concept Clusters

The implementation for evaluating this quality issue makes use of the JGraphT library to ensure efficient computation. We build a graph processable by JGraphT by iterating over all concepts and find all relations (i.e., subproperties of `skos:semanticRelation`) to other resources. We add them as nodes and edges to a directed multigraph and use the `ConnectivityInspector` to find all *connected sets*<sup>13</sup> which make up the result set of this evaluation.

#### 4.2.2.12 Cyclic Hierarchical Relations

We use a two-step approach in our implementation to evaluate this quality issue:

1. *Building an hierarchy graph*: We generate a directed graph processable by JGraphT containing all resources (as nodes) being hierarchically related to each other by subproperties of `skos:broaderTransitive` or `skos:narrowerTransitive` (as edges) that point to the broader concept.
2. *Finding cycles*: We use JGraphT's `CycleDetector` class to find the “set of all vertices which participate in at least one cycle in this graph”<sup>14</sup>. For each of the vertices (constituting the vocabulary's concepts) we find the strongly connected components they are part of. They contain the potentially problematic concepts of each cycle.

The data structure that results from this computation is the set of strongly connected components with their members (i.e., vocabulary concepts) being part of at least one hierarchical cycle.

#### 4.2.2.13 Valueless Associative Relations

This issue can be evaluated using a single SPARQL query. The resulting data structure is a set of pairs of resources that are related to each other by `skos:related` and have the same broader concept.

---

<sup>13</sup>A connected set is a set of vertices of a graph that are in the same maximally connected component.

<sup>14</sup>`CycleDetector` documentation: [http://jgrapht.org/javadoc/org/jgrapht/alg/CycleDetector.html#findCycles\(\)](http://jgrapht.org/javadoc/org/jgrapht/alg/CycleDetector.html#findCycles()). Retrieved 2015-06-23.

#### 4.2.2.14 Solely Transitively Related Concepts

For the evaluation of this issue we issue two SPARQL queries returning pairs of concepts that are related by either `skos:broaderTransitive` or `skos:narrowerTransitive` properties but not by `skos:broader` or `skos:narrower`. The result of the evaluation is a set of pairs of affected resources.

#### 4.2.2.15 Unidirectionally Related Concepts

*qSKOS* currently does not support the `owl:inverseOf` property, so we hardcoded this functionality according to the SKOS schema. We issue multiple SPARQL queries against the vocabulary for finding pairs of authoritative concepts that are related to each other but omit the reciprocal relation using the corresponding inverse SKOS property. The resulting data structure is a map holding the pairs of problematic unidirectionally related concepts as well as the affected omitted inverse relation property.

#### 4.2.2.16 Omitted Top Concepts

We iterate over all concept schemes and collect those that do not have another resource associated by the properties `skos:hasTopConcept` or `skos:topConceptOf`, making up the resulting list of the evaluation.

#### 4.2.2.17 Top Concepts Having Broader Concepts

This issue can be evaluated by a single SPARQL query, resulting in a list of affected top concepts.

#### 4.2.2.18 Hierarchical Redundancy

It turned out that computation of this issue using only SPARQL queries does not scale very well with large vocabularies. Therefore, we decided to use the JGraphT hierarchy graph as in Issue *Cyclic Hierarchical Relations*. We iterate over each pair of vertices connected by an edge. We temporarily remove this edge and compute the shortest path between the vertices using JGraphT's `DijkstraShortestPath` implementation. If a path is found this means that another hierarchical path between these two concepts exists, introducing hierarchical redundancy. For the next iterations the previously removed edge is inserted into the graph again. The resulting data structure is the set of pairs of resources that are redundantly hierarchically related.



#### 4.2.2.19 Reflexively Related Concepts

We iterate over all authoritative concepts and use SPARQL ASK queries to determine if the concept is related to itself by a subproperty of `skos:semanticRelation`. The resulting data structure is the set of all triples asserting a reflexive relation.

#### 4.2.2.20 Mapping Relations Misuse

To evaluate this quality issue we iterate over all pairs of concepts related by (a subproperty of) `skos:mappingRelation`. For each pair we check if the two concepts are either members of the same concept scheme or both concepts do not belong to any concept scheme in the vocabulary. If one of these conditions is met, we add the pair to the resulting list.

#### 4.2.2.21 Missing Incoming Links

We estimate the number of incoming links by iterating over all authoritative concepts and query the Sindice<sup>15</sup> and datahub<sup>16</sup> remote SPARQL endpoints to find out if these concepts are referenced by at least one other vocabulary on the Web. For each authoritative concept we check the remote SPARQL endpoints if a resource (identified by an HTTP URI) exists that references the concept using any property. Empty query results are indicators for missing incoming links. The resulting data structure of the computation is the set of resources that are not referenced by any triple in the remote datasets.

#### 4.2.2.22 Missing Outgoing Links

Unlike the evaluation of *Missing Incoming Links*, utilization of dataset registries is not necessary for this quality issue because outgoing links can be identified locally by comparing URI namespaces. For each authoritative concept in the analyzed Web vocabulary we find resources that reference (or are being referenced by) the concept. If all of these resources are authoritative resources, the concept has no relations to other vocabularies on the Web and thus misses outgoing links. Consequently, the resulting data structure of this evaluation is the set of such authoritative concepts.

---

<sup>15</sup>Sindice (<http://sindice.com/>, retrieved 2015-06-23) indexes the Web of Data, which is composed of pages with semantic markup in RDF, RDFa, Microformats, or Microdata. Currently it covers approximately 230M documents with over 11 billion triples.

<sup>16</sup>Datahub (<http://datahub.io/>, retrieved 2015-06-23) is a community-run catalog of currently 5045 datasets, many of them following the Linked Data guidelines.

#### 4.2.2.23 Broken Links

Evaluation of broken links is performed in two steps:

1. *Collecting all HTTP URIs*: We create a list of all HTTP(S) URIs specified in the vocabulary by iterating over all triple subject, predicate and object components. Fragment parts in the URIs are pruned if found.
2. *Dereferencing URIs*: We iterate over the collected URIs and look them up using the Apache HttpComponents HTTP client. We follow possible redirects and allow the response to arrive within a one minute timeout. If the HTTP status code of the response is 200, the link is considered dereferencable, otherwise we add it to the resulting data structure of this evaluation: the list of undereferencable (i.e., broken) URIs.

#### 4.2.2.24 Undefined SKOS Resources

In order to find all used deprecated skos properties we issue a single SPARQL query covering a hardcoded list of deprecated SKOS properties<sup>17</sup>. For finding non-existent SKOS resources, we use a SPARQL query to identify all URIs in the subject, predicate, or object position of the vocabulary triples that are defined in the SKOS namespace but are not members of the official SKOS data schema. The resulting data structure is the list of found problematic URIs.

#### 4.2.2.25 HTTP URI Scheme Violation

We iterate over the subject components of all triples of the analyzed vocabulary and find those that do not use the HTTP or HTTPS URI scheme name. The list of found resources are the result of this evaluation.

#### 4.2.2.26 Relation Clashes

From practical experience with large Web vocabularies we found that evaluation of this quality issue can be very expensive to compute when implemented only with SPARQL queries. We therefore use a combined approach:

---

<sup>17</sup>List of outdated SKOS elements: <http://www.w3.org/TR/skos-reference/#namespace>. Retrieved 2015-06-23.

1. Issue a SPARQL query to find all pairs of concepts that are related by `skos:related` or `skos:relatedMatch`.
2. For each two concepts from the first step we use the hierarchy graph (see Section 4.2.2.12) and check if a connecting path between the concepts exists (using JGraphT's `DijkstraShortestPath` class). If this is the case, a relation clash has been detected.

The resulting data structure of this issue is the set of all pairs involved in a relation clash.

#### 4.2.2.27 Mapping Clashes

We implemented evaluation of this issue using a single SPARQL query that finds pairs of concepts that are connected by (chains of) `skos:exactMatch` relations as well as `skos:broadMatch`, `skos:narrowMatch`, or `skos:relatedMatch`. All pairs of concepts meeting this condition are added to the evaluation's result list.

#### 4.2.2.28 Inconsistent Preferred Labels

We iterate over all triples relating a resource to a literal by the `skos:prefLabel` property and create a map that assigns each resource the set of its assigned preferred labels. We can then find all pairs of preferred labels of a resource that have identical language tags or no language tag assigned at all. The resulting data structure of the evaluation is a list of all such conflicts for each concept.

#### 4.2.2.29 Disjoint Labels Violation

For evaluating this quality issue, we iterate over all triples that relate a resource to a literal value by any SKOS label property. We create a map that assigns each literal the list of concepts that use the same literal for one of their labels. By iterating over the list of concepts for each literal, we can detect a disjoint label violation if we find concepts that are in this list twice with different label types. The result of this evaluation is a map with the conflicting label as key and the affected resources as values.

### 4.2.3 PoolParty Product Integration

We integrated an adapted version of the *qSKOS* library into the *PoolParty Thesaurus Server* (PPTS). The evaluation algorithms had to be adapted because PPTS stores the

Web vocabularies in repositories without forward-chaining RDFS inferencing available. As PPTS is a commercial product, a strong emphasis lies on providing an easy-to-use user interface, which we had to implement for each quality issue evaluation result separately. As a starting set we focused on seven quality issues that we observed most frequently in our case studies and therefore would be most beneficial for the customers. Table 4.1 shows the equivalence of the quality issue names in this thesis and their names in the user interface of PPTS (Figure 4.1). The quality issue “No Broaders and no Top Concept” is not implemented in *qSKOS* because it focuses on a constraint in the way PPTS organizes concepts: any concept managed by PPTS must have a broader concept assigned, except if it is directly related to a concept scheme using `skos:topConceptOf` or `skos:hasTopConcept`.

Quality Issue Name	PPTS Quality Check Name
Cyclic Hierarchical Relations	Hierarchical Cycles
Disjoint Labels Violation	Non-Disjoint Labels
Overlapping Labels	Same Label for different concepts
-	No Broaders and no Top Concept
Inconsistent Preferred Labels	Inconsistent Preferred Labels
Omitted or Invalid Language Tags	Omitted or Invalid Language Tags
Relation Clashes	Relation Clashes

TABLE 4.1: Equivalence in quality issue naming between the PPTS user interface and the quality issue catalog.

In PPTS, evaluation of the quality checks can be triggered manually for the whole project. Once finished, the findings are shown in an overview area so vocabulary developers can see what issues occur and to what extent (Figure 4.1). Each time the project changes (e.g., new concepts are created, relations are changed, or are labels reformulated) the developers are required to regenerate this quality report. Additional screenshots of the PPTS quality reports user interface are shown in Appendix A.

Details for each quality issue can be viewed when clicking at the quality issue heading. Depending on the quality issue they are displayed in a different way. For labeling and documentation issues, results are provided in textual form, presenting the URIs of the affected concepts and their potentially problematic properties, as depicted in Figure A.6 with an exemplary Web vocabulary.

For issues that return a subgraph of the vocabulary like *Relation Clashes*, the affected concepts are displayed in a graphical form (Figure A.7). Each occurrence of the quality issue is presented as a list entry which is composed of the preferred labels of the identified problematic concepts. When this “headline” is clicked, a graphical representation is automatically rendered that displays these concepts and the affected relations. Concepts are displayed in ellipses using the preferred label in the project’s primary language.

## Reegle Thesaurus + Corpus

1DBC8732-A250-0001-3594-1A9212B91CE1

Metadata & Statistics	Concepts	Triples	SPARQL	Autopopulate	Visualization	Quality Report
<b>Last Generated:</b> 27.10.2014 - 17:16						
Hierarchical Cycles (0)						
Non-Disjoint Labels (0)						
<input checked="" type="checkbox"/> <b>Inconsistent Preferred Labels (4)</b>						
No Broaders and no Top Concept (0)						
Omitted or Invalid Language Tags (0)						
<input checked="" type="checkbox"/> <b>Same Label for different concepts (105)</b>						
<input checked="" type="checkbox"/> <b>Relation Clashes (280)</b>						

FIGURE 4.1: Overview of the seven integrated quality issues (three of them are detected in this case).

Clicking the concept URIs in textual reports or the concept labels in the graphical reports exposes an editor where all properties (e.g., labels and relations to other resources) can be changed by the vocabulary developer to resolve the quality issue.

### 4.2.4 Online SKOS Vocabulary Quality Checker

We provide a Web frontend of the *qSKOS* tool that allows everyone to upload vocabularies and receive the generated quality report as downloadable text file or by email. Since we focused on simplicity, we do not provide the configuration options described in Section 4.2.2 and omit checking for *Broken Links* and *Missing Incoming Links* which can take very long to evaluate.

The first version of the *online SKOS Quality Checker* went online in December 2013 and was publicly announced on Twitter<sup>18</sup> and mailing lists<sup>19</sup>. Since then it can be reached at <http://qskos.poolparty.biz/>. Screenshots of the user interface can be found in Appendix A.

Users of the *online SKOS Quality Checker* are required to log in with their existing Google, LinkedIn, XING, or Twitter accounts (Figure A.1). After having logged in, they enter their personal overview area, listing all previously uploaded vocabularies together

<sup>18</sup>Initial tweet on 14 Jan 2014, announcing the *online SKOS Quality Checker*: [https://twitter.com/PoolParty\\_Team/status/423115019748794368](https://twitter.com/PoolParty_Team/status/423115019748794368). Retrieved 2015-06-23.

<sup>19</sup>Announcement of the *online SKOS Quality Checker* on public mailing lists: <http://lists.w3.org/Archives/Public/public-esw-thes/2014Jan/0011.html>. Retrieved 2015-06-23.

with their quality analysis reports (Figure A.2). Requiring to log in is also necessary to prevent the service from being abused and to keep track of users in order to contact them for collection of feedback regarding the service.

When uploading their Web vocabularies, users are required to additionally provide a name for them. This name serves as some kind of “project name”, intended to hold different versions of the same Web vocabulary. As upload format we support RDF in the serialization formats N3, N-Triples, RDF/XML, TriG, TriX, or Turtle and a maximum file size of 100 Megabyte. Users can optionally provide their email address to which the report is sent when the quality analysis is complete.

As soon as the vocabulary upload is complete, the *qSKOS* tool is invoked to perform the analysis (Figure A.3). During creation of the quality issues report, the currently processed issue and the percentage finished (for some quality issues) is displayed. The analysis can be canceled any time or closed, which takes the user back to the upload interface.

When the creation of the quality report is finished, a summary of the found issue occurrences is displayed (Figure A.4). The detailed report is made available on the upload screen in the overview area of checked vocabularies but can also be downloaded immediately by clicking the provided download button.

### 4.3 rsine - On-change Vocabulary Quality Checking

The LOD2 project was a four-year project within the EU’s FP7 Information and Communication Technologies Work Programme. Its main goal was to provide an integrated stack of tools that support, e.g., Linked Data creation (extraction and linking), storage, revision, enrichment, and exploration<sup>20</sup>.

In the course of Work Package five, Task 5.3 of the LOD2 project we developed a stack component that enables users to subscribe for notifications on changes of any RDF dataset, based on predefined filter criteria. Our solution, *rsine* (Resource Subscription and Notification sErvice), is a publisher/subscriber notification framework that can be installed independently from other stack components. It is available online<sup>21</sup> under an open-source license, thus it can also be used outside of the LOD2 context.

The framework notifies subscribers whenever assertions to resources in which they are interested in are created, updated, or removed. It is based on the W3C standards RDF

<sup>20</sup>LOD2 tool stack: <http://stack.lod2.eu/blog/>. Retrieved 2015-06-23.

<sup>21</sup>GitHub repository of the *rsine* source code: <https://github.com/rsine/rsine>. Retrieved 2015-06-23.

and SPARQL and is designed to be used alongside with existing triple storage solutions that support these technologies. Subscriptions to assertions involving resources of interest can be expressed on the triple level as SPARQL queries, which allows to accurately define the scope of notifications. The modular approach is capable of utilizing various types of notification channels like email, Twitter, or creation of log files for documentation of dataset changes. Furthermore, instances of the proposed notification framework can forward resource assertions to each other. This establishes the possibility to subscribe for assertions that are created or modified on any dataset on the Web, provided that it is configured to use the proposed *rsine* framework.

There are several user groups or positions that can benefit from such a notification framework. In the following we list some of them. One person can here not only occupy one but several roles, e.g.:

- The owner and/or creator of content who wants to be informed in case of any change to prevent misunderstandings or abuse of the data,
- A consumer of a specific subset of the content, e.g., a project manager or visualization designer who is interested to be informed about the latest changes as they could have an impact on their product,
- A quality manager who needs to check changes with regard to any violation of predefined data validation criteria or quality rules,
- Automated scripts that generate metadata for the monitored dataset, based on certain metrics.

This list is not complete, it is however possible that more than these beneficiaries might come up gradually – especially as additional methods of notification channels are supported in future which broadens the range of potential application scenarios for the framework. In this section, we cover two main application areas:

The first one is based on requirements of the current Wolters Kluwer Germany (WKD) controlled vocabulary development and publication process. We introduce the concrete usage scenarios defined by WKD and report on the general applicability of our approach. In this context, the first two roles in the list above will profit most from the availability of instant notifications.

The second application area of a Linked Data notification framework is integrated quality management in the controlled Web vocabulary development process. As we have shown in our previous work [[MHI12](#), [SM13](#)], potential quality problems of controlled

Web vocabularies can be detected from patterns (i.e., quality issues) in the underlying RDF graph of the vocabulary. We believe that immediate notification of the responsible contributors on the introduction of such quality issues will lead to faster development and higher quality of the created vocabularies.

### 4.3.1 Requirements and Design Considerations

Taking into account the number and diversity of the LOD2 stack components (15 at the time *rsine* development started) with their various responsibilities in the Linked Data lifecycle (e.g., storage, authoring, quality analysis), we aimed for two main goals when designing *rsine*:

1. *Easy Integration into Stack Components*: Design of a custom notification API in, e.g., Java, that needs to be integrated and supported by the component owners would be too costly in both time and effort. Instead, we decided that *rsine* observes the underlying triple changes directly when they are incorporated into the datasets written by the stack components. This way any kind of change can be detected and disseminated to the subscriber as a notification. Implementation as a stand-alone service that can be used via REST-like HTTP requests ensures that subscriptions for notifications can be registered without changing the stack component's source code. As a consequence, this approach requires detailed knowledge about the data changes performed by the stack component on the triple level. Otherwise a significant amount of reverse-engineering is needed to formulate subscription queries.
2. *Flexibility*: By using SPARQL as a well-established and powerful method for querying RDF graphs, we achieve a maximum level of flexibility to express the kind of changes a subscriber is interested in. *rsine* also stores metadata of the triple changes such as timestamp or type (addition or removal) in a separate RDF graph, allowing for even more sophisticated queries like stating a specific timely order in which changes have been introduced. This flexibility in querying is accompanied by a powerful template-based approach for formulating human-readable notification messages: projected values from the notification queries can be directly used in the notification messages sent out to the subscribers.

### 4.3.2 Approach

Figure 4.2 describes the architecture of the notification framework (depicted with a green frame) that was designed and implemented as proof-of-concept in the course of the project



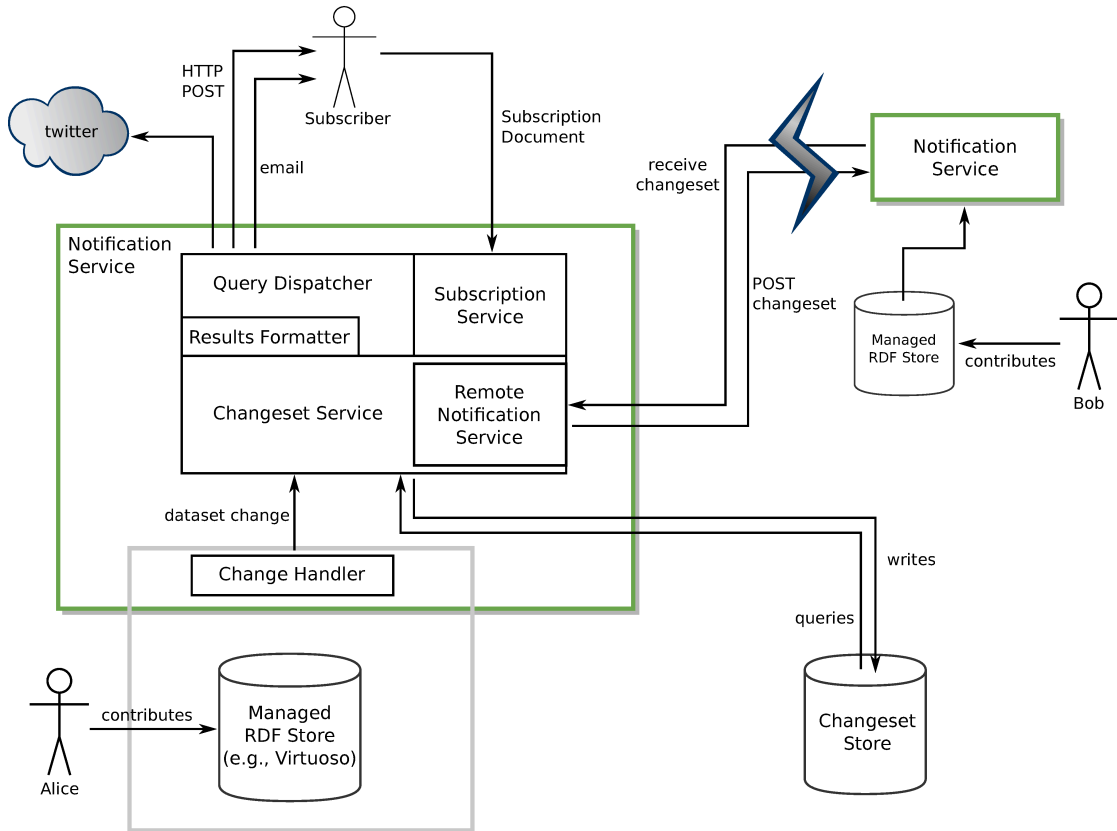


FIGURE 4.2: Conceptual overview on rsine architecture.

deliverables. The *Notification Service* on the left side of the figure is intended to give an overview on how the components interact internally. The *Notification Service* on the right side is an identical second instance of the framework installed on a remote location on the Web to illustrate notification forwarding (Section 4.3.3.2). Both instances are identical but internal details of the remote instance have been omitted to improve conciseness.

In order to fulfill the use cases described above, the framework continuously observes changes to the data held by a *Managed RDF Store* on the triple level, i.e., every time a triple is added, updated, or removed the framework is triggered by the *Change Handler*. Detection of changes in the managed RDF store works independently from the various methods data may be manipulated in the managed RDF store (e.g., SPARQL update queries via HTTP or connector libraries). Observed triple changes are converted to *changesets* expressed in RDF using a standard ontology. These changes are then persisted to an internal *Changeset Store*.

The gray box around the *Change Handler* and the *Managed RDF Store* should emphasize the fact that the *Change Handler* needs to be implemented on top of the used RDF store. In our proof-of-concept implementation we provide a *Change Handler* that can be used for Virtuoso Servers. However, in environments that rely on different RDF storage backends

such as Sesame, a custom Change Handler that fits to the internals of the used storage solution must be implemented. Due to this storage-dependent components, two parts of the frameworks need to be deployed separately: the Change Handler matching the utilized RDF storage technology and the generic part, consisting of the Query Dispatcher, Results Formatter, Registration Service, Changeset Service, and Remote Notification Service.

### 4.3.3 Subscribing for Notifications

A subscriber who is interested in receiving notifications can register at the framework by providing an RDF *subscription document* that consists of SPARQL queries, the preferred notification channel (e.g., logfile, email, Twitter) and, optionally, a template for the textual message that is disseminated to the subscribers. The SPARQL queries select resources the subscriber is interested in and access both the data contained in the Managed RDF Store as well as in the Changeset Store. The results of the query are then disseminated through the desired channels. Before dissemination, the framework formats the query results into human-readable form if a template is provided by the subscriber on registration.

In the following sections we describe the internal workflow of the proposed framework. Thereby we distinguish between “local notifications” and “notification forwarding” use cases. The former are processed only within one instance of the framework, as typically installed on a local development environment. The latter denote notifications that are disseminated to the contributors as a result of changes being communicated between two distinct installations of the framework, possibly residing on different systems on the Web.

#### 4.3.3.1 Local Notifications

This workflow involves only one instance of the notification framework. It targets the scenario that an RDF dataset is developed and managed by one *rsine* instance with subscribers registering at this instance to receive notifications for changes introduced to this dataset.

1. Every time a triple is added, updated, or removed in the Managed RDF Store, the Change Handler calls the Changeset Service.
2. The Changeset Service creates an RDF representation of the change and persists it to the local Changeset Store. This Changeset Store can be implemented as a separate RDF store or a named graph located in the Managed RDF Store.

3. The Changeset Service triggers the Query Dispatcher which iterates over every registered user and executes their SPARQL queries as provided at registration time. If stated in the subscription document, a Results Formatter creates a human-readable document from the query results. A common usage for the Results Formatter is, e.g., to translate validation query results to appropriate error messages and format the causes of the error in a way that they can be easily understood.
4. Based on the notification channels the subscriber specified in the subscription document, the Query Dispatcher is also responsible to send the (formatted) query results to the subscriber.

#### 4.3.3.2 Notification Forwarding

Here we address the scenario that vocabulary developers are typically interested in the number and kind of “incoming links”, i.e., relations introduced to any dataset on the Web that reference resources of their locally developed dataset. We illustrate this workflow based on an example with these prerequisites:

- Alice develops a controlled vocabulary on her local system (using the namespace `alice`), that is managed by the proposed notification framework.
- Bob does the same on his local system (using the namespace `bob`).
- Both systems publish 5\* Open Data<sup>22</sup>.

Now suppose, for example, that Alice references a concept stored in Bob’s vocabulary by adding the triple `alice:alicesConcept skos:exactMatch bob:thirdPartyConcept` to her managed RDF store. The following steps are executed within the *rsine* framework:

1. The Changeset Service on Alice’s system detects that `bob:thirdPartyConcept` denotes a resource on a remote system by having a namespace other than `alice`.
2. In addition to storing an RDF representation of the change to the Changeset Store, it is also passed to the Remote Notification Service (RNS) of Alice’s *rsine* installation.
3. The RNS marks the changeset as originating from Alice’s system by adding her namespace as a `dcterms:source` property. Afterwards, it sends it to Bob’s RNS by an HTTP POST.

---

<sup>22</sup>The five-star deployment scheme for Open Data: <http://5stardata.info/>. Retrieved 2015-06-23.

4. Bob's RNS passes the changeset to the local Changeset Service which writes it to his Changeset Store. As a result it is possible to register for changes that have been introduced in remote systems in the same way as for local changes.

Step three requires a strategy for Alice's system to know where Bob's RNS is located, in order to notify him about the remote change. One way to accomplish this is to pack this information into a custom HTTP response header. When Alice creates a link to a concept residing on Bob's system, she looks up `bob:thirdPartyConcept` and parses the response header for a special property, e.g., `X-Rsine-Location` which points to the URI of Bob's RNS. This requires Bob to configure his Webserver to add the required header whenever resources of his vocabulary are dereferenced. This method is inspired by common "linkback" methods like Pingback<sup>23</sup> or Webmention<sup>24</sup> that are implemented in, e.g., blogging systems to help authors in keeping track about who is linking their articles.

#### 4.3.4 Implementation

We implemented a proof-of-concept of the *rsine* framework and published it into two repositories hosted at GitHub<sup>25</sup>. One repository<sup>26</sup> contains the generic part of the framework that works independently from the utilized RDF storage backend. It is implemented as a standard Java 1.7 application that can be built with Maven 3 and uses the Spring Framework<sup>27</sup> for dependency injection and exposing a REST-like HTTP interface. Rsine uses OpenRDF Sesame to manage an internal changeset store and for querying the managed store SPARQL endpoint. The Velocity template engine<sup>28</sup> is used to provide meaningful notification messages that can access the projected values of the SPARQL notification queries.

In order to enable notifications to work with Virtuoso<sup>29</sup> as the Managed RDF Store, we provide an extension package for Virtuoso at a separate repository<sup>30</sup>. It is implemented as a Virtuoso-specific `.vad` package that can be installed using the graphical Virtuoso

<sup>23</sup>Pingback specification: <http://www.hixie.ch/specs/pingback/>. Retrieved 2015-06-23.

<sup>24</sup>GitHub site of the Webmention project: <http://indiewebcamp.com/Webmention>. Retrieved 2015-06-23.

<sup>25</sup>*rsine* project development resources: <https://github.com/rsine>. Retrieved 2015-06-23.

<sup>26</sup>Storage-independent part of the *rsine* framework: <https://github.com/rsine/rsine>. Retrieved 2015-06-23.

<sup>27</sup>Spring application development framework: <http://projects.spring.io/spring-framework/>. Retrieved 2015-06-23.

<sup>28</sup>Apache Velocity Engine: <http://velocity.apache.org/engine/index.html>. Retrieved 2015-06-23.

<sup>29</sup>Virtuoso Universal Server: <http://virtuoso.openlinksw.com/>. Retrieved 2015-06-23.

<sup>30</sup>Extension to connect *rsine* to Virtuoso as storage backend: <https://github.com/rsine/rsineVad>. Retrieved 2015-06-23.

Conductor management interface. On installation an SQL “script” is executed that creates a database trigger which in turn calls a stored procedure that forwards the affected triples to a running *rsine* instance using HTTP GET. As input parameters the extension needs the location and port where the *rsine* instance runs (`localhost:8080` per default) and the URI(s) of the named graph(s) whose triple additions and removals are passed to *rsine*.

In addition to the two repositories mentioned above that form our main implementation effort, *rsine* has also received attention from developers of the GeoKnow project<sup>31</sup>. The project focuses on improving creation, reuse, and exploitation of geospatial data and implemented an improved Change Handler for *rsine* that replaces the Virtuoso extension package described above. Instead of detecting triple changes in Virtuoso using database triggers it parses Virtuoso’s transaction log which avoids negative impact on the performance of RDF data processing.

Listing 4.1 shows an exemplary *rsine* subscription document that notifies the subscriber by email with a proper message whenever a `dcterms:creator` is added to a resource changeset ontology. It is intended for use with *PoolParty Thesaurus Server* (PPTS) which, according to its business logic, sets the `dcterms:creator` of a concept if it is newly created. We can therefore notify the user that a new concept has been created when a value is assigned to a resource using this property.

---

```
rsine:query [
  spin:text "SELECT ?concept ?creator WHERE {
    ?cs a cs:ChangeSet .
    ?cs cs:createdDate ?csdate .
    ?cs cs:addition ?addition .

    ?addition rdf:subject ?concept .
    ?addition rdf:predicate dcterms:creator .
    ?addition rdf:object ?creator .

    FILTER (?csdate >
      'QUERY_LAST_ISSUED'^^<http://www.w3.org/2001/XMLSchema#dateTime>)
  }";

  rsine:formatter [
    a rsine:vtlFormatter;
    rsine:message "A new concept with URI '$bindingSet.getValue('concept')' has
      been created by '$bindingSet.getValue('creator')'";
  ];
];
```

---

LISTING 4.1: Example *rsine* subscription document.

---

<sup>31</sup>GeoKnow project website: <http://geoknow.eu/Project.html>. Retrieved 2015-06-23.

#### 4.3.4.1 Integration

In order to showcase the capabilities of *rsine*, we integrated it with two exemplary components of the LOD2 stack: PPTS, a tool for domain experts to develop controlled Vocabularies and publish them as Linked Data using standardized schemas (e.g., SKOS) and *Pebbles*, a Web application that provides a graphical user interface to manage RDF metadata for XML documents. Both applications are operated by WKD in a production environment.

PPTS builds on the OpenRDF Sesame infrastructure for persisting RDF data. All triple additions and removals are therefore performed using a `RepositoryConnection` object. In order to provide interoperability between PPTS and *rsine*, we need a means to forward these triple changes to *rsine*. Therefore we implemented a subclass of `RepositoryConnectionListenerAdapter` that acts as a proxy between the OpenRDF repository and PPTS. It intercepts the triple changes and, before handing them down to the OpenRDF repository for persistence, announces them to *rsine*.

Pebbles uses Virtuoso as storage backend. The task of integrating *rsine* with Pebbles was thus limited to deploy the *rsine* .vad extension package to the Virtuoso instance. As a result of this step, all triple changes Pebbles performs to its underlying dataset are communicated to *rsine*, establishing the basis for integration with the service.

We configured and tested the integration of *rsine* with these LOD2 components using only local notifications as described in Section 4.3.3.1. Notification Forwarding is currently in an experimental state and has not been tested or used within the context of the LOD2 project.

#### 4.3.5 Management-related Notifications

For deliverable 5.3.2 and 7.3 of the LOD2 project, WKD contributed a list of usage scenarios for notifications that are required to be supported by *rsine*. The scenarios mainly focus on supporting general management-related tasks of the Web vocabulary development process, such as monitoring

1. changes (i.e., creations/deletion/linking/editing) of concepts,
2. editing activities of a specific hierarchy branch in the thesaurus,
3. broken links to external resources,
4. changes in the structure of the thesaurus by merging and decomposing concepts,  
and

5. reuse of thesaurus concepts within other datasets on the Web.

We were able to cover all but one (item number three) of these requirements. However, in this thesis we do not go into more detail on the implementation of these issues (see Deliverable 5.3.2 for further information) because we focus on exploiting instant notifications for detecting quality issues in this thesis.

### 4.3.6 Quality Notifications

Based on the catalog introduced in Chapter 3, we identified nine quality issues that are capable for being checked on each triple change. These are:

- Issues focusing on hierarchical relations: *Cyclic Hierarchical Relations*, *Hierarchical Redundancy*, *Top Concepts Having Broader Concepts*.
- Issues focusing on associative relations: *Valueless Associative Relations*, *Relation Clashes*.
- Issues focusing on mapping relations: *Mapping Relations Misuse*, *Mapping Clashes*.
- Issues focusing on SKOS label properties: *Overlapping Labels*, *Disjoint Labels Violation*.

For each of these quality issues we provide an *rsine* subscription document that contains the logic for issue evaluation, formatting, and dissemination of the created notifications. They can be reviewed by browsing *rsine*'s source code repository<sup>32</sup>.

#### 4.3.6.1 Subscribing for a Notification

In this section, we describe how *rsine* can be used for detection of the exemplary quality issue *Mapping Relations Misuse*, which we formally introduced in Section 3.1.2.11. As already stated, the necessary information required by *rsine* for detecting the quality issue, assembling a notification message and dissemination of this message is encapsulated in subscription documents. Therefore, the Listings 4.2-4.5 show different parts of the content of a possible subscription document for detecting occurrences of *Mapping Relations Misuse*, alongside with an explanation of the functionality of the respective code. For brevity reasons we omit namespace declarations in the listings; the abbreviations can be found in Table 4.2.

<sup>32</sup>Source directory containing the subscription documents for all implemented quality issues: <https://github.com/rsine/rsine/tree/master/src/test/resources/quality>. Retrieved 2015-06-23.

Namespace URI	Abbreviation
<a href="http://www.w3.org/1999/02/22-rdf-syntax-ns#">http://www.w3.org/1999/02/22-rdf-syntax-ns#</a>	rdf
<a href="http://purl.org/vocab/changeset/schema#">http://purl.org/vocab/changeset/schema#</a>	cs
<a href="http://spinrdf.org/sp/">http://spinrdf.org/sp/</a>	spin
<a href="http://www.w3.org/2004/02/skos/core#">http://www.w3.org/2004/02/skos/core#</a>	skos

TABLE 4.2: Namespaces and their abbreviations used in the subscription document examples.

Listing 4.2 shows how the set of all received triple changes (changesets) in the Changeset Store is constrained only to those which involve additions of the SKOS mapping relations. Filtering for *csdate* and the placeholder string `QUERY_LAST_ISSUED` is currently required for *rsine*'s internal processing and must be included in every subscription document.

---

```
spin:text
"SELECT ?concept ?newMappingType ?mappedConcept WHERE {
  ?cs a cs:ChangeSet .
  ?cs cs:createdDate ?csdate .
  ?cs cs:addition ?addition .

  ?addition rdf:subject ?concept .
  ?addition rdf:predicate ?newMappingType .
  ?addition rdf:object ?mappedConcept .

  FILTER (?newMappingType IN (skos:exactMatch, skos:broadMatch, skos:narrowMatch,
    skos:relatedMatch, skos:closeMatch) && ?csdate >
    'QUERY_LAST_ISSUED'^^<http://www.w3.org/2001/XMLSchema#dateTime>)
}";
```

---

LISTING 4.2: Filtering changesets for mapping property additions.

The condition that determines if a mapping relation is misused is shown in Listing 4.3. If the SPARQL ASK query evaluates to the same value as stated in `rsine:expect`, it is considered fulfilled and the quality issue is detected. Conditions defined as `rsine:condition` can use the value bindings for the variables evaluated in the changeset filter (see Listing 4.2 above).

---

```
rsine:condition [
  spin:text
  "ASK {
    ?concept skos:broader*/skos:topConceptOf ?cs .
    ?mappedConcept skos:broader*/skos:topConceptOf ?cs .
    ?cs a skos:ConceptScheme .
  }";
  rsine:expect true;
];
```

---

LISTING 4.3: Condition query for detecting the structural pattern.



Listing 4.4 shows the formatter template that defines the wording of the notification message. Just as the condition definition which we exemplified above, formatters can use the variable bindings from the changeset filter and access them using `$bindingSet.getValue()` calls.

---

```
rsine:formatter [
  a rsine:vtlFormatter;
  rsine:message "The concepts '$bindingSet.getValue('concept')' and
    '$bindingSet.getValue('mappedConcept')' are in the same concept scheme and
    should not be associated by a mapping relation";
];
```

---

LISTING 4.4: Template for formatting a notification message.

Listing 4.5 illustrates the definition of notifiers which name the methods and targets that specify how and where the notification messages should be disseminated to. In this example, two notifiers are configured. One is of type `rsine:loggingNotifier` which internally uses the class `LoggingNotifier` that passes all notification messages to *rsine*'s logging system. The second notifier is implemented by the class `EmailNotifier` which encapsulates the notification messages into an email and sends them to the provided recipient using functionality from the `javax.mail.internet` package.

---

```
rsine:notifier [
  a rsine:loggingNotifier;
];

rsine:notifier [
  a rsine:emailNotifier;
  foaf:mbox <mailto:c.mader@semantic-web.at>
].
```

---

LISTING 4.5: Defining notification dissemination.

## 4.4 Summary

In this chapter we introduced two approaches that are suitable for computing occurrences of the quality issues from our catalog provided in Chapter 3. Each approach was implemented by a different tool and published online under open-source licenses.

We can assume that the tools are perceived well by the Linked Data community. For example, we know that the Leibniz-Informationszentrum Wirtschaft<sup>33</sup> use *qSKOS* to check the Standard Thesaurus Wirtschaft (STW), which they author, for quality issues.

---

<sup>33</sup>Leibniz-Informationszentrum Wirtschaft: <http://zbw.eu/de/>. Retrieved 2015-06-23.

The integrated quality checking methods of the *PoolParty Thesaurus Server*, which are based on *qSKOS*, are received well by customers who can purchase this functionality as an additional software feature. Finally, the high number of users of the *online SKOS Quality Checker* shows the value of *qSKOS* to the community.

Also *rsine* received very positive feedback from the LOD2 project partners. The fact that it is used and even extended in the context of another EU-funded project indicates its benefit for the Linked Data community. When used for quality evaluation of Web vocabularies, *rsine* was considered “an appropriate instrument to support human developers in a thesaurus development process”<sup>34</sup>.

---

<sup>34</sup>See LOD2 Project Deliverable 7.3.

## Chapter 5

# Case Studies and Findings

In this chapter, we provide a detailed overview on our findings regarding the relevance of the quality issues we introduced in Chapter 3. Our contributions are as follows:

- We present the results of a survey for receiving expert feedback among developers and users of controlled (Web) vocabularies. From the survey we infer recommendations of best practices and how to improve existing and future versions of Web vocabulary quality assessment tools. We already published this contribution as part of our earlier work [MH13].
- We present our findings on the occurrence of the quality issues in vocabularies that are currently published online. We already performed Web vocabulary analysis in our earlier work [MH11, MHI12, SM13]. Here we present our findings based on the set of Web vocabularies we identified in [SM13] but cover additional quality issues implemented in the latest version of *qSKOS*.
- We report on our results of investigating how the quality issues and their automatic assessment during the development process are perceived by vocabulary developers. We set up a case study among students assigned with the task of creating a Web vocabulary. This case study serves us to get insights into (i) the practicability of the integration and (ii) if and how Web vocabularies can be improved by our approach. This contribution has also been published as part of our earlier work [MW14].
- We furthermore analyze the data gathered from the *online SKOS Quality Checker* to find out about (i) what quality issues occur in the uploaded vocabularies and (ii) if and what quality issues are addressed between different versions of the vocabularies.

## 5.1 Expert Perception of Quality Issues

We performed a survey among developers and users of controlled (Web) vocabularies to learn how our catalog of potential quality issues is perceived. In particular, we were interested in the usefulness of the quality issues and under what circumstances they typically need to be addressed and where they are less relevant. Furthermore, we sought to find out about new quality issues and how to improve those we already identified. We also evaluated opinions of our participants regarding the impact of quality issues on different vocabulary usage scenarios. The survey and results have been published in our earlier work [MH13].

### 5.1.1 Survey Structure and Question Design

We designed a questionnaire which consisted of four parts in which we (i) presented introductory information, (ii) collected general domain and usage information, (iii) presented open and closed-ended questions targeting vocabulary quality, and (iv) collected information about the participants. The analytic and explorative nature of the survey is reflected in the third part: To find out about the usefulness of existing quality issues, closed-ended questions that can be analyzed automatically were used. For exploring additional quality issues or improving existing ones we included open-ended questions that allowed us to, e.g., infer rationales for rating decisions or information on the development processes.

We employed two different kinds of closed-ended questions: First, we used multiple choice checkboxes (including an “other” option), e.g., for selecting the domain or usage scenarios of Web vocabularies. Second, we formulated explicit *quality statements* (e.g., “Concepts should not be hierarchically related to themselves.”) based on the issues identified in Chapter 3 and asked participants to express their level of agreement on a 5-point Likert scale, including a neutral option. For each quality statement, it was possible to give no answer. To learn about the participant’s decision rationale, every closed-ended question was complemented by a free-text field for providing the decision’s rationale. We used a similar symmetric 5-point Likert scale to find out about the relevance of the quality issues in relation to a vocabulary usage scenario, as shown in Table 5.1. Participants were asked to select one of the categories *very important*, *important*, *neither*, *less important*, or *not important*. In case they were not able to provide an answer, we added the option *no answer/don’t know*.

We organized the quality statements in three groups (*Labeling and Documentation Issues*, *Structural Issues* and *Linked Data Specific Issues*) and follow this structure in the discussion of our findings in the Sections 5.1.4, 5.1.5, 5.1.6, and 5.1.9.

Name	Description
Manual/Intellectual Indexing	Performed by domain experts who process a corpus of documents and extract relevant concepts
Automatic Indexing	Algorithmic extraction of common words in a text corpus based on statistical measures (frequency, co-occurrence,...)
Tagging	The vocabulary is used by end users to do subject indexing of a collection of items (text corpus, images,...)
Classification/Categorization	The vocabulary defines categories that can be assigned to items of a collection (text corpus, images,...)
Faceted Search	Facets describe content from multiple perspectives, by forming a mutually exclusive classification based on the indexed items
Multilingual Search	The vocabulary contains textual descriptions of concepts in multiple languages
Document Suggestion (Recommendation)	Based on the search query, similar documents are included in the search result
Spelling Suggestions and Corrections/Autocompletion	User input (at search and indexing time) is matched with the vocabulary terms and corrections are suggested
Term Suggestions	Based on the structural organization of the vocabulary and the user input, additional terms are suggested
Query expansion and refinement	Based on a controlled vocabulary structure, a user query is broadened or narrowed to adjust search recall
Navigation	Visual guidance for exploring information resources (e.g., websites, collections,...)
Search results grouping/ranking	Vocabulary-supported optimization of the visual representation of search results
Linking (Data Integration)	The controlled vocabulary is created as an intermediate step to provide compatibility with another data source
Publication	The controlled vocabulary is made available “as is” online for reuse by others to view or download

TABLE 5.1: Vocabulary usage scenarios.

### 5.1.2 Survey Response Analysis Methodology

We believe that due to our chosen survey distribution channels, we can trust our participants’ expertise. We intentionally did not require the participants to have a background in SKOS so we could reach a wider target audience. Since a meaningful quantitative analysis and statistical interpretation would require a much larger, but hard to collect sample, we concentrated on a qualitative analysis based on the received responses.

To identify the usage scenarios that have been rated as most important (average median value less than three) for the provided quality statements, we computed and sorted the agreement ratings for each statement by ascending median, mode, mean and standard

deviation<sup>1</sup>. Based on these results, we identified the three most important issues for each usage scenario. Furthermore, we computed the arithmetic mean of usage scenario importance over all quality statements and sorted them accordingly.

To find out the level of agreement for each quality statement, we calculated the relative number for each possible choice on the Likert scale (from *strongly agree* to *strongly disagree* and *no answer*) based on the total number of participants who answered the respective question. The rationales provided by the participants were analyzed qualitatively. We compared them to the agreement ratings, and collected those that overlap or contradict in meaning or are of other interest.

For further studies we provide the anonymized data collected in the survey online<sup>2</sup>.

### 5.1.3 Usage Scenarios

Since our survey focuses on the practical usefulness and implication of our defined quality issues, we first describe their relation to the identified vocabulary usage scenarios. We focus on eight issues which our participants considered to be most important for the selected six usage scenarios. We then present the participants' agreement levels on quality issues, summarize their decision rationales, and discuss the findings we can derive from their answers.

The closed-ended question on what usage scenario the participants intend to support with the developed controlled vocabulary was answered by 76 respondents. "Classification/Categorization" was mentioned most often (58), followed by "Manual/Intellectual Indexing" (52) and "Faceted Search" (45). Multiple selections were allowed and only 5 participants selected "other" as a usage scenario.

For each usage scenario we collected the importance rankings over all quality statements as stated by the participants. By assigning a numerical value to each level of importance ("very important" → 1, "important" → 2, etc.) and calculating median and mean values, we were able to determine which quality statements were perceived as most important for each vocabulary usage scenario. Based on this data we found that "Publication", "Navigation", and "Linking" were mentioned as most important usage scenarios.

Table 5.2 provides a summary of our findings for the usage scenarios mentioned above by showing the three most important quality issues for each scenario. "1" indicates the quality issue that has been stated as most important, "2" indicates the second important

<sup>1</sup>For the analysis of the received answers, we treat the Likert scale as a balanced interval scale and can therefore use descriptive statistics.

<sup>2</sup>Survey Data: <http://tinyurl.com/oc24r3o>. Retrieved 2015-06-23.

and “3” the third important quality issue. Gaps indicate that the quality issue is not among the three most important issues for the usage scenario and the value has therefore been omitted in order not to clutter the table.

Usage Scenario	Omitted or Invalid Language Tags	Label Conflicts	Undocumented Concepts	Number of Synonyms and Non-descriptors	Cyclic Hierarchical Relations	Orphan Concepts	Missing Out-Links to Other Vocabularies	Missing Out-Links to Other Resources
Classification / Categorization	2		3		1			
Faceted Search	1	3			2			
Linking	3						1	2
Manual Indexing		3		2	1			
Navigation	1				2	3		
Publication		1	2		3			

TABLE 5.2: Importance of quality issues for usage scenarios.

#### 5.1.4 Labeling and Documentation Issues

This group of issues was considered as most important for the selected vocabulary usage scenarios, as shown in Table 5.2.

##### 5.1.4.1 Omitted or Invalid Language Tags

In our previous study we observed that language tags in documentary concept properties (e.g., labels, notes) are either used consistently for all concepts or are omitted completely. This raised the question whether inclusion of language tags is a commonly desired feature in Web vocabularies. To learn about the participants’ perception of omitted or invalid language tags, we included the statement “*Textual descriptions of concepts (e.g., labels) should make use of language tags*” in our survey. The majority of the participants (80.4% of the 56 respondents) agreed with that statement, 5.4% disagreed, and the rest selected neutral or gave no answer.

Participants who provided a rationale for their decision stated that using language tags is highly useful in multilingual and/or multicultural environments. It supports language independence and interoperability and enables the vocabulary to be utilized for translation use cases. Usability has also been pointed out as a benefit of making the used languages explicit. However, one contributor states that the user interface should inform the user about a vocabulary's language(s) instead of showing abbreviated codes used for language tags attached to the labels. Another argues that language tags might be superfluous for monolingual vocabularies.

#### 5.1.4.2 Label Conflicts

In *qSKOS* we also defined a function to detect *Overlapping Labels* (cf. Section 3.1.1) on a more general level than outlined in the SKOS primer [IS09]. This definition is expected to provide hints to duplicated concepts or misspelled labels. In our study we could observe that 8 of 15 reviewed vocabularies contain pairs of distinct concepts that have identical descriptors or non-descriptors. Thus, we included the statement “*Different concepts should not be labeled identically (i.e., their descriptors, non-descriptors, or synonyms should not overlap)*”. From the total 39 answers approximately 67% of the respondents agreed, 10% disagreed, and 23% gave no answer or voted for neutral.

Respondents who disagreed with this statement pointed out that identical labels cannot always be avoided, e.g., in case of homographs or when the set of indexing terms must not be changed. One contributor claimed label ambiguity to be beneficial for exploring an information system because it would lead to new search questions. Others perceive non-overlapping labels as important for human communication and automated processing, e.g., Natural Language Processing. Confusion (users select incorrect concepts) and decreased manageability have also been mentioned.

#### 5.1.4.3 Undocumented Concepts

Documentation is often considered beneficial for human users who work with a Web vocabulary. However, documentation can be provided on various levels. Options are, for instance, documenting on the vocabulary level (e.g., content overview or intended usage), documentation of certain groups of concepts, or documenting at the concept level (e.g., scope or history notes, definitions). In our survey we focused on the last case by asking participants to rank their agreement with the statement “*Every concept should be documented (by, e.g., scope notes, definitions, history notes)*”. More than 77% agreed, 9% disagreed, the rest selected neutral.



Contributors who agree with this statement mention that labels alone are often insufficient for disambiguation and understandability. Concept-level documentation provides additional context which has been identified as essential for indexing and tagging usage scenarios, as well as for establishing mappings between terms and auto-categorization techniques. Three contributors point out the importance of providing history notes for documenting a vocabulary's evolution. Contributors who disagreed argued that not every concept needs documentation and that documentation causes maintenance overhead that could be avoided by providing scope by means of adequate labels and relationships. Also, for some usage scenarios like "large-scale indexing of general-interest content", providing documentation for every concept is perceived as impractical and unnecessary by one contributor.

#### 5.1.4.4 Number of Synonyms and Non-descriptors

Web vocabularies differ widely in their support and quantity of synonyms and non-descriptors. Web vocabularies like DBpedia categories, for example, define only preferred labels and no alternative or hidden labels. Only 5,450 of over 170,000 concepts in GTAA<sup>3</sup> have alternative labels whereas AGROVOC provides on average more than four alternative labels per concept<sup>4</sup>. To find out if and in what cases synonyms and non-descriptors (lexical variants) are important, we included the statement "*The more synonyms and non-descriptors are defined per concept, the more useful is the controlled vocabulary*" in the survey. More than 60% agreed, 10.5% disagreed, and a relatively large number selected neutral (18.4%) or gave no answer (10.5%).

Again, the additional context given by a higher number of synonyms and non-descriptors has been pointed out as beneficial. One contributor stated that more synonyms improve usability whereas more non-descriptors (i.e., lexical variants) have the potential to improve interoperability with other sources. Similarly, it has been noted that synonyms enable more accurate searches and offer more choices in concept selection. Thus, the availability of synonyms and non-descriptors is seen as highly usage-scenario dependent. They may be more useful in text-focused applications but not for Linked Data applications. A rich number of synonyms has furthermore been mentioned as beneficial for manually mapping vocabulary terms. Contributors also stated that the quality of the included synonyms is crucial. They should be unambiguous and fit to the content, i.e., non-required synonyms should be excluded. Furthermore, the quantity may increase complexity and can increase recall and reduce precision. One contributor even argues

<sup>3</sup><http://datahub.io/en/dataset/gemeenschappelijke-thesaurus-audiovisuele-archieven>. Retrieved 2015-06-23.

<sup>4</sup>Numbers are taken from the dataset of our previous work, available at <https://github.com/cmader/qSKOS-data>. Retrieved 2015-06-23.

that adding many synonyms is a waste of time because natural language dictionaries already exist for this task.

### 5.1.5 Structural Issues

Our participants rated structural issues as important for five out of the six usage scenarios we focus on.

#### 5.1.5.1 Cyclic Hierarchical Relations

The negative aspects of cycles in hierarchical relations between concepts have been addressed in numerous tutorials and guidelines on vocabulary development ([ABG03, Hed10]). Nevertheless, in our previous vocabulary study, we could find cycles in hierarchical relations in three out of 15 vocabularies. This led to the question on the relevance of cycles for vocabulary quality and therefore included the statement “*Controlled vocabularies should not contain circular hierarchical dependencies between concepts*” in our questionnaire. 80% of the 30 respondents to this statement agreed (50% strongly agreed), 10% disagreed, 3.3% voted neutral and 6.7% provided no answer or did not know.

More specifically, we observed two kinds of cyclic relations: those that involve only one concept (reflexive cycles) and those that involve multiple concepts. Thus we included the two statements “*Concepts should not be hierarchically related to themselves*” and “*Controlled vocabularies should not contain circular hierarchical dependencies between concepts*” in our questionnaire.

Concerning the first statement, more than 77% of a total 31 respondents agreed (more than 51% even strongly agreed) that concepts should not be hierarchically related to themselves. 6.4% disagreed, 9.7% were neutral, and 6.5% gave no answer or did not know.

Although cycles may not turn out as problems in some scenarios (e.g., if hierarchically related to others and not top concept), reflexive cycles are perceived as unintuitive and increase the complexity of a vocabulary because they do not add any value. They may represent a degenerated cycle and contributors stated that they cannot imagine scenarios where reflexive cycles could be a requirement. One contributor stated that cycles can be a sign of “lack of care by the vocabulary publisher”. Others point out possible technical problems due to these “loops”.

Contributors have argued similarly for cycles involving multiple concepts. They are also perceived to decrease coherence, increase complexity, and are confusing and unintuitive.

However, as one participant noted, cycles are only an issue if hierarchical relations are interpreted transitively. Others state that cycles might be caused by misuse of hierarchical relations and suggest the use of other constructs (e.g., alternative labels) in order to avoid cycles.

#### 5.1.5.2 Orphan Concepts

Checking for orphan concepts, i.e., concepts that are not linked to other concepts, is a frequently employed quality assurance method. However, we experienced a high number of orphans in several Web vocabularies (e.g., GTAA, LCSH<sup>5</sup>, DBpedia categories<sup>6</sup>). Thus, we wanted to know how such structures are perceived in general in the Linked Data context and formulated the statement “*Every concept should be linked (e.g., associatively, hierarchically or equivalently) to at least one other concept of the controlled vocabulary*” in our questionnaire. Approximately 65% of 37 total respondents agreed to the statement and a small number (2.7%) provided no answer. A relatively large number of participants disagreed with the statement (22%) and 11% voted for neutral.

From the provided rationale the main concern about orphan concepts was their lack of scope and context which impacts the user’s understanding in a negative way. Furthermore orphan concepts are of “little automated usage” and make it easier to navigate through the vocabulary. However, orphan concepts sometimes cannot be avoided because some usage scenarios do not require relations between concepts (e.g., glossaries). In these cases, unnecessary relations for the purpose of circumventing orphans should not be “invented”.

### 5.1.6 Linked Data Specific Issues

Although the survey analysis indicates the importance of interlinking Web vocabularies for various usage scenarios, Linked Data specific issues have been considered as most important only for the usage scenario *Linking*.

#### 5.1.6.1 Links to Other Vocabularies

Establishing links to other vocabularies on the Web is a core Linked Data design principle and also suggested in controlled vocabulary development standards and guidelines (e.g., [ISO11b, ABG03]). However, it is currently unclear, how the value of links between online vocabularies of different provenance are perceived from a quality point of view. Thus, we

<sup>5</sup><http://id.loc.gov/authorities/subjects.html>. Retrieved 2015-06-23.

<sup>6</sup>downloadable at <http://wiki.dbpedia.org/Downloads#3>. Retrieved 2015-06-23.

included the statement “*Good-quality vocabularies reference (link) to other vocabularies on the web*” in our questionnaire. More than 64% of total 28 respondents who gave feedback on the statement agreed, 11% disagreed and 21% voted for neutral. The rest selected no answer/don’t know.

Additional scope and the ability to “share” resources are benefits of linking to other vocabularies on the Web. One participant meant that this is especially important for navigation, browsing and retrieval use cases. Other contributors noted that linking to other vocabularies “Allows better cross resource searching” and that it increases trust and understandability. However, contributors also mentioned that linked vocabularies must also meet a high quality standard like reasonably established vocabularies such as LCSH or AGROVOC. Three contributors argued that vocabularies can be of very good quality on their own and that links to other vocabularies are not an indicator of quality.

#### 5.1.6.2 Links to Other Resources

Linked Data allows for linking to any other kind of resource on the Web such as web pages that provide additional information about a concept. To find out the impact of such links on vocabulary quality, we included the statement “*Concepts should be linked to other resources on the Web (to, e.g., refer to additional information about the concept)*” in our questionnaire. More than 78% agreed with the statement, 3.6% disagreed (no participant strongly disagreed) and the rest voted for neutral.

The decision rationales are very similar to those discussing linking to third-party vocabularies. Linking to resources on the Web provides additional context, rendering it “useful for end-users and automatic extraction methods” as one contributor stated. Context has also been mentioned important to assist users in choosing an appropriate term. However, link stability has been a concern of three participants. Linked resources should be permanently available and no broken links should be introduced. Those who do not agree mention that vocabularies should be complete on their own and that links to other resources provide additional values but are no substitute for good vocabulary-internal descriptions and definitions.

#### 5.1.7 Responses

In total we received 163 responses with varying coverage because only a few questions were mandatory and some participants did not complete the survey. From the 25 participants who indicated their role, 12 were vocabulary managers, one was a contributor and four identified themselves as users. Two of these 25 participants chose to select the option

“No answer” and six stated other roles with two of them giving no exact role description. The maximum number of responses we received for a quality-relevant question was 56, decreasing towards the end of the survey with 28 being the minimum. The majority of the responses came from the US (39), followed by the UK (15) and Italy (10).

### 5.1.8 Summarized Findings

From the answers and results presented above we can infer the summarized findings listed below. They target the covered quality issues and their relevance according to the usage scenarios can be inferred from Table 5.2.

- Although not essential in a strictly monolingual context, language tags in RDF literals enhance understandability and usability of the vocabulary.
- It is generally desirable to have all concepts labeled in each supported language. However, this is not always possible due to missing equivalents in some languages.
- Presence of documentation on the concept-level is appreciated but costly and not always needed.
- Whenever possible, identical concept labels have to be avoided to maintain unambiguity and avoid confusion.
- If a vocabulary is intended to organize and contextualize concepts, orphans should generally be avoided.
- Circular hierarchical dependencies are unintuitive and may indicate or lead to errors.
- When judging the quality of a published and linked Web vocabulary, also the quality of the linked resources has to be taken into account.
- Link stability (changing availability and semantics) is perceived as a risk when interconnecting vocabularies on the Web.

### 5.1.9 Recommendations for Best Practices

Although there are many tutorials for creating and publishing SKOS vocabularies, such as the SKOS Primer [IS09], there are some aspects of the publishing that could benefit from more explicitly specified best practices. In particular, the question of which relationships to explicitly assert in the published vocabulary and which to leave for the vocabulary user

to infer is not always clear. All SKOS semantic relationships between concepts are either symmetric (e.g., `skos:related` and `skos:exactMatch`) or have an inverse counterpart (e.g., `skos:broader` and `skos:narrower`, and their transitive and mapping variants). In principle, a rather small set of relationships can be used to specify the whole vocabulary, and the remaining (redundant) ones inferred using RDFS and OWL inference. This may be a good strategy for editing SKOS vocabularies: minimal assertions are used during editing, and the rest are inferred and materialized only in the published vocabulary. This way, some inconsistent assertions involving inferred relationships, such as the instances of *Solely Transitively Related Concepts* we found in some vocabularies, can be avoided.

In practice, inference is not always possible or desirable for vocabulary users. Applications making use of SKOS vocabularies may benefit from explicitly asserted relations, even if they are in principle redundant and could have been inferred. We thus propose the following guidelines for the inclusion of SKOS relationships in vocabularies published on the Web of Data:

1. Explicitly declare the types of SKOS `skos:Concept`, `skos:ConceptScheme` and `skos:Collection` instances, even if they could be inferred. This is in line with the recommendation by Abdul Manaf et al. [AMBS12a].
2. Include one or more concept schemes describing your vocabulary and label them appropriately. Assert the full set of both `skos:topConceptOf` and `skos:hasTopConcept` relationships. Make sure `skos:inScheme` relationships are asserted for every concept.
3. Assert the full set of both `skos:broader` and `skos:narrower` relationships. This is also in line with the recommendation by Abdul Manaf et al. [AMBS12a]. However, do not include the `skos:broaderTransitive` and `skos:narrowerTransitive` relationships, as they are only likely to be useful in special scenarios, may add a lot of new assertions to the vocabulary, and may be inferred by the vocabulary user when necessary.
4. Assert `skos:related` properties reciprocally.
5. Assert mapping relationships only one way, with concepts from your own vocabulary as the subjects. This is to avoid “SKOS vocabulary hijacking”, i.e., the assertion of facts about vocabularies published by others, which is similar to *ontology hijacking* [HHP<sup>+</sup>10].

In this section, we combined the findings from the survey with our practical experience in working with controlled vocabularies and implementation of tools such as *qSKOS*.

We now provide suggestions and guidelines for quality checking functionality in Web vocabulary development tools.

#### 5.1.9.1 Labeling and Documentation Issues

When a user creates concept labels or free-text literals, vocabulary development tools should (semi-)automatically add *language tags*. The appropriate tag can be determined from sensitive default settings or by employing existing language detection tools. Vocabulary development tools should also provide language information in a meaningful way on the user-interface level to, for instance, assist users in search term disambiguation. Label suffixes such as “@de” could confuse users who are not familiar with RDF-based technologies and should thus be hidden in favor of a clearer language presentation. In cases where identical labels cannot be avoided, we suggest to structure the vocabulary by making use of the SKOS extension for Labels<sup>7</sup> that allows modeling labels as resources instead of literals. As a consequence, additional information such as scope notes for disambiguation or context-specific usage information can be directly attached to labels when needed. When reporting overlapping labels to the vocabulary creators, conflicts between alternative and preferred labels should be reported with higher priority than conflicts that occur between alternative labels of different concepts. When unique preferred labels cannot be avoided (e.g., in case of homographs), the vocabulary development software should prompt the user to add documentation (e.g., scope notes) for further disambiguation, especially when links between concepts are sparse. Furthermore, by monitoring search queries and user behavior, frequent mistakes can be (semi-) automatically added as hidden labels.

A common motivation for creating controlled vocabularies is to support translation-related use cases and, as a consequence, controlled vocabulary development software must support creation of labels in multiple languages. If used in a multilingual setting, each concept should be labeled in every relevant language. However, this is often not possible because direct equivalents of concepts in different languages sometimes do not exist. Thus, at least one “default” language should be supported, i.e., one language for which each concepts *must* have a label. Vocabulary development software should tolerate these gaps in language support, but should prompt the user to provide at least a documentation property in the “missing” languages to provide orientation for human users. Also, developers should have the choice to handle similar language tags equally, e.g., concepts labeled in **en-GB** should not require a label for **en-US**.

Concepts lacking context are problematic for using and adopting Web vocabularies. The best way to provide context is by establishing relations between concepts and linking

---

<sup>7</sup>SKOS-XL: <http://www.w3.org/TR/skos-reference/skos-xl.html>. Retrieved 2015-06-23.

to other (external) resources. Thus, vocabulary development software should encourage users to amend labels or documentation to concepts that still lack these interconnections.

Since detecting *conflicting labels* requires domain expertise and human input, vocabulary development and navigation interfaces should reveal a concept’s surrounding (e.g., hierarchical) structure to support the user in manual disambiguation. This is also important for supporting resolution of conflicts which can be done by merging or renaming and provide hierarchical or associative links to other concepts. Tools could also apply predefined rules for automatic label rewriting, e.g., by including the broader terms’ labels and helping in resolving conflicts. If a greater number of label conflicts occur in a vocabulary, another possibility is to (automatically) split the vocabulary into two, separately managed parts with their own, clearly defined scope.

### 5.1.9.2 Structural Issues

As stated by respondents of our survey, *circular hierarchical relations* often root in misuse of hierarchical properties. Users might, for instance, interpret a concept hierarchy either as “has-a” or “is-a” relations. To some degree, tools could suggest replacement of such circularities. Suominen et al. [SH12] introduce strategies to remove different kinds of hierarchical cycles between concepts. This approach could possibly be extended by providing feedback to the vocabulary developer and suggesting replacement with an associative relation. Taking into account transitivity when checking for cycles is an important operation in this case because computation of cycles in the transitive closure can be omitted if the user perceives hierarchical links as not being transitive.

*Orphan concepts* decrease cohesiveness of vocabularies and lack context. Vocabulary development tools should suggest semantically related concepts (e.g., inferred from existing popular resources on the Web) that orphan concepts could reference by mapping relations. Tools that automatically identify orphan concepts could also order them by degree of documentation. Orphans without additional documentation properties are more likely to constitute an error or being misinterpreted than those with adequate documentation. Furthermore, context can also be provided by other orphans being members of the same concept schemes or collections.

Whether or not orphan concepts affect the quality of a vocabulary also depends on the vocabulary type and use case. As we observed in the survey results, orphan concepts are more critical for, e.g., navigation usage scenarios and less severe for glossaries. Thus, automatic quality assessment tools could use classification methods (e.g., based on structural properties as suggested in [NPM11]) to infer these types and report orphan concepts only if necessary for the vocabulary type at hand.



### 5.1.9.3 Linked Data Specific Issues

Being able to assess the quality of a vocabulary's linked resources was another desire expressed by our survey participants. Therefore, tools that analyze vocabulary quality should offer the option to run this process also on vocabularies that are (i) *linked by* or (ii) *linked to* the main vocabulary. To avoid undesired effects on the semantics of third-party content, tools should recognize if the developer performs substantial changes to a concept that is linked by these resources. This is manageable in local settings but clearly more difficult in distributed settings, such as Linked Data. To find incoming links in the latter, one has to rely on dataset registries<sup>8</sup> or metadata descriptors like VoID<sup>9</sup>. Given the changing nature of the Web, checks for broken links should be performed automatically on a regular basis and developers should be notified accordingly.

Outgoing links are generally perceived as a method to provide additional scope to concepts, even though they are not strictly necessary for most usage scenarios our participants want to support. Concepts that lack “internal” description and documentation should therefore be reported by vocabulary development tools with a higher priority. To effectively check the quality of linked vocabularies, they should be accessible via a SPARQL endpoint and described in a machine-friendly way, e.g., by a VoID dataset descriptor.

A common concern among our contributors was the increased responsibility when introducing changes to a vocabulary that is linked to others. Providing history notes (rationale of changes) and methods for tracking changes (e.g., keeping multiple versions of concepts and vocabularies) is thus an important feature of vocabulary development tools. Participants of our survey have stated the need for provenance (*who* changed *what* and *when*) of controlled vocabularies. This information should be automatically gathered and attached to the vocabulary. Keeping “historical” data is perceived essential by some of our participants because compatibility with existing systems should be maintained.

## 5.2 Quality Analysis of Existing Vocabularies

In this section, we analyze currently published Web vocabularies by using the *qSKOS* tool. We provide a detailed list on the number and kinds of quality issues we found in each vocabulary. In our explanation of these findings we concentrate on giving examples that illustrate typical or curious findings. For further information, both the analyzed

<sup>8</sup>e.g., Sindice (<http://sindice.com/>), datahub (<http://datahub.io/>), both retrieved 2015-06-23.

<sup>9</sup>VoID vocabulary for describing linked datasets: <http://www.w3.org/TR/void/>. Retrieved 2015-06-23.

vocabularies as well as detailed reports of the analysis results are available online for download<sup>10</sup>.

The methodology of this section is largely based on our earlier work [MHI12], that was later extended by inclusion of a richer set of analyzed Vocabularies, an updated catalog of quality issues and a technique to automatically repair some quality issues [SM13]. As automated repair strategies for the found quality issues are outside of the scope of this thesis, we focus on providing and reviewing the reports generated by *qSKOS*. However, in this section we extend our earlier work by

- covering additional quality issues that were later implemented in *qSKOS* and some of which are based on feedback from our expert survey (Section 5.1),
- updating our findings according to the current version of *qSKOS*, and
- providing more detailed coverage for some of the found issues.

Despite of using an up-to-date version of *qSKOS* (1.4.3) for creating the quality issue reports, we use the same set of Web vocabularies as in our earlier work [SM13]. We downloaded each vocabulary that was provided as one or more RDF files and also included mappings to other vocabularies, in case the vocabulary publisher provided them. For vocabularies that were only available as SPARQL endpoints, we used a script<sup>11</sup> to query all the triples in the store and serialized them into files. We converted each vocabulary to a single merged file in Turtle syntax using the `rdcat` utility from the *Apache Jena*<sup>12</sup> distribution.

Further pre-processing was necessary for some vocabularies to make them compliant with the used RDF parsers in order to analyze them successfully. Missing namespace declarations were added manually for UMBEL. In NYTL, the invalid language tag `fr_1793` was manually changed into `fr-1793` in order to comply with BCP47 and the Turtle specification. In Reegle, an unparseable line in the original RDF dump was manually removed. For GEMET, the source file containing Arabic labels was excluded as it contained labels with improper Unicode encoding that caused the Jena toolkit to fail in parsing it. For ODT, STW and SSW, an URI pattern was explicitly specified to identify authoritative concepts. DDC contained triples asserting dates in an unparseable format, using the properties `dct:modified` and `dct:date`. We removed these lines because they do not have an effect on the analysis results.

<sup>10</sup>Analyzed vocabularies and reports: <http://tinyurl.com/ka5ellv>. Retrieved 2015-06-23.

<sup>11</sup>The script `sparqldump.py` is included in the *Skosify* distribution: <https://code.google.com/p/skosify/downloads/list>. Retrieved 2015-06-23.

<sup>12</sup><http://jena.apache.org>, retrieved 2015-06-23.

### 5.2.1 Vocabulary Statistics

Table 5.3 summarizes some basic statistical properties of our vocabulary selection, such as the number of concepts and authoritative concepts, concept labels (i.e., `skos:prefLabel`, `skos:altLabel`, and `skos:hiddenLabel` relations involving concepts), semantic relations (i.e., pairs of resources related by a subproperty of `skos:semanticRelation`), and URIs that use the HTTP scheme.

	All Concepts	Authoritative Concepts	Concept Labels	Semantic Relations	Concept Schemes	Collections	HTTP URIs
ODT	233	107	326	512	6	0	493
GeoNames	680	680	3241	0	9	0	179
Reegle	2952	1447	3665	29456	12	0	5480
PXV	2112	1686	3628	2693	1	0	2770
NYTL	1920	1920	1920	0	1	0	64461
SSW	2656	1939	3487	14042	10	0	4389
IPTC	2061	2061	1128	2241	0	0	2065
UNESCO	2509	2509	7512	5740	1	0	2515
Plant	6492	3246	3581	28576	3	0	11405
IPSV	4732	4732	7945	13843	3	0	4771
NYTP	4979	4979	4979	0	1	0	29341
GEMET	14112	5209	165890	22129	1	79	14198
STW	25107	6789	58441	71200	3	0	25171
Eurovoc	6797	6797	457788	14289	128	0	403936
LVAk	13411	13411	17250	16346	0	0	13414
EARTH	26137	14351	30403	48038	1	0	26161
UMBEL	26427	26389	88621	72330	0	0	26922
AGROVOC	52893	32291	624776	86641	1	0	666573
SNOMED	102614	102614	150964	265483	1	0	9
GTAA	171991	171991	178776	50889	9	0	172005
RAMEAU	355158	207272	470392	465688	0	0	1648701
DDC	251977	251977	158162	302331	70	0	284817
LCSH	503943	408923	750219	659885	1	408923	503537
DBpedia	865902	865902	862826	1727029	0	0	865904

TABLE 5.3: Vocabulary statistics as determined by *qSKOS*, ordered by the number of authoritative concepts

From these properties we can see that approximately 3 000 DBpedia categories concepts are missing labels (e.g., `Category:South_Korean_social_scientists`), which is a consequence of missing natural language descriptions in some Wikipedia categories. Also, many concepts in DDC are not labeled in natural language but have a `skos:notation` literal defined instead.

We can also determine the type of the vocabulary from the number of `skos:semantic-Relation` relations to some extent. GeoNames, NYTL and NYTP are mainly intended as authoritative lists and do not define, e.g., hierarchical or associative relations between concepts.

The reason for only nine HTTP URIs found in SNOMED is that the concepts are identified by URI fragments (e.g., [http://Snomed3\\_5.fr#C-7087](http://Snomed3_5.fr#C-7087)) which are to be evaluated on the client side and thus treated as one URI ([http://Snomed3\\_5.fr](http://Snomed3_5.fr)) by *qSKOS*.

It is furthermore remarkable that only two (GEMET and LCSH) of the 24 vocabularies of our representative set of Web vocabularies are structured by assigning concepts to a `skos:Collection`. However, most vocabularies (19 of 24) define at least one `skos:ConceptScheme` to aggregate concepts and impose an additional level of structure.

### 5.2.2 Labeling and Documentation Issues

Table 5.4 shows the result of our vocabulary analysis focusing on labeling and documentation related issues. They focus on inconsistencies and omitted values for properties such as the SKOS label relations (`skos:prefLabel`, `skos:altLabels`, `skos:hiddenLabel`) but also include properties from other data schemas like `dc:title`. All of them have in common that they are used to annotate resources with literals holding either natural language information designed for human interpretation and interaction or typed data intended to reference the resource in third-party knowledge organization systems. The group of labeling and documentation issues encompasses in total eight quality issues. We found occurrences of at least two of them in each of the reviewed Web vocabularies.

#### 5.2.2.1 Omitted or Invalid Language Tags

Occurrences of this quality issue can be observed in 14 of the 24 vocabularies. In ODT this issue only occurs in three blank nodes of the VoID dataset descriptor describing `void:TechnicalFeatures`. This is also the case for Plant, Reegle, and SSW which all were created with the *PoolParty Thesaurus Manager*.

Eurovoc describes 218 countries which have a `skos:altLabel` consisting of two characters (e.g., “PT” for the Portuguese Republic) without a language tag. Additionally, one language tag is missing for the preferred label of the `skos:ConceptScheme` definition.

PXV and LVAk omit language tags with their labeling properties, LCSH with documentation properties (e.g., `skos:note`, `skos:editorialNote`, `skos:example`). STW uses

	Omitted or Invalid Language Tags	Incomplete Language Coverage	No Common Language	Undocumented Concepts	Overlapping Labels	Missing Labels	Empty Labels	Unprintable Characters in Labels	Ambiguous Notation References
ODT	3	16	-	35	2	0	0	0	0
GeoNames	0	43	-	60	162	9	0	1	0
Reegle	3	1450	-	3	22	0	3	0	2
PXV	1578	0	×	1492	7	2	0	0	0
NYTL	0	0	-	1862	0	1	0	0	0
SSW	4	1143	-	1324	39	1	0	0	0
IPTC	0	0	-	933	1	933	0	0	0
UNESCO	0	0	-	2509	227	5	0	0	0
Plant	1	0	-	220	54	0	0	0	0
IPSV	0	0	-	2899	0	2	0	0	0
NYTP	0	0	-	4094	0	1	0	0	0
GEMET	4	894	-	1	3638	0	35	1	0
STW	45	25050	×	5290	10123	216	0	0	0
Eurovoc	219	6370	-	5341	62	0	0	0	0
LVak	13411	0	×	13411	13	0	0	0	0
EARTH	1	313	-	7840	2100	1	189	0	0
UMBEL	25793	0	-	2848	5207	558	0	0	0
AGROVOC	0	32060	×	29820	2666	232	233	8457	0
SNOMED	102600	0	×	102614	229	16	0	0	0
GTAA	0	0	-	96850	11894	1	0	0	0
RAMEAU	116343	140860	×	70358	5539	34803	0	168	0
DDC	0	158161	×	251977	40729	93886	25	4	88966
LCSH	100316	0	-	308607	7766	1	0	0	17572
DBpedia	0	0	-	865902	765	3076	0	0	0

TABLE 5.4: Validation results using *qSKOS*, Part 1: *Labeling and Documentation Issues*. The **×**-marks indicate that no common language (cf. Section 5.2.2.3) could be detected in the corresponding vocabularies.

many **@x-other** language tags, which are considered invalid by *qSKOS*, and additionally does not use language tags with two instances of **skos:definition**, which have apparently been copied from the SKOS RDF schema.

SNOMED completely omits language tags for concepts. They are only used for the description and license statement of the vocabulary, expressed with the **dc:description** and **dc:rights** properties.

RAMEAU uses language tags predominantly with **skos:prefLabel**, **skos:altLabel**, **skos:scopeNote**, and **skos:inScheme** attributes, although the use of the latter does not conform with the SKOS schema (RAMEAU uses a literal instead of a **skos:ConceptScheme**

resource as the object of the `skos:inScheme` statement). Furthermore, literals of `dcterms:description` in some cases also have assigned language tags, mostly if the description is given in natural language, e.g., “Suite lithographique pour illustrer l’oeuvre de Shakespeare”@fr but not for position descriptions such as “383-[1] p.”. Also, literals of the `dcterms:title` property are sparsely annotated with language tags.

### 5.2.2.2 Incomplete Language Coverage

Incomplete Language Coverage was spotted in 11 of the 24 vocabularies. Most concepts in ODT are described with English and German preferred labels, except 16 which lack the German `skos:prefLabel`.

Nearly all of the 6 370 incompletely covered concepts in Eurovoc omit the Irish and Maltese languages (language tags @ga and @mt); in six cases Hungarian (@hu) is missing. Apparently, translation into these languages has not been performed yet, which is reflected by the SKOS-XL<sup>13</sup> labels that state `eu:toBeTranslated` properties with the literals “ga” or “mt” as objects.

AGROVOC contains literals in 25 different languages but 32 060 concepts are not labeled in all languages. From these, 19 concepts lack labels for only two languages whereas others do not cover up to 24 languages.

STW, which is expressed mainly in English and German, has many concepts with incomplete language coverage because it (i) links to non-authoritative concepts that are only labeled in German and (ii) uses the private, but valid language tag @x-other with some of its concept labels.

158 161 concepts in DDC have incomplete descriptions in exactly 13 languages. This happens because concepts are defined separately for different languages. For example, the concepts `ddc:class/746.44/2007/02/about.it` and `ddc:class/955/2009/03/about.de` have only an Italian or German `skos:prefLabel` defined. Also, many concepts in DDC only have English labels.

### 5.2.2.3 No Common Language

We found that in seven of the 24 analysed vocabularies concepts are not described in one common language. LVAk, PXV, and SNOMED have no language tags attached to literal values at all, therefore no common language can be extracted from these vocabularies.

---

<sup>13</sup>SKOS-XL is an extension schema to SKOS that enhances the labeling capabilities by treating labels as resources and not as literals.

For STW, the cause for not finding a common language lies in the `@x-other` language tags that are assigned to some label literals, although the labels of the remaining concepts are all equipped with either an English or German language tag. In AGROVOC, a single common language for all concepts is missing although each concept is described by literals having valid language tags.

RAMEAU uses 114 distinct languages to label the concepts it defines. However, many concepts are described only in a subset of these languages. On smaller scale with only 14 distinct languages this was also observed for DDC.

#### 5.2.2.4 Undocumented Concepts

All of the 24 vocabularies that we reviewed contained *Undocumented Concepts*. The vocabularies with the least occurrences of this issue are GEMET and Reegle.

GEMET provides a `skos:definition` for each concept except one (“wood resource”). Similarly, Reegle assigns a `skos:definition` to its 1 444 authoritative concepts. Although these definitions in 642 cases contain only a placeholder text (“No reegle definition available”), definitions are omitted for three concepts. Also ODT makes heavy use of `skos:definition` properties. However, we could find 35 concepts lacking these or other SKOS documentation properties.

The most widely used documentation properties in Eurovoc are `skos:scopeNote` but there are 5 341 of 6 797 concepts that remain undocumented. Also, all other vocabularies have a significant number of undocumented concepts.

#### 5.2.2.5 Overlapping Labels

*Overlapping Labels* were observed in 21 vocabularies, the least of them in IPTC (1) and ODT (2).

Overlapping labels in IPTC and the UNESCO vocabulary are caused by the same reason. They only occur between `skos:prefLabel` values because the other SKOS labeling properties `skos:altLabel` and `skos:hiddenLabel` are not used. The overlap arises because the IPTC and UNESCO are hierarchical classifications where the categories are implicitly qualified by their surrounding context, but the context is not expressed in the label itself. For example, the preferred label “freestyle” is used for separate concepts that are in one case narrower concepts of a concept describing the “wrestling” sport and once in a “swimming” context. In UNESCO, *Theory* appears both under *General demography* and *General sociology*. There are also many categories with the label “Other (specify)”@en.

ODT shows two cases of label overlap due to the use of the same abbreviations as alternative labels in different concepts.

The 765 overlapping labels in DBpedia are caused by duplicate categories which differ only in case, e.g., “Visual Arts” and “Visual arts”.

Abbreviations are a source for overlap also in Eurovoc. For example, the concepts with the preferred label “United Nations High Commissioner for Human Rights” has an alternative label “UNHCR” in Polish language. However, there also exists a concept with an alternative label “United Nations High Commissioner for Refugees” which has been assigned the same abbreviation as the preferred label. Besides these abbreviation-related overlaps we could also observe identical labels being used for different concepts, e.g., “hooldushüvitis”@et defined both as a `skos:prefLabel` for the concept `eurovoc:7946` and as an `skos:altLabel` for the concept `eurovoc:4209`.

In the same way, PXV uses the string “primary peroxisomal enzyme deficiency” with two concepts in the same concept scheme, but once with a `skos:prefLabel` and in another case with a `skos:altLabel` property.

There are over 10 000 overlapping labels in STW. They arise because the vocabulary includes mappings to other vocabularies, and the mappings include the labels of the foreign concepts. The current version of qSKOS cannot distinguish between authoritative and non-authoritative concepts when looking for overlapping labels.

#### 5.2.2.6 Missing Labels

We observed missing labels in 18 of 24 vocabularies. A common type of resource that lacks label information is `skos:ConceptScheme` which is the case for Geonames, NYTL, NYTP, IPSV, and LCSH.

In the EARTH vocabulary, the only concept with a missing label is at the same time an orphan concept that has no additional information asserted (despite being of type `skos:Concept`).

The two occurrences in PXV are caused by two concepts having assigned an `rdfs:label` instead of a `skos:prefLabel`, as it is the case for all other concepts in this vocabulary.

In SNOMED both the single defined concept scheme as well as the top concepts assigned to it lack textual label information.

Most of the concepts in STW without labels are deprecated concepts (as described in the `skos:historyNotes` of these concepts) and only have `rdfs:label` properties assigned instead of SKOS labels.



In RAMEAU, SSW, and GTAA many concepts are defined as broader or narrower concepts of other concepts, but no additional facts about these resources are contained in the vocabulary, leading to detection of missing labels. This is possibly caused by an incomplete data dump. Also DBpedia sets concepts into hierarchical or associative relations to other concepts which do not have any additional information asserted. The same can be observed for AGROVOC and IPTC with concepts mapped by, e.g., `skos:exactMatch` or `skos:broadMatch` properties. In UNESCO four concepts related by using the `skos:related` property do not have any additional facts asserted. One statement furthermore relates the concept labeled “Water chemistry” to an URI consisting only of a namespace (`unesco6`), which is most likely not intended.

#### 5.2.2.7 Unprintable Characters in Labels

In AGROVOC we found 8 457 occurrences of the “zero-width non-joiner” (Unicode character 200C) control character in the languages Farsi and Hindi. This character is used to ensure correct typography by suppressing ligatures. In RAMEAU we spotted 168 occurrences of this quality issue, mostly caused by newline characters and, with 11 occurrences, Unicode control character “right-to-left mark” (200F) and “right-to-left embedding” (202B) in the languages Hebrew and Yiddish but also in labels with German language tags and without language tags at all. In DDC we found unprintable characters in four Vietnamese (`@vi`) labels and one occurrence of the quality issue in both GEMET (Unicode Character “left-to-right mark” 200E in a Maltese language label) and GeoNames (newline character).

While some of the spotted unprintable characters control the correct representation of labels in user interfaces, we consider newline characters a quality problem, probably introduced by a conversion process or the application used for editing the vocabulary. In any case, applications that make use of these labels must be aware that byte-wise string comparison algorithms against user input may not give the expected results.

#### 5.2.2.8 Empty Labels

We found empty labels in only five of the 24 Web vocabularies we analyzed. In EARTH and AGROVOC many preferred labels in Italian are empty. Although Italian labels are assigned to many concepts, for some of them they are missing and just an empty string is assigned `""@it` as preferred label. A similar pattern can be found in GEMET where preferred labels tagged with `@bg` are mostly empty.

In Reegle we found empty alternative and hidden labels for both Spanish and English language. Interestingly, the concepts that had empty English alternative and hidden labels assigned, in addition had also proper English labels (e.g., “hydropower plant”) assigned. So these empty labels are probably the remains of manual label deletion actions by some vocabulary contributor and thus constitute superfluous information. In DDC all 25 empty labels are English preferred labels (“”@en).

### 5.2.2.9 Ambiguous Notation References

Eight vocabularies make use of the `skos:notation` property, namely DDC, GEMET, GeoNames, GTAA, LCSH, REEGLE, STW, and UNESCO. We found ambiguous notation references in three of them.

In Reegle we found in total two occurrences of this quality issue. One concept has both an empty notation (a string literal with zero length) and one valid notation (“HP 01.02”). One notation (“FC 04.01”) is used by two concepts.

Ambiguous notation references were spotted in LCSH 17 572 times. 7 331 concepts have multiple distinct notations assigned and 10 241 notations are assigned to more than one concept.

In DDC we also found a high number of occurrences for this issue, 88 966 in total. 33 758 concepts have multiple annotations assigned, whose literal representations are very similar. Many of them, e.g., <http://dewey.info/class/113/> have two notations with identical textual information (e.g., “113”) but different datatypes (either `file:schema-terms/Notation` or `ddc:Notation`). Other concepts have two notations assigned that only differ by one character. For example, the concept with the preferred label “Saren” was assigned the notations “T2-4947644” and “2-4947644”, both of type `ddc:Notation`. 55 208 notation literals are used for more than one concept. The reason for this is that the creators of DDC decided to keep multiple versions of one resource in the vocabulary. Therefore, one notation is used in all versions of the resource, e.g., the notation “641.3441” is assigned to the resources [641.3441/e23](#), [641.3441/e23/2012-08-08](#), [641.3441/e23/2012-08-01](#), and [641.3441/e23/2012-06-14](#), which are versions of the concept [641.3441](#)<sup>14</sup>.

	Orphan Concepts	Disconnected Concept Clusters	Cyclic Hierarchical Relations	Valueless Associative Relations	Solely Transitivity Related Concepts	Omitted Top Concepts	Top Concepts Having Broader Concepts	Unidirectionally Related Concepts	Hierarchical Redundancy	Reflexively Related Concepts	Mapping Relations Misuse
ODT	4	7	0	7	0	0	2	0	46	0	2
GeoNames	680	0	0	0	0	9	0	0	0	0	0
Reegle	4	2	0	2013	842	1	0	93	9787	0	14
PXV	2	10	0	0	0	0	1	2295	2	0	0
NYTL	1920	0	0	0	0	1	0	0	0	0	0
SSW	6	1	0	118	22	0	0	3	2576	0	16
IPTC	0	10	0	0	1113	0	0	2241	52	0	1128
UNESCO	0	1	0	19	0	0	0	124	0	0	0
Plant	0	22	0	3463	0	0	44	0	126	0	0
IPSV	0	1	0	253	0	0	0	25	2	0	4035
NYTP	4979	0	0	0	0	1	0	0	0	0	0
GEMET	0	5	0	31	0	1	0	0	25	0	0
STW	70	141	0	5004	0	2	0	7	4772	0	0
Eurovoc	7	4	0	6	0	1	0	14289	2652	0	0
LVAk	21	11	5	5	0	0	0	16344	637	0	0
EARTH	2288	354	0	1124	0	0	0	61	2	0	0
UMBEL	2936	86	5	0	36535	0	0	740	3482	0	0
AGROVOC	0	234	0	281	0	0	0	25	1	0	0
SNOMED	0	1	0	119	0	0	0	60396	1	4	0
GTAA	162000	621	0	9448	0	9	0	18804	5355	0	0
RAMEAU	86137	24927	4	5118	0	0	0	83987	4597	1	0
DDC	97294	2087	0	0	0	30	1812	4761	0	0	0
LCSH	173149	22343	0	0	0	1	0	74	13100	0	0
DBpedia	103877	1174	1133	9021	0	0	0	1713339	171637	1482	0

TABLE 5.5: Validation results using *qSKOS*, Part 2: *Structural Issues*.

### 5.2.3 Structural Issues

Table 5.5 summarizes our findings regarding the structure of the analyzed Web vocabularies. This group contains in total eleven issues that focus on certain patterns of semantic relations between multiple resources.

<sup>14</sup>We omitted the prefix <http://dewey.info/class/> for the URIs in this example to improve readability.

### 5.2.3.1 Orphan Concepts

*Orphan Concepts* occur in 17 of the 24 vocabularies. In the GeoNames, NYTL, and NYTP vocabularies, all concepts are orphan concepts, which means that these vocabularies are authority files rather than thesauri or taxonomies. This also implies that these vocabularies have no disconnected concept clusters. GTAA is a mixture of name authority file (approx. 162 000 concepts) and thesaurus (approx. 10 000 concepts). The 70 orphan concepts in STW are deprecated concepts and marked as such with the `skos:historyNote` property.

All four orphan concepts of ODT are top concepts within the same concept scheme with the `rdfs:label` “Regions”, i.e., they are not used with any `skos:semanticRelations` in the vocabulary. These concepts may be very infrequently used which could also be indicated by the so far uncorrected typing error in the preferred label “Ocenania” of the concept <http://vocabulary.semantic-web.at/OpenData/Ocenania>.

Similarly, all seven orphan concepts of Eurovoc are top concepts that do not participate in any `skos:semanticRelation`.

The large number of orphan concepts in DDC are caused by the way different versions of a concept are organized. For example, the orphan concept <http://dewey.info/class/2--499/e23/> is only related to its versioned counterparts, e.g., <http://dewey.info/class/2--499/e23/2012-08-08/>, by the property `dct:hasVersion`. These versioned concepts are then organized in an hierarchical structure.

### 5.2.3.2 Disconnected Concept Clusters

*Disconnected Concept Clusters* (DCCs) are found in 21 vocabularies. Three vocabularies show no DCCs because all concepts are orphan concepts and thus no relations between them are established. Four vocabularies (IPSV, SNOMED, UNESCO, and SSW) consist of only one “giant component”, which is for some cases considered an optimal vocabulary structure because every concept can be reached from each other concept by following a path of `skos:semanticRelations`.

STW forms one giant component (containing 24 572 concepts), but has also 140 additional DCCs, which all consist of authoritative concepts mapped to third-party vocabularies. All other vocabularies split into several clusters of semantically related concepts, each of which represents a certain subtopic.

Eurovoc has four DCCs, consisting of 6 775, 6, 5, and 4 concepts. In the large DCC (the “main” cluster) a custom ontology is used to organize numerous micro-thesauri and

domains and cross-connects concepts by `skos:related` properties. However, this is not the case for the three small DCCs, which might possibly indicate a quality flaw.

GTAA consists of 621 highly unbalanced DCCs. One component contains 8 413 subjects from a thesaurus with carefully curated semantic relations. Most other components contain fewer than 10 entities from other categories, e.g., locations, person names, and genres.

PXV consists of ten topic-related DCCs, such as “deficiencies”, “defects”, or “signals”. Some of the eleven concept clusters contained in the LVAk thesaurus are obviously forgotten test data.

### 5.2.3.3 Cyclic Hierarchical Relations

Only four vocabularies contain *Cyclic Hierarchical Relations* which is a comparatively small number. Also, the number of cycles within the vocabularies is small (4 or 5), except for DBpedia which contains 1 133 cycles.

Four of the five cycles in UMBEL involve only two concepts, one cycle involves three concepts. Also in LVAk the cycles are rather small with five involved concepts at maximum. RAMEAU has one cycle involving 20 concepts whereas the other three cycles, contain only 2–3 concepts.

Also the cycles found in LVAk are rather small, involving 2–5 concepts. They seem to be accidentally created and could, in our opinion, be resolved by deleting hierarchical relations or replacing them with associative relations or synonym definitions.

In the collaboratively created DBpedia vocabulary, many cycles are caused by concepts that have reflexive `skos:broader` relations (see also Section 5.2.3.10). The DBpedia authors are aware of this, noting that the “categories do not form a proper topical hierarchy, as there are cycles in the category system and as categories often only represent a rather loose relatedness between articles” [BLK<sup>+</sup>09].

### 5.2.3.4 Valueless Associative Relations

*Valueless Associative Relations* have been detected in 16 vocabularies. Some of the potentially valueless associative relations could possibly be fixed by reconsidering the structure and replacing some associative relations by hierarchical ones. This could be observed, e.g., in LVAk and GEMET. The latter defines the concept labeled “leukaemia” as `skos:related` to the concept labeled “cancer” with a common parent labeled “human disease”@en. Here an hierarchical structure might be worth considering.

In general, the total number of occurrences of this issue is relatively low compared to the number of all semantic relations in the respective vocabularies. Still, in large vocabularies occurrences of this issue can rise to thousands, making revision of the affected relations unmanageable for a single thesaurus manager.

#### 5.2.3.5 Solely Transitively Related Concepts

*Solely Transitively Related Concepts* were found in four vocabularies. UMBEL only uses `skos:broaderTransitive` and `skos:narrowerTransitive` properties and completely omits `skos:broader` and `skos:narrower` properties. IPTC only uses `skos:broaderTransitive` relations to create an hierarchical structure.

The other two vocabularies being affected by this issue are SSW and Reegle with 22 and 842 occurrences, respectively. Both vocabularies were developed using the *PoolParty Thesaurus Server* which can be configured to automatically infer `skos:broaderTransitive` and `skos:narrowerTransitive` relations and include them in the vocabulary. Speaking to the developers of the PoolParty system, we were informed that this functionality is now discontinued. However, the exact causes of these “superfluous” transitive relations remain to be investigated.

#### 5.2.3.6 Omitted Top Concepts

*Omitted Top Concepts* were found in 10 of the 24 reviewed vocabularies. NYTL, NYTP, LCSH, GEMET, GTAA, and GeoNames omit top concepts in all the concept schemes they define. Eurovoc uses 128 concept schemes but has one without a top concept, which simply contains all concepts defined in the vocabulary. Such an “umbrella concept scheme” without a top concept is also present in LCSH, NYTL, NYTP, and GEMET. The only concept scheme in Reegle that omits a top concept is automatically created by the *PoolParty Thesaurus Server* and does not contain any concepts. The two omitted top concepts in STW are introduced by the AGROVOC and GESIS<sup>15</sup> mapping files. Both of them assign concepts from their originating vocabulary to a concept scheme also in this vocabulary which seem to be copied statements from the original publication.

---

<sup>15</sup>TheSoz Thesaurus for the Social Sciences, <http://datahub.io/dataset/gesis-thesoz>. Retrieved 2015-06-23.

### 5.2.3.7 Top Concepts Having Broader Concepts

In our selection of vocabularies, only four vocabularies feature *Top Concepts Having Broader Concepts*. ODT defines 29 top concepts, but only two of them have broader concepts. However, the broader concepts of these two concepts are again top concepts.

In its current version, PXV is affected by one top concept that has broader concepts. In earlier versions more of them could be found which were, according to the vocabulary creator, abandoned but still available in the triple store, probably caused by some bug in the vocabulary management software.

All three concept schemes defined in Plant have associated top concepts. From these, 44 are related to broader concepts.

### 5.2.3.8 Unidirectionally Related Concepts

*Unidirectionally Related Concepts* are contained in all except for six vocabularies (ODT, GeoNames, NYTL, GEMET, NYTP, Plant) which assert the complete set of reciprocal relations.

In SSW three occurrences involve one concept that is hierarchically related to other concepts but no additional facts are asserted for this one concept. This means that neither a label nor a contributor or a creation date is specified although these are the standard properties created by *PoolParty Thesaurus Server* used for creating the vocabulary. Also IPSV, AGROVOC and EARTH lack reciprocal relations for the assertion of concept scheme memberships (`skos:inScheme`) where reciprocal relations are omitted in EARTH also with `skos:related`. SNOMED asserts `skos:related` relations only in one direction.

STW includes all reciprocal relations except for the seven top concepts where the `skos:topConceptOf` relation is missing. LVAk, Eurovoc and DBpedia generally omit assertion of reciprocal relations at all.

### 5.2.3.9 Hierarchical Redundancy

Five vocabularies, GeoNames, NYTL, NYTP, UNESCO, DDC, have no redundant hierarchical relations asserted. In combination with the lack of *Cyclic Hierarchical Relations* this reflects a tree structure of these vocabularies, suitable, e.g., for classification use cases because redundant hierarchical relations would break the strict hierarchy.

For some vocabularies this issue could be observed for only a small fracture of the total hierarchical relations. SNOMED and AGROVOC, for example, each have one pair of

concepts related hierarchically redundantly. While the redundancy in SNOMED involves only authoritative concepts, in AGROVOC also mapped concepts of linked vocabularies are affected by the property `skos:broadMatch`: two authoritative concepts are hierarchically linked by `skos:broader` and both of them are mapped to the same concept using `skos:broadMatch`.

While EARTH, IPSV, and PXV have a high number of immediate hierarchical relations (over 10 000 for EARTH, over 3 000 for IPSV, and over 1 600 for PXV) each of these vocabularies has only two pairs of concepts involving redundant hierarchical relations.

### 5.2.3.10 Reflexively Related Concepts

We observed concepts related to themselves only in three vocabularies, SNOMED, RAMEAU, and DBpedia. In SNOMED four concepts are asserted to be `skos:related` with themselves which, in a strict logical sense, is not wrong but could be considered redundant. Similarly, RAMEAU contains one reflexively related concept using the `skos:related` relation. In DBpedia, reflexive relations occur 1 482 times with two kinds of relations, `skos:related` and `skos:broader`. The latter are also reported as *Cyclic Hierarchical Relations* and are likely to constitute a quality problem, because an intuitive understanding of the “Wikipedia categories” (which are expressed in the DBpedia dataset), would suggest a tree-like classification structure.

### 5.2.3.11 Mapping Relations Misuse

We spotted occurrences of this issue in five vocabularies: ODT, Reegle, IPSV, IPTC, and SSW. In all of them we found concepts that are mapped using `skos:mappingRelation` but which are not asserted to be a member of any concept scheme.

## 5.2.4 Linked Data Specific Issues

In Table 5.6 we give an overview about issues we consider relevant for online publication and interoperability with other vocabularies. We did not include figures of *Missing In-links* and *Broken Links* for LVAk because this vocabulary is not yet published online.

### 5.2.4.1 Missing Incoming Links

For 22 of the 24 analyzed vocabularies, the number of missing incoming links is very close to the number of authoritative concepts. This means that for these vocabularies



	Missing Incoming Links	Missing Outgoing Links	Broken Links	Undefined SKOS Resources	HTTP URI Scheme Violation
ODT	111	31	37	1	0
GeoNames	24	680	11	0	0
Reegle	1447	809	321	1	9
PXV	1686	1046	107	0	0
NYTL	1892*	0	1376*	0	0
SSW	1941	1606	285	1	1
IPTC	2061	933	2	1	0
UNESCO	2509	2509	1	0	0
Plant	3246	0	662	0	0
IPSV	4731	4732	1	1	0
NYTP	4965	0	9	0	0
GEMET	3290*	584	40*	0	0
STW	6781	1463	504	0	0
Eurovoc	6170*	6797	120790*	0	0
LVAk	-	13411	-	0	0
EARTH	14349	9558	410	0	0
UMBEL	26110*	0	130*	0	0
AGROVOC	31680*	17286	160*	0	0
SNOMED	102610*	0	5*	0	0
GTAA	171990*	171991	740*	0	0
RAMEAU	207260*	34803	132333*	0	0
DDC	250790*	458	110*	0	0
LCSH	408920*	347560	2640*	0	0
DBpedia	865566*	865902	11400*	0	0

TABLE 5.6: Validation results using *qSKOS*, Part 3: *Linked Data Specific Issues*. Values marked with an asterisk (\*) have been extrapolated from a randomly sampled subset of the concepts.

only a small number of the defined concepts are referenced by other vocabularies on the Web. Only for two vocabularies, GeoNames and GEMET, we can find a significant percentage of their concepts referenced by other vocabularies (96% for GeoNames and 37% for GEMET).

However, it is important to take into consideration that, as described in Section 4.2.2.21, we evaluated this issue using data from existing Linked Data indexes which can be incomplete. As a consequence, the values may not be representative for the Web of Data as a whole.

#### 5.2.4.2 Missing Outgoing Links

The difference between the number of concepts and the number of authoritative concepts in Table 5.3 already indicates which vocabularies contain outgoing links to other SKOS vocabularies. Closer examination shows that every authoritative concept in NYTL, NYTP, and Plant is linked to other resources on the Web. UMBEL and SNOMED are also reported to define an outlink for every concept, but this is caused by multiple type definitions (e.g., every concept in UMBEL is also explicitly typed as `owl:NamedIndividual` and `owl:Class`), and should be considered in future versions of the tool. In a similar way, DDC defines most concepts as being of type `owl:Thing`.

Eurovoc, GeoNames, IPSV, GTAA, UNESCO, and DBpedia do not define outgoing links for any of the authoritative concepts they specify. Most other vocabularies, e.g., RAMEAU, AGROVOC, STW, and GEMET expose a significant difference in the number of authoritative concepts and missing outgoing links. This means that most of the concepts they define reference related third-party resources on the Web.

#### 5.2.4.3 Broken Links

Even though we could not determine the exact number of *Broken Links* because of the large number of links to resolve (over 400 000 in Eurovoc, over 500 000 in LCSH), we found that broken links are a common issue in most vocabularies. However, some vocabularies (IPSV, UNESCO, IPTC) contain very few links that could not be dereferenced at the time of testing. For others, e.g., Eurovoc, we were not able to dereference one third of all HTTP URIs mentioned in the vocabulary, including authoritative concepts. This was possibly caused by a misconfiguration of the vocabulary data server.

#### 5.2.4.4 Undefined SKOS Resources

We were able to spot *Undefined SKOS Resources* in five vocabularies. IPSV uses the deprecated `skos:prefSymbol` property. ODT, Reegle, and SSW still contain the deprecated `skos:subject` property. IPTC states top concepts using the property `skos:HasTopConcept`, which does not match the property definition in the SKOS ontology.

#### 5.2.4.5 HTTP URI Scheme Violation

URIs that have a schema part that is not equal to HTTP or HTTPs were observed in only two vocabularies. Reegle and SSW use URNs for free concepts, i.e., resources that have a

preferred label assigned but are no SKOS concepts and do not (yet) have any hierarchical relationship to concepts from the vocabulary. Also, one `skos:ConceptScheme` is defined in Reegle by using an URN. Furthermore, two resources are specified with the schema part of the URI being “info”. These resources serve an unknown purpose since they do not have any type assigned but state a contributor name, a version, and a modification date.

### 5.2.5 Adherence to SKOS Integrity Conditions

The SKOS integrity conditions S14, S13, S27, and S46 correspond to the *qSKOS* quality issues for *Inconsistent Preferred Labels*, *Disjoint Labels Violation*, *Relation Clashes*, and *Mapping Clashes*, respectively (cf. Section 3.1.4). Table 5.7 gives an overview of our findings of integrity condition violations as implemented by *qSKOS*. We found that 18 of the 24 Web vocabularies we checked were affected by at least one issue and six vocabularies (Eurovoc, NYTL, IPTC, NYTP, UNESCO, and Plant) stand out by not violating any of the integrity conditions.

#### 5.2.5.1 Relation Clashes

*Relation Clashes* occur in 13 of the 24 reviewed vocabularies. We could observe that the associative relations span various hierarchy levels. For LVAk and PXV the maximum level is one, i.e., concepts that are connected by `skos:related` are also directly connected by `skos:broader`. However, there are also occurrences over multiple levels that are harder to spot like those we observed in Reegle and IPSV, spanning three or four hierarchy levels. The highest number of hierarchy levels that were connected by associative relations were found in SNOMED (7), RAMEAU (26), and DBpedia (38).

#### 5.2.5.2 Mapping Clashes

*qSKOS* could find *Mapping Clashes* only in the Reegle vocabulary, where two clashes could be detected. They were caused by mappings to GEMET and DBpedia.

#### 5.2.5.3 Inconsistent Preferred Labels

*Inconsistent Preferred Labels* could be found only in 5 out of the 24 reviewed vocabularies. A reason could be that this issue is stated as an integrity condition in the SKOS reference and is also covered by thesaurus guidelines [NIS05, Hed10] in a similar way. Thus, vocabulary developers might already check their vocabularies against it.

	Relation Clashes	Mapping Clashes	Inconsistent Preferred Labels	Disjoint Labels Violation
ODT	0	0	0	1
GeoNames	0	0	1	0
Reegle	317	2	0	3
PXV	2	0	0	4
NYTL	0	0	0	0
SSW	4	0	0	16
IPTC	0	0	0	0
UNESCO	0	0	0	0
Plant	0	0	0	0
IPSV	5	0	0	21
NYTP	0	0	0	0
GEMET	2	0	0	3
STW	5	0	214	0
Eurovoc	0	0	0	0
LVak	1	0	0	0
EARTh	61	0	0	69
UMBEL	0	0	2	1
AGROVOC	1	0	0	2424
SNOMED	1234	0	0	202
GTAA	37	0	0	0
RAMEAU	337	0	0	33066
DDC	0	0	1	0
LCSH	0	0	669	206
DBpedia	10219	0	0	0

TABLE 5.7: Validation results using *qSKOS*, Part 4: *SKOS Consistency Issues*.

UMBEL has two inconsistently labeled resources which may be caused by a misunderstanding of the `skos:prefLabel` usage because one of the labels is a longer narrative description of the concept that might be better expressed by using one of the `skos:note` properties.

The only occurrence of this issue in GeoNames is caused by inconsistent usage of upper/lowercase in `skos:prefLabel` literals: one concept has both the labels “language school” and “Language School”.

All inconsistently labeled resources of STW are resources from DBpedia that are assigned multiple German preferred labels within the STW vocabulary. For example, the resource

dbpedia:Agritourism has two labels, “Turismo rural”@de and “Agrotourismus”@de.

Inconsistent labels also occur in a greater quantity in LCSH, mostly with minor differences in labeling. The same concept is, e.g., labeled with the preferred labels “Nation-state–Congresses”, “National state–Congresses” and “National-state–Congresses”.

#### 5.2.5.4 Disjoint Labels Violation

Compared to the total numbers of concepts in the vocabularies, *Disjoint Labels Violations* seem to be a minor issue that is already handled well by the vocabulary developers. A higher number of occurrences of this issue can be found in RAMEAU (over 30 000) and AGROVOC (over 2 400). All other vocabularies show up to approximately 200 occurrences which is an amount that can be handled by manual correction(s).

ODT and UMBEL each have one concept labeled identically as `skos:prefLabel` and `skos:altLabel`. The same pattern can be observed with EARTH, SNOMED, AGROVOC, and RAMEAU which also do not make use of hidden labels.

### 5.3 Online Vocabulary Checker

In this section, we present our findings from analyzing the quality reports generated by the *online SKOS Quality Checker* described in Section 4.2.4. The reports were generated by *qSKOS* version 1.2.2 and cover all vocabularies uploaded since launching the service in December 2013 and October 14<sup>th</sup> 2014. By analyzing the reports we seek to answer the following questions:

- What quality issues are most commonly observed?
- Do users upload multiple updated (improved) versions of their vocabularies and, if yes,
- In what way do these versions differ from a quality perspective?

#### 5.3.1 Methodology

For this case study we took into account the vocabularies uploaded to the service as well as the generated reports that were provided to the users after vocabulary analysis was done. Since we aimed to get insights on how controlled vocabulary quality checking services are accepted and used, we intentionally did not instruct or encourage users to

revise their uploaded vocabularies based on the received quality report and resubmit them again.

It turned out that not all reports generated by the *online SKOS Quality Checker* were usable. We only took into account all non-empty reports of uploaded Web vocabularies that did not contain the extension “.xml” in their filename. This was necessary due to a limitation of the RDF parser built into the deployed version of *qSKOS*. It rendered *qSKOS* unable to correctly detect the vocabulary’s serialization format and thus produced erroneous reports. However, among the total 470 reports generated by the *online SKOS Quality Checker* we were able to identify 287 suitable for analysis.

In order to detect changes in the reports of different uploaded versions of the same vocabularies, we had to first identify the corresponding versions of the generated reports. This was accomplished by analyzing the report filenames as stored by the *online SKOS Quality Checker*. They consist of a base name which is identical to the filename of the uploaded vocabulary as well as an identification number and a timestamp. It was therefore possible to identify two or more reports for one vocabulary that reflected the analysis results of the various uploaded versions. For the findings in this section we only take into account the first and the last version of the quality reports. From this data, which we refer to as vocabulary (report) pairs in the following, it was possible to track changes in quality issue occurrences, i.e., to identify what issues were changed (improved, degraded) or stayed constant. From the 287 usable quality reports we were able to identify in total 37 of such vocabulary pairs.

Between the time we launched the *online SKOS Quality Checker* service and October 14<sup>th</sup> 2014, 118 users logged in, 99 used their Google credentials to log in, 10 LinkedIn and 9 Twitter. In total, 354 files were submitted for quality checking and 287 were actually processable by *qSKOS*. The size of the analyzed uploaded files ranged from 108 Bytes (a partial file with just rdf/xml header) to 48 Megabytes. Some of them (58) contained no `skos:Concepts` at all, the largest files made assertions for up to 82 722 concepts (76 249 and 32 035 in the second and third largest files). Also, the use of `skos:ConceptSchemes` varied. The file using them most extensively defined 2 196 `skos:ConceptSchemes` whereas 63 files do not make use of `skos:ConceptSchemes`.

### 5.3.2 Overall Issue Occurrences

Table 5.8 shows the different kinds of quality issues that were detected by analyzing the generated reports of 287 uploaded Web vocabularies. Overall, we observed 23 kinds of quality issues. Version 1.2.2 of *qSKOS* supports in total 26 quality issues, however, for performance reasons the *online SKOS Quality Checker* does not check for *Broken*

Quality Issue	Number of Vocabularies
Unidirectionally Related Concepts	157
Missing Out-Links	156
Undocumented Concepts	143
Disconnected Concept Clusters	116
No Common Languages	100
Orphan Concepts	90
Omitted Top Concepts	74
Overlapping Labels	72
Omitted or Invalid Language Tags	68
Hierarchical Redundancy	56
Missing Labels	54
Valueless Associative Relations	45
Disjoint Labels Violation	23
Top Concepts Having Broader Concepts	22
Incomplete Language Coverage	19
Relation Clashes	19
Cyclic Hierarchical Relations	14
Undefined SKOS Resources	12
Solely Transitively Related Concepts	9
HTTP URI Scheme Violation	7
Mapping Relations Misuse	6
Inconsistent Preferred Labels	5
Empty Labels	4

TABLE 5.8: Detected quality issues and their presence in the total number of uploaded vocabularies.

*Links* and *Missing Incoming Links*. *Mapping Clashes* was the only quality issue that was checked by *qSKOS* but could never be observed in any of the uploaded vocabularies.

Each of the three quality issues that occur most often is a member of a different issue category (as defined in Chapter 3), i.e., one of *Labeling and Documentation Issues*, *Structural Issues*, and *Linked Data Specific Issues*. The two most common quality issues, *Unidirectionally Related Concepts* and *Missing Outgoing Links*, occur in 2/3 of the uploaded vocabularies. The four quality issues that occurred least often were spotted in less than three percent of the uploaded vocabularies.

### 5.3.3 Changes in Quality Issue Occurrences

Focusing on the 37 identified vocabulary pairs, we found that in 24 vocabularies at least one quality issue was improved, i.e., we spotted less occurrences of the quality issue in the first uploaded version than in the later version. In 13 vocabularies at least one quality issue degraded, i.e., the number of occurrences of that kind of issue increased in the latest uploaded version. We also observed that many vocabularies in which occurrences of some kinds of issues have decreased, at the same time experienced increasing occurrences of

other issues, which means that for these issues the quality degraded. We show these changes per vocabulary in detail in Figure 5.3.

Quality Issue	First Version	Latest Version	Difference
Omitted or Invalid Language Tags	13	5	8
Omitted Top Concepts	13	5	8
Undocumented Concepts	22	17	5
Missing Labels	10	5	5
No Common Languages	16	11	5
Unidirectionally Related Concepts	19	14	5
Missing Out-Links	23	20	3
Disconnected Concept Clusters	16	14	2
Orphan Concepts	15	13	2
Overlapping Labels	12	11	1
Valueless Associative Relations	6	5	1
Cyclic Hierarchical Relations	3	2	1
HTTP URI Scheme Violation	1	0	1
Hierarchical Redundancy	9	8	1
Solely Transitively Related Concepts	2	1	1
Top Concepts Having Broader Concepts	5	5	0
Empty Labels	1	1	0
Mapping Relations Misuse	1	1	0
Relation Clashes	2	2	0
Disjoint Labels Violation	3	3	0
Incomplete Language Coverage	2	3	-1

TABLE 5.9: Detected quality issues and their presence in the number of vocabularies of both first and latest uploaded version.

Table 5.9 lists all kinds of quality issues together with the number of vocabularies they occurred in, separated into the first and latest uploaded version. For example, the quality issue *Cyclic Hierarchical Relations* occurred in three vocabularies in their first uploaded version but only in the latest uploaded version of two vocabularies. We can observe that two issues (*Undefined SKOS Resources* and *Inconsistent Preferred Labels*) were found when analyzing all quality reports but did not occur in our analysis of vocabulary pairs. In total, we found 21 kinds of quality issues in the pair analysis.

Figure 5.1 shows the changes in occurrences of different kinds of quality issues in a scatterplot. Each kind of quality issue is represented by a dot. Depending on the coordinates of the dots we can see the number of vocabularies in which the quality issue was observed in the first (x-axis) and the last version (y-axis). All quality issues represented by dots positioned on the diagonal line stayed constant, i.e., their number of occurrences did not change between the versions. Thus, we can see that in total five quality issues did not change (not taking into account the three issues that were never observed). One issue, *Incomplete Language Coverage*, was observed in the first version of two vocabularies but in the latest version of three vocabularies. This issue was the only one whose occurrence increased between consecutive uploaded versions. All other issues



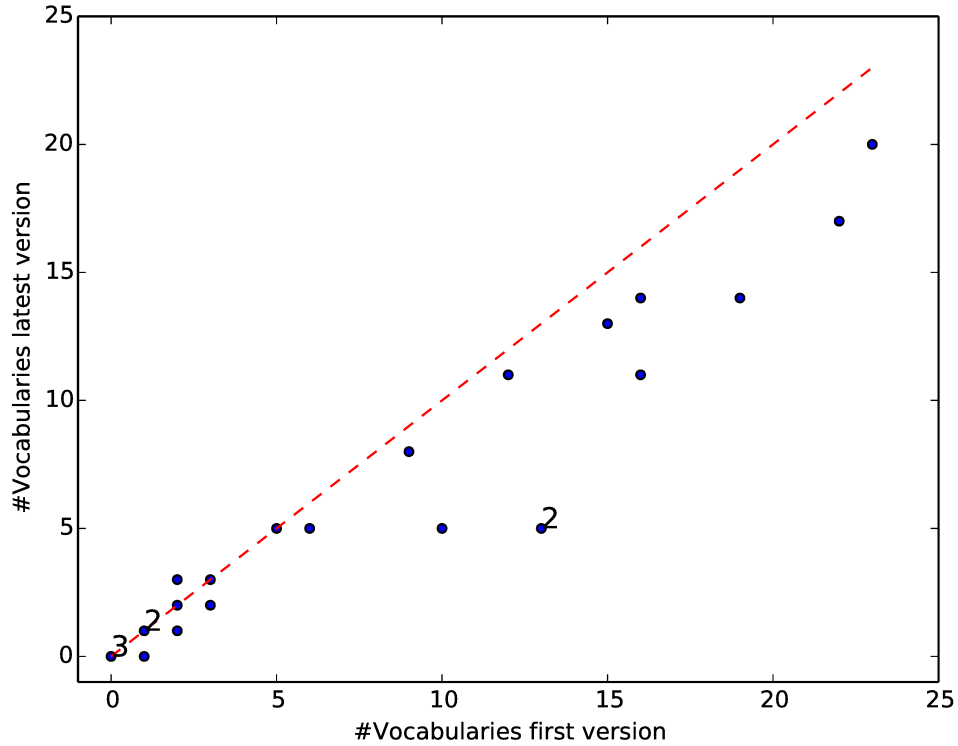


FIGURE 5.1: Number of vocabularies affected by quality issues in the first and last uploaded version.

(15 in total) are located below the diagonal line which means that less vocabularies in their latest version were affected by the quality issue than in their first version. The two issues that were improved in the most vocabulary pairs (8) are *Omitted or Invalid Language Tags* and *Omitted Top Concepts*.

Detailed numbers on the type and amount of changes of each quality issue are visualized in Figure 5.2. We can see that all but one of the 21 observed quality issues in the analyzed vocabulary pairs were improved in at least one vocabulary. Some quality issues were improved in up to seven vocabularies, however for 16 of the 21 quality issues we could also detect degradations in up to five vocabularies which means that these quality issues were detected more often in the latest version of the vocabulary than in the first uploaded version.

Figure 5.3 shows the amount of quality issue changes for each vocabulary pair. On the x-axis we provide the abbreviated base name of the uploaded vocabulary and on the y-axis the number of affected quality issues. We can see that from the 37 vocabulary pairs 24 were improved in up to 11 quality issues. However, also 13 vocabularies degraded in

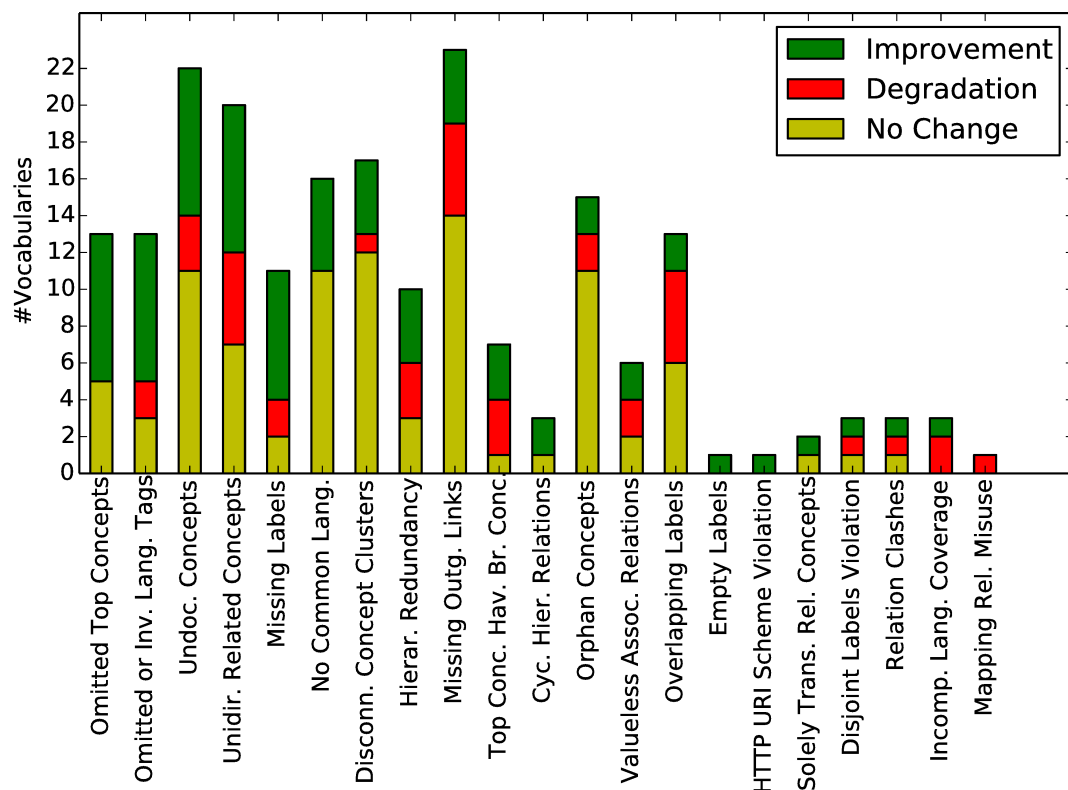


FIGURE 5.2: Quality changes by issue.

up to 12 quality issues and the number of occurring quality issues did not change in six vocabularies.

### 5.3.4 Degradations and Unchanged Vocabularies

For two vocabularies that show the highest number of degradations of quality issues we can observe that (i) they have only been uploaded two times and (ii) several months have passed between these two uploads. The initial version of vocabulary “yso-skos.ttl” was uploaded on April 22<sup>nd</sup> 2014 and the latest version was submitted on October 7<sup>th</sup> 2014. Although three issues, *Empty Labels*, *Incomplete Language Coverage* and *Disconnected Concept Clusters* were improved, twelve issues degraded. We observed that in the latest version some concepts were introduced that were described by a `skos:scopeNote` as “deprecated on” followed by a date. These literals have no language tag specified, and are thus responsible for the degradation of the issue *Omitted or Invalid Language Tags*. Also, e.g., the issue *Orphan Concepts* degraded, caused by newly introduced deprecated concepts that are not linked anymore by a `skos:semanticRelation`. The degraded issue *Relation Clashes* also involves deprecated concepts in 6 of the 8 occurrences.

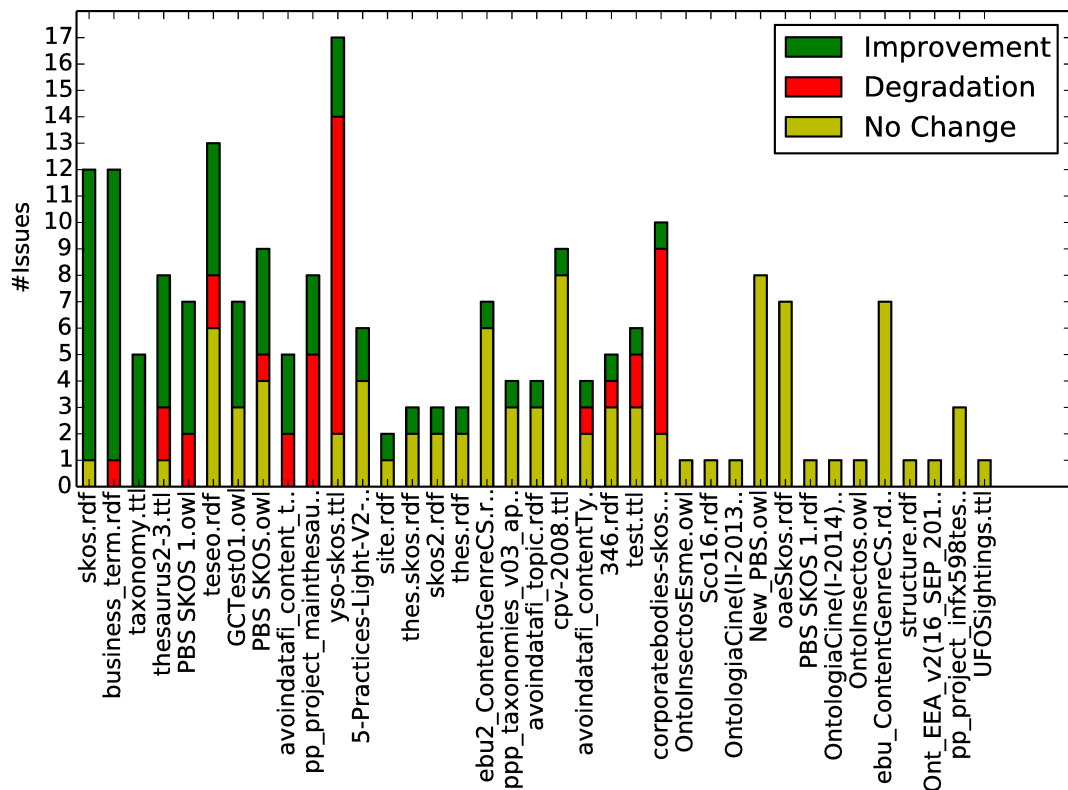


FIGURE 5.3: Quality changes by Web vocabulary.

Also the upload dates of the two versions of the vocabulary “corporatebodies” differ significantly (January 15<sup>th</sup> 2014 vs. July 31<sup>st</sup> 2014). The increase of, e.g., the quality issue *Incomplete Language Coverage* is caused by the introduction of new concepts that have assigned only English literals and omit the other languages (about 25) used for describing most of the remaining concepts. These concepts are furthermore marked as deprecated and, as it was the case with the “yso-skos.ttl” vocabulary, are partly responsible for the increase of occurrences of other quality issues, e.g., *Disconnected Concept Clusters* and *Unidirectionally Related Concepts*. Newly introduced concepts in the latest uploaded version have also caused an increase of *Orphan Concepts*.

From the files where we could observe no changes in quality issue occurrences, five were no Web vocabularies, i.e., they did not contain any SKOS resources. In two cases we observed problems with the SKOS namespace. In one file the SKOS namespace was an invalid HTTP URL (<http://www.w3.org/2004/02/skos/core#>) and in another one the SKOS namespace was defined as <http://www.w3.org/2008/05/skos#> instead of <http://www.w3.org/2004/02/skos/core#>.

### 5.3.5 Service Usage

Table 5.10 shows the upload count of each Web vocabulary that has been uploaded at least two times. We can generally observe that vocabularies that feature the most issue improvements according to Figure 5.3 were uploaded most often. An exception is “PBS SKOS 1.rdf” that has been uploaded 14 times but never improved because *qSKOS* was not able to detect SKOS concepts due to an erroneous SKOS namespace declaration in the file.

We can assume that the *online SKOS Quality Checker* service was used in some cases to iteratively improve the uploaded Web vocabularies. For example, in the file “skos.rdf” 11 quality issues were improved between the first and the latest uploaded version. It was uploaded 18 times in total with a timespan of 25 minutes between the first and the second upload. The following uploads were performed at a frequency of 11 minutes at maximum. A similar usage of our service was observed with the vocabulary file “business\_term.rdf” which was initially uploaded and analyzed on April 3<sup>rd</sup> 2014 with multiple further uploads on the following days.

## 5.4 Automated Quality Checking in a Teaching Context

In university courses in which thesaurus construction in general or more specifically SKOS is taught, it is hard to make students aware of problems that arise when working on large real-world thesauri. Often students make exercises only with small vocabularies. Many of the typical problems arise when a thesaurus becomes larger and when it is not possible anymore to view the complete thesaurus structure at a glance. For the course teacher it will, on the other hand, become difficult and time consuming to identify all issues in a number of non-trivial thesauri.

Therefore we believe that integration of automated quality checks in the thesaurus development process can (i) help students in a teaching environment to increase awareness for common quality problems and (ii) help developers to improve the quality of their thesauri.

In the following, we report on the results of a case study performed in the spring term 2013 at the University of Applied Sciences in Hannover. Students with a background on thesaurus construction were required to develop thesauri covering businesses in different economic sectors. After a first development iteration, an automatically generated quality report was created and the results were handed out to the students. After a second development iteration, another quality report was generated. We compared the results

Vocabulary Name	Number of Uploads
PBS SKOS 1.owl	32
skos.rdf	18
business_term.rdf	16
PBS SKOS 1.rdf	14
GCTest01.owl	12
taxonomy.ttl	9
avoindatafi_content_type.rdf	8
PBS SKOS.owl	7
skos2.rdf	5
346.rdf	5
New_PBS.owl	4
avoindatafi_topic.rdf	4
site.rdf	3
OntologiaCine(II-2013).owl	3
OntologiaCine(I-2014).owl	3
ebu2_ContentGenreCS.rdf	3
OntoInsectos.owl	3
structure.rdf	3
ppp_taxonomies_v03_application.rdf	3
pp_project_mainthesauruseschemeversion.ttl	3
teseo.rdf	3
cpv-2008.ttl	3
OntoInsectosEsme.owl	2
5-Practices-Light-V2-18mai-xls.ttl	2
Sco16.rdf	2
avoindatafi_contentType.rdf	2
thes.skos.rdf	2
yso-skos.ttl	2
oaeSkos.rdf	2
thesaurus2-3.ttl	2
thes.rdf	2
ebu_ContentGenreCS.rdf	2
Ont_EEA_v2(16_SEP_2013).owl	2
corporatebodies-skos.rdf	2
test.ttl	2
pp_project_infx598test.ttl	2
UFOSightings.ttl	2

TABLE 5.10: Upload count of each vocabulary.

of these reports and performed an in-depth analysis to find out about the quality issues that occurred and if and how they were addressed in the subsequent vocabulary version.

#### 5.4.1 Methodology

We used version 0.9.5 of *qSKOS* which checks for 21 potential quality issues. Checking for missing incoming links has been omitted in this study because the created vocabularies were not published online, leading to 20 checked issues. All submitted vocabularies and

the generated reports can be retrieved online<sup>16</sup>. In total we analyzed 26 vocabularies, 13 for each submission containing between 43 and 111 concepts.

#### 5.4.1.1 Data Acquisition and Studied Vocabularies

Construction of a SKOS vocabulary was an obligatory part of both the bachelor's and the master's course. For the bachelor students this was a classroom exercise, that was done in small groups of two or three students. The master students constructed a thesaurus as an individual exercise as part of the examination.

Students from the master course had to select an economic sector and find websites of 20 companies in this sector. All selected companies are located in Germany and have a German Web site. In the next step we collected characteristic words from these websites. These lists of words should then form the base for the construction of a thesaurus. Students had the freedom to remove irrelevant words from the list and add important missing terms. Table 5.11 lists all chosen domains alongside with their German reference that will be used throughout this paper.

Domain	Reference
Fashion and Clothing Industry	Bekleidung
Pharmacy	Pharma
Goat Farms	Ziegenhof
Library Information Systems	Bibliotheksoftware
Wind Energy	Wind
Mechanical Engineering	Maschinenbau
Shipbuilding	Werften
Orthopedic Technology	Orthopaedie
Paper and Board Industry	Papierindustrie
Medical Technology	Medtechnik

TABLE 5.11: Thesaurus domains with German references.

We used crawler4j<sup>17</sup> to crawl the websites. For a few companies the crawling was not successful and no pages could be retrieved. Since some companies have a site with a high number of pages, we limited the number of pages to be retrieved to 120. The limitation serves the practical goal of keeping the size of the corpus moderate, but also has more fundamental reasons: We expect that even a large company should be described rather well on the first two levels of a web site. If we crawl in a breadth first way, as we do, we might expect that at some point we have seen the core information of a company. If more pages follow, we might get more and more specific information on detailed topics, that even could obscure the more important and central information. The limit of 120

<sup>16</sup>Vocabularies and quality reports created in the study: <http://tinyurl.com/mv8vocs>. Retrieved 2015-06-23.

<sup>17</sup>Crawler4j project website: <http://code.google.com/p/crawler4j/>. Retrieved 2015-06-23.

is rather arbitrary and turned out to be a size that allows us for almost all companies in our list to crawl the complete site. In total, 14 673 pages were retrieved, which averages to 70.5 pages per company, with a total amount of about 4.6 million words. Almost the same corpus was used for the keyword extraction experiments described in [WGA13].

We did not do any boiler-plate removal since it turned out that in many cases relevant and interesting words would be removed. For example, a list of products or departments is often given as a menu that might be removed. The whole corpus is tokenized and all words are lemmatized and tagged with their part of speech by the TreeTagger [Sch95].

Each of the sub-corpora, i.e., the texts collected from the websites of the companies belonging to one economic sector, was used to construct a list of domain-specific terms. As candidates for the thesaurus, all words are selected that (i) are tagged as common noun, (ii) occur at least 5 times in the sub-corpus, and (iii) have a relative frequency that is higher than the frequency in the general DeWaC [BBFZ09] corpus.

Due to the fact that we did not use boiler-plate removal, there are a lot of single words that are not part of a well formed sentence. Consequently a lot of errors are made by the part-of-speech tagger and a number of words that are not common nouns end up in the lists of term candidates. Moreover, many words are included, that are not typical for the economic sector of the sub-corpus, but for website texts. This is mainly a consequence of the completely different strategies for collecting texts for our corpus and for the DeWaC reference corpus. Despite these errors, each of the lists contained enough relevant words that could serve as a basis for a domain specific thesaurus.

#### 5.4.1.2 Vocabulary Construction

After completion of the list of candidate terms each of the master students could work on the thesaurus on the chosen economic sector. Students were instructed to use all relevant words from the generated list, where judgment of relevance was left to the students own opinion. Moreover they were allowed to add a limited number of terms, if these terms are necessary to construct the thesaurus. Two criteria were mentioned explicitly: (i) a term might be necessary to represent an otherwise missing intermediate level or natural more general term to a number of more specific terms, and (ii) if otherwise a more general term would only have just one daughter. Finally, they were instructed to find matching terms in another thesaurus for at least 10 terms.

Since TopBraid Composer<sup>18</sup> was used throughout the course, all students used this tool for the thesaurus construction as well. In order to check the quality of the thesaurus

---

<sup>18</sup>TopBraid Composer Standard Edition: <http://www.topquadrant.com/tools/modeling-topbraid-composer-standard-edition/>. Retrieved 2015-06-23.

they could send it once to the course teacher in order to get the *qSKOS* report. The differences between this first submission and the thesaurus that they finally submitted as a part of their assignment is used below to get insight in the value of *qSKOS*.

The bachelor students were given the same instructions, but without the requirement to find matching terms in other vocabularies. These students worked in small groups of two or three students on two afternoons in a classroom setting. The lists of words used by these students were the same as those used by the master students. Only three thesauri of this course are included in this study. Some thesauri could not be used since the students have used *qSKOS* as a tool to continuously improve the thesaurus quality. Furthermore, some thesauri did not achieve a level of maturity that allows for a useful application of *qSKOS*.

#### 5.4.2 *qSKOS* Quality Analysis Results

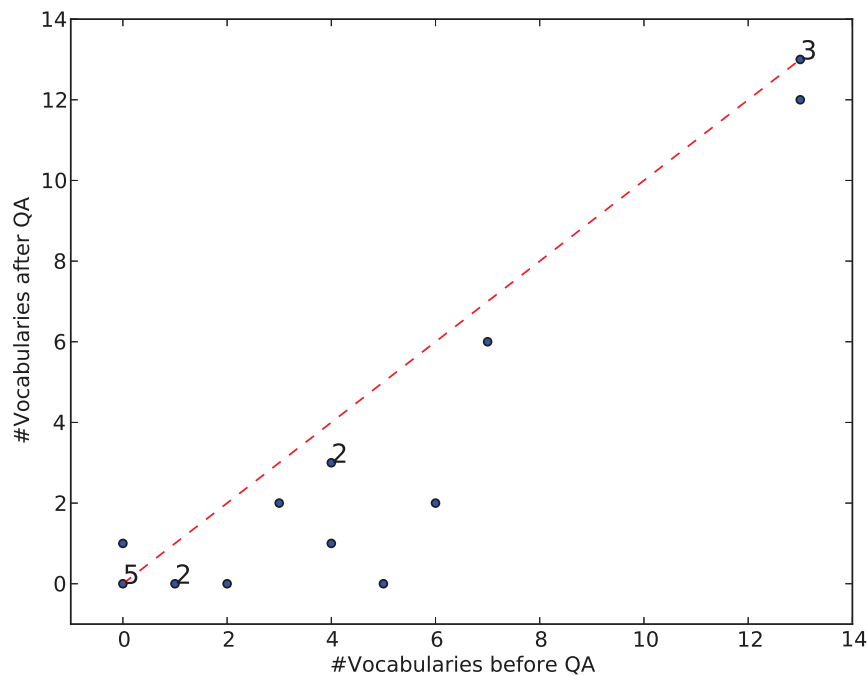


FIGURE 5.4: Number of vocabularies affected by quality issues before and after QA.

We counted the number of vocabularies that show a specific quality issue before *qSKOS* quality assessment (QA) and afterwards. Figure 5.4 shows that 11 of all 20 assessed quality issues lie in the right side of the dotted line, i.e., after the quality check less vocabularies were affected by these issues than before. Eight issues either did not occur in any vocabulary or occurred in all vocabularies and did not improve after QA. One issue



Quality Issue	Before QA	After QA	Difference
Orphan Concepts	5	0	5
Valueless Associative Relations	6	2	4
Relation Clashes	4	1	3
Inconsistent Preferred Labels	2	0	2
Omitted or Invalid Language Tags	3	2	1
Incomplete Language Coverage	7	6	1
Undocumented Concepts	13	12	1
Overlapping Labels	4	3	1
Disconnected Concept Clusters	4	3	1
Mapping Clashes	1	0	1
Disjoint Labels Violation	1	0	1
Solely Transitively Related Concepts	0	0	0
Omitted Top Concepts	0	0	0
Top Concepts Having Broader Concepts	0	0	0
Missing Out-Links	13	13	0
Broken Links	13	13	0
Undefined SKOS Resources	0	0	0
Unidirectionally Related Concepts	13	13	0
HTTP URI Scheme Violation	0	0	0
Cyclic Hierarchical Relations	0	1	-1

TABLE 5.12: Occurrences of quality issues before and after QA.

is positioned at the left side of the line, because it did not occur in any vocabulary before QA but showed up in one vocabulary afterwards. Table 5.12 shows issue occurrences in more detail.

To get a more detailed impression of the way the vocabularies were influenced by each quality issue, we calculated the issue occurrences for each vocabulary before and after QA. Figure 5.5 shows the number of vocabularies where less (improvements), more (degradation) or equal (no change) issue occurrences were spotted in the revised version.

In the following we elaborate on the quality issue changes in detail. Our findings can be summarized as follows:

- 13 of 20 checked quality issues were improved in at least one vocabulary.
- Seven quality checks led to improvements in up to five vocabularies.
- For eight other quality checks, besides improvements in up to seven vocabularies also degradations in up to four vocabularies were found.

#### 5.4.2.1 Improvements without Degradations

*Orphan Concepts* were resolved in all five vocabularies where they occurred. The affected concepts were either removed from the vocabularies or hierarchically related to existing

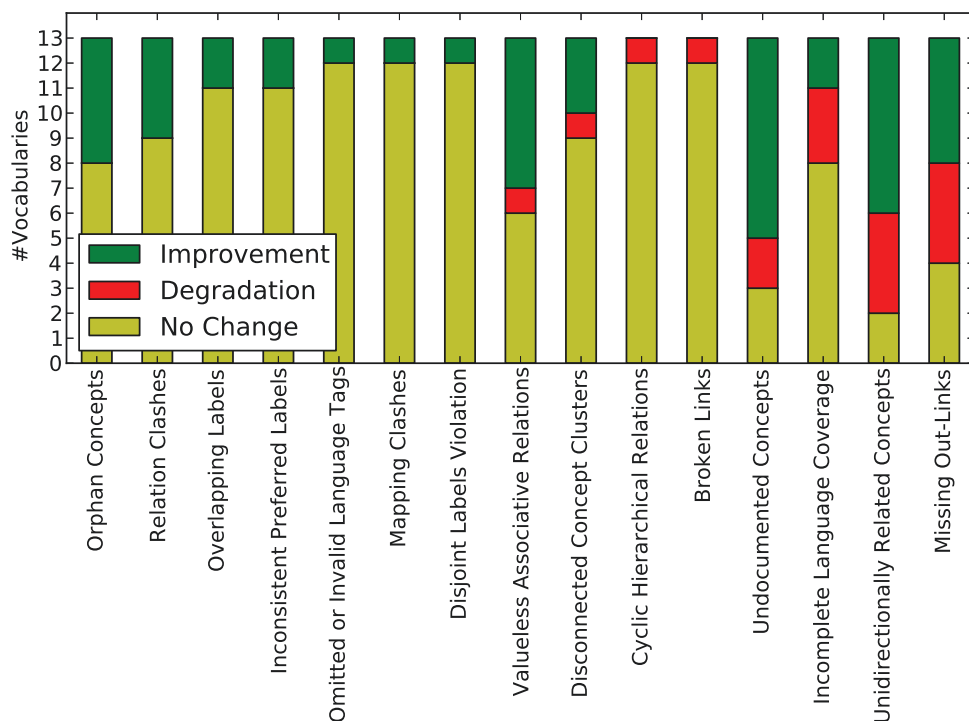


FIGURE 5.5: Quality changes by issue.

concepts. In some cases (Pharma) concepts were merged in a way that they became alternative labels of other concepts and were removed from the vocabulary afterwards.

*Relation Clashes* were resolved by the contributors for four vocabularies. In three vocabularies all occurrences of this issue were fixed and for one vocabulary (Medtechnik) the occurrences could be reduced from 25 to six. The applied resolution strategies were to remove the associative relations (Maschinenbau, Wind), change them to reference other concepts (Wind2), or replace them with an hierarchical relation (Wind2). Some clashes (Medtechnik) were resolved by changing the hierarchical structure of the affected concepts. However, we assume that these substantial changes of the Medtechnik vocabulary led to the introduction of the remaining six occurrences of this issue.

Occurrences of *Overlapping Labels* could be observed in four vocabularies and were improved in two of them. For one vocabulary (Bibliotheksoftware), all four occurrences were addressed by rephrasing preferred labels and removing alternative labels. The other improved vocabulary (Papierindustrie) initially showed conflicts between preferred and alternative labels of two concepts but only the latter was addressed in the subsequent version.

*Inconsistent Preferred Labels* occurred in two vocabularies and were fixed in both of them by either removing or rephrasing conflicting labels or by conversion to alternative labels.

Three vocabularies were affected by *Omitted or Invalid Language Tags*, but two vocabularies showed no change and in only one vocabulary (Orthopaedie) this issue was improved. In this vocabulary, all issue occurrences were fixed by adding language tags to the three `rdfs:labels` where they were missing.

*Mapping Clashes* were observed in only one vocabulary (Wind2) with one concept which was mapped by both `skos:exactMatch` and `skos:broadMatch` to the same “external” resource. The issue was resolved by removing the latter. The same vocabulary was the only one which showed *Disjoint Labels Violations* for one concept, which were fixed by rephrasing the preferred label.

#### 5.4.2.2 Improvements with Degradations

*Valueless Associative Relations* were observed in six vocabularies before QA. In five vocabularies, all occurrences were fixed, in one vocabulary (Medtechnik) all but one occurrence was fixed. This unfixed issue did not occur in the initial version of the vocabulary, thus it has been introduced by changes in the hierarchical structure of the improved vocabulary. One vocabulary (Wind2) in the first version did not show any valueless associative relations, but one such relation was introduced by addition of an hierarchical relation.

*Disconnected Concept Clusters* were observed in four vocabularies before QA and improved in three of them. After the quality check the number of disconnected concept clusters were reduced to one “giant component” in two vocabularies. In one vocabulary (Pharma) the number of disconnected concept clusters decreased substantially from 18 to five. However, one vocabulary introduced an additional disconnected concept clusters after the check that was not present before. This cluster defines an hierarchical branch of materials and consists of eight of the 12 new concepts that have been newly introduced after QA.

None of the vocabularies showed *Cyclic Hierarchical Relations* before initial QA. However, in one vocabulary (Bekleidung) one cycle was identified in the final version. It seems that the introduction of these issues was a side-effect when reducing the number of disconnected concept clusters from three to one. One concept in a cluster of only two concepts was hierarchically reorganized which caused the cycle.

In each of the created vocabularies we were able to spot at least one *Broken Link*. They were caused by the XML root namespace definition (set to, e.g., <http://hs-hannover.de/maschinenbau#>) for the created concepts. These links did not resolve because the vocabularies were not published online. One vocabulary (Papierindustrie) contained

three external links to DBpedia and AGROVOC which returned an HTTP status other than 200. After the check, another link to DBpedia was introduced that also did not resolve.

*Undocumented Concepts* occurred in all vocabularies and generally changed very few after QA. The changing numbers are mainly caused by concepts that were removed or newly added. However, in four vocabularies we actually noted intentional manual additions of `skos:scopeNotes` and `skos:definitions`. In one vocabulary (Maschinenbau2) all but four concepts were undocumented and these undocumented concepts were fixed after QA with `skos:definitions`.

*Incomplete Language Coverage* was spotted in seven vocabularies by QA and improved in two vocabularies in the final version. In one vocabulary (Maschinenbau2), two English labels were provided for concept that only had German labels. Another vocabulary (Werften) initially contained 80 concepts with German labels. Only two concepts were labeled in English. In the final version, the English alternative label was removed and the language tag of the preferred label “Cruises” was changed from `@en` to `@de`. These changes can actually be considered a degradation because correct information in the original vocabulary was removed and changed to be incorrect. In three other vocabularies, minor degradations could be observed that were caused by additions and removals of new concepts and labels as it was also the case for the issue of *Undocumented Concepts*.

*Unidirectionally Related Concepts* were explicitly not required to be treated by the experiment contributors. Thus, for all vocabularies that showed issues of this kind before the quality check, they remained nearly constant with changes only being indirectly caused by concept additions and removals.

Improvements for *Missing Out-Links* could be observed in five vocabularies. In three of them (Wind2, Werften, Pharma), the experiment contributors deliberately introduced mapping relations to external resources on the web. Removals and additions of concepts led to the side-effect of improvements of this issue in two vocabularies and degradations for another two vocabularies. For some reason in two vocabularies (Bekleidung2, Medtechnik) seemingly correct mapping relations were removed by the participants.

## 5.5 Summary

In this chapter, we performed a survey for receiving expert feedback among developers and users of controlled (Web) vocabularies. They provided us with valuable information on the relevance of the quality issues we identified and show directions of future research

and improvements. Many of the findings and suggestions from the survey were used to improve the catalog of quality issues as well as their implementation in *qSKOS*.

We performed an in-depth analysis of currently available Web vocabularies, carving out in detail what kind of quality issues occur, to what extent they occur, and elaborated on possible causes.

To complement our analysis of already published Web vocabularies, we studied the vocabularies submitted to the *online SKOS Quality Checker*. These are mostly vocabularies still in development or in the review process, some of them not yet published, incomplete or experimental. However, we found out that the service is actively used by the Linked Data community and can be helpful during the development process to reduce the number of occurring quality issues.

In a case study with students educated in the creation of controlled Web vocabularies we were able to gain insights on how automated quality assessment could be integrated into the vocabulary development process and how it can support contributors to controlled vocabularies in achieving a higher level of quality.



## Chapter 6

# Conclusions and Future Work

In this chapter, we summarize the content and contributions of this thesis, discuss our findings from the previous section and draw conclusions on the impact of our work. We furthermore elaborate on the limitations of our contributions and provide an outlook on our planned future work.

### 6.1 Summary

In this thesis we first introduced controlled vocabularies and provided an historical outline to help obtaining an understanding of their usefulness for certain problem domains. We introduced the various types of controlled vocabularies as found in existing literature and discussed their differences in content, intended usage, and structure. Based on these, we explained the role of controlled vocabularies published as Linked Data on the Web and defined the notion of “Web vocabularies” which are the main subject of discourse in this thesis. We introduced the notion of quality of Web vocabularies by discussing existing work on quality assurance for “traditional” controlled vocabularies and existing quality assurance approaches for linked datasets. We reviewed approaches that combine these two topics and found that these are often subjective or do not focus on the specific requirements of Web vocabularies. Furthermore, work that focused on automatically assessable quality measures for Web vocabularies was still underrepresented. Therefore, we defined the establishment of such measures and the integration of their assessment in controlled vocabulary development approaches as the problem space which we contribute to in this thesis.

We chose an approach that is based on a catalog of 29 quality issues, i.e., formally defined patterns in Web vocabularies that can potentially cause a quality problem. The

catalog is based on existing work, expert discussion, and review of currently available Web vocabularies. We formally described each quality issue so that it can be automatically evaluated. Because of the multiple other factors that influence Web vocabulary quality such as usage scenario, development methodology, or personal preferences, we decided that the final judgment on validity and correction of the found quality problems must be up to the vocabulary developers.

This is reflected in the two different implementation approaches which we contributed for checking occurrences of quality issues in Web vocabularies. The first method is designed to check a Web vocabulary as a whole, i.e., it applies each quality issue check from the catalog on a Web vocabulary that is provided as a file. The output of this implementation is a detailed quality report that states the found quality issues alongside with the affected resources that cause the quality issue. We integrated this approach into *PoolParty Thesaurus Server*, a commercial thesaurus development software product. The second method which we contributed focuses on on-change checking of Web vocabularies and targets situations where the vocabulary is currently edited by developers. We developed the approach in the course of the EU-funded LOD2 project and designed it to be used with the quality issue catalog introduced in this thesis. From a user perspective, the on-change checking approach notifies developers as soon as they introduce a potentially problematic change to the Web vocabulary that may cause a quality problem. From the feedback we received from customers and the Linked Data community, we know that both implementation approaches are used or adopted by third-party developers and research projects.

We performed a survey among experts in the field of vocabulary construction to find out about the usefulness of the quality issues we defined in our catalog. The survey contained both open- and closed-ended questions in order to allow us to refine the quality issues and learn about potential additional issues. As a result of the survey, we found that the identified quality issues are valid and of practical significance. Based on our findings, we were able to infer recommendations and best practices for the development of good quality Web vocabularies. In another case study we performed, we report to what extent the quality issues we identified in our catalog occur in existing vocabularies that are available on the Web. For this purpose we compiled a representative set of Web vocabularies, differing in, e.g., size, complexity, or covered domain. We used our catalog and implementation to provide a detailed quality analysis of these vocabularies. We found violations of each quality issue in at least one of the analyzed vocabularies and pointed out potential for future improvement. In another case study which we contributed, we assigned students with the task of creating a controlled vocabulary for a specific domain. We focused on studying differences in two versions of the same vocabularies: the first version was developed without automated quality assessment support,



whereas the second version has been revised by the students, based on the quality report which we produced with our implementation. We found that integrating quality assessment methods in the development process leads to less quality issues. Furthermore, we contributed an analysis of the data gathered from a Web frontend to our quality assessment implementation. This Web application provides a means for any interested user to upload her Web vocabularies and generate quality reports. Based on this data, we gave an overview on the quality issues that occur most often and if and how they have been addressed in subsequently checked versions of the same vocabulary. The data indicates that our quality analysis approach is used in practical settings by controlled vocabulary developers and helps in reducing the number of occurring quality issues.

## 6.2 Discussion

In this section we reflect on our findings from the case studies and discuss them in detail.

### 6.2.1 Expert Perception of Quality Issues

We reported the results of a survey we conducted to learn about how curators and users of Web vocabularies perceive vocabulary quality. We asked the participants to express their opinion and experience on quality issues we identified in our previous work. Our findings clearly reflect the subjective dimension of data quality and point out controversial approaches and opinions. It is therefore important to have tools that can automatically check against these controversially perceived quality issues. This way it is possible to find out, e.g., if two Web vocabularies “fit together”, i.e., the developers of each vocabulary follow similar approaches regarding vocabulary quality. However, from the responses we can conclude that existing tools could support taxonomists in producing higher quality vocabularies by providing semi-automated labeling, documentation, and relationship creation support. Based on the survey results and decision rationales from our participants, we gave recommendations on possible extensions and improvements of quality assessment tools. Thus, these tools could avoid and possibly fix quality problems in Web vocabularies.

### 6.2.2 Quality Analysis of Existing Vocabularies

To find out how to measure the quality of Web vocabularies, we reviewed existing literature, asked for feedback of Web vocabulary users and developers, and brought in our own experience in working with Web vocabularies. We came up with a catalog comprising

29 potential quality issues for SKOS vocabularies that can be assessed automatically. In order to get an impression if and to what extent these issues occur in currently published Web vocabularies, we computed them against a representative set of 24 vocabularies.

In this vocabulary analysis we found occurrences of potential quality issues in all of the reviewed Web vocabularies, which is in line with the findings of our earlier studies of SKOS vocabulary quality [MH11, MHI12]. The number of vocabularies that were affected by a specific quality issue varied widely. For example, each of the reviewed vocabularies had at least one undocumented concept while we found occurrences of *Mapping Clashes* in only one vocabulary.

A remarkable finding was that 18 of the 24 analyzed vocabularies were found to violate the SKOS integrity conditions. For example, we found that the SKOS integrity condition S27, which specifies that the `skos:related` relationship is disjoint with the `skos:broaderTransitive` relationship, is violated by the majority of the vocabularies we examined. In some cases, such as the complex hierarchies in LCSH, the invalid `skos:related` relationships bridge many levels of the concept hierarchy.

Such findings may not be surprising, considering that RDF data published online has been found to contain many errors in previous studies [DF06, HHP<sup>+</sup>10]. However, these studies did not look specifically at the validity of SKOS vocabularies or considered only a small number of OWL modeling issues [AMBS12a]. On the other hand, despite only some of them being defined in a strict formal way, the SKOS integrity conditions are part of the SKOS modeling scheme and vocabulary development applications should implement them as “baseline” checks. We therefore assume that our findings are caused by the current lack of tools providing these kinds of quality checks. As the tools developed in the course of this thesis implement evaluation of quality issues that go beyond basic SKOS consistency issues, we can consider them a valuable contribution to fill this gap and help decreasing occurrences of quality issues in Web vocabularies. *qSKOS* and *online SKOS Quality Checker* are so far the only available (online) tools that provide a broad range of quality issue checks that have been selected or developed specifically for the analysis of Web vocabulary quality.

### 6.2.3 Online Vocabulary Checker

From the number of uploaded Web vocabularies and generated reports we can witness a wide interest in the *online SKOS Quality Checker* tool by the controlled vocabulary development and Linked Data community. However, many of the uploaded vocabulary files were not usable because either the service was not used correctly (due to, e.g., an unsuitable file format or content) or because *qSKOS* had difficulties detecting the RDF

serialization format of files with the extension *.xml*. This was caused by the RDF parser (OpenRDF) internally used by *qSKOS*, which interpreted the content of these files as *TriX* format and not as *RDF/XML*. We fixed this behavior in version 1.4.7 of *qSKOS*. However, 287 uploaded Web vocabularies were valid and could be analyzed.

In the uploaded vocabularies we found occurrences of 23 quality issues we defined in our catalog. Some users of the service apparently used it to improve their vocabularies by uploading it multiple times during the development process. As the generated quality issues show, the developers were able to reduce the number of found quality issue with each iteration. 15 quality issues were fixed in at least one vocabulary that was uploaded in multiple versions. This shows that the identified and reported quality issues actually denote quality problems as perceived by the vocabulary developer and that iterative quality analysis on each vocabulary change improves the quality of Web vocabularies.

#### 6.2.4 Automated Quality Checking in a Teaching Context

Our study showed that quality issues were found in each of the vocabularies that have been created by the participants. However, not all quality issues that we checked against were also observed. Only 15 of 20 kinds of quality issues did occur in the vocabularies because of two reasons: Either some SKOS constructs like concept schemes or top concepts were not used by the participants or, in the case of *Undefined SKOS Resources* and *HTTP URI Scheme Violation*, the workflow of the used development tool (TopBraid Composer) prevented such kind of errors.

From the 15 occurring kinds of quality issues, 13 were reduced in at least one vocabulary after the quality check. However, for eight kinds of quality issues we also noted increased occurrences (i.e., degradations) which were mainly caused by side effects of other changes like addition or removal of concepts or changes in the hierarchical structure. These degradations could have probably been reduced by requiring the participants to perform a quality check before finally submitting the vocabulary.

The mentioned side-effects also show a weakness of our approach of counting the number of improvements for each quality issue: For *Unidirectionally Related Concepts*, we could observe a relatively high number of both improvements and degradations although participants were instructed not to take this issue into account. However, detailed examinations of the vocabularies and used issue resolution strategies show that the majority of all improvements were caused by the participants in explicitly resolving the respective quality issues.

In one case (Werften) changes were apparently introduced to achieve improvements for *Incomplete Language Coverage* although they actually cause a loss of information in favor of “better” issue occurrence values (i.e., a resulting value of zero for this quality issue). The reason for this kind of change could have been a misinterpretation of the experiment’s goal by the participants and concerns of achieving a lower grade if some issues are not completely fixed.

## 6.3 Limitations of Our Approach

Due to our focus on computable, data-oriented quality issues, we leave out more intellectual criteria, such as “appropriate specificity” of the vocabulary or the meaning of semantic relations. Thus, our findings and the automated corrections may be judged by some domain experts as inappropriate or even wrong for a specific usage scenario or requirement. We believe that our approach reveals its full potential in assisting human experts in their vocabulary development tasks, comparable to source code checks performed in integrated development environments for programming languages or as spell checkers do in word processors.

### 6.3.1 Catalog of Quality Issues and On-demand Vocabulary Quality Checking

Our catalog of quality issues and the tools we developed analyze the vocabularies as isolated entities. In reality, a controlled vocabulary is most often used in conjunction with other resources, e.g., a corpus of documents that is indexed with terms from the vocabulary. On the Web, however, it is rarely possible to evaluate the vocabularies in relation with their associated corpus, because it is not available for download or does not (only) cover digital objects. We therefore currently focus on “intrinsic” quality of Web vocabularies and omit corpora-related quality issues.

Furthermore, the implementation of evaluating *Missing In-links* and *Broken Links* shows poor performance for most Web vocabularies as they rely on dereferencing and querying external resources. Therefore, special care must be taken when integrating checks against these quality issues into vocabulary development environments as they may block user interaction or increase network load.

### 6.3.2 rsine - On-change Vocabulary Quality Checking

*rsine* is designed with the goal in mind to subscribe for notifications without (or only minimally) modifying source code of existing applications that process data in RDF format such as the LOD2 stack components *PoolParty* and *Pebbles*. Therefore our approach is based on monitoring all triple additions and removals in an RDF store and filter for change patterns by using SPARQL queries. This involves a considerable amount of reverse-engineering, i.e., finding out the changes in the RDF store caused by specific user interactions with the application that manipulates the data (e.g., the stack component).

This tight coupling of the definition of the notifications and the RDF data representation can be problematic if the business logic of the application that writes the data into the managed store changes. Suppose, for example, a notification is defined to be triggered on addition of a concept and the notification message contains the human-readable `rdfs:label` of the concept. During the development process of the thesaurus management application, specifications change and labels of concepts are represented by `skos:prefLabel` rather than `rdfs:label`. As a consequence, the notification will break, causing the notification specification to be adapted to using the `skos:prefLabel` label property.

However, the above mentioned weakness brings up a new potential use case of *rsine*. It can easily be integrated into a black-box testing approach for applications writing RDF data. A common strategy for testing Web applications (cf. Selenium<sup>1</sup>) is, e.g., to automatically execute specific tasks in the application and compare the resulting Web pages with the expected result of the task. *rsine* can be used to bring this methodology to the data level. A notification could, for example, specify the patterns that must be written to the managed store after a specific user interaction has been performed. If the notification fires, the observed application behavior is identical to the expected behavior and thus the test can be considered successful. Otherwise an implementation bug might have been introduced. Recent publications [KWA<sup>+</sup>14] and projects<sup>2</sup> propose approaches for (automatically) evaluating ontologies using a set of predefined quality patterns and focus on a unit-test-like approach.

In some cases such as for notifications on *Inconsistent Preferred Labels* and *Overlapping Labels* additional restrictions might be necessary because the former is a special case of the latter and thus multiple notifications on a single dataset change will be disseminated. Furthermore, inferring the type of action a user performed in the application by observing

---

<sup>1</sup>Selenium browser automation framework for testing purposes: <http://docs.seleniumhq.org/>. Retrieved 2015-06-23.

<sup>2</sup>RDFUnit: <https://github.com/AKSW/RDFUnit/>. Retrieved 2015-06-23.

changes in the underlying RDF data proved to require complex queries in some cases. This leads to queries that are hard to understand and maintain.

Furthermore, one requirement from Wolters Kluwer Germany (WKD) was to get notified on broken links to “external” sources on the Web, e.g., DBpedia. *rsine* cannot address this requirement because it is designed to notify subscribers as soon as data changes within a managed RDF store occur. Broken links can be inflicted by the linked sources which is not within the scope of control of the Web vocabulary developers. Therefore periodic checks of the whole Web vocabulary encompassing all links pointing to external sources would be necessary. Nevertheless, it would be partly possible to address the issue of *Broken Links* with *rsine*. An approach would be to listen for changesets that describe additions of triples with links to external sources as subject or object. An independent process (e.g., based on *qSKOS*) could then dereference these collected links and trigger subscriber notifications.

As outlined in the *rsine* implementation description (Section 4.3.2), changes to the managed store are monitored and persisted to the changeset store in a standardized format. The data in the changeset store therefore constitutes the basis against which all subscription queries are evaluated against. As a consequence, there is no way of providing an option to the notification message recipient which allows to block or veto a certain change because at the time the notification message is disseminated, the change has already been committed to the managed store.

### 6.3.3 Usability Issues

After activating on-change notification on thesaurus changes, some usability issues were brought to our attention by WKD who evaluated the feasibility of the approach. One of these issues was that a specific change (the creation of a new concept in *PoolParty Thesaurus Server*) created multiple notifications. This is due to the fact that creation of a concept involves multiple triple additions like asserting the type `skos:Concept` to a newly created URI, assigning a `skos:prefLabel` to that URI as well as integrating the new concept into the thesaurus structure by stating the `skos:broader` concept(s). As a consequence, also other notifications are triggered that listen to RDF dataset changes which are a subset of actions involving more complex changes, e.g., when introducing an additional `skos:prefLabel` to an existing concept. While in some environments this would be no problem, during testing our notification approach within the LOD2 project in the WKD context, this was pointed out as usability issue.

Another issue that was brought up was that the concepts which were actually affected by a dataset change were not clearly named in the notification messages. We addressed

this by introducing “auxiliary queries” that are used to select additional information from the managed store which can be integrated into the message. This enables notification messages to contain, e.g., the actual preferred labels of affected concepts improving readability.

A similar concern was that not all notification messages provided information about the contributor who caused the change action. This can be addressed by extending the respective subscription documents and include the creator of the changeset into the notification message. Some applications (e.g., PoolParty), persist the most recent contributor to a resource in the dataset, so adding this information to the notification message can easily be achieved for thesaurus development scenarios. However, we did not yet investigate this for the Pebbles application in the LOD2 use case.

Another important part that was missing in the notification messages from a usability point of view was that the actual cause of the notification was not clear enough. We addressed this by providing the possibility to define a description for each notification subscription, explaining its purpose. This information can then be appended to the notification message. An exemplary notification message that has been formulated based on the improvements received by feedback from WKD is provided in Listing 6.1.

---

```
The concepts <a href='http://vocabulary.semantic-web.at/semweb/119'>IBM</a> and
<a href='http://vocabulary.semantic-web.at/semweb/1520'>ILOG</a> form an
hierarchical cycle. You receive this notification because of subscription
'http://example.org/chr' (Notification on circular hierarchical relations)
```

---

LISTING 6.1: An exemplary notification message with improvements based on user feedback.

Regarding the *qSKOS* tool we received feedback that it is often not evident what a certain quality issue is about. Although we provide the name and description of each found quality issue in the generated textual reports, this point of criticism seems to prevail. We therefore also include URIs to the *qSKOS* Wiki<sup>3</sup> that describes all quality issues in greater detail into the quality reports and plan to update it with even more detailed information.

## 6.4 Future Work

We found that some quality issues in our catalog can be extended and improved to better fit some special cases that can be experienced when applying them to existing

---

<sup>3</sup>*qSKOS* Wiki: <https://github.com/cmader/qSKOS/wiki/Quality-Issues>. Retrieved 2015-06-23.

vocabularies on the Web. One example is to improve the handling of private language tags such as `x-other` that carry no agreed-upon meaning and proved to be problematic when checking for the issue *No Common Language*. Furthermore, a user of the tool suggested additional configuration options for the issue *Incomplete Language Coverage*<sup>4</sup>. We also plan on extending the assessment of the quality issue *Missing Incoming Links* by configuration options for the used SPARQL endpoints and support for Linked Data dumps such as LOD Laundromat<sup>5</sup>.

In Web vocabulary development we can observe that SKOS is used as a starting point, but as the vocabulary evolves, often the need for more fine-granular modeling arises. Developers enrich semantics by, e.g., reusing classes from other ontologies or define their own. We expect that this increased semantic complexity leads to introduction of new quality issues that we do not yet cover, as we are focusing on the SKOS semantic model. Additional semantic properties and type information can, for example, help in detecting inconsistencies in hierarchical relations such as using generic-specific relations and whole-part relations in the same hierarchy branch. These issues can currently not be reliably detected when vocabulary analysis is limited to the SKOS schema.

From a usability perspective, we plan to express the generated quality reports in a standard format using common ontologies. Recently developed frameworks like Luzzu<sup>6</sup> provide ways for expressing findings from quality metrics in a standardized way and can profit from the quality issues we defined in this work. We also plan to evaluate how *qSKOS* and *rsine* can be used in the area of Web vocabulary regression tests and if and how it can integrate with existing solutions such as RDFUnit.

We currently do not provide a user interface that enables dataset curators and Web vocabulary developers to easily subscribe for notifications. In our test installation in the context of the LOD2 project we set up *rsine* with all notifications enabled and with a predefined email address as notification recipient. One major future contribution will be to provide a graphical user interface for selection of predefined subscriptions as well as support for creation of custom subscriptions. Future directions of this work will target (i) simplification of notification subscription queries and (ii) extending expressiveness to support assessment of quality issues that exceed the potential of SPARQL queries.

In the evaluation of *rsine* we performed in the context of the LOD2 project, we only set up one instance of PoolParty to gather some preliminary user feedback. Thus, we were not able to test *rsine*'s capability to forward notification messages and monitor the reuse of data between distinct systems, which will be part of our future work.

---

<sup>4</sup>Feature request for *Incomplete Language Coverage* improvement: <https://github.com/cmader/qSKOS/issues/36>. Retrieved 2015-06-23.

<sup>5</sup>LOD Laundromat: <http://lodlaundromat.org/>. Retrieved 2015-06-23.

<sup>6</sup>Luzzu quality assessment framework: <http://eis-bonn.github.io/Luzzu/>. Retrieved 2015-06-23.



# Bibliography

- [ABG03] J. Aitchison, D. Bawden, and A. Gilchrist. *Thesaurus Construction and Use: A Practical Manual*. Taylor & Francis, 2003.
- [AH11] D. Allemang and J. Hendler. *Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL*. Morgan Kaufmann, 2011.
- [AMBS12a] Nor Azlinayati Abdul Manaf, Sean Bechhofer, and Robert Stevens. Common modelling slips in SKOS vocabularies. In Pavel Klinov and Matthew Horridge, editors, *OWLED*, volume 849 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2012.
- [AMBS12b] Nor Azlinayati Abdul Manaf, Sean Bechhofer, and Robert Stevens. The current state of SKOS vocabularies on the Web. In Elena Simperl, Philipp Cimiano, Axel Polleres, Oscar Corcho, and Valentina Presutti, editors, *ESWC*, volume 7295 of *Lecture Notes in Computer Science*, pages 270–284. Springer, 2012.
- [BBFZ09] Marco Baroni, S. Bernardini, A. Ferraresi, and E. Zanchetta. The WaCky Wide Web: A Collection of Very Large Linguistically Processed Web-Crawled Corpora. *Language Resources and Evaluation* 43(3): 209–226, 43(3):209–226, 2009.
- [BFF08] Diego Berrueta, Sergio Fernández, and Iván Frade. Cooking HTTP content negotiation with Vapour. 2008.
- [BLK<sup>+</sup>09] Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. DBpedia - a crystallization point for the web of data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3):154 – 165, 2009.
- [BS06] Carlo Batini and Monica Scannapieca. Data Quality Dimensions. *Data Quality: Concepts, Methodologies and Techniques*, pages 19–49, 2006.
- [BvHH<sup>+</sup>04] Sean Bechhofer, Frank van Harmelen, Jim Hendler, Ian Horrocks, Deborah McGuinness, Peter Patel-Schneijder, and Lynn Andrea Stein. OWL Web Ontology Language Reference. Recommendation, World Wide Web Consortium (W3C), February10 2004. See <http://www.w3.org/TR/owl-ref/>.
- [DA09] Rim Djedidi and Marie-Aude Aufaure. ONTO-EVOAL an Ontology Evolution Approach Guided by Pattern Modeling and Quality Evaluation. In Sebastian Link and Henri Prade, editors, *FoIKS*, volume 5956 of *Lecture Notes in Computer Science*, pages 286–305. Springer, 2009.

- [dCWF<sup>+</sup>09] Sherri de Coronado, Lawrence W. Wright, Gilberto Fragoso, Margaret W. Haber, Elizabeth A. Hahn-Dantona, Francis W. Hartel, Sharon L. Quan, Tracy Safran, Nicole Thomas, and Lori Whiteman. The NCI Thesaurus quality assurance life cycle. *J. Biomed. Inform.*, 42(3):530–539, 2009.
- [DF06] Li Ding and Tim Finin. Characterizing the Semantic Web on the Web. *Electrical Engineering*, 4273(August):5–9, 2006.
- [FF06] Martin Fowler and Matthew Foemmel. Continuous integration. *Thought-Works*) <http://www.thoughtworks.com/Continuous Integration.pdf>, 2006.
- [FH10a] Christian Fürber and Martin Hepp. Using semantic web resources for data quality management. In *Proceedings of the 17th international conference on Knowledge engineering and management by the masses, EKAW’10*, pages 211–225, Berlin, Heidelberg, 2010. Springer-Verlag.
- [FH10b] Christian Fürber and Martin Hepp. Using SPARQL and SPIN for data quality management on the semantic web. In *Business Information Systems*, pages 35–46. Springer, 2010.
- [FS96] Norbert E Fuchs and Rolf Schwitter. Attempto controlled english (ace). *arXiv preprint cmp-lg/9603003*, 1996.
- [GCCL05] Aldo Gangemi, Carola Catenacci, Massimiliano Ciaramita, and Jens Lehmann. Ontology evaluation and validation: an integrated formal model for the quality diagnostic task. Technical report, Laboratory of Applied Ontologies – CNR, Rome, Italy, 2005. [http://www.loa-cnr.it/Files/OntoEval4OntoDev\\_Final.pdf](http://www.loa-cnr.it/Files/OntoEval4OntoDev_Final.pdf).
- [GHKP12] Birte Glimm, Aidan Hogan, Markus Krötzsch, and Axel Polleres. OWL: yet to arrive on the web of data? In *WWW2012 Workshop on Linked Data on the Web, Lyon, France, 16 April, 2012*, 2012.
- [GPS11] Rafael S. Gonçalves, Bijan Parsia, and Ulrike Sattler. Analysing the evolution of the NCI Thesaurus. In *CBMS*, pages 1–6. IEEE, 2011.
- [Har10] P. Harpring. *Introduction to Controlled Vocabularies: Terminology for Art, Architecture, and Other Cultural Works*. Getty publications. Getty Research Institute, 2010.
- [Hay04] Patrick Hayes. RDF Semantics. W3C Recommendation, 2004.
- [HB11] Tom Heath and Christian Bizer. *Linked Data: Evolving the Web into a Global Data Space*. Morgan & Claypool, 2011.
- [Hed10] H. Hedden. *The Accidental Taxonomist*. Information Today, Incorporated, 2010.
- [HH10] Harry Halpin and Patrick J. Hayes. When owl:sameAs isn’t the same: An analysis of identity links on the semantic web. April 2010.
- [HHP<sup>+</sup>10] Aidan Hogan, Andreas Harth, Alexandre Passant, Stefan Decker, and Axel Polleres. Weaving the Pedantic Web. In *Proc. WWW2010 Workshop on Linked Data on the Web (LDOW)*, 2010.

- [HKR10] Pascal Hitzler, Markus Krötzsch, and Sebastian Rudolph. *Foundations of Semantic Web Technologies*. CRC Press, 2010.
- [IS09] Antoine Isaac and Ed Summers. SKOS Simple Knowledge Organization System Primer. Working Group Note, W3C, 2009.
- [Iso11a] ISO 25964-1: Information and documentation – Thesauri and interoperability with other vocabularies – Part 1: Thesauri for information retrieval. Norm, International Organization for Standardization, 2011.
- [ISO11b] ISO 25964-2: Information and documentation – Thesauri and interoperability with other vocabularies – Part 2: Interoperability with other vocabularies. Norm, International Organization for Standardization, 2011.
- [IWYZ11] Antoine Isaac, William Waites, Jeff Young, and Marcia Zeng. Library Linked Data Incubator Group: Datasets, Value Vocabularies, and Metadata Element Sets. *W3C Incubator Group Report*, 25, 2011.
- [KIT<sup>+</sup>05] Asanee Kawtrakul, Aurawan Imsombut, Aree Thunkijjanukit, Dagobert Soergel, Anita Liang, Margherita Sini, Gudrun Johannsen, and Johannes Keizer. Automatic term relationship cleaning and refinement for AGROVOC. In *Workshop on The Sixth Agricultural Ontology Service*, pages 247–260, 2005.
- [KM10] Daniel Kless and Simon Milton. Towards Quality Measures for Evaluating Thesauri. In Salvador Sánchez-Alonso and Ioannis N. Athanasiadis, editors, *Metadata and Semantic Research*, volume 108 of *Communications in Computer and Information Science*, pages 312–319. Springer Berlin Heidelberg, 2010.
- [KSVdS<sup>+</sup>13] Martin Klein, Robert Sanderson, Herbert Van de Sompel, Simeon Warner, Bernhard Haslhofer, Carl Lagoze, and Michael L Nelson. A technical framework for resource synchronization. *D-Lib Magazine*, 19(1):3, 2013.
- [KWA<sup>+</sup>14] Dimitris Kontokostas, Patrick Westphal, Sören Auer, Sebastian Hellmann, Jens Lehmann, Roland Cornelissen, and Amrapali Zaveri. Test-driven evaluation of linked data quality. In *Proceedings of the 23rd international conference on World Wide Web*, pages 747–758. International World Wide Web Conferences Steering Committee, 2014.
- [LS99] Ora Lassila and Ralph R. Swick. Resource Description Framework (RDF) Model and Syntax Specification. W3C recommendation, W3C, February 1999.
- [Mad12] Christian Mader. Quality Assurance in Collaboratively Created Web Vocabularies. In *PhD symposium of ESWC2012*, Greece, January 2012. Springer.
- [MB09] Alistair Miles and Sean Bechhofer. SKOS Simple Knowledge Organization System Reference. Recommendation, W3C, 2009.

- [MH11] Christian Mader and Bernhard Haslhofer. Quality Criteria for Controlled Web Vocabularies. In *International Conference on Theory and Practice of Digital Libraries 2011, NKOS Workshop*, Berlin, Germany, August 2011.
- [MH13] Christian Mader and Bernhard Haslhofer. Perception and Relevance of Quality Issues in Web Vocabularies. In *I-SEMANTICS 2013*, Graz, AUT, 2013.
- [MHI12] Christian Mader, Bernhard Haslhofer, and Antoine Isaac. Finding Quality Issues in SKOS Vocabularies. In *TPDL 2012 Theory and Practice of Digital Libraries*, Germany, May 2012.
- [MHP11] Christian Mader, Bernhard Haslhofer, and Niko Popitsch. The MEKE-TREpository - Middle Kingdom Tomb and Artwork Descriptions on the Web. In *Theory and Practice of Digital Libraries*, Germany, March 2011. Springer.
- [MMS14] Christian Mader, Michael Martin, and Claus Stadler. Facilitating the Exploration and Visualization of Linked Data. In *Linked Open Data - Creating Knowledge Out of Interlinked Data*, pages 90–107. Springer, August 2014.
- [MW14] Christian Mader and Christian Wartena. Supporting Web Vocabulary Development by Automated Quality Assessment: Results of a Case Study in a Teaching Context. In *Workshop on Human-Semantic Web Interaction (HSWI2014)*, CEUR Workshop Proceedings, May 2014.
- [Neu09] Joachim Neubert. Bringing the “Thesaurus for Economics” on to the web of linked data. In *Proceedings of the WWW2009 Workshop on Linked Data on the Web (LDOW2009)*, 2009.
- [NIS05] NISO. *ANSI/NISO Z39.19 - Guidelines for the Construction, Format, and Management of Monolingual Controlled Vocabularies*, 2005.
- [NPM11] Helmut Nagy, Tassilo Pellegrini, and Christian Mader. Exploring Structural Differences in Thesauri for SKOS-based Applications. In *I-SEMANTICS 2011*, Graz, Austria, September 2011.
- [OYE07] Anthony M. Orme, Haining Yao, and Letha H. Etzkorn. Indicating ontology data quality, stability, and completeness throughout ontology evolution. *Journal of Software Maintenance*, 19(1):49–75, 2007.
- [PH10] Niko P. Popitsch and Bernhard Haslhofer. DSNotify: handling broken links in the web of data. In *Proc. 19th Int. Conf. World Wide Web (WWW)*, pages 761–770, 2010.
- [PM10] Alexandre Passant and Pablo N Mendes. sparqlPuSH: Proactive Notification of Data Updates in RDF Stores Using PubSubHubbub. In *SFSW*, 2010.
- [PPW06] George Papamarkos, Alexandra Poulouvasilis, and Peter T Wood. Event-condition-action rules on RDF metadata in P2P environments. *Computer Networks*, 50(10):1513–1532, 2006.

- [PS08] Eric Prud'hommeaux and Andy Seaborne. SPARQL Query Language for RDF. W3C Recommendation, 2008.
- [PVdCSFGP12] María Poveda-Villalón, María del Carmen Suárez-Figueroa, and Asunción Gómez-Pérez. Validating Ontologies with OOPS! In Annette Teije, Johanna Völker, Siegfried Handschuh, Heiner Stuckenschmidt, Mathieu d'Aquin, Andriy Nikolov, Nathalie Aussenac-Gilles, and Nathalie Hernandez, editors, *EKAU*, volume 7603 of *Lecture Notes in Computer Science*, pages 267–281. Springer, 2012.
- [Sch95] Helmut Schmid. Improvements in Part-of-Speech Tagging with an Application to German. In *Proceedings of the ACL SIGDAT-Workshop*, 1995.
- [SH12] Osmo Suominen and Eero Hyvönen. Improving the quality of SKOS vocabularies with Skosify. In *Proceedings of the 18th international conference on Knowledge Engineering and Knowledge Management, EKAU'12*, pages 383–397, Berlin, Heidelberg, 2012. Springer-Verlag.
- [Shi12] A. Shiri. *Powering Search: The Role of Thesauri in New Information Environments*. ASIS&T monograph series. American Society for Information Science and Technology by Information Today, Incorporated, 2012.
- [SLW97] Diane M Strong, Yang W Lee, and Richard Y Wang. Data quality in context. *Communications of the ACM*, 40(5):103–110, 1997.
- [SM13] Osmo Suominen and Christian Mader. Assessing and Improving the Quality of SKOS Vocabularies. *Journal on Data Semantics*, 2(2), 2013.
- [Soe97] Dagobert Soergel. Functions of a thesaurus/classification/ontological knowledge base. *College of Library and Information Services. University of Maryland*, 1997.
- [Soe02] D. Soergel. Thesauri and ontologies in digital libraries: tutorial. In *Proc. 2nd Joint Conf. on Digital libraries (JCDL)*, 2002.
- [Spe08] S. Spero. LCSH is to Thesaurus as Doorbell is to Mammal: Visualizing Structural Problems in the Library of Congress Subject Headings. In *Proc. Int. Conf. on Dublin Core and Metadata Applications (DC)*, 2008.
- [SPG<sup>+</sup>07] Evren Sirin, Bijan Parsia, Bernardo Grau, Aditya Kalyanpur, and Yarden Katz. Pellet: A practical OWL-DL reasoner. *Journal of Web Semantics*, 5(2):51–53, 2007.
- [SPL04] K. Supekar, C. Patel, and Y. Lee. Characterizing Quality of Knowledge on Semantic Web. In *Proceedings of AAAI Florida AI Research Symposium (FLAIRS-2004)*, Miami Beach, Florida, May 17-19 2004.
- [ST08] Dariusz Strasunskas and Stein L. Tomassen. Empirical Insights on a Value of Ontology Quality in Ontology-Driven Web Search. In Robert Meersman and Zahir Tari, editors, *OTM Conferences (2)*, volume 5332 of *Lecture Notes in Computer Science*, pages 1319–1337. Springer, 2008.

- [Sve] Elaine Svenonius. *Design of Controlled Vocabularies*, chapter 107, pages 822–838.
- [Sve03] E. Svenonius. Design of controlled vocabularies. *Encyclopedia of Library and Information Science*, 45:822–838, 2003.
- [TA07] Samir Tartir and I. Budak Arpinar. Ontology Evaluation and Ranking using OntoQA. *International Conference on Semantic Computing*, 0:185–192, 2007.
- [TAM<sup>+</sup>05] Samir Tartir, I. Budak Arpinar, Michael Moore, Amit P. Sheth, and Boanerges Aleman meza. OntoQA: Metric-based ontology quality analysis. In *IEEE Workshop on Knowledge Acquisition from Distributed, Autonomous, Semantically Heterogeneous Data and Knowledge Sources*, 2005.
- [TDM09] Jiao Tao, Li Ding, and Deborah L. McGuinness. Instance Data Evaluation for Semantic Web-Based Knowledge Management Systems. In *HICSS*, pages 1–10. IEEE Computer Society, 2009.
- [VdSSK<sup>+</sup>12] Herbert Van de Sompel, Robert Sanderson, Martin Klein, Michael L Nelson, Bernhard Haslhofer, Simeon Warner, and Carl Lagoze. A perspective on resource synchronization. *D-Lib Magazine*, 18(9):3, 2012.
- [Vra10] Denny Vrandečić. *Ontology Evaluation*. PhD thesis, KIT, Fakultät für Wirtschaftswissenschaften, Karlsruhe, 2010.
- [71] World Wide Web Consortium (W3C). OWL 2 Web Ontology Language. Structural Specification and Functional-Style Syntax, 2009.
- [WGA13] Christian Wartena and Montserrat Garcia-Alsina. Challenges and Potentials for Keyword Extraction from Company Websites for the Development of Regional Knowledge Maps. In Kecheng Liu, editor, *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval and the International Conference on Knowledge Management and Information Sharing*, pages 241–248. Scitepress, 2013.
- [Yat66] F.A. Yates. *The Art of Memory*. A Phoenix book. University of Chicago Press, 1966.

# Appendices





## Appendix A

# Implemented Tools Output and Configuration

---

This is the quality report of file /home/christian/diss/journalpaper/odt/odt-combined.ttl, generated by qSKOS on So, 28 Sep 2014 19:04:52 +0200

\* Summary of Quality Issue Occurrences:

Orphan Concepts: FAIL (4)  
Disconnected Concept Clusters: FAIL (7)  
Cyclic Hierarchical Relations: OK (no potential problems found)  
Valueless Associative Relations: FAIL (7)  
Solely Transitively Related Concepts: OK (no potential problems found)  
Omitted Top Concepts: OK (no potential problems found)  
Top Concepts Having Broader Concepts: FAIL (2)  
Unidirectionally Related Concepts: OK (no potential problems found)  
Hierarchical Redundancy: FAIL (46)  
Reflexively Related Concepts: OK (no potential problems found)  
Mapping Relations Misuse: FAIL (2)

\* Detailed coverage of each Quality Issue:

--- Orphan Concepts

Description: Finds all orphan concepts, i.e. those not having semantic relationships to other concepts  
Detailed information: <https://github.com/cmader/qSKOS/wiki/Quality-Issues#orphan-concepts>  
orphan-concepts  
count: 4  
<http://vocabulary.semantic-web.at/OpenData/Africa>  
<http://vocabulary.semantic-web.at/OpenData/Asia>  
<http://vocabulary.semantic-web.at/OpenData/Americas>  
<http://vocabulary.semantic-web.at/OpenData/Ocenania>

--- Disconnected Concept Clusters

Description: Finds sets of concepts that are isolated from the rest of the vocabulary  
Detailed information: <https://github.com/cmader/qSKOS/wiki/Quality-Issues#disconnected-concept-clusters>  
disconnected-concept-clusters

```

count: 7
Cluster 1, size: 201
Cluster 2, size: 3
Cluster 3, size: 2
Cluster 4, size: 14
Cluster 5, size: 2
Cluster 6, size: 3
Cluster 7, size: 4
Cluster 1, size: 201
  http://rdf.freebase.com/ns/m/0cmcæn6
  http://vocabulary.semantic-web.at/OpenData/open_data_formats
  http://vocabulary.semantic-web.at/semweb/1233
  http://rdf.freebase.com/ns/guid.9202a8c04000641f80000000000a173c
[...]
Cluster 2, size: 3
  http://vocabulary.semantic-web.at/OpenData/scrapper
  http://dbpedia.org/resource/Web_scraping
  http://rdf.freebase.com/ns/m/07ykb5
Cluster 3, size: 2
  http://vocabulary.semantic-web.at/OpenData/goverment
  http://vocabulary.semantic-web.at/OpenData/budget
Cluster 4, size: 14
  http://vocabulary.semantic-web.at/OpenData/Central_Europe
  http://www4.wiwi.fu-berlin.de/factbook/resource/Austria
[...]
Cluster 5, size: 2
  http://vocabulary.semantic-web.at/OpenData/visualiser
  http://vocabulary.semantic-web.at/OpenData/PoolParty
Cluster 6, size: 3
  http://ckan.net/tag/dentistry
  http://vocabulary.semantic-web.at/OpenData/Dentistry
  http://vocabulary.semantic-web.at/OpenData/Health_care_and_medicine
Cluster 7, size: 4
  http://dbpedia.org/resource/Parsing
  http://vocabulary.semantic-web.at/OpenData/parser
  http://umbel.org/umbel/ne/wikipedia/Parsing
  http://rdf.freebase.com/ns/guid.9202a8c04000641f800000000001c86fe

--- Cyclic Hierarchical Relations
Description: Finds concepts that are hierarchically related to each other
Detailed information: https://github.com/cmader/qSKOS/wiki/Quality-Issues#
cyclic-hierarchical-relations
count: 0

--- Valueless Associative Relations
Description: Finds sibling concept pairs that are also connected by an associative
relation
Detailed information: https://github.com/cmader/qSKOS/wiki/Quality-Issues#
valueless-associative-relations
count: 7
(http://vocabulary.semantic-web.at/OpenData/structured_data, http://vocabulary.
semantic-web.at/OpenData/unstructured_data)
(http://vocabulary.semantic-web.at/OpenData/structured_data, http://vocabulary.
semantic-web.at/OpenData/semistructured_data)
(http://vocabulary.semantic-web.at/OpenData/OpenStreetMap_Geodata_License, http://

```

```
vocabulary.semantic-web.at/OpenData/AttributionShareAlike_20_Generic)
(http://vocabulary.semantic-web.at/OpenData/Public_Domain_Dedication_and_License,
http://vocabulary.semantic-web.at/OpenData/Public_Domain_Dedication)
(http://vocabulary.semantic-web.at/OpenData/Web_API, http://vocabulary.semantic-
web.at/OpenData/mashup)
(http://vocabulary.semantic-web.at/OpenData/semistructured_data, http://
vocabulary.semantic-web.at/OpenData/unstructured_data)
(http://vocabulary.semantic-web.at/OpenData/structured_data, http://vocabulary.
semantic-web.at/OpenData/linked_data)
```

#### --- Solely Transitively Related Concepts

Description: Concepts only related by skos:broaderTransitive or skos:
narrowerTransitive

Detailed information: <https://github.com/cmader/qSKOS/wiki/Quality-Issues#solely-transitively-related-concepts>

count: 0

#### --- Omitted Top Concepts

Description: Finds skos:ConceptSchemes that don't have top concepts defined

Detailed information: <https://github.com/cmader/qSKOS/wiki/Quality-Issues#omitted-top-concepts>

count: 0

#### --- Top Concepts Having Broader Concepts

Description: Finds top concepts internal to the vocabulary hierarchy tree

Detailed information: <https://github.com/cmader/qSKOS/wiki/Quality-Issues#top-concepts-having-broader-concepts>

count: 2

[http://vocabulary.semantic-web.at/OpenData/energy\\_data](http://vocabulary.semantic-web.at/OpenData/energy_data)

<http://vocabulary.semantic-web.at/OpenData/mashup>

#### --- Unidirectionally Related Concepts

Description: Concepts not including reciprocal relations

Detailed information: <https://github.com/cmader/qSKOS/wiki/Quality-Issues#unidirectionally-related-concepts>

count: 0

#### --- Hierarchical Redundancy

Description: Finds broader/narrower relations over multiple hierarchy levels

Detailed information: <https://github.com/cmader/qSKOS/wiki/Quality-Issues#hierarchical-redundancy>

count: 46

[http://vocabulary.semantic-web.at/OpenData/SPARQL\\_API](http://vocabulary.semantic-web.at/OpenData/SPARQL_API), [http://vocabulary.semantic-web.at/OpenData/Application\\_Programming\\_Interface](http://vocabulary.semantic-web.at/OpenData/Application_Programming_Interface)

<http://vocabulary.semantic-web.at/OpenData/Austria>, <http://vocabulary.semantic-web.at/OpenData/Europe>

<http://vocabulary.semantic-web.at/OpenData/KML>, [http://vocabulary.semantic-web.at/OpenData/data\\_formats](http://vocabulary.semantic-web.at/OpenData/data_formats)

[...]

#### --- Reflexively Related Concepts

Description: Finds concepts that are related to themselves

Detailed information: [https://github.com/cmader/qSKOS/wiki/Quality-Issues#wiki-Reflexive\\_Relations](https://github.com/cmader/qSKOS/wiki/Quality-Issues#wiki-Reflexive_Relations)

count: 0

---

--- Mapping Relations Misuse

Description: Finds concepts within the same concept scheme that are related by a mapping relation

Detailed information: <https://github.com/cmader/qSKOS/wiki/Quality-Issues#mapping-relations-misuse>

count: 2

([http://vocabulary.semantic-web.at/OpenData/linked\\_open\\_data](http://vocabulary.semantic-web.at/OpenData/linked_open_data), <http://www.w3.org/2004/02/skos/core#mappingRelation>, [http://vocabulary.semantic-web.at/OpenData/five\\_stars\\_data](http://vocabulary.semantic-web.at/OpenData/five_stars_data)) [null]

([http://vocabulary.semantic-web.at/OpenData/five\\_stars\\_data](http://vocabulary.semantic-web.at/OpenData/five_stars_data), <http://www.w3.org/2004/02/skos/core#mappingRelation>, [http://vocabulary.semantic-web.at/OpenData/linked\\_open\\_data](http://vocabulary.semantic-web.at/OpenData/linked_open_data)) [null]

---

LISTING A.1: Quality report generated by *qSKOS* for structural issues (shortened).

---

```
@prefix spin: <http://spinrdf.org/sp/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix rsine: <http://lod2.eu/rsine/> .
@prefix dcterms: <http://purl.org/dc/terms/>.

<http://example.org/chr> a rsine:Subscription;
rsine:query [
  dcterms:description "Notification on circular hierarchical relations";

  spin:text "
    PREFIX cs:<http://purl.org/vocab/changeset/schema#>
    PREFIX spin:<http://spinrdf.org/sp/>
    PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
    PREFIX skos:<http://www.w3.org/2004/02/skos/core#>

    SELECT ?concept ?otherConcept WHERE {
      ?cs a cs:ChangeSet .
      ?cs cs:createdDate ?csdate .
      ?cs cs:addition ?addition .

      ?addition rdf:subject ?concept .
      ?addition rdf:predicate skos:broader .
      ?addition rdf:object ?otherConcept .

      FILTER (?csdate >
        'QUERY_LAST_ISSUED'^^<http://www.w3.org/2001/XMLSchema#dateTime>)
    }";

  rsine:condition [
    spin:text "PREFIX skos:<http://www.w3.org/2004/02/skos/core#>
      ASK {
        ?concept skos:broader+ ?otherConcept .
        ?otherConcept skos:broader+ ?concept
      }";
    rsine:expect true;
  ];
];

rsine:auxiliary [
  spin:text "PREFIX skos:<http://www.w3.org/2004/02/skos/core#>"
```

---

```

    SELECT ?conceptLabel WHERE {
      ?concept skos:prefLabel ?conceptLabel .
      FILTER(langMatches(lang(?conceptLabel), 'en'))
    }";
spin:text "PREFIX skos:<http://www.w3.org/2004/02/skos/core#>
    SELECT ?otherConceptLabel WHERE {
      ?otherConcept skos:prefLabel ?otherConceptLabel .
      FILTER(langMatches(lang(?otherConceptLabel), 'en'))
    }";
];

rsine:formatter [
  a rsine:vttlFormatter;
  rsine:message
    "The concepts <a href='$bindingSet.getValue('concept')'>
      $bindingSet.getValue('conceptLabel').getLabel()</a> and
      <a href='$bindingSet.getValue('otherConcept')'>
        $bindingSet.getValue('otherConceptLabel').getLabel()
      </a> form a hierarchical cycle";
];

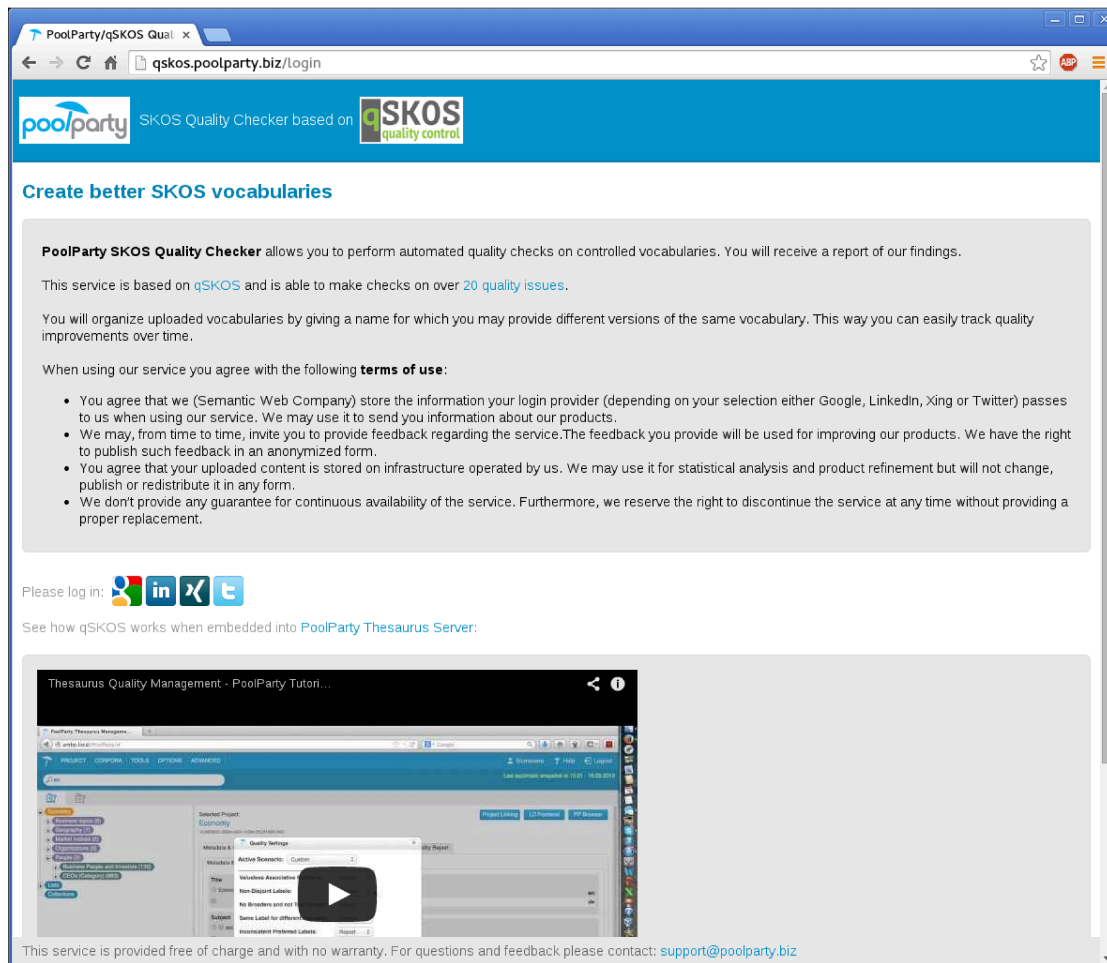
rsine:notifier [
  a rsine:loggingNotifier;
];


rsine:notifier [
  a rsine:emailNotifier;
  foaf:mbox <mailto:c.mader@semantic-web.at>
];


```

---

LISTING A.2: *rsine* subscription document for getting notified on introduction of circular hierarchical relations.

FIGURE A.1: Start screen of the *online SKOS Quality Checker* Web application.



SKOS Quality Checker based on 

Provide a name for the vocabulary you want to check if it does not yet exist in the list below:

### Your vocabularies

test


Datei auswählen


Keine ausgewählt

**Allowed formats:** n3, ntriples, rdf, rdfxml, trig, trix, turtle; maximum size 100MB


☐ Send the quality report to this email address:

Previous analysis results

  
aaaa.rdf  
Analyzed: Di, 14 Jan 2014 12:54:06

  
aaaa.rdf  
Analyzed: Di, 14 Jan 2014 12:54:32

  
stw.rdf  
Analyzed: Di, 14 Jan 2014 13:08:25

  
aaaa.rdf  
Analyzed: Di, 14 Jan 2014 13:08:25


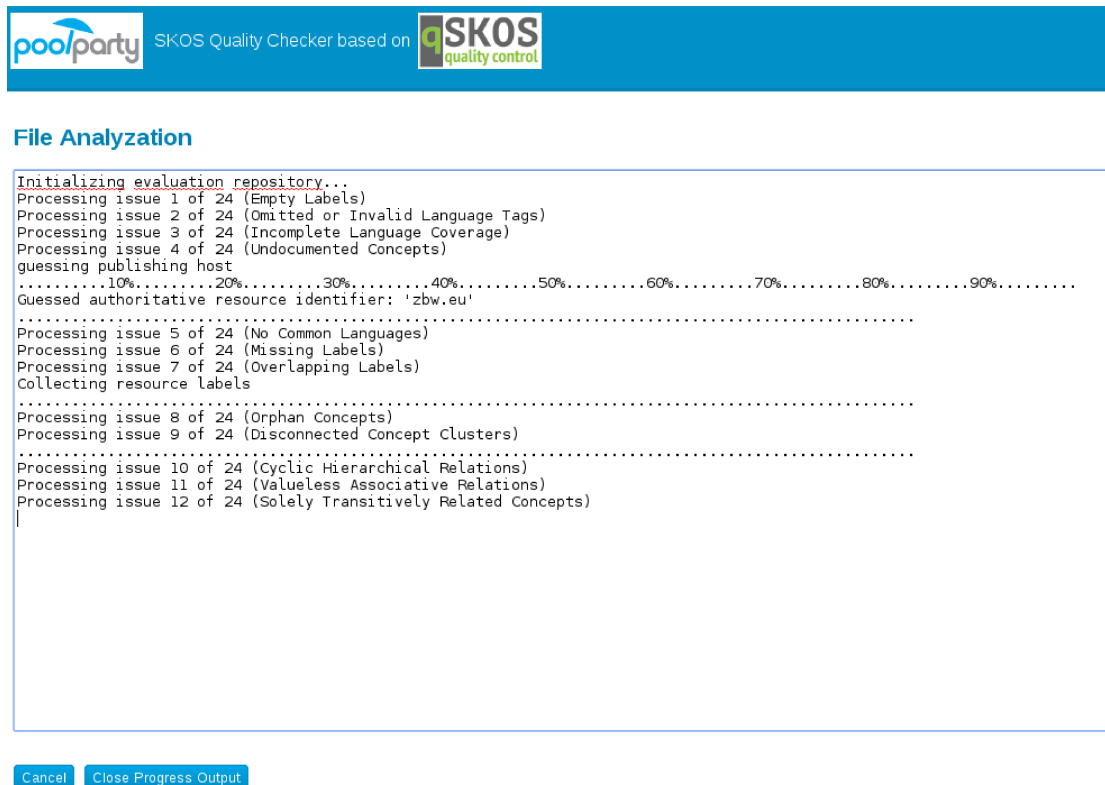
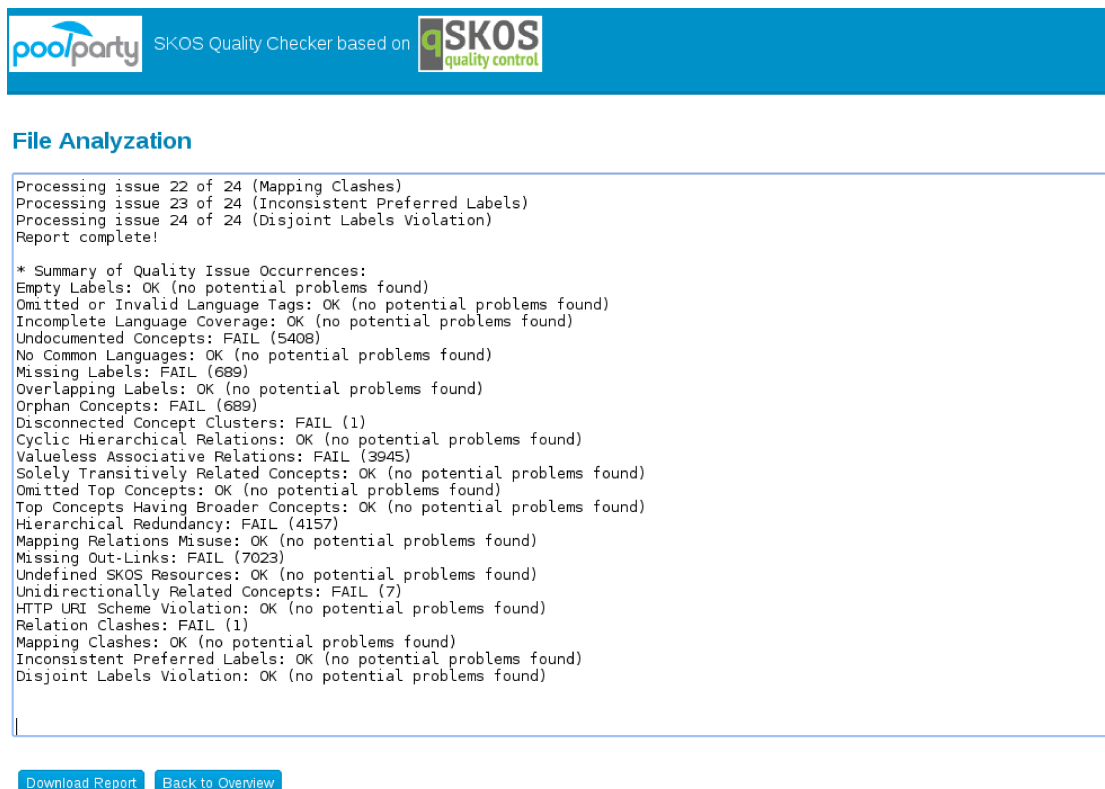
  
stw.rdf  
Analyzed: Mo, 24 Feb 2014 16:06:16

FIGURE A.2: Vocabulary upload and overview interface of the *online SKOS Quality Checker*.

FIGURE A.3: Vocabulary analysis in progress using *online SKOS Quality Checker*.FIGURE A.4: Vocabulary analysis finished in *online SKOS Quality Checker*.



**Reegle Thesaurus + Corpus**  
IDBCB732-A250-0001-3594-1A9212B91CE1

Metadata & Statistics | Concepts | Triples | SPARQL | Autopopulate | Visualization | **Quality Report**

Last Generated: 27.10.2014 - 17:16

- Hierarchical Cycles (0)
- Non-Disjoint Labels (0)
- Inconsistent Preferred Labels (4)**
- No Broaders and no Top Concept (0)
- Omitted or Invalid Language Tags (0)
- Same Label for different concepts (105)**
- Relation Clashes (280)**

FIGURE A.5: Overview of all detected quality issues in *PoolParty Thesaurus Server*.

**Reegle Thesaurus + Corpus**  
IDBCB732-A250-0001-3594-1A9212B91CE1

Metadata & Statistics | Concepts | Triples | SPARQL | Autopopulate | Visualization | **Quality Report**

Last Generated: 27.10.2014 - 17:16

- Hierarchical Cycles (0)
- Non-Disjoint Labels (0)
- Inconsistent Preferred Labels (4)**
- No Broaders and no Top Concept (0)
- Omitted or Invalid Language Tags (0)
- Same Label for different concepts (105)**
- Relation Clashes (280)**

<http://reegle.info/glossary/185>  
(en) Alternative Label  
<http://reegle.info/glossary/685>  
(en) Hidden Label

<http://reegle.info/glossary/588>  
**Windturbine** (de) Preferred Label  
<http://reegle.info/glossary/808>  
**Windturbine** (de) Alternative Label

<http://reegle.info/glossary/2454>  
**efficient buildings** (en) Alternative Label  
<http://reegle.info/glossary/2508>  
**efficient buildings** (en) Preferred Label

<http://reegle.info/glossary/631>  
**aguas residuales** (es) Preferred Label  
<http://reegle.info/glossary/1570>  
**aguas residuales** (es) Preferred Label

FIGURE A.6: List of overlapping labels found by *PoolParty Thesaurus Server*.

The screenshot displays the Reegle Thesaurus + Corpus web application. The interface includes a top navigation bar with tabs: PROJECT, CORPORA, TOOLS, OPTIONS, and ADVANCED. A search bar is present below the navigation bar. On the left, a tree view shows the hierarchy of the thesaurus, with categories like Climate Compatible Development Glossary (10), Energy Efficiency Glossary (7), and Renewable Energy Glossary (11). The main panel on the right shows the 'Reegle Thesaurus + Corpus' title and a list of relation clashes. The 'Last Generated' date is 27.10.2014 - 17:16. The list of clashes includes: Hierarchical Cycles (0), Non-Disjoint Labels (0), Inconsistent Preferred Labels (4), No Broaders and no Top Concept (0), Omitted or Invalid Language Tags (0), Same Label for different concepts (105), and Relation Clashes (280). A focused issue is shown below the list, illustrating a relation clash between 'hydrogen production from primary en...' and 'hydrogen production' (broader) and 'partial oxidation' (related).

Reegle Thesaurus + Corpus  
IDBCB732-A250-0001-3594-1A9212B91CE1

Metadata & Statistics Concepts Triples SPARQL Autopopulate Visualization Quality Report

Last Generated: 27.10.2014 - 17:16

Hierarchical Cycles (0)

Non-Disjoint Labels (0)

Inconsistent Preferred Labels (4)

No Broaders and no Top Concept (0)

Omitted or Invalid Language Tags (0)

Same Label for different concepts (105)

Relation Clashes (280)

direct evaporators - evaporators - air-water evaporators

direct evaporators - evaporators - ground-water evaporators

hydrogen production from secondary ... - water electrolysis - hydrogen production

hydrogen production from primary en... - natural gas reformation - hydrogen production

compressor designs - compressors - open compressors

compressor designs - semi-hermetic compressors - compressors

hydrogen production from primary en... - partial oxidation - hydrogen production

hydrogen production from primary en...

hydrogen production

partial oxidation

broader

related

compressor designs - compressors - hermetically sealed compressors

FIGURE A.7: List of relation clashes with one focused issue shown in *PoolParty Thesaurus Server*.

# Vita

Christian Mader is a PhD student at the Research Group Multimedia Information Systems (MIS), Faculty of Computer Science, University of Vienna since 2012. During his studies, he was responsible for the IT part of the MeKeTRE project<sup>1</sup>, a FWF-funded research project conducted at the Institute of Egyptology in cooperation with MIS between 2009 and 2013. He received his Diploma in Computer Science from the Vienna University of Technology in 2005. He currently works at Semantic Web Company<sup>2</sup> where he is responsible for software development tasks as well as conducting research in various scientific projects. His research interests focus on Linked Data, controlled vocabularies, and data quality.

---

<sup>1</sup><http://meketre.org>, retrieved 2015-06-23.

<sup>2</sup><http://www.semantic-web.at/>, retrieved 2015-06-23.