

# Management von Metadatenstandards in Geodateninfrastrukturen

DIPLOMARBEIT

zur Erlangung des akademischen Grades

**Diplom-Ingenieur**

im Rahmen des Studiums

**Software Engineering & Internet Computing**

eingereicht von

**Georg Ramler, BSc**

Matrikelnummer 0427641

an der Fakultät für Informatik  
der Technischen Universität Wien

Betreuung: Privatdoz. Dipl.-Ing. Dr.techn. Gerhard Navratil

Wien, 18. März 2025

---

Georg Ramler

---

Gerhard Navratil



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar  
The approved original version of this thesis is available in print at TU Wien Bibliothek.

# Management of Metadata Standards in Geodata Infrastructures

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

**Diplom-Ingenieur**

in

**Software Engineering & Internet Computing**

by

**Georg Ramler, BSc**

Registration Number 0427641

to the Faculty of Informatics

at the TU Wien

Advisor: Privatdoz. Dipl.-Ing. Dr.techn. Gerhard Navratil

Vienna, 18<sup>th</sup> March, 2025

---

Georg Ramler

---

Gerhard Navratil



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar  
The approved original version of this thesis is available in print at TU Wien Bibliothek.

# Erklärung zur Verfassung der Arbeit

Georg Ramler, BSc

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 18. März 2025

---

Georg Ramler



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar  
The approved original version of this thesis is available in print at TU Wien Bibliothek.

# Danksagung

Diese Arbeit wäre ohne die Unterstützung vieler Menschen nicht möglich gewesen.

Mein besonderer Dank gilt meinem Betreuer Privatdoz. Dipl.-Ing. Dr.techn. Gerhard Navratil, der mich mit wertvollem Feedback, konstruktiver Kritik und viel Geduld während der gesamten Arbeit begleitet hat.

Ein großes Dankeschön geht an Bettina und meine beiden Kinder Elisa und Jakob. Danke für euer Verständnis und die Zeit, die ihr mir gegeben habt, um mich dieser Arbeit zu widmen.

Besonders bedanken möchte ich mich auch bei meinen Eltern, die mir durch ihre beständige Unterstützung das Studium ermöglicht haben.



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar  
The approved original version of this thesis is available in print at TU Wien Bibliothek.

# Acknowledgements

This thesis would not have been possible without the support of many people.

My special thanks go to my advisor, Priv.-Doz. Dipl.-Ing. Dr. techn. Gerhard Navratil, who guided me through this work with valuable feedback, constructive criticism, and great patience.

A heartfelt thank you to Bettina and my two children, Elisa and Jakob. Thank you for your understanding and for giving me the time I needed to focus on this thesis.

I would also like to express my deep gratitude to my parents, whose unwavering support made my studies possible.



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar  
The approved original version of this thesis is available in print at TU Wien Bibliothek.

# Kurzfassung

Die Verwaltung von Metadaten in Geodateninfrastrukturen ist, insbesondere im Hinblick auf die Interoperabilität zwischen verschiedenen Standards wie ISO 19115, INSPIRE und DCAT-AP, eine zentrale Herausforderung im Geodatenmanagement. Die fehlende direkte Kompatibilität zwischen diesen Standards führt dazu, dass Geodatenstellen Metadaten mehrfach in unterschiedlichen Werkzeugen und Formaten erfassen müssen, was Redundanzen, Inkonsistenzen und einen hohen Verwaltungsaufwand verursacht. Zudem überfordert die Komplexität dieser Standards viele Anwender, insbesondere Datenproduzenten, die für die initiale Metadatenerfassung verantwortlich sind, jedoch oft nicht über die erforderliche Expertise im Metadatenmanagement verfügen.

Diese Arbeit entwickelt eine ontologiebasierte Lösung zur Harmonisierung der Metadatenstandards ISO 19115, INSPIRE und DCAT-AP. Ziel ist es, eine einheitliche semantische Repräsentation der Metadaten zu schaffen, die eine standardübergreifende Verwaltung und eine automatisierte Transformation zwischen den Formaten ermöglicht. Durch die zentrale Erfassung der Metadaten in einem ontologiegestützten Modell werden manuelle Mehrfacherfassungen überflüssig, die Datenkonsistenz verbessert und die Interoperabilität zwischen verschiedenen Geodateninfrastrukturen gestärkt.

Die entwickelte Ontologie wird in eine prototypische Implementierung eingebettet, die den gesamten Publikationsprozess unterstützt. Das vorgeschlagene Regelwerk ermöglicht dabei die automatisierte Übertragung von Informationen zwischen den Standards und zeigt am Beispiel einer Publikation eines Datensatzes über einen Dienst, die Möglichkeit einer automatisierten Anreicherung der Metadaten zwischen den Ressourcen. Dadurch wird nicht nur die Qualität der Metadaten verbessert, sondern auch der Aufwand für die Erfassung und Pflege erheblich reduziert.

Die Ergebnisse dieser Arbeit tragen zur Weiterentwicklung der semantischen Integration in Geodateninfrastrukturen bei, indem sie die Brücke zwischen XML-basierten Geodatenstandards, RDF-basierten Geodatenstandards und Wissensgraphen schlagen. Sie bieten eine praktische Lösung für die Harmonisierung heterogener Metadatenmodelle und leisten somit einen Beitrag zur nachhaltigen Verwaltung, Wiederverwendbarkeit und Auffindbarkeit von Geodaten in vernetzten Informationssystemen.



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar  
The approved original version of this thesis is available in print at TU Wien Bibliothek.

# Abstract

The management of metadata in geospatial data infrastructures presents a central challenge, particularly regarding interoperability between different standards such as ISO 19115, INSPIRE, and DCAT-AP. The lack of direct compatibility between these standards often requires geospatial data providers to record metadata multiple times across various tools and formats, leading to redundancies, inconsistencies, and significant administrative overhead. Additionally, the complexity of these standards can overwhelm many users, particularly data producers responsible for the initial metadata creation, who often lack expertise in metadata management.

This thesis develops an ontology-based solution to harmonize the metadata standards ISO 19115, INSPIRE, and DCAT-AP. The objective is to create a unified semantic representation of metadata that enables cross-standard management and automated transformation between formats. By capturing metadata centrally within an ontology-driven model, redundant manual entries are eliminated, data consistency is improved, and interoperability across different geospatial data infrastructures is enhanced.

The developed ontology is integrated into a prototype implementation that supports the entire publication process. The proposed rule-based framework facilitates the automated transfer of information between standards and demonstrates, using the example of dataset publication through a service, how metadata can be automatically enriched across resources. This approach not only improves metadata quality but also significantly reduces the effort required for metadata creation and maintenance.

The findings of this thesis contribute to the advancement of semantic integration in geospatial data infrastructures by bridging the gap between XML-based metadata standards, RDF-based metadata standards and knowledge graphs. The proposed solution provides a practical approach for harmonizing heterogeneous metadata models, offering a contribution to the sustainable management, reusability, and discoverability of geospatial data in interconnected information systems.



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar  
The approved original version of this thesis is available in print at TU Wien Bibliothek.

# Inhaltsverzeichnis

<b>Kurzfassung</b>	<b>xi</b>
<b>Abstract</b>	<b>xiii</b>
<b>Inhaltsverzeichnis</b>	<b>xv</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Problemstellung . . . . .	2
1.3 Beitrag der Arbeit . . . . .	4
1.4 State of the art und related work . . . . .	4
1.5 Aufbau der Arbeit . . . . .	6
<b>2 Einführung in Wissensmanagement mit Ontologien</b>	<b>7</b>
2.1 Relevanz und Anwendung . . . . .	7
2.2 Grundkonzepte und Begriffe . . . . .	8
2.3 Implementierung von Ontologien . . . . .	16
2.4 Abfrage und Validierung von Ontologien . . . . .	18
2.5 Ontologiearten . . . . .	20
2.6 Methoden zur Ontologieentwicklung . . . . .	21
2.7 Werkzeuge . . . . .	24
2.8 Einsatz von Ontologien in der Arbeit . . . . .	26
<b>3 Metadatenstandards in der Geoinformation</b>	<b>29</b>
3.1 Übersicht und historische Entwicklung der ISO 19100-Serie . . . . .	29
3.2 Aufbau eines minimalen ISO 19115 Metadatensatzes . . . . .	38
3.3 Aufbau eines minimalen INSPIRE Metadatensatzes . . . . .	41
3.4 Aufbau eines minimalen DCAT-AP Metadatensatzes . . . . .	44
<b>4 Harmonisierung der Metadatenstandards</b>	<b>47</b>
4.1 Vorgehen . . . . .	47
4.2 Klassen und Klassenhierarchie . . . . .	49
4.3 Eigenschaften und Einschränkungen . . . . .	56
4.4 Instanzen und Validierung . . . . .	57
	xv

4.5	Implementierung . . . . .	60
<b>5</b>	<b>Zusammenfassung und Ausblick</b>	<b>65</b>
5.1	Erreichte Ergebnisse . . . . .	66
5.2	Herausforderungen und nicht vollständig gelöste Aspekte . . . . .	66
5.3	Zukünftige Potenziale und Erweiterungsmöglichkeiten . . . . .	67
<b>6</b>	<b>Anhang</b>	<b>69</b>
6.1	Ontologie Begriffsdefinitionen . . . . .	69
6.2	Ontologie Klassenhierarchie . . . . .	73
6.3	Ontologie Objekt- und Dateneigenschaften . . . . .	75
6.4	Ontologie Axiome . . . . .	78
	<b>Abbildungsverzeichnis</b>	<b>83</b>
	<b>Listings</b>	<b>85</b>
	<b>Literaturverzeichnis</b>	<b>87</b>

# Einleitung

## 1.1 Motivation

Geografische Metadaten beschreiben Informationen und Ressourcen mit einer räumlichen Ausdehnung.[ISO14] Sie sind in Geodateninfrastrukturen eine wesentliche Informationsquelle, da sie sowohl für Menschen als auch für Maschinen verständliche Beschreibungen der Inhalte, der Entstehungshistorie, von Kontaktmöglichkeiten und technischen Eigenschaften von Datensätzen und Diensten bieten. Die Dokumentation dieser Eigenschaften gewährleistet die Wiederverwendbarkeit von Datenbeständen und Dienstinfrastrukturen.

Um ein gemeinsames Verständnis der zu erfassenden Metainformationen zu schaffen und die Austauschbarkeit bei der Datenweitergabe zu ermöglichen, wurden verschiedene Metadatenstandards entwickelt. Diese Standards normieren die Semantik, die Syntax sowie die verpflichtend und optional zu erfassenden Informationen. Allerdings hat die Vielzahl dieser Standards und deren unterschiedliche technologische Grundlagen zu einer steigenden Komplexität in der Erfassung und Verwaltung von Metadaten geführt.

Der breite Einsatz von Metadatenstandards in der Geoinformation begann 2003, als das technische Komitee ISO/TC 211 die Metadaten-Norm “ISO 19115:2003 - Geographic Information Metadata” veröffentlichte.[ISO03a] Dieser Standard definiert über 400 Metadatenattribute zur Beschreibung von Geodaten und Diensten, wobei je nach Typ der Ressource etwa 22 Elemente verpflichtend zu befüllen sind. ISO 19115 legte den Grundstein für die globale Interoperabilität von Geodaten, indem es ein einheitliches Modell zur Beschreibung von Geoinformationen und der Infrastruktur, die Geodaten bereitstellt, bietet.

Mit der Einführung der “RICHTLINIE 2007/2/EG DES EUROPÄISCHEN PARLAMENTS UND DES RATES zur Schaffung einer Geodateninfrastruktur in der Europäischen Gemeinschaft (INSPIRE)”[PC07] im Jahr 2007 erlangte die Harmonisierung von Geodaten auf europäischer Ebene eine neue Bedeutung. INSPIRE, ein Akronym für

**Infrastructure for Spatial Information in the European Community**, zielt darauf ab, eine gemeinsame verteilte Geodaten- und Dienstinfrastruktur in der Europäischen Union zu schaffen. Dazu wurde der Metadatenstandard “Technical Guidance for the implementation of INSPIRE dataset and service metadata based on ISO/TS 19139:2007” [INS24] entworfen der auf ISO 19115:2003 [ISO03a] und ISO/TS 19139:2007 [ISO07], den damals gültigen ISO Normen für Geodaten und Dienste-Metadaten, basiert. Allerdings gab es Unterschiede in den Anforderungen, was dazu führte, dass die Kompatibilität zwischen den Standards nicht vollständig gegeben ist.

Parallel dazu trieben politische Initiativen, wie der “BESCHLUSS DER KOMMISSION über die Weiterverwendung von Kommissionsdokumenten (2011/833/EU)” [Com07], die Bereitstellung von frei lizenzierten Dokumenten und Daten in der Europäischen Union voran. Daraus resultierten verschiedene neue Metadatenstandards auf nationaler und internationaler Ebene, wobei sich DCAT-AP [DCA20] für die Beschreibung der Datenkataloge, die für die Umsetzung der Entscheidung auf EU-Ebene bereitgestellt werden, durchgesetzt hat. Da Geodaten einen relevanten Bestandteil dieser Portale bilden, müssen auch hier Metadaten erfasst werden, jedoch bieten DCAT-AP, ISO 19115 und INSPIRE nur begrenzte Interoperabilität.

Im Jahr 2014 wurde die ISO 19115:2003 Norm aktualisiert und durch ISO 19115:2014 [ISO14] ersetzt. Diese Version ist nicht kompatibel mit ISO 19115:2003 und auch nicht kompatibel mit den INSPIRE Metadaten Technical Guidelines oder DCAT-AP.

Um Geodaten in Open Data-Portalen besser auffindbar und zugänglich zu machen, wurde 2016 GeoDCAT-AP [Geo20] entwickelt. GeoDCAT-AP ist eine Erweiterung von DCAT-AP, die Mappings zwischen INSPIRE Metadaten und DCAT-AP bereitstellt. Dieser Standard soll somit die Lücke zwischen INSPIRE Metadaten und DCAT-AP Metadaten schließen, berücksichtigt jedoch nicht die Inkompatibilität zu ISO 19115.

Die historische Einführung und Nutzung mehrerer Metadatenstandards in den verschiedenen Kontexten führt dazu, dass Geodatenstellen in der Europäischen Union in der Praxis mindestens drei Standards vorhalten sollten: ISO 19115, INSPIRE und DCAT-AP. Hinzu kommen oft interne organisations- oder fachspezifische Anforderungen an die Metadaten, welche in den Standards nicht abgebildet werden. Diese Fragmentierung und die Notwendigkeit, dieselben Informationen in mehreren Standards zu dokumentieren, steigern die Komplexität der Erfassung und Verwaltung erheblich und beeinträchtigen die Qualität und Konsistenz der Geodaten-Metadaten.

### 1.2 Problemstellung

Die Erfassung und Verwaltung von Metadaten in Geoinformationsinfrastrukturen ist komplex, da die etablierten Standards nicht nur Unterschiede in den Inhalten aufweisen, sondern auch auf unterschiedlichen technischen Grundlagen basieren.

ISO 19115 und INSPIRE basieren auf einem Datenmodell zur Beschreibung von Datensätzen und Diensten mit Raumbezug. Ein Metadatensatz beschreibt immer nur eine

Ressource, also einen Datensatz oder einen Dienst. Die Kodierung der Information basiert auf der Extensible Markup Language (XML).

DCAT-AP beschreibt ganze Datenkataloge und Ressourcen, die entweder Datensätze oder Dienste sein können. Dabei ist es im Gegensatz zu ISO19115 und INSPIRE möglich nicht nur Ressourcen mit Raumbezug zu beschreiben, sondern jegliche Art von Daten wie auch Dokumente. Der DCAT-AP-Standard basiert auf dem Resource Description Framework (RDF) welches für die semantische Beschreibung von Daten im Web entwickelt wurde auf Verlinkung und Interoperabilität ausgerichtet ist.

Metadatenkataloge, in denen Metadaten erstellt und verwaltet werden, unterstützen meist nur einen der genannten Standards. Daher kann es erforderlich sein, die relevanten Metadaten in mehreren Metadatenkatalogen zu erfassen, um sowohl die Anforderungen der ISO/INSPIRE-Standards als auch die der RDF-basierten DCAT-AP-Spezifikation zu erfüllen. Dies erhöht die Komplexität und den Aufwand für die Verwaltung der Metadaten.

Metadatenstandards sind in erster Linie darauf ausgelegt, den Austausch von Daten zwischen verschiedenen Organisationen zu erleichtern, insbesondere mit Blick auf die Veröffentlichung im Web. Die Publikation von Daten für organisationsexterne Akteure erfolgt dabei üblicherweise über etablierte Datenportale wie z.B.: das INSPIRE Geoportal<sup>1</sup> oder das Open Data Portal für Daten zu Europa<sup>2</sup>. Die Portale konzentrieren sich dabei die Informationen einem öffentlichen Nutzerkreis niederschwellig zu Verfügung zu stellen, bieten jedoch keine Möglichkeit zur Erfassung oder Verwaltung von Metadaten.

Ein weiteres Problem besteht darin, dass die Erheber der Geodaten meist jener Nutzerkreis sind, der auch die relevanten inhaltlichen Informationen zur Erstellung der zugehörigen Metadaten besitzt. Die Erheber sind jedoch in den meisten Fällen keine Metadatenexperten und die Komplexität der Metadatenstandards überfordert die Akteure. Dies führt dazu, dass der Prozess der Erfassung von Metadaten oft fehleranfällig ist, relevante Informationen nicht oder nicht vollständig befüllt werden oder auf die Erfassung der Metadaten gänzlich verzichtet wird. Die Mehrfacherfassung ähnlicher Informationen in verschiedenen Standards erzeugt dabei nicht nur Mehrarbeit, sondern auch Fragmentierungen und Inkonsistenzen, welche die Qualität der Metadaten weiter mindern und dadurch Wiederverwendbarkeit der Daten beeinträchtigen.

Die zentrale Problemstellung dieser Arbeit besteht darin, dass es bisher keine integrierte Lösung gibt, die eine einmalige Erfassung von Metadaten ermöglicht und gleichzeitig den unterschiedlichen Anforderungen der Metadatenstandards, ISO 19115, INSPIRE und DCAT-AP, gerecht wird. Aktuell verfügbare Implementierungen erfordern oft eine mehrfache Erfassung von Metadaten, um sowohl den Anforderungen öffentlicher Datenportale, als auch den internen Bedürfnissen einer Organisation zu entsprechen.

---

<sup>1</sup><https://inspire-geoportal.ec.europa.eu/>

<sup>2</sup><https://data.europa.eu/de>

### 1.3 Beitrag der Arbeit

Der Beitrag dieser Arbeit liegt in der Entwicklung einer Ontologie, die in der Lage ist, die Mindestanforderungen der Metadatenstandards ISO 19115, INSPIRE und DCAT-AP zu harmonisieren und die Verwaltung dieser Metadaten zu unterstützen. Die Ontologie bietet eine flexible, standardübergreifende semantische Datenhaltung, die es ermöglicht, Metadaten einmal zentral zu erfassen und diese dann in die verschiedenen Formate der relevanten Standards zu transformieren. Dadurch wird der Bedarf an manuellen Mehrfacherfassungen eliminiert, was die Effizienz der Metadatenverwaltung erheblich steigert.

Durch die Einbettung der Ontologie in eine python-Implementierung wird ein Publikationsprozess ermöglicht, bei dem gezeigt wird, wie durch die Implementierung eines Regelwerks und den Einsatz von logischen Schlussfolgerungen Informationen automatisch zwischen Ressourcen übertragen werden. Dies führt nicht nur zu einer Steigerung der Konsistenz, sondern auch zu einem erhöhten Informationsgehalt in den erstellten Metadatensätzen.

Die Arbeit liefert nicht nur eine theoretische Grundlage für die Harmonisierung der Metadatenstandards, sondern beinhaltet auch eine praktische Implementierung der Ontologie. Dabei wird gezeigt, wie Metadaten, die den Standards ISO 19115, INSPIRE und DCAT-AP entsprechen, einmalig erfasst und in die entsprechenden Formate überführt werden können.

### 1.4 State of the art und related work

In der Arbeit "Geographic Information Metadata - An Outlook from the International Standardization Perspective" von Brodeur et al. [BCD<sup>+</sup>19] wird die Entwicklung und Standardisierung von Metadaten für Geoinformationen untersucht. Die Autoren betonen, dass Metadaten bereits in der analogen Kartografie eine wichtige Rolle spielten und mit der Digitalisierung zunehmend in standardisierten Formaten erfasst wurden. Dabei wird die Arbeit der ISO/TC 211 hervorgehoben, die mit Normen wie ISO 19115 eine Grundlage für die Beschreibung, den Austausch und die Nutzung von Geodaten geschaffen hat. Dennoch bleibt die Interoperabilität zwischen verschiedenen nationalen und domänenspezifischen Metadatenprofilen eine Herausforderung. Zur Lösung dieses Problems wurden spezifische Profile wie das North American Profile, das DGIWG-Profil (für den Verteidigungsbereich) und das INSPIRE-Profil entwickelt, um eine einheitliche Nutzung von Metadaten in unterschiedlichen Anwendungsbereichen zu ermöglichen. Neben Standardisierungsbemühungen werden auch technologische Fortschritte bei der Verwaltung und Nutzung von Metadaten diskutiert. Dazu gehören XML basierte Ansätze, Ontologien für die semantische Interoperabilität sowie Web-Technologien wie DCAT, die den Austausch und die Integration von Metadaten erleichtern. Darüber hinaus stellt die Arbeit verschiedene Softwarelösungen wie GeoNetwork, Esri Geoportal Server und Catalogue Service for the Web (CSW) vor, mit denen Metadaten erfasst und bereit-

gestellt werden können. Trotz dieser Fortschritte bleibt die Pflege von Metadaten eine Herausforderung, da sie oft als zu aufwendig betrachtet wird. Die Autoren schlagen daher automatisierte Methoden zur Metadatengenerierung und Qualitätsbewertung vor und betonen die Notwendigkeit, Metadaten besser in den Semantic Web-Kontext einzubinden, um ihre Auffindbarkeit und Nutzbarkeit weiter zu verbessern.

Die Arbeit “Describing Geospatial Assets in the Web of Data: A Metadata Management Scenario” [FPO<sup>+</sup>16] behandelt die Verwaltung von Metadaten für geografische Daten im Kontext moderner Webtechnologien. Metadaten sind entscheidend für die Entdeckung und Nutzung dieser Daten, insbesondere weil geografische Informationen meist nicht als Text bereitgestellt werden und klassische Suchmaschinen hier an ihre Grenzen stoßen. Die Autoren präsentieren ein flexibles, template-basiertes Bearbeitungstool, das verschiedene XML-Schemen unterstützt und externe Datenquellen für automatische Vervollständigung einbindet. Durch die Nutzung von RDF basierten Metadaten werden neue semantische Such- und Verwaltungsmöglichkeiten geschaffen, die über klassische ISO-Standards hinausgehen. Damit wird der Übergang von starren, textbasierten Metadaten hin zu dynamischen, vernetzten Informationen erleichtert. Anhand von zwei Anwendungsfällen zeigen die Autoren, wie ihr System externe Datenquellen nutzt: Einerseits zur Verwaltung von Kontaktinformationen, andererseits zur Integration von Schlagworten aus kontrollierten Vokabularen. Dabei wird deutlich, dass redundante Metadatenangaben vermieden und gleichzeitig die Konsistenz verbessert werden kann. Das System ermöglicht es, Metadaten automatisch zu aktualisieren und mit bestehenden Datenbeständen zu verknüpfen.

In dem Paper “Raising Semantics-Awareness in Geospatial Metadata Management” von Fugazza et al. [FPO<sup>+</sup>18] wird die Problematik veralteter und ineffizient kodierter Geodaten Metadaten behandelt. Die Autoren zeigen auf, dass bestehende Metadatenmodelle oft isolierte und wenig interoperable Datensätze erzeugen, welche die semantische Verknüpfung und eine automatisierte Verarbeitung erschweren. Ihr Ansatz sieht eine verteilte Metadatenverwaltung vor, indem Metadaten nicht mehr als statische Dokumente, sondern als vernetzte RDF-Strukturen modelliert werden. Das ermöglicht eine automatische Aktualisierung von Metadaten, indem relevante Entitäten über URIs eindeutig referenziert werden. Durch die Anwendung dieser Methodik auf INSPIRE-Metadaten demonstrieren die Autoren, dass semantisch beschriebene Metadaten nicht nur die Konsistenz erhöhen, sondern auch die Wiederverwendbarkeit und Interoperabilität in Geodateninfrastrukturen verbessern. Der vorgestellte EDI-Editor erlaubt dabei eine nutzerfreundliche Erstellung und Verwaltung von Metadaten, indem er semantische Informationen direkt in die Bearbeitung einbindet. Das Paper zeigt, dass eine Verknüpfung von Metadaten mit RDF-Quellen eine neue, dynamischere Form der Metadatenverwaltung ermöglicht, die über klassische XML-basierte Modelle hinausgeht.

“Describing Geospatial Assets in the Web of Data: A Metadata Management Scenario” ebenfalls von Fugazza et al. [FPO<sup>+</sup>16] beschreibt eine flexible und semantisch angereicherte Lösung für die Verwaltung von Geodaten Metadaten. Die Autoren argumentieren, dass herkömmliche Metadatenmodelle oft starr und wenig anpassbar sind, wodurch sich die Verwaltung und Aktualisierung erschwert. Sie schlagen daher ein templatebasiertes

Metadaten Editorframework vor, das sich auf verschiedene XML-basierte Metadaten-schemata anpassen lässt und gleichzeitig eine RDF-basierte Repräsentation erzeugt. Das ermöglicht die Integration externer Datenquellen zur semantischen Anreicherung und Autovervollständigung von Metadaten, wodurch eine dynamischere und interoperablere Metadatenverwaltung entsteht. Durch die Anwendung in einem italienischen Forschungsprojekt (RITMARE) demonstrieren die Autoren, wie ihr Ansatz Metadaten in Echtzeit aktualisieren und konsistenter halten kann, insbesondere wenn sich Organisationsstrukturen oder Terminologien ändern. Die Kombination aus INSPIRE-kompatibler Metadatenverwaltung und Semantic-Web-Technologien erleichtert nicht nur die Datenauffindbarkeit, sondern eröffnet auch neue Möglichkeiten der semantischen Suche und Datenverknüpfung.

### 1.5 Aufbau der Arbeit

#### **Kapitel 1: Einleitung**

Die Einleitung gibt einen Überblick über die Relevanz der Thematik und stellt die Motivation hinter dieser Arbeit dar. Es werden die Problemstellung und die Herausforderungen bei der Verwaltung von Metadatenstandards in Geodateninfrastrukturen erläutert. Zudem wird der wissenschaftliche Beitrag der Arbeit beschrieben und ein Überblick über den aktuellen Stand der Forschung gegeben.

#### **Kapitel 2: Einführung in Wissensmanagement mit Ontologien**

Dieses Kapitel bietet eine Einführung in Ontologien als Methode des Wissensmanagements. Es erläutert grundlegende Konzepte und Begriffe sowie Techniken zur Implementierung und Abfrage von Ontologien. Zudem werden verschiedene Arten von Ontologien und Methoden zu deren Entwicklung diskutiert.

#### **Kapitel 3: Metadatenstandards in der Geoinformation**

Hier werden die wesentlichen Metadatenstandards ISO 19115, INSPIRE und DCAT-AP detailliert vorgestellt. Die historische Entwicklung sowie die strukturellen und semantischen Unterschiede dieser Standards werden analysiert. Zudem wird dargestellt, wie ein minimaler Metadatensatz für jeden dieser Standards aufgebaut ist.

#### **Kapitel 4: Harmonisierung der Metadatenstandards**

In diesem Kapitel wird der methodische Ansatz zur Harmonisierung der drei Metadatenstandards beschrieben. Es wird ein Ontologiemodell, einschließlich der notwendigen Klassenhierarchien, Eigenschaften und Einschränkungen vorgestellt. Die Implementierung sowie die Validierung der harmonisierten Metadatenmodelle werden ebenfalls behandelt.

#### **Kapitel 5: Zusammenfassung und Ausblick**

Dieses Kapitel fasst die Ergebnisse der Arbeit zusammen und reflektiert die erzielten Fortschritte im Bereich der Metadatenharmonisierung. Darüber hinaus werden offene Fragen und mögliche zukünftige Forschungsrichtungen aufgezeigt.

# Einführung in Wissensmanagement mit Ontologien

## 2.1 Relevanz und Anwendung

Ontologien dienen als Mittel zur Wissensrepräsentation und bieten eine strukturierte Form zur Beschreibung von Daten, Konzepten und deren Beziehungen zueinander. Sie helfen Daten und Konzepte zu standardisieren, zu organisieren und maschinenlesbar zu machen. Eine Ontologie definiert dabei nicht nur die Begriffe einer bestimmten Domäne, sondern auch deren Hierarchien, Attribute und Beziehungen. Über Annotationen lassen sich die einzelnen Bausteine einer Ontologie auch noch näher beschreiben und definieren, dadurch ist es möglich das Wissensmodell zusätzlich mit Metainformationen zu versehen. Durch diese Strukturierung des Wissens wird es möglich, Information konsistent zu repräsentieren, unabhängig von ihrer Herkunft oder dem Kontext ihrer Erzeugung.

Das Konzept der Ontologie hat seinen Ursprung in der Philosophie, wo sie dazu dient, die Natur von Existenz und Kategorien zu definieren. In der Informatik und im Bereich der Wissensrepräsentation hat sich dieser Begriff weiterentwickelt und wird genutzt, um einen formalen Rahmen zur Modellierung von Wissen zu bieten. Eine Ontologie kann beispielsweise festlegen, dass ein *Patient* eine Unterklasse von *Person* ist und dass ein *Arzt* eine *Person* mit bestimmten Fähigkeiten und Qualifikationen ist. *Arzt* und *Patient* sind daher spezielle Personen. Diese formale Struktur erlaubt es, Beziehungen zwischen diesen Konzepten zu definieren, wie z.B. *Ein Arzt behandelt Patienten*.

Ontologien spielen eine zentrale Rolle in einer Vielzahl von Anwendungsbereichen, da sie eine gemeinsame und standardisierte Sprache bieten, um Wissen und Daten zu beschreiben. Eine der wichtigsten Anwendungen von Ontologien ist das Semantic Web, das von der

Idee getrieben wird, das Internet in ein semantisches Netzwerk zu verwandeln, in dem Informationen nicht nur durch Menschen, sondern auch durch Maschinen genutzt werden können. Ontologien bilden hier das Rückgrat, da sie definieren, wie Daten maschinenlesbar und semantisch verknüpft werden können. Durch die Verwendung von Ontologien im Semantic Web können Suchanfragen und Datenaustausch über Domänengrenzen hinweg gewährleistet werden.

Im Bereich der Datenintegration und Dateninteroperabilität ermöglichen Ontologien die Zusammenführung heterogener Datenquellen. Dies ist besonders in der Wissenschaft und Forschung von Bedeutung, wo Daten aus unterschiedlichen Disziplinen und Systemen zusammengeführt werden müssen. Eine Ontologie kann hier die Gemeinsamkeiten und Unterschiede zwischen den Datenquellen abbilden und so eine harmonisierte Sicht auf die Daten ermöglichen. Dies ist beispielsweise im Gesundheitswesen entscheidend, wo Ontologien dazu beitragen können, Patientendaten aus verschiedenen Krankenhäusern, Labors und Forschungsprojekten zu integrieren und zu harmonisieren, um eine umfassendere Sicht auf den Gesundheitszustand des Patienten zu ermöglichen.

Im E-Commerce ermöglichen Ontologien die Verbesserung der Produktsuche und -empfehlung, indem sie ein semantisches Verständnis von Produkten und deren Eigenschaften ermöglichen. So können Ontologien z.B. Kategorien und Beziehungen zwischen Produkten definieren, wodurch intelligente Suchfunktionen und Empfehlungsalgorithmen möglich werden. Das britische BBC verwendet Ontologien, um die Inhalte seines Webauftritts semantisch zu beschreiben, miteinander zu verknüpfen und durchsuchbar zu machen.<sup>1</sup>

### 2.2 Grundkonzepte und Begriffe

Ontologien setzen sich aus mehreren Bausteinen zusammen, die gemeinsam Wissen strukturiert und formal abbilden. Diese Bausteine lassen sich in zwei Hauptkategorien unterteilen: die sogenannte **Terminological Box (TBox)** und die **Assertional Box (ABox)**. Die TBox enthält das konzeptionelle Wissen, das die Struktur und die Regeln der Ontologie definiert, während die ABox konkrete Informationen über Instanzen, Eigenschaften deren Beziehungen enthält.

Folgender Text repräsentiert konkretes Wissen (ABox): *Frau Huber wohnt in Melk. Sie ist Patientin Krankenhaus Melk, wo sie wegen einer starken Erkältung behandelt wird. Ihr behandelnder Arzt, Dr. Gruber, hat ein Medizinstudium abgeschlossen und arbeitet seit vielen Jahren im Krankenhaus Melk als Arzt. Dr. Gruber ist ein erfahrener Allgemeinmediziner und therapiert Frau Huber. Der Chefarzt des Krankenhauses, Dr. Schmidt, ist Dr. Grubers Vorgesetzter und leitet das Krankenhaus Melk. Da Frau Huber an einer speziellen Atemwegserkrankung leidet, überweist Dr. Gruber sie zu Kollegen Dr. Meier, einen Spezialisten für Atemwegserkrankungen, der als Facharzt ebenfalls im Krankenhaus Melk arbeitet.*

---

<sup>1</sup><https://www.bbc.co.uk/ontologies>

Dieser Text kann verallgemeinert werden, sodass man nicht mehr konkret von Frau Huber, Dr. Gruber oder einer Erkältung spricht, sondern von Patienten, Ärzten und Krankheiten (TBox): *Es gibt Personen, von denen einige Ärzte sind. Ärzte haben ein Medizinstudium abgeschlossen und arbeiten in Krankenhäusern. Sie behandeln Patienten, die an Krankheiten leiden. Jeder Arzt hat einen Vorgesetzten, der Chefarzt genannt wird. Der Chefarzt ist Leiter des Krankenhauses. Manchmal müssen Ärzte ihre Patienten zu anderen Ärzten mit einer Spezialisierung überweisen, diese Ärzte nennt man Spezialisten oder Fachärzte. Innerhalb eines Krankenhauses sind alle Ärzte untereinander Kollegen und bilden so das Kollegium.*

### 2.2.1 Klassen und Hierarchien

**Klassen** repräsentieren die Konzepte einer Ontologie und bilden die Typen von Objekten, die in einer bestimmten Domäne von Interesse sind. In unserem Beispieltext wären das z.B.: Personen, Patienten, Ärzte, Krankheiten, Krankenhäuser, Chefarzte, Spezialisten und Kollegen. Für die Beschreibung der Existenz eines Konzepts der Klasse *Person* kann die Prädikatenlogik verwendet werden:

$$\exists x (\text{Person}(x))$$

Der Ausdruck besagt: Es existiert mindestens eine Entität  $x$ , die eine *Person* ist.

Eine Auflistung von Konzepten mit deren Definition wird auch **kontrolliertes Vokabular** genannt. Ein kontrolliertes Vokabular bildet die Grundstruktur jeder Ontologie. Das kontrollierte Vokabular für unseren Beispieltext kann folgendermaßen aussehen:

$$\exists x (\text{Person}(x)) = \text{def. ein menschliches Wesen.}$$

$$\exists x (\text{Arzt}(x)) = \text{def. eine Person mit Medizinstudium.}$$

$$\exists x (\text{Patient}(x)) = \text{def. eine Person die an einer Krankheit leidet.}$$

$$\exists x (\text{Krankheit}(x)) = \text{def. ein gesundheitlicher Zustand, der eine Person beeinträchtigt.}$$

$$\exists x (\text{Krankenhaus}(x)) = \text{def. eine Einrichtung für medizinische Behandlungen von Patienten.}$$

$$\exists x (\text{Chefarzt}(x)) = \text{def. ein Arzt der ein Krankenhaus leitet.}$$

$$\exists x (\text{Spezialisierung}(x)) = \text{def. eine spezielle fachliche Ausbildung.}$$

$$\exists x (\text{Spezialist}(x)) = \text{def. eine Arzt mit einer Spezialisierung.}$$

$$\exists x (\text{Facharzt}(x)) = \text{def. ein Arzt mit einer Spezialisierung.}$$

$$\exists x (\text{Kollege}(x)) = \text{def. ein Arzt der im selben Krankenhaus arbeitet.}$$

**Hierarchien** sind ein Konzept zur Strukturierung von Wissen und zur Darstellung von Beziehungen zwischen verschiedenen Konzepten. Klassen in Ontologien können durch Hierarchien organisiert werden, die durch *is\_a*-Beziehungen (Subklassen-Beziehungen) ausgedrückt werden. Diese *is\_a*-Beziehungen zeigen eine Spezialisierung an, wobei eine untergeordnete Klasse eine spezifischere Form der übergeordneten Klasse darstellt.

$$\forall x (\text{Arzt}(x) \Rightarrow \text{Person}(x))$$

Der Ausdruck besagt: Für jede Entität  $x$  gilt, dass, wenn  $x$  ein *Arzt* ist, dann ist  $x$  auch eine *Person*. Dies bedeutet, dass alle Ärzte Personen sind, oder anders ausgedrückt, die Klasse *Arzt* ist vollständig in der Klasse *Person* enthalten.

Die Strukturierung eines kontrollierten Vokabulars wird auch **Taxonomie** genannt. Die Taxonomie für das Beispiel sieht folgendermaßen aus:

$$\forall x (\text{Patient}(x) \Rightarrow \text{Person}(x))$$

$$\forall x (\text{Arzt}(x) \Rightarrow \text{Person}(x))$$

$$\forall x (\text{Chefarzt}(x) \Rightarrow \text{Arzt}(x))$$

$$\forall x (\text{Spezialist}(x) \Rightarrow \text{Arzt}(x))$$

$$\forall x (\text{Facharzt}(x) \Rightarrow \text{Arzt}(x))$$

$$\forall x (\text{Kollege}(x) \Rightarrow \text{Arzt}(x))$$

In der Modellierung des Beispiels wurde ein Kollege als *ein Arzt der im selben Krankenhaus arbeitet* definiert, daher wurde *Kollege* als eine Spezialisierung von *Arzt* modelliert. Würde man die Definition verallgemeinern z.B.: *def. eine Person die in derselben Organisation arbeitet*, könnte man *Kollege* als Spezialisierung von *Person* modellieren.

### Äquivalenz

Zwei Klassen können als **äquivalent** betrachtet werden, wenn sie dieselben Individuen teilen. Das bedeutet, jedes Individuum, das zur ersten Klasse gehört, gehört auch zur zweiten Klasse, und umgekehrt. Im Beispiel werden *Facharzt* und *Spezialist* synonym verwendet und haben auch die gleiche Definition, daher können sie als äquivalent betrachtet werden.

$$\forall x (\text{Facharzt}(x) \Leftrightarrow \text{Spezialist}(x))$$

Dieser Ausdruck besagt: Für jede Entität  $x$  gilt, dass  $x$  ein *Facharzt* ist, wenn und nur wenn  $x$  ein *Spezialist* ist.

### Disjunktion

Zwei Klassen sind **disjunkt**, wenn sie keine gemeinsamen Individuen haben. Das bedeutet, dass ein Individuum nicht gleichzeitig zu beiden Klassen gehören kann. Die Klassen *Arzt* und *Krankheit* sind disjunkt, da ein Individuum nicht gleichzeitig ein *Arzt* und eine *Krankheit* sein kann:

$$\forall x (\text{Arzt}(x) \Rightarrow \neg \text{Krankheit}(x))$$

Dieser Ausdruck besagt: Für jede Entität  $x$  gilt, wenn  $x$  ein Arzt ist, dann ist  $x$  keine Krankheit.

### 2.2.2 Objekteigenschaften

**Objekteigenschaften (Object Properties)** beschreiben die Beziehungen zwischen den Klassen einer Ontologie und definieren, wie diese Konzepte miteinander interagieren. Jedes Objekteigenschaft hat eine **Domäne (Domain)** und einen **Zielbereich (Range)**, die angeben, welche Klassen durch die Beziehung miteinander verbunden sind. Die Domain definiert die Klasse, aus der das **Subjekt** der Beziehung stammt, während die Range die Klasse bestimmt, zu der das **Objekt** gehört. Die Beziehung selbst bildet das **Prädikat**. Zum Beispiel könnte die Objekteigenschaft *therapiert* in der Ontologie die Domain *Arzt* und Range *Patient* haben. Das bedeutet, dass nur Individuen der Klasse *Arzt* Subjekte der *therapiert*-Beziehung sein können und nur Individuen der Klasse *Patient* als Objekte in dieser Beziehung stehen können. Das bedeutet im Umkehrschluss, dass alles was *therapiert* wird automatisch ein *Patient* ist und jeder der *therapiert* automatisch ein *Arzt* ist. Formal kann dies wie folgt ausgedrückt werden:

$$\forall x (\text{Arzt}(x) \Rightarrow \exists y (\text{therapiert}(x, y) \wedge \text{Patient}(y)))$$

Dieser Ausdruck besagt: Für jede Entität  $x$  gilt, wenn  $x$  ein *Arzt* ist, dann existiert eine Entität  $y$ , die  $x$  *therapiert*, und  $y$  ist ein *Patient*. Mit anderen Worten, ein *Arzt* *therapiert* *Patienten*.

#### Funktionale Objekteigenschaften

Eine Eigenschaft ist **funktional**, wenn jedes Subjekt dieser Eigenschaft mit höchstens einem Objekt verbunden sein kann. Zum Beispiel kann jeder *Chefarzt* nur einem *Krankenhaus* zugeordnet sein:

$$\forall x (\text{Chefarzt}(x) \Rightarrow \exists^1 y (\text{arbeitetIn}(x, y) \wedge \text{Krankenhaus}(y)))$$

Dieser Ausdruck besagt: Für jede Entität  $x$  gilt, wenn  $x$  ein *Chefarzt* ist, dann existiert genau ein  $y$ , in dem  $x$  arbeitet, und  $y$  ist ein *Krankenhaus*. Mit anderen Worten, jeder *Chefarzt* *arbeitet in genau einem Krankenhaus*.

Eine **invers funktionale** Objekteigenschaft bedeutet, dass jedes Objekt mit höchstens einem Subjekt verbunden sein kann. Zum Beispiel kann man damit ausdrücken das jedes

Krankenhaus nur einen Chefarzt haben kann, ohne dafür eine neue Relation einführen zu müssen:

$$\forall y (\text{Krankenhaus}(y) \Rightarrow \exists^1 x (\text{Chefarzt}(x) \wedge \text{arbeitetIn}(x, y)))$$

Dieser Ausdruck besagt: Für jede Entität  $y$  gilt, wenn  $y$  ein *Krankenhaus* ist, dann existiert genau eine Entität  $x$ , die  $y$  als *Chefarzt* zugeordnet ist und in  $y$  arbeitet. Mit anderen Worten, jedes *Krankenhaus* hat genau einen *Chefarzt*.

Ist ein Property gleichzeitig funktional und invers funktional, dann drückt das eine 1:1 Beziehung aus.

### Transitive Objekteigenschaften

Eine Eigenschaft ist **transitiv**, wenn gilt: Wenn  $x$  mit  $y$  und  $y$  mit  $z$  durch die Eigenschaft verbunden ist, dann ist  $x$  auch mit  $z$  verbunden. Zum Beispiel könnte eine Hierarchiebeziehung zwischen *Arzt* und *Chefarzt* transitiv sein:

$$\forall x \forall y \forall z (\text{Chefarzt}(x) \wedge \text{istVorgesetzterVon}(x, y) \\ \wedge \text{istKollegeVon}(y, z) \Rightarrow \text{istVorgesetzterVon}(x, z))$$

Dieser Ausdruck besagt: Wenn  $x$  ein *Chefarzt* ist und *Vorgesetzter* von  $y$  ist, und  $y$  ein *Kollege* von  $z$  ist, dann ist auch  $x$  *Vorgesetzter* von  $z$ . Mit anderen Worten, ein *Chefarzt* ist automatisch der *Vorgesetzte* der anderen *Ärzte* des Krankenhauses.

### Symmetrische Objekteigenschaften

Eine Eigenschaft ist **symmetrisch**, wenn gilt: Wenn  $x$  mit  $y$  durch die Eigenschaft verbunden ist, dann ist auch  $y$  mit  $x$  verbunden. Ein Beispiel wäre die Beziehung *istKollegeVon*:

$$\forall x \forall y (\text{istKollegeVon}(x, y) \Rightarrow \text{istKollegeVon}(y, x))$$

Dieser Ausdruck besagt: Wenn ein *Arzt*  $x$  ein *Kollege* von *Arzt*  $y$  ist, dann ist auch *Arzt*  $y$  ein *Kollege* von *Arzt*  $x$ . Mit anderen Worten, die Beziehung *istKollegeVon* ist symmetrisch, da Kollegialität immer in beide Richtungen gilt.

### Asymmetrische Objekteigenschaften

Eine Eigenschaft ist **asymmetrisch**, wenn gilt: Wenn  $x$  mit  $y$  durch die Eigenschaft verbunden ist, dann kann  $y$  nicht mit  $x$  verbunden sein. Zum Beispiel ist die Beziehung *istVorgesetzterVon* asymmetrisch:

$$\forall x \forall y (\text{istVorgesetzterVon}(x, y) \Rightarrow \neg \text{istVorgesetzterVon}(y, x))$$

Dieser Ausdruck besagt: Wenn eine *Person*  $x$  der *Vorgesetzte* von  $y$  ist, dann kann  $y$  nicht der *Vorgesetzte* von  $x$  sein. Mit anderen Worten, die Beziehung *istVorgesetzterVon* ist asymmetrisch, da Vorgesetztenverhältnisse nicht in beide Richtungen gelten können, da man nicht selbst sein Vorgesetzter sein kann.

### Reflexive Objekteigenschaften

Eine Eigenschaft ist **reflexiv**, wenn jedes Individuum mit sich selbst verbunden ist. Zum Beispiel könnte die Eigenschaft *therapiert* reflexiv sein:

$$\forall x (\text{Arzt}(x) \Rightarrow \text{therapiert}(x, x))$$

Dieser Ausdruck besagt: Für jede Entität  $x$ , die ein *Arzt* ist, gilt, dass sie sich selbst therapieren kann. Mit anderen Worten, die Eigenschaft *therapiert* ist reflexiv, da ein *Arzt* in der Lage ist, sich selbst zu therapieren.

### Irreflexive Objekteigenschaften

Eine Eigenschaft ist **irreflexiv**, wenn kein Individuum mit sich selbst verbunden sein kann. Zum Beispiel ist *istVorgesetzterVon* irreflexiv:

$$\forall x (\text{Person}(x) \Rightarrow \neg \text{istVorgesetzterVon}(x, x))$$

Dieser Ausdruck besagt: Für jede Entität  $x$ , die eine *Person* ist, gilt, dass sie nicht ihr eigener *Vorgesetzter* sein kann. Mit anderen Worten, die Eigenschaft *istVorgesetzterVon* ist irreflexiv, da niemand gleichzeitig sein eigener Vorgesetzter sein kann.

### Äquivalenz von Objekteigenschaften

Zwei Objekteigenschaften können **äquivalent** sein, wenn sie dieselbe Beziehung beschreiben. Dies bedeutet, dass, wenn die eine Beziehung gilt, auch die andere gilt. Zum Beispiel können die Beziehungen *behandelt* und *therapiert* äquivalent sein:

$$\forall x \forall y (\text{behandelt}(x, y) \Leftrightarrow \text{therapiert}(x, y))$$

Dieser Ausdruck besagt: Für jede Entität  $x$  und  $y$  gilt, wenn  $x$   $y$  *behandelt*, dann *therapiert*  $x$   $y$  und umgekehrt. Mit anderen Worten, die Beziehungen *behandelt* und *therapiert* sind äquivalent, da sie dieselbe Beziehung zwischen den Entitäten  $x$  und  $y$  ausdrücken.

### Disjunkte Objekteigenschaften

Zwei Objekteigenschaften sind **disjunkt**, wenn sie keine gemeinsamen Paare von Subjekten und Objekten haben können. Zum Beispiel können die Beziehungen *istVorgesetzterVon* und *istUntergebenerVon* disjunkt sein:

$$\forall x \forall y (\text{istVorgesetzterVon}(x, y) \Rightarrow \neg \text{istUntergebenerVon}(x, y))$$

Dieser Ausdruck besagt: Für jede Entität  $x$  und  $y$  gilt, wenn  $x$  der Vorgesetzte von  $y$  ist, dann kann  $x$  nicht gleichzeitig der Untergebene von  $y$  sein. Mit anderen Worten, die Beziehungen *istVorgesetzterVon* und *istUntergebenerVon* sind disjunkt, da eine Entität nicht gleichzeitig Vorgesetzter und Untergebener derselben Entität sein kann.

### 2.2.3 Dateneigenschaften

**Dateneigenschaften (Data Properties)** sind Eigenschaften, die spezifische Merkmale oder Attribute einer Klasse beschreiben. Sie liefern zusätzliche Informationen, die für die Individuen der Klassen relevant sind. Zum Beispiel hat eine Person einen Namen. Diese Eigenschaften können über *is\_a*-Beziehungen vererbt werden. Wenn eine *Person* einen Namen hat und ein *Arzt* eine Unterklasse von *Person* ist, dann hat auch ein *Arzt* einen Namen.

Dateneigenschaften können funktional sein, was bedeutet, dass jede Instanz für ein bestimmtes Attribut nur einen Wert haben darf. Zum Beispiel hat jede Person genau einen Namen. Dies wird durch folgende formale Aussage beschrieben:

$$\forall x (\text{Person}(x) \Rightarrow \exists^=1 y (\text{hatName}(x, y) \wedge \text{string}(y)))$$

Der Ausdruck besagt, dass jedes Individuum  $x$ , die eine *Person* ist, genau einen Namen  $y$  hat, der ein String ist.

Dateneigenschaften haben wie Objekteigenschaften eine Domain und eine Range, wobei die Range nur primitive Datentypen wie Strings oder Zahlen enthalten darf und nicht auf andere Klassen zeigen darf. Auch die Äquivalenz und Disjunktheit sind gleich zu verstehen.

### 2.2.4 Individuals

**Individuen (Individuals)** sind konkrete Objekte einer Klasse und repräsentieren Instanzen realer Entitäten. Sie gehören zur ABox, welche spezifische Fakten über die Welt enthält. Beispielsweise kann *Frau Huber* eine Instanz der Klasse *Patient* sein, während *Dr. Gruber* eine Instanz der Klasse *Arzt* ist. **Anonyme Individuals** dienen als Platzhalter für beliebige Instanzen, die nicht spezifiziert oder identifiziert sind. **Named Individuals**

repräsentieren konkrete Objekte und sind durch eine eindeutige ID oder einen Namen identifiziert. Beispiele für Named Individuals in einer Ontologie sind:

Patient(*FrauHuber*)

Arzt(*Dr.Gruber*)

Chefarzt()

Diese Ausdrücke bedeuten: *Frau Huber* ist ein Individuum der Klasse *Patient*, und *Dr. Gruber* ist ein Individuum der Klasse *Arzt*. Weiters gibt es noch ein anonymes Individuum *Chefarzt*, das zwar existiert, aber nicht näher spezifiziert ist.

### 2.2.5 Axiome

**Axiome** definieren allgemeine Regeln und Beziehungen zwischen den Klassen innerhalb einer Ontologie. Sie legen fest, unter welchen Bedingungen bestimmte logische Schlussfolgerungen getroffen werden können. Ein Beispiel für ein Axiom, das beschreibt, dass jedes Individuum einer *Person*, die an einer *Krankheit* leidet, auch ein *Patient* ist, lautet:

$$\forall x (\text{Person}(x) \wedge \exists y (\text{hatKrankheit}(x, y)) \Rightarrow \text{Patient}(x))$$

Dieser Ausdruck besagt: Für jede Entität  $x$ , die eine *Person* ist, und für die eine *Krankheit*  $y$  existiert, an der  $x$  leidet, folgt, dass  $x$  ein *Patient* ist. Mit anderen Worten, wenn eine Person eine Krankheit hat, ist diese Person auch automatisch ein Patient.

### 2.2.6 Restriktionen

**Restriktionen** schränken die zulässigen Beziehungen oder Werte von Klassen und Instanzen ein und sorgen für die Konsistenz innerhalb der Ontologie. Sie stellen sicher, dass bestimmte Bedingungen für ein Individuum einer Klasse erfüllt werden müssen. Zum Beispiel könnte eine Restriktion vorschreiben, dass ein *Patient* nur an einen Spezialist überwiesen werden darf:

$$\forall x (\text{Patient}(x) \Rightarrow \forall y (\text{überweist}(x, y) \Rightarrow \text{Spezialist}(y)))$$

Dieser Ausdruck besagt: Für jede Entität  $x$ , die ein *Patient* ist, und für jede Entität  $y$ , an die  $x$  überwiesen wird, muss gelten, dass  $y$  ein Spezialist ist. Mit anderen Worten, Patienten dürfen nur an Ärzte überwiesen werden, die Spezialisten sind.

### 2.2.7 Reasoning

Ein **Reasoner** ist ein logisches System, das verwendet wird, um implizites Wissen in einer Ontologie abzuleiten und die Konsistenz der Ontologie zu überprüfen. Er führt die definierten Regeln aus und ermöglicht es so, automatisch Schlussfolgerungen zu ziehen, die nicht explizit in der Ontologie definiert sind, sich aber aus dem Regelwerk ergeben.

Es gibt verschiedene Reasoner, die sich in ihrer Performance, Funktionsumfang und den unterstützten OWL-Profilen unterscheiden. Bekannte Implementierungen sind Hermit, ELK und Pellet.

**Hermit**<sup>2</sup> [GHM<sup>+</sup>14] ist ein leistungsstarker Reasoner, der die vollständige OWL 2 DL-Spezifikation unterstützt. Er eignet sich besonders für komplexe Ontologien, die logische Abhängigkeiten, Kardinalitätsbeschränkungen und disjunkte Klassen enthalten. Hermit verwendet ein Tableau-Verfahren, das es ermöglicht, auch anspruchsvolle logische Konstrukte zu verarbeiten. Allerdings kann Hermit bei sehr großen Ontologien langsam arbeiten, da das Verfahren rechenintensiv ist.

**ELK**<sup>3</sup> [KKS12] ist ein schneller Reasoner, der auf das OWL 2 EL-Profil spezialisiert ist, einer Teilmenge von OWL 2 DL. ELK ist für sehr große Ontologien ideal, die einfache Inferenzregeln und Hierarchien verwenden, aber keine komplexen logischen Konstrukte wie disjunkte Klassen oder inverse Eigenschaften erfordern. Aufgrund seiner Effizienz wird ELK häufig für Ontologien verwendet, die eine hohe Performanz bei der Klassifizierung und Konsistenzprüfung erfordern.

**Pellet**<sup>4</sup> [SPG<sup>+</sup>07] ist ein Reasoner, der die vollständige OWL 2 DL-Spezifikation unterstützt. Pellet kann sowohl auf der Individuenebene (ABox-Reasoning) als auch auf der konzeptionellen Ebene (TBox-Reasoning) logische Schlussfolgerungen ziehen, was es besonders für komplexe Ontologien mit strikten Konsistenzanforderungen geeignet macht. Es unterstützt auch erweiterte Schlussfolgerungen über Datentypen und komplexe logische Ausdrücke.

## 2.3 Implementierung von Ontologien

### RDF

Die Implementierung von Ontologien erfolgt durch spezialisierte Ontologiesprachen, die es ermöglichen, Konzepte, Relationen und Axiome formal zu spezifizieren und Wissen strukturiert darzustellen. Eine der grundlegenden Sprachen hierfür ist das **Resource Description Framework (RDF)**. RDF basiert auf einer Tripelstruktur, die aus einem Subjekt, einem Prädikat und einem Objekt besteht. Diese Struktur ermöglicht es, einfache Aussagen über Ressourcen zu treffen. Allerdings bietet RDF allein keine Informationen

---

<sup>2</sup><http://www.hermit-reasoner.com/>

<sup>3</sup><https://liveontologies.github.io/elk-reasoner/>

<sup>4</sup><https://github.com/stardog-union/pellet>

über die semantische Struktur der Daten. Alle Aussagen zu Hierarchien, Äquivalenzen, Restriktionen, Disjunktionen, ... können nicht abgebildet werden.

## RDFS

**RDF Schema (RDFS)** erweitert RDF, indem es die Definition von Klassen und Relationen ermöglicht. Während RDF primär einfache Datenbeziehungen beschreibt, erlaubt RDFS die Modellierung von Metadaten und die Definition von Klassenstrukturen. Klassen können über Hierarchien abgebildet werden. Relationen können mit *range* und *domain* eingeschränkt werden. Die Äquivalenz von Klassen, Disjunktionen, Restriktionen oder funktionale oder transitive Eigenschaften einer Relation können nicht angegeben werden.

## OWL - Web Ontology Language

Eine noch mächtigere Ontologiesprache ist die **Web Ontology Language (OWL)**. OWL basiert auf RDF und erweitert dessen Ausdrucksmöglichkeiten deutlich, indem es logisches Schließen (Reasoning) unterstützt. In OWL können komplexe Regeln und Beziehungen modelliert werden. OWL bietet auch die Möglichkeit, Restriktionen zu definieren. Durch die Unterstützung von logischen Regeln kann ein OWL-Reasoner zusätzliche Informationen ableiten, die nicht explizit in der Ontologie angegeben sind.

Ein wesentlicher Vorteil von OWL ist seine Fähigkeit, mit der **Open World Assumption** umzugehen. Diese Annahme besagt, dass das Wissen in einer Ontologie prinzipiell unvollständig ist und dass das Fehlen einer Information nicht bedeutet, dass die Information falsch ist. Beispielsweise könnte in der Ontologie festgehalten sein, dass *Dr. Gruber* aktuell keinen *Patient* hat. Nach der Open World Assumption bedeutet dies jedoch nicht, dass *Dr. Gruber* niemals einen Patienten haben wird, sondern nur, dass diese Information zum jetzigen Zeitpunkt nicht erfasst wurde. Im Gegensatz dazu steht die **Closed World Assumption**, die in traditionellen Datenbanksystemen angewendet wird. Diese Annahme besagt, dass alle relevanten Informationen vollständig erfasst sind, und wenn eine Information fehlt, wird sie als falsch angesehen. In der Beispielontologie würde das Fehlen eines *therapiert* für *Dr. Gruber* bedeuten, dass er tatsächlich keinen Patienten hat und würde als Fehler ausgeworfen werden, da laut der Restriktion in der Ontologie jeder Arzt mindestens einen Patienten therapieren muss.

OWL bietet verschiedene Profile, die eine unterschiedliche Balance zwischen Ausdruckskraft und Berechenbarkeit darstellen. Die drei Hauptprofile sind OWL Lite, OWL DL und OWL Full. **OWL Lite** ist das einfachste Profil und eignet sich für Ontologien mit grundlegenden Klassifikationsanforderungen. Es unterstützt Beziehungen und ist leicht zu verarbeiten, eignet sich jedoch nicht für komplexere Modellierungen. **OWL DL** (Description Logic) erweitert die Ausdrucksmöglichkeiten von OWL Lite erheblich, indem es komplexere logische Konstrukte und Restriktionen ermöglicht, während es gleichzeitig sicherstellt, dass die Inferenzprozesse entscheidbar bleiben. Daher eignet sich OWL DL für anspruchsvollere Ontologien, bei denen eine präzise Modellierung und effizientes

Reasoning erforderlich sind. **OWL Full** bietet die umfassendste Ausdruckskraft von OWL, indem es keine der Einschränkungen von OWL DL auferlegt. Dadurch kann OWL Full sehr komplexe Ontologien modellieren, allerdings ohne die Garantie, dass Reasoner vollständig arbeiten, da die Inferenzprobleme in diesem Profil im Allgemeinen nicht entscheidbar sind.

## 2.4 Abfrage und Validierung von Ontologien

### 2.4.1 SPARQL

**SPARQL (SPARQL Protocol and RDF Query Language)** ist eine standardisierte Abfragesprache, die speziell für das Abrufen und Manipulieren von RDF-Daten entwickelt wurde.

In der folgenden SPARQL-Abfrage werden die Namen der Patienten, die von einem bestimmten Arzt behandelt werden, sowie die zugehörigen Krankheiten aus der Ontologie abgerufen.

```

1  PREFIX rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns#
2  PREFIX ex: http://example.org/ontology#
3
4  SELECT ?patientName ?disease
5  WHERE {
6      ?doctor rdf
7          ex
8          . ?doctor ex
9          "Dr._Weise" . ?doctor ex
10     ?patient . ?patient ex
11     ?patientName . ?patient ex
12     ?disease .
13 }

```

Listing 2.1: Beispiel SPARQL-Abfrage

### 2.4.2 SHACL

**SHACL (Shapes Constraint Language)** ist eine W3C-Empfehlung, die verwendet wird, um Einschränkungen und Validierungsregeln für RDF-Daten zu definieren. SHACL ermöglicht es, Struktur, Datenwerte und Beziehungen von RDF-Daten zu überprüfen und sicherzustellen, dass diese den gewünschten Regeln entsprechen. Es wird häufig eingesetzt, um sicherzustellen, dass RDF-Daten bestimmten Ontologien oder Schemata folgen.

Mit SHACL können sogenannte *Shapes* definiert werden, die als Muster oder Einschränkungen für RDF-Ressourcen dienen. Diese Shapes definieren, welche Eigenschaften und Werte für bestimmte RDF-Klassen erlaubt oder erforderlich sind.

Im folgenden Beispiel wird eine SHACL-Validierung dargestellt, die überprüft, ob alle Patienten einer Ontologie einen Namen und eine Krankheit haben.

```

1 @prefix sh: <http://www.w3.org/ns/shacl#> .
2 @prefix ex: <http://example.org/ontology#> .
3
4 ex:PatientShape
5   a sh:NodeShape ;
6   sh:targetClass ex:Patient ;
7   sh:property [
8     sh:path ex:hatName ;
9     sh:datatype xsd:string ;
10    sh:minCount 1 ;
11    sh:maxCount 1 ;
12    sh:message "Jeder Patient muss genau einen Namen haben." ;
13  ] ;
14  sh:property [
15    sh:path ex:hatKrankheit ;
16    sh:minCount 1 ;
17    sh:message "Jeder Patient muss mindestens eine Krankheit haben." ;
18  ] .

```

Listing 2.2: Beispiel SHACL-Shape zur Validierung einer Ontologie

In diesem SHACL-Shape wird definiert, dass jedes Individuum der Klasse *ex:Patient* genau einen Namen vom Datentyp *xsd:string* haben muss. Zusätzlich wird überprüft, ob jeder *Patient* mindestens eine *Krankheit* hat.

### 2.4.3 SWRL

**SWRL (Semantic Web Rule Language)** ist eine regelbasierte Sprache, die entwickelt wurde, um logische Regeln für OWL-Ontologien zu definieren. Mit SWRL können Bedingungen festgelegt werden, die, wenn sie erfüllt sind, zur Ableitung neuer Fakten in einer Ontologie führen. SWRL erweitert die Ausdrucksmöglichkeiten von OWL, indem es Regeln in der Form von *wenn-dann*-Aussagen erlaubt.

SWRL-Regeln bestehen aus einem Prämissenteil (*wenn*) und einem Konklusionsteil (*dann*). Der Prämissenteil legt Bedingungen fest, und wenn diese Bedingungen erfüllt sind, werden die in der Konklusion definierten Konsequenzen abgeleitet.

Im folgenden Beispiel wird eine SWRL-Regel gezeigt, die besagt, dass wenn ein Patient von einem Arzt behandelt wird und eine Krankheit hat, dann der Arzt diese Krankheit behandelt.

```

1 @prefix ex: <http://example.org/ontology#> .
2 @prefix swrlb: <http://www.w3.org/2003/11/swrlb#> .
3
4 ex:Arzt(?a) ^ ex:behandelt(?a, ?p) ^ ex:Patient(?p) ^ ex:hatKrankheit(?p, ?k)
5   -> ex:behandeltKrankheit(?a, ?k) .

```

Listing 2.3: Beispiel SWRL-Regel

In dieser SWRL-Regel wird definiert, dass für jeden Arzt  $?x$ , der einen Patienten  $?p$  behandelt und wenn dieser Patient eine Krankheit  $?k$  hat, der Arzt auch als behandelnder Arzt für diese Krankheit abgeleitet wird.

### 2.5 Ontologiearten

**Top-Level-Ontologien (Upper Ontologies)** genannt, sind die allgemeinsten Ontologien und definieren abstrakte Konzepte, die domänenübergreifend verwendet werden können. Sie bieten eine grundlegende Struktur, die als Ausgangspunkt für die Entwicklung spezifischerer Ontologien dient. Zu den Konzepten, die in Top-Level-Ontologien häufig vorkommen, gehören Entitäten wie z.B.: Entity, Physical, Abstract und Relation. Diese Konzepte sind so abstrakt, dass sie in nahezu jeder Domäne Bedeutung haben. Ein prominentes Beispiel für eine solche Ontologie ist die Suggested Upper Merged Ontology (SUMO) [NP01], die darauf abzielt, allgemeine Konzepte zu vereinheitlichen und so eine gemeinsame Grundlage für den Austausch von Wissen zwischen verschiedenen Ontologien zu schaffen. Eine weitere bedeutende Top-Level-Ontologie ist die Basic Formal Ontology (BFO) [ASS15], die insbesondere im biomedizinischen Bereich verwendet wird und sich auf allgemeine Konzepte wie Entity, Continuant, Occurrent, Process, Role, ... und deren Relationen konzentriert. Diese Ontologien bieten eine breite Anwendbarkeit, was sie zu einem wertvollen Instrument für die Schaffung eines gemeinsamen Verständnisses über verschiedene Disziplinen hinweg macht. Ihre allgemeine Natur bedeutet jedoch auch, dass sie oft weniger detailliert sind und daher nicht alle Anforderungen spezialisierter Domänen erfüllen können.

Im Gegensatz dazu sind **domänen-spezifische Ontologien (Domain Ontologies)** darauf ausgelegt, Wissen in einem bestimmten Anwendungsbereich detailliert und präzise zu modellieren. Diese Ontologien bieten eine viel höhere Granularität als Top-Level-Ontologien, da sie speziell auf die Bedürfnisse und Anforderungen einer bestimmten Domäne zugeschnitten sind. Ein Beispiel für eine domänen-spezifische Ontologie ist SNOMED CT [Bha15], ein standardisiertes Vokabular für klinische Begriffe, das in der Medizin weit verbreitet ist. SNOMED CT ist eine der umfassendsten Ontologien im Bereich der Medizin und enthält über 300.000 medizinische Begriffe. Sie dient als standardisierte Terminologie für klinische Begriffe und wird weltweit in elektronischen Gesundheitsakten eingesetzt, um die konsistente Erfassung, Verarbeitung und Analyse von klinischen Daten zu ermöglichen. Eine der umfassendsten Ontologien in den Biowissenschaften ist die Gene Ontology [Con04], die sich auf die Beschreibung von Genen und deren biologischen Funktionen konzentriert und damit ein wichtiges Werkzeug für die Genomforschung darstellt. Domänen-spezifische Ontologien sind im Vergleich zu Top-Level Ontologien weniger flexibel in anderen Bereichen einsetzbar, gehen aber sehr spezifisch auf die Bedürfnisse der jeweiligen Domäne ein.

**Anwendungsorientierte Ontologien (Application Ontologies)** hingegen zeichnen sich dadurch aus, dass sie speziell für die Unterstützung bestimmter Aufgaben oder Prozesse entwickelt wurden. Sie sind oft weniger umfassend als domänen-spezifische Ontologien,

konzentrieren sich jedoch auf die Modellierung von Wissen, das zur Lösung einer spezifischen Aufgabe erforderlich ist. Diese Ontologien werden z.B. für die Automatisierung von Geschäftsprozessen, die Verwaltung von Sensornetzwerken oder die Unterstützung von Entscheidungsfindungsprozessen in einem klar abgesteckten Anwendungsfeld eingesetzt.

## 2.6 Methoden zur Ontologieentwicklung

Die Entwicklung von Ontologien ist ein systematischer Prozess, der darauf abzielt, Wissen in einer strukturierten und formalen Weise darzustellen. Es gibt verschiedene etablierte Methoden zur Ontologieentwicklung, die sich durch ihre Schritte und Vorgehensweisen unterscheiden.

Der Top-Down-Ansatz beginnt mit der Festlegung allgemeiner Prinzipien und Konzepte, die dann schrittweise in spezifischere Unterkonzepte und Beziehungen unterteilt werden. Dieser Ansatz eignet sich besonders gut für die Entwicklung von Ontologien, die auf etablierten Theorien oder Modellen basieren. Er ermöglicht eine klare Hierarchie und Struktur und fördert die Konsistenz innerhalb der Ontologie.

Ein Nachteil des Top-Down-Ansatzes ist, dass er möglicherweise nicht alle spezifischen Details und Besonderheiten einer Domäne berücksichtigt, insbesondere wenn diese nicht von Anfang an vollständig bekannt sind.

Der Bottom-Up-Ansatz geht von den konkreten Daten und Informationen aus, die in der Ontologie abgebildet werden sollen. Konzepte und Beziehungen werden aus der Analyse dieser Daten abgeleitet und dann zu einer umfassenderen Struktur zusammengeführt. Dieser Ansatz ist besonders nützlich, wenn die Ontologie eine spezifische, datengetriebene Domäne abbilden soll, bei der die Details der Daten eine entscheidende Rolle spielen.

Ein Nachteil des Bottom-Up-Ansatzes ist, dass die resultierende Ontologie weniger strukturiert sein kann und es schwieriger sein kann, allgemeine Prinzipien oder übergreifende Zusammenhänge zu identifizieren.

Das Paper “A survey on ontology creation methodologies” von Christani et al. [CC05] und das Paper “An analysis of ontology engineering methodologies: A literature review” von Iqbal et al. [IMM<sup>+</sup>13] geben einen Überblick über etablierte Methoden.

Der Ontology 101-Ansatz [Noy01] richtet sich an Entwickler, die ihre erste Ontologie erstellen möchten. Dieser Top-Down Ansatz legt besonderen Wert auf die methodische und iterative Entwicklung von Ontologien.

Die Schritte des Ontology 101-Ansatzes umfassen:

- **Zielbestimmung:** Festlegung des Anwendungsbereichs und der Ziele der Ontologie, um sicherzustellen, dass sie den Anforderungen der Domäne entspricht.
- **Klassenidentifikation:** Bestimmung der wichtigsten Konzepte der Ontologie, die die relevanten Klassen der Domäne darstellen.

- **Hierarchiebildung:** Strukturierung der Klassen in einer Hierarchie.
- **Eigenschaftenbestimmung:** Definition der Attribute oder Merkmale der Klassen und ihrer Instanzen.
- **Einschränkungen:** Festlegung von Einschränkungen für die Eigenschaften, wie z.B. zulässige Werte oder Datentypen.
- **Instanzenerstellung:** Erstellung von konkreten Instanzen der definierten Klassen, die reale Objekte der Domäne repräsentieren.
- **Evaluierung und Verfeinerung:** Überprüfung und Verbesserung der Ontologie durch iterative Evaluierungen, um sicherzustellen, dass sie konsistent, korrekt und nützlich ist.

Der METHONTOLOGY-Ansatz [FLGPJ97] ist eine der ältesten Methoden zur Ontologieentwicklung und kombiniert Top-Down als auch Bottom-Up. Dieser Ansatz wurde in den 1990er Jahren entwickelt und bietet ein umfassendes Framework, das den gesamten Lebenszyklus einer Ontologie abdeckt, von der Planung über die Entwicklung bis hin zur Wartung. Ziel des METHONTOLOGY-Ansatzes ist es, den Übergang von der Ontologieentwicklung als Kunst hin zu einem ingenieurmäßigen Prozess zu erleichtern. Dabei werden systematische Aktivitäten, Techniken und Werkzeuge verwendet, die den Entwicklungsprozess standardisieren und strukturieren.

Die zentralen Schritte des METHONTOLOGY-Ansatzes umfassen:

- **Planung:** In dieser Phase wird das Ziel der Ontologie definiert und die benötigten Ressourcen identifiziert. Es geht darum, eine klare Vorstellung davon zu bekommen, warum die Ontologie entwickelt wird, welche Endbenutzer und Anwendungen davon profitieren sollen und welche Ressourcen benötigt werden.
- **Spezifikation:** Die Spezifikationsphase beinhaltet die Erstellung eines Ontologieanforderungsdokuments. In diesem Dokument werden der Anwendungsbereich und die wichtigsten Anforderungen der Ontologie festgelegt, die entweder informell, halbformell oder formell dokumentiert werden können.
- **Wissensakquisition:** Diese Phase beinhaltet die Erhebung und Sammlung von domänenspezifischem Wissen, das für die Ontologie erforderlich ist. Dazu können Techniken wie Interviews mit Fachexperten, Literaturstudien oder die Wiederverwendung vorhandener Ontologien verwendet werden.
- **Konzeptualisierung:** Diese Phase umfasst die Strukturierung des erworbenen Wissens in ein konzeptuelles Modell. Dabei werden die wichtigsten Konzepte, deren Eigenschaften und die Beziehungen beschrieben. Hier kommen oft Zwischenrepräsentationen wie kontrollierte Vokabulare und Thesauri zum Einsatz.

- **Formalisierung:** In dieser Phase wird das konzeptuelle Modell in eine formale Repräsentation überführt, die maschinenlesbar ist. Dies erfordert die Verwendung formaler Beschreibungslogiken.
- **Implementierung:** In dieser Phase wird die Ontologie in einer formalen Ontologiesprache wie OWL implementiert.
- **Evaluierung:** Die Evaluierung umfasst die technische Überprüfung der Ontologie, um deren Konsistenz, Korrektheit und Anwendbarkeit zu gewährleisten. Die Evaluierung erfolgt anhand eines definierten Rahmens und umfasst sowohl Verifikations- als auch Validierungsprozesse.
- **Dokumentation:** Eine umfassende Dokumentation ist entscheidend, um die Wiederverwendung und Wartung der Ontologie zu erleichtern. In jeder Phase des Entwicklungsprozesses sollte dokumentiert werden, welche Entscheidungen getroffen wurden, welche Techniken verwendet wurden und wie die Ontologie strukturiert ist.
- **Wartung:** Nach der Fertigstellung der Ontologie ist die kontinuierliche Wartung erforderlich, um Änderungen oder Erweiterungen vorzunehmen, die sich aus neuen Anforderungen oder Technologien ergeben.

Der NeOn-Ansatz [SFGPFL11] wurde im Rahmen des NeOn-Projekts der Europäischen Union entwickelt und fokussiert sich auf die kollaborative Entwicklung von Netzwerk-Ontologien in verteilten Umgebungen. Im Gegensatz zu traditionellen Ansätzen, die einem festen, sequentiellen Workflow folgen, bietet die NeOn-Methodologie flexible Wege zur Entwicklung von Ontologien, insbesondere durch die Wiederverwendung und das Reengineering vorhandener Ressourcen. Der Ansatz betont die kollaborative Entwicklung, bei der verschiedene Akteure an unterschiedlichen Teilen der Ontologie arbeiten und diese später miteinander vernetzen.

Die wichtigsten Schritte des NeOn-Ansatzes umfassen:

- **Anforderungsanalyse:** Zunächst müssen die Ziele und Anforderungen der zu entwickelnden Ontologie festgelegt werden. Dies erfolgt durch die Spezifikation von Ontologieanforderungen, ähnlich wie in traditionellen Ontologie-Methoden.
- **Ontologie-Wiederverwendung:** Ein zentrales Merkmal des NeOn-Ansatzes ist die Wiederverwendung bestehender Ontologien oder Ontologie-Module. Diese Wiederverwendung kann durch die Identifikation geeigneter Ressourcen in Ontologie-Repositories und deren Anpassung an die aktuelle Ontologie erfolgen.
- **Kollaborative Entwicklung:** Ein weiteres Schlüsselmerkmal ist die Zusammenarbeit zwischen verschiedenen Entwicklern und Domänenexperten. Der Ansatz ermöglicht die parallele Entwicklung und spätere Integration der Ontologie-Module.

- **Evolutionsmanagement:** Da Ontologien häufig in dynamischen Umgebungen eingesetzt werden, in denen sich die Anforderungen ändern können, legt die NeOn-Methodologie Wert auf ein Evolutionsmanagement. Die Ontologie wird kontinuierlich weiterentwickelt, um neue Anforderungen zu erfüllen.
- **Dokumentation und Wartung:** Wie bei jeder Methode zur Ontologieentwicklung sind eine umfassende Dokumentation und die fortlaufende Wartung der Ontologie unerlässlich, um die Konsistenz und Verwendbarkeit zu gewährleisten.

### 2.7 Werkzeuge

Ontologie-Werkzeuge haben sich in den letzten Jahrzehnten erheblich weiterentwickelt und bieten wesentliche Unterstützung bei der Entwicklung, Visualisierung und Verwaltung von Ontologien. Diese Werkzeuge ermöglichen es komplexe Ontologien zu entwerfen und zu pflegen. Das Paper “Towards a Modern Ontology Development Environment” von Stadnicki et al. [SPB20] gibt eine Übersicht über die historische Entwicklung, aktuelle Werkzeuge, Trends und Herausforderungen.

#### 2.7.1 Historische Entwicklung der Ontologie-Werkzeuge

Das Konzept der Ontologie im Bereich der Informatik begann in den frühen 1990er Jahren an Bedeutung zu gewinnen. Zu den ersten bedeutenden Werkzeugen gehörten *Ontolingua* und *Chimaera*, die an der Stanford University entwickelt wurden. Diese spielten eine Schlüsselrolle bei der Definition, wie Ontologien erstellt und zwischen Forschern und Entwicklern geteilt werden konnten. *Ontolingua* war eine der ersten kollaborativen Ontologie-Entwicklungsumgebungen und ermöglichte es mehreren Nutzern, gemeinsame ontologische Strukturen zu erstellen und zu verfeinern.

In den späten 1990er und frühen 2000er Jahren entstanden neue Werkzeuge, die die Fähigkeiten von Ontologie-Entwicklungsumgebungen erweiterten. Eines der bedeutendsten ist *Protege*<sup>5</sup>, ebenfalls an der Stanford University entwickelt, das schnell zu einer der am weitesten verbreiteten Ontologie-Entwicklungsplattformen wurde. *Protege* bietet eine grafische Benutzeroberfläche, die es den Nutzern ermöglichte, Ontologien zu erstellen, zu bearbeiten und zu visualisieren, ohne umfangreiche Programmierkenntnisse zu benötigen. Im Laufe der Zeit entwickelte es sich zu einem robusten Werkzeug, das mehrere Ontologiesprachen, einschließlich OWL, unterstützt und ein reichhaltiges Plugin-Ökosystem für erweiterte Funktionalität bietet.

Ein weiteres bedeutendes Werkzeug ist das *NeOn Toolkit*<sup>6</sup>, das 2006 eingeführt wurde und die Unterstützung für musterbasierte Ontologie-Designs einführte. Es ermöglichte die Verwendung kollaborativer Methoden wie dem eXtreme Design. Trotz seines anfänglichen Potenzials stagnierte jedoch die Entwicklung des NeOn Toolkits, und das Werkzeug wird heute nicht mehr aktiv gewartet.

---

<sup>5</sup><https://protege.stanford.edu/>

<sup>6</sup><http://neon-toolkit.org/>

## 2.7.2 Aktuelle Ontologie-Werkzeuge

In der heutigen Landschaft bleiben *Protege* und sein webbasierter Ableger *WebProtege*<sup>7</sup> die dominierenden Werkzeuge in der Ontologieentwicklung. Protege verfügt über eine große und aktive Benutzercommunity mit über 350.000 registrierten Nutzern. Es wird weiterhin aktiv weiterentwickelt und bietet Funktionen wie OWL-Unterstützung, kollaborative Bearbeitung und umfangreiche Anpassungsmöglichkeiten durch Plugins.

Ein neueres Werkzeug, das an Popularität gewonnen hat, ist *VocBench*<sup>8</sup>. Es handelt sich um eine Open-Source-Plattform, die für das kollaborative Management von Ontologien, Thesauri und Linked Data-Vokabularen entwickelt wurde. VocBench legt besonderen Wert auf Mehrsprachigkeit und wird häufig in Bereichen eingesetzt, in denen die Wissensrepräsentation mehrere Sprachen und kulturelle Kontexte unterstützen muss.

Ein weiteres kürzlich entwickeltes Werkzeug ist der *Fluent Editor*<sup>9</sup>, der eine Bearbeitung von Ontologien in natürlicher Sprache ermöglicht. Dadurch können Nutzer Ontologien mit vereinfachten sprachlichen Beschreibungen erstellen, was die Eintrittsbarriere für die Ontologieentwicklung senkt und es Domänenexperten erleichtert, ohne formale Schulung in der Ontologie-Entwicklung aktiv zu werden.

Ergänzend zu den vollständigen Ontologie-Entwicklungsumgebungen ist eine breite Palette unterstützender Werkzeuge entstanden, um spezifische Bedürfnisse im Ontologieentwicklungsprozess abzudecken. Beispielsweise bietet *WebVOWL*<sup>10</sup> interaktive Visualisierungen von OWL-Ontologien, während *WIDOCO*<sup>11</sup> (Wizard for Documenting Ontologies) die automatisierte Generierung von Ontologiedokumentationen einschließlich Metadatenextraktion und Inhalten unterstützt.

Neben den genannten Ontologie-Entwicklungswerkzeugen gibt es auch mehrere Tools, Bibliotheken und Datenbanken, die spezifische Funktionen im Ontologie- und Wissensmanagementprozess abdecken. Einige davon sind:

*RDFLib*<sup>12</sup> ist eine python-Bibliothek, die das Arbeiten mit RDF-Daten vereinfacht. RDFLib bietet Unterstützung für das Parsen, Speichern, Serialisieren und Abfragen von RDF-Daten und ist besonders nützlich, um Ontologien in RDF-Format in python-Projekten zu integrieren.

*Owready2*<sup>13</sup> ist eine python-Bibliothek, die das Laden, Abfragen und Modifizieren von OWL-Ontologien ermöglicht. Sie unterstützt den Zugriff auf OWL-Klassen und -Eigenschaften durch eine API und erleichtert die Integration von Ontologien in python-Anwendungen. Zudem bietet Owready2 Schnittstellen zu Reasonern wie Pellet, was eine automatische Konsistenzprüfung und logische Inferenz erlaubt.

<sup>7</sup><https://webprotege.stanford.edu/>

<sup>8</sup><https://vocbench.uniroma2.it/>

<sup>9</sup><https://www.cognitum.eu/semantics/fluenteditor/>

<sup>10</sup><https://github.com/VisualDataWeb/WebVOWL>

<sup>11</sup><https://github.com/dgarijo/Widoco>

<sup>12</sup><https://rdflib.readthedocs.io/en/stable/>

<sup>13</sup><https://owready2.readthedocs.io/en/latest/>

*Jena*<sup>14</sup> ist ein Open-Source-Framework für die Verarbeitung und das Management von RDF-Daten. Es unterstützt neben RDF auch OWL und SPARQL und ermöglicht Entwicklern, umfangreiche Ontologien zu verarbeiten und semantische Abfragen durchzuführen. *Fuseki*<sup>15</sup>, der SPARQL-Server von Jena und dient als Schnittstelle, um SPARQL-Abfragen über HTTP durchzuführen und ermöglicht damit die Verwaltung von RDF-Datenbanken.

*GraphDB*<sup>16</sup> ist eine kommerzielle hochleistungsfähige RDF-Datenbank, die sich auf die Speicherung und Abfrage von Ontologien spezialisiert hat. Sie unterstützt effiziente semantische Abfragen und kann mit großen Wissensgraphen arbeiten, was sie ideal für skalierbare Ontologie- und Linked Data-Projekte macht.

*Stardog*<sup>17</sup> ist eine kommerzielle Wissensgraphen-Plattform, die Ontologien und semantische Technologien zur Datenintegration, -speicherung und -abfrage nutzt. Stardog bietet fortschrittliche Abfrage- und Inferenzmöglichkeiten, die es erlauben, verschiedene Datenquellen in Wissensgraphen zu integrieren und logische Schlussfolgerungen auf Basis von Ontologien zu ziehen.

*OntoGraph*<sup>18</sup> ist ein Werkzeug zur Visualisierung von Ontologien. Es generiert grafische Repräsentationen von Ontologie-Hierarchien und Beziehungen, was es einfacher macht, die Struktur und die Zusammenhänge innerhalb einer Ontologie zu verstehen.

### 2.8 Einsatz von Ontologien in der Arbeit

Geodateninfrastrukturen verwenden Metadaten aus aus einer Vielzahl von heterogenen Datenquellen, die unterschiedliche Metadatenstandards verwenden, um Informationen mit Raumbezug zu beschreiben. Diese Heterogenität erschwert den interoperablen Austausch von Metadaten zwischen verschiedenen Systemen und Organisationen, da unterschiedliche Begriffe und Klassifikationen für ähnliche oder identische Konzepte verwendet werden.

Durch den Einsatz von Ontologien können diese Unterschiede überbrückt werden, indem eine gemeinsame semantische Grundlage geschaffen wird, die es ermöglicht, verschiedene Metadatenstandards zu harmonisieren. Ontologien bieten eine formale und explizite Spezifikation von Konzepten und deren Beziehungen zueinander, was es ermöglicht, unterschiedliche Begrifflichkeiten aus verschiedenen Standards zu verbinden und aufeinander abzustimmen.

Ein weiterer Vorteil von Ontologien in diesem Kontext ist ihre Fähigkeit, logische Beziehungen zwischen Konzepten zu modellieren. Dadurch wird es möglich, Informationen aus Metadaten automatisch zu verknüpfen, zu integrieren und abzuleiten, auch wenn sie ursprünglich in unterschiedlichen Metadatenformaten vorliegen.

---

<sup>14</sup><https://jena.apache.org/>

<sup>15</sup><https://jena.apache.org/documentation/fuseki2/>

<sup>16</sup><https://graphdb.ontotext.com/>

<sup>17</sup><https://www.stardog.com/>

<sup>18</sup><https://github.com/NinePts/OntoGraph>

In dieser Arbeit wird eine *Application Ontology* entwickelt, die die Metadatenmodelle ISO, DCAT-AP und INSPIRE in eine gemeinsame Wissensbasis überführt. Die Vorgehensweise zur Ontologieerstellung orientiert sich am *Ontology 101-Ansatz*, da dessen Top-Down Methode und die Möglichkeit zur iterativen Entwicklung einen intuitiven und niederschweligen Zugang ermöglichen. Für den Entwurf und die Validierung der Ontologie wird *Protege* eingesetzt, da die grafische Oberfläche auch bei komplexen Strukturen eine gute visuelle Übersicht bietet. Die prototypische Implementierung wird mit *Owlready2* in python realisiert, um die nahtlose Überführung der XML- und RDF-Enkodierungen der Standards in die Ontologie und umgekehrt zu gewährleisten.



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar  
The approved original version of this thesis is available in print at TU Wien Bibliothek.

# Metadatenstandards in der Geoinformation

## 3.1 Übersicht und historische Entwicklung der ISO 19100-Serie

Die Normen der ISO 19100-Serie werden von der *International Organization for Standardization (ISO)* bereitgestellt, um Informationen und Ressourcen mit räumlichem Bezug zu beschreiben. Die Entwicklung dieser Normen soll die Interoperabilität zwischen Geodatenquellen und Diensten verbessern. Im Folgenden werden die relevanten Standards der Serie chronologisch aufgeführt und ihre Entwicklungen beschrieben.

### 3.1.1 ISO 19115 - Geographic Information - Metadata

- ISO 19115:2003 - Geographic Information - Metadata [ISO03a]
- ISO 19115:2003/Cor 1:2006 - Geographic Information - Metadata - Technical Corrigendum 1 [ISO03b]
- ISO 19115-1:2014 - Geographic Information - Metadata - Part 1: Fundamentals [ISO14]
- ISO 19115-1:2014/Amd 1:2018 - Amendment 1 [ISO18a]
- ISO 19115-1:2014/Amd 2:2020 - Amendment 2 [ISO20]

Die erste Version von *ISO 19115* wurde 2003 veröffentlicht. Sie definiert eine Struktur von Metadatenattributen zur Beschreibung von Geodatensätzen und -diensten. Die Norm unterscheidet zwischen verpflichtenden und optionalen Elementen, die je nach Ressourcentyp

und den spezifischen Anforderungen variieren. Zu den verpflichtenden Elementen gehören unter anderem Informationen zur geografischen Ausdehnung, Kontaktinformationen des Datenbereitstellers sowie räumliche und zeitliche Referenzen.

Mit *ISO 19115:2003/Cor 1:2006* wurden Mehrdeutigkeiten der ursprünglichen Version behoben, insbesondere bei der Interpretation von Metadatenfeldern wie räumliche Referenzsysteme und zeitliche Abdeckung.

Im Jahr 2014 wurde der Standard mit der Veröffentlichung von *ISO 19115-1:2014* aktualisiert. Diese Version führte eine Anpassung der Metadatenelemente ein, um moderne technologische und semantische Anforderungen besser zu unterstützen. Die Beschreibung von Datensätzen, Datenserien und grundlegenden Informationen zu Diensten wurde umstrukturiert, während die detaillierte Dienstbeschreibung weiterhin in *ISO 19119* spezifiziert bleibt. Die Kompatibilität zu *ISO 19115:2003* wurde aufgrund der Neustrukturierung von Metadatenelementen und der Einführung neuer Konzepte aufgehoben.

Die Änderungen betreffen unter anderem die Präzisierung der Metadatenstrukturen zur Erfassung räumlicher Referenzsysteme, Datenqualität und zeitlicher Aspekte. Außerdem wurde die Möglichkeit zur Erstellung nationaler Profile verstärkt, sodass spezifische Anpassungen auf Grundlage der *ISO 19115*-Konzepte entwickelt werden können.

*ISO 19115-1:2014* legt allgemeine Metadatenstrukturen für Geodatenätze und Dienste fest, berücksichtigt jedoch keine spezifischen Datentypen wie Punktwolken oder Sensordaten. Diese Formate wurden in späteren Erweiterungen wie *ISO 19115-2* behandelt, das zusätzliche Anforderungen und Metadaten zur Erfassung und Verarbeitung von Rasterdaten, Bilddaten und Sensordaten spezifiziert. Diese Erweiterungen fügen spezifische Metadaten zur detaillierten Beschreibung von Sensordaten hinzu, einschließlich Informationen zu Erfassungstechnologien, Datenquellen sowie zu Verarbeitungsprozessen wie Kalibrierung und Korrekturmethode.

Das *Amendment 1* von 2018 fügte Metadaten zur Sicherheitsklassifikation hinzu, um den Schutzstatus von Geodaten zu dokumentieren. Es enthält Angaben zur Vertraulichkeit und Zugriffskontrolle. Das *Amendment 2* von 2020 führte Metadaten zur Dokumentation von Lizenzen und rechtlichen Bedingungen ein, um Nutzungsrechte und rechtliche Rahmenbedingungen für die Weitergabe und Wiederverwendung von Geodaten anzugeben.

#### **ISO 19115-2 - Extensions for acquisition and processing**

- ISO 19115-2:2009 - Geographic Information - Metadata - Part 2: Extensions for imagery and gridded data [ISO09]
- ISO 19115-2:2019 - Geographic Information - Metadata - Part 2: Extensions for acquisition and processing [ISO19b]
- ISO 19115-2:2019/Amd 1:2022 - Amendment 1 [ISO22]

*ISO 19115-2:2009* wurde als Erweiterung von ISO 19115 veröffentlicht, um Metadaten für die Erfassung und Verarbeitung von Bild- und Rasterdaten bereitzustellen. Diese Version war speziell auf Fernerkundungsdaten ausgerichtet, die von Satelliten, Luftbildern und anderen Fernerkundungssystemen erfasst wurden. Ergänzend zu den allgemeinen Metadaten in ISO 19115 wurden spezifische Metadatenfelder eingeführt, die Informationen über Erfassungsmethoden, Datenverarbeitungsprozesse sowie die Qualität der Bilddaten und Rasterdaten dokumentieren. Diese Metadaten umfassen beispielsweise die Beschreibung der Sensorplattform, des Erfassungszeitraums und der räumlichen Auflösung.

Die überarbeitete Version *ISO 19115-2:2019* führte zusätzliche Metadatenelemente ein, um die detaillierte Dokumentation der Erfassung und Verarbeitung von Bild-, Raster- und Sensordaten zu verbessern. Zu den wesentlichen Neuerungen gehört die Einführung von Metadaten zur genauen Beschreibung der Position von Sensoren, der verwendeten Technologie sowie der Genauigkeit der Erfassungsmethoden. Darüber hinaus wurden die Metadaten zur Beschreibung der Datenverarbeitung erweitert, um Kalibrierungs- und Korrekturverfahren präziser zu dokumentieren. Diese zusätzlichen Felder ermöglichen eine genauere Nachvollziehbarkeit der Verarbeitungsprozesse, insbesondere bei Sensordaten.

Das *Amendment 1* von 2022 ergänzte weitere Metadaten zur detaillierten Beschreibung der Erfassungs- und Verarbeitungsprozesse, speziell für Sensordaten. Die neuen Metadatenfelder umfassen insbesondere die Kalibrierung von Sensoren und die angewendeten Verarbeitungsmethoden, um die Qualität und Verlässlichkeit der Sensordaten sicherzustellen. Diese Erweiterung ist besonders relevant für Daten aus mehrstufigen Erfassungssystemen, wie sie bei Drohnen oder Satelliten vorkommen, bei denen die genaue Kalibrierung und Verarbeitung entscheidend für die Datenqualität ist.

#### **ISO 19119 - Geographic Information - Services**

- ISO 19119:2005 - Geographic Information - Services [ISO05]
- ISO 19119:2005/Amd 1:2008 - Amendment 1 [ISO08]
- ISO 19119:2016 - Geographic Information - Services [ISO16]

*ISO 19119:2005* spezifizierte die standardisierte Beschreibung und Kategorisierung von Geodiensten. Diese Norm definiert eine Grundlage zur Dokumentation verschiedener Arten von Geodiensten, einschließlich Web-Diensten wie Web Map Services (WMS) und Web Feature Services (WFS). Sie ermöglicht die systematische Beschreibung und Veröffentlichung von Geodiensten in Metadatenkatalogen, um deren Auffindbarkeit und Nutzung zu erleichtern. Der Standard umfasst Geodienste, die der Bereitstellung, Verarbeitung und Transformation von Geodaten dienen.

Das *Amendment 1* von 2008 ergänzte die Norm um detailliertere Kategorien zur Beschreibung von Dienstoperationen. Es führte neue Metadatenfelder ein, die eine genauere Dokumentation der Aktionen und Interaktionen ermöglichen, die ein Geodienst ausführen

kann. Zudem wurden Kategorien eingeführt, um die Beschreibung von Diensten, die auf verteilten Plattformen ausgeführt werden, zu standardisieren und die Interoperabilität zwischen verschiedenen Systemen zu verbessern.

Mit der 2016 veröffentlichten Version *ISO 19119:2016* wurde die Spezifikation von Geodiensten aktualisiert, um den Anforderungen moderner technischer Entwicklungen besser gerecht zu werden. Die plattformunabhängige Beschreibung von Geodiensten, die bereits in der Version von 2005 definiert wurde, blieb ein zentrales Element und wurde durch verbesserte Ansätze zur Unterstützung der Interoperabilität erweitert. Diese Version berücksichtigt zudem Cloud-basierte Umgebungen und erweitert die Definition von Dienstkategorien, um neueren Technologien und Architekturen wie Service-oriented Architectures (SOA) gerecht zu werden.

#### **ISO 19157 - Geographic Information - Data quality**

- ISO 19157:2013 - Geographic Information - Data quality [ISO13]
- ISO 19157:2013/Amd 1:2018 - Amendment 1 [ISO18b]
- ISO 19157-1:2023 - Geographic Information - Data quality - Part 1: General requirements [ISO23]

*ISO 19157:2013* definiert ein Modell zur Bewertung der Qualität von Geodaten. Die Norm legt Qualitätsparameter fest, darunter Vollständigkeit, Positionsgenauigkeit, thematische Genauigkeit, logische Konsistenz und zeitliche Genauigkeit. Diese Parameter ermöglichen es, die Eignung von Geodaten in verschiedenen Anwendungsfällen zu bewerten und zu dokumentieren. Der Standard beschreibt zudem, wie Qualitätsinformationen erfasst und in Metadaten integriert werden, um eine konsistente und nachvollziehbare Qualitätsbewertung der Daten sicherzustellen.

Das *Amendment 1* von 2018 ergänzte Methoden zur erweiterten Beschreibung der Qualität von Raster- und Coverage-Daten. Diese Erweiterungen erleichtern die Bewertung der Vollständigkeit und Positionsgenauigkeit von kontinuierlichen Datensätzen, wie beispielsweise digitalen Höhenmodellen oder Fernerkundungsdaten. Sie bieten präzisere Werkzeuge zur Dokumentation der Datenqualität in diesen speziellen Bereichen, insbesondere bei der Analyse und Nutzung von Rasterdaten.

Mit der Veröffentlichung von *ISO 19157-1:2023* wurden die allgemeinen Anforderungen an die Qualitätsbewertung von Geodaten weiter präzisiert. Diese Version konzentriert sich auf eine standardisierte Dokumentation der Datenqualität über verschiedene Geodatenquellen hinweg und bietet Regeln zur Sicherstellung der Konsistenz in der Qualitätsbewertung.

#### ISO/TS 19139 - Geographic Information - Metadata - XML schema implementation

- ISO/TS 19139:2007 - Geographic Information - Metadata - XML schema implementation [ISO07]
- ISO/TS 19139-1:2019 - Geographic Information - XML schema implementation - Part 1: Encoding rules [ISO19a]

*ISO/TS 19139:2007* spezifizierte das XML-Schema zur Implementierung der Metadaten, die in *ISO 19115:2003* definiert wurden. Diese Technische Spezifikation ermöglichte eine standardisierte XML-Kodierung von Metadaten, um den konsistenten und maschinenlesbaren Austausch von Metadaten zwischen verschiedenen Systemen zu gewährleisten. Sie definierte Regeln zur Abbildung der in ISO 19115 definierten UML-Konzepte in XML, um sicherzustellen, dass die Struktur und Semantik der Metadaten korrekt in XML umgesetzt werden.

Mit der Veröffentlichung von *ISO/TS 19139-1:2019* wurden neue Kodierungsregeln eingeführt, die auf dem in *ISO 19115-1:2014* definierten UML-Profil basieren. Diese Version bietet erweiterte Anleitungen zur Implementierung von XML-Schemata und stellt sicher, dass die Metadaten interoperabel und maschinenlesbar sind. Die Kodierungsregeln wurden aktualisiert, um den Anforderungen moderner Metadatenstandards gerecht zu werden und eine größere Flexibilität bei der Integration von Metadaten in verschiedene Systeme zu ermöglichen.

#### 3.1.2 INSPIRE

Folgend wird ein Überblick über die INSPIRE EU-Richtlinie gegeben. Der Überblick wird aus den gesetzlichen Grundlagen

- DIRECTIVE 2007/2/EC OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL establishing an Infrastructure for Spatial Information in the European Community (INSPIRE)[PC07]
- COMMISSION REGULATION (EC) No 1205/2008 implementing Directive 2007/2/EC of the European Parliament and of the Council as regards metadata[PC08]
- Geodateninfrastrukturgesetz - Bundesgesetz über eine umweltrelevante Geodateninfrastruktur des Bundes (Geodateninfrastrukturgesetz - GeoDIG)[GEO10]

und den technischen und rechtlich nicht verbindlichen Dokumenten

- INSPIRE Generic Conceptual Model[INS14]

- INSPIRE Metadata Implementing Rules: Technical Guidelines based on EN ISO 19115 and EN ISO 19119[INS24]

zusammengefasst.

INSPIRE ist eine EU Richtlinie die unter der Kennzahl 2007/2/EG vom Europäischen Parlament und vom Europäischen Rat am 14.04.2007 verabschiedet wurde. Diese Richtlinie hat den Zweck in der Europäischen Gemeinschaft eine gemeinsame Infrastruktur für geographische Daten aufzubauen.

Die Idee einer gemeinsamen Geodateninfrastruktur in der EU ist im Zuge der Berichtspflichten der Mitgliedsstaaten im Umweltbereich entstanden. Um politische Entscheidungen im Umweltbereich treffen zu können, bedarf es gemeinsamer gesamteuropäischer Datengrundlagen. Jeder Mitgliedsstaat betreibt Umwelt Monitoring Systeme, wie z.B.: Luftgütemessnetze, Aufnahmen der Gewässerqualität, Niederschlagsmessungen, Meßnetze für Radioaktivität, etc. wo Daten mit unterschiedlichen Qualitäten, Messverfahren, Ausprägungen und Datenmodellen aufgenommen werden. Für die Erzeugung eines gesamteuropäischen Lagebilds, müssen daher die Daten der verschiedenen Nationalstaaten harmonisiert, publiziert und zu einem Gesamtdatenbestand vereinigt werden. Da für die Interpretation der Daten des Umwelt Monitorings auch Grundlagendaten wie Höhenmodelle, Straßennetze, Koordinatensysteme, etc. benötigt werden, trifft die INSPIRE Richtlinie nicht nur die Veröffentlichung von umweltspezifischen Datensätzen, sondern auch den Großteil der Grundlagendatensätze die in den INSPIRE-pflichtigen Organisationen existieren.[PC07]

Die 32 INSPIRE-pflichtigen Datenthemen sind in den Anhängen zur Richtlinie 2007/2/EG angeführt. Anhang 1 definiert Grundlagendaten wie Koordinatenreferenzsysteme, Geografische Bezeichnungen, Verwaltungseinheiten, Verkehrsnetze, etc. Ebenfalls grundlegende Daten wie Höhenmodelle, Orthofotografie, Geologie, etc. sind in Anhang 2 der Richtlinie gelistet. Sehr umfassend ist der Anhang 3 mit insgesamt 21 Datenthemen, wie z.B.: Umweltüberwachung, Statistische Einheiten oder Verteilung der Bevölkerung - Demografie.[PC07]

Die zeitliche Umsetzung der Richtlinie orientiert sich im Ablauf ebenfalls nach diesen Datenthemen. Der Bereich der Metadaten war für alle Datenthemen der Anhänge 1,2 und 3 bereits bis 03.12.2013 umzusetzen gewesen. Für alle "Invocable data services", also Dienste die keine harmonisierten Daten anbieten, aber INSPIRE relevant sind, waren die Metadaten bis 10.12.2015 zu bereitzustellen.

Die rechtliche Grundlage für INSPIRE ist in der Richtlinie 2007/2/EG definiert, die von allen Nationalstaaten der Europäischen Union in die nationale Gesetzgebung umgesetzt wurde. Die Richtlinie 2007/2/EG wird wiederum durch Verordnungen zur Umsetzung (Implementing Rules) genauer ausgeführt. Die für Metadaten relevante gesetzliche Grundlage ist die Verordnung Nr. 1205/2008 der Kommission vom 3. Dezember 2008. Hier werden rechtlich bindend verpflichtende und optionale Metadatenelemente, deren Beziehungen und deren Wertedomänen vorgeschrieben.[PC08] In Österreich wurde als

rechtliche Implementierung der EU-Richtlinie und den zugehörigen EU-Verordnungen 2010 das “Geodateninfrastrukturgesetz (GeoDIG)”[GEO10] als nationale Umsetzung beschlossen.

Für die technische Umsetzung der EU-Richtlinie und der EU-Verordnungen wurden auch sogenannte “Technical Guidelines” aufgelegt, die eine konkrete technische Implementierung den sogenannten “Implementing Rules” vorschlagen. Die “Technical Guidelines” haben im Vergleich zu der Richtlinie und den Verordnungen nur informativen Charakter, d.h. sie sind nicht rechtlich bindend. [INS24] Eine konkrete technische Implementierung der Richtlinie und der Verordnungen ist den Mitgliedsstaaten somit selbst überlassen, es wird jedoch empfohlen sich bei der technischen Implementierung an den “Technical Guidelines” zu orientieren, da hier die höchstmögliche Interoperabilität zwischen den teilnehmenden Organisationen gewährleistet ist.

Die INSPIRE “Technical Guidelines” Dokumente beschreiben in ihrer Gesamtheit eine vollständige Architektur wie eine INSPIRE-Geodateninfrastruktur aufgebaut werden kann. Die technische Implementierung der Metadaten wird im Dokument “INSPIRE Metadata Implementing Rules: Technical Guidelines based on EN ISO 19115 and EN ISO 19119”[INS24] beschrieben. Es befinden sich in den Technical Guidelines zu den Implementierungen der jeweiligen Datenthemen auch Unterkapitel zu erweiterten Metadaten. Eine Zusammenfassung der erweiterten Metadaten die alle Datenthemen betreffen befindet sich im Dokument “INSPIRE Generic Conceptual Model”, das auch als gemeinsame Grundlage der Spezifikationen der jeweiligen Datenthemen dient.[INS14]

Die “INSPIRE Metadata Implementing Rules” basieren auf den ISO Standards EN ISO 19115 aus 2003 und EN ISO 19119 aus 2005. Die Spezifikation wurde aus den beiden ISO Standards abgeleitet und ist daher im Aufbau und in der Ausprägung sehr vieler Metadatenelemente sehr ähnlich, jedoch insgesamt nicht kompatibel. Das Befüllen aller Pflichtelemente des ISO Metadatenstandards ergibt somit nicht automatisch einen INSPIRE konformen Metadatenatz. Umgekehrt ergibt die Befüllung aller Pflichtelemente eines INSPIRE konformen Metadatenatzes auch nicht automatisch einen konformen ISO 19115 Metadatenatz. Jeder der beiden Standards definiert zum Teil andere verpflichtende Kernelemente. INSPIRE definiert darüber hinaus teilweise auch zusätzliche bzw. auch eingeschränkte Wertedomänen bei Metadatenelementen. Dies betrifft insbesondere Angaben zu Datenthemen, Konformität zu den relevanten INSPIRE Verordnungen und lizenzrechtliche Aspekte bezüglich Zugang und Weiterverwendung der Datensätze.[INS24]

Aufgrund der zusätzlichen Erweiterung bzw. Einschränkungen der Metadatenelemente in Bezug auf die ISO Standards EN ISO 19115 und EN ISO 19119 und der daraus resultierenden Inkompatibilitäten, ergibt sich somit für INSPIRE-pflichtige Organisationen die Herausforderung einen zusätzlichen Metadatenstandard zu befüllen und zu verwalten.

#### 3.1.3 Data Catalog Vocabulary - DCAT-AP und GeoDCAT

In diesem Unterkapitel wird folgend die Entstehung und der Zusammenhang zwischen “Open Government Data” und der “Public Sector Information” und den daraus resultie-

renden Metadatenstandards erläutert. Die Informationen wurden aus den gesetzlichen Grundlagen

- DIRECTIVE 2003/98/EC OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL on the re-use of public sector information[PC13a]
- DIRECTIVE 2013/37/EU OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL amending Directive 2003/98/EC on the re-use of public sector information[PC13b]
- Bundesgesetz über die Weiterverwendung von Informationen öffentlicher Stellen (Informationsweiterverwendungsgesetz - IWG)[IWG15]

und den Dokumenten

- Rahmenbedingungen für Open Government Data Plattformen / Open Government Data - 1.1.0[OGD12]
- OGD Metadaten - 2.3[OGD17]
- Data Catalog Vocabulary (DCAT)[DCA14]
- Technisch/organisatorische Umsetzung der PSI Richtlinie / upsir-1.0.0[PSI15b]
- Klassifikation von Informationen für PSI-Umsetzung / psi-klassifikation 1.0.0[PSI15a]
- Erläuterung der XML-Spezifikation für die Metadaten Schnittstelle / metadaten-md 1.0.0[PSI14]

entnommen.

Im Jahr 2011 wurde die *Cooperation Open Government Data Österreich* gegründet. Diese Kooperation verfolgt das Ziel durch gemeinsame Standards und Rahmenbedingungen die Bereitstellung freier Verwaltungsdaten zu fördern. Unter Verwaltungsdaten versteht man Datensätze die im Zuge der Tätigkeit einer Organisation des öffentlichen Sektors anfallen. Freie Verwaltungsdaten Daten werden von der *Cooperation Open Government Data Österreich* folgendermaßen definiert: “*Open Government Data sind jene nicht - personenbezogenen und nicht -infrastrukturkritischen Datenbestände, die im Interesse der Allgemeinheit ohne jedwede Einschränkung zur freien Nutzung, zur Weiterverbreitung und zur freien Weiterverwendung frei zugänglich gemacht werden.*”[OGD12] Für die Lizenzierung und Nutzungsbedingungen für Datensätze die unter dieser Prämisse publiziert werden wurde die Lizenz *Creative Commons Lizenz CC-BY-AT* definiert, die eine freie Weiterverwendung unter Namensnennung der Bezugsquelle definiert. Im Zuge dieser Kooperation wurde auch ein Metadatenstandard erschaffen der in dem White-Paper *OGD-Metadaten* veröffentlicht wurde. Aktuell liegt dieser nationale Standard in der

Version 2.3 vor.[OGD17] Der *OGD-Metadaten Standard* wurde aus dem DCAT-AP RDF Vokabular abgeleitet, der für die Interoperabilität von Datenkatalogen im Web erschaffen wurde und mittlerweile als W3C Empfehlung für den Austausch von Metadaten zwischen Datenkatalogen im Web geführt wird.[DCA14] Aufgrund des Status eines White-Papers ist dieser Standard jedoch nur informativ und hat für Organisationen des öffentlichen Sektors keine rechtlich bindende Relevanz. Auch die Veröffentlichung von freien Verwaltungsdaten auf Basis von *Open Government Data* passierte durch die Organisationen freiwillig. Um auf dem nationalen Metadatenkatalog für offene Verwaltungsdaten data.gv.at gelistet zu werden war die Einhaltung dieses Standards jedoch eine Voraussetzung.[OGD12][OGD17]

Durch die EU-Richtlinie mit der Kennzahl 2003/98/EG[PC13a] und die dazugehörige Änderung der Richtlinie 2013/37/EU[PC13b], welche die *Weiterverwendung von Informationen des öffentlichen Sektors* (PSI) der Nationalstaaten in der Europäischen Union bestimmt, wurde der Veröffentlichung von Verwaltungsdaten auch eine rechtliche Grundlage gegeben. Die Richtlinie definiert, dass alle Dokumente die im Zuge des öffentlichen Auftrages einer Organisation oder Behörde anfallen, sofern diese nicht aus Gründen der Geheimhaltung, des Datenschutzes o.ä. ausgenommen sind, soweit möglich und sinnvoll in offenen und maschinenlesbaren Datenformaten, gemeinsam mit den zugehörigen Metadaten, für die kommerzielle und nicht-kommerzielle Nutzung publiziert werden sollen.[PC13b] Für die im Zuge der Publikation anfallenden Kosten, können auch sogenannte “Grenzkosten”, verlangt werden, die die Möglichkeit bieten sollen die nötigen Aufwände für die Bereitstellung abzugelten. Eine konkrete Definition eines Dokuments wird in der EU-Richtlinie nicht näher ausgeführt. Aufgrund der Intention der Weiterverbreitung von allen Daten von Organisationen des öffentlichen Sektors die unter keine der in der Richtlinie aufgelisteten Ausnahmen fallen, kann davon ausgegangen werden, dass dies auch Datensätze aus Geoinformationsinfrastrukturen betrifft.[PC13a][PC13b]

Für die Umsetzung der PSI-Richtlinie wurde in Österreich rechtlich durch das *Bundesgesetz über die Weiterverwendung von Informationen öffentlicher Stellen (Informationsweiterverwendungsgesetz - IWG)* implementiert.[IWG15] Die technische Implementierung wurde durch die Arbeitsgruppe UPSIR (Umsetzung PSI Richtlinie) des Bund-Länder-Städte-Gemeinden Gremiums durchgeführt. [PSI15b] Im Zuge dieser Umsetzung wurde auch eine nationale XML Kodierung für die Metadatenbeschreibung für PSI-Dokumente beschlossen der aus dem bereits etablierten “OGD-Metadaten” Metadatenstandard abgeleitet wurde. Durch den Beschluss des BLSG Gremiums ist die Verwendung der PSI-XML Kodierung und durch die darin enthaltenen Referenzierungen auch der *OGD Metadaten - 2.2 Standard* für Organisationen des öffentlichen Sektors in Österreich verbindlich.[PSI14] Um Datensätze die für die Weitergabe auf Basis der PSI-Richtlinie geeignet sind zu identifizieren wurde auch eine Klassifikation entwickelt mit der dieser Ablauf dokumentiert werden kann.[PSI15a] Die Dokumentation der Klassifikation selbst ist kein Teil der veröffentlichten Metadaten, jedoch für das Publikationsmanagement notwendig, da hier auch Gründe angeführt werden warum ein Datensatz auf Basis der PSI-Richtlinie nicht publiziert wurde.

### 3.2 Aufbau eines minimalen ISO 19115 Metadatensatzes

ISO19115 erlaubt die Beschreibung von einzelnen geografischen Daten, Diensten und Datenserien, definiert jedoch keinen minimalen Metadatensatz.

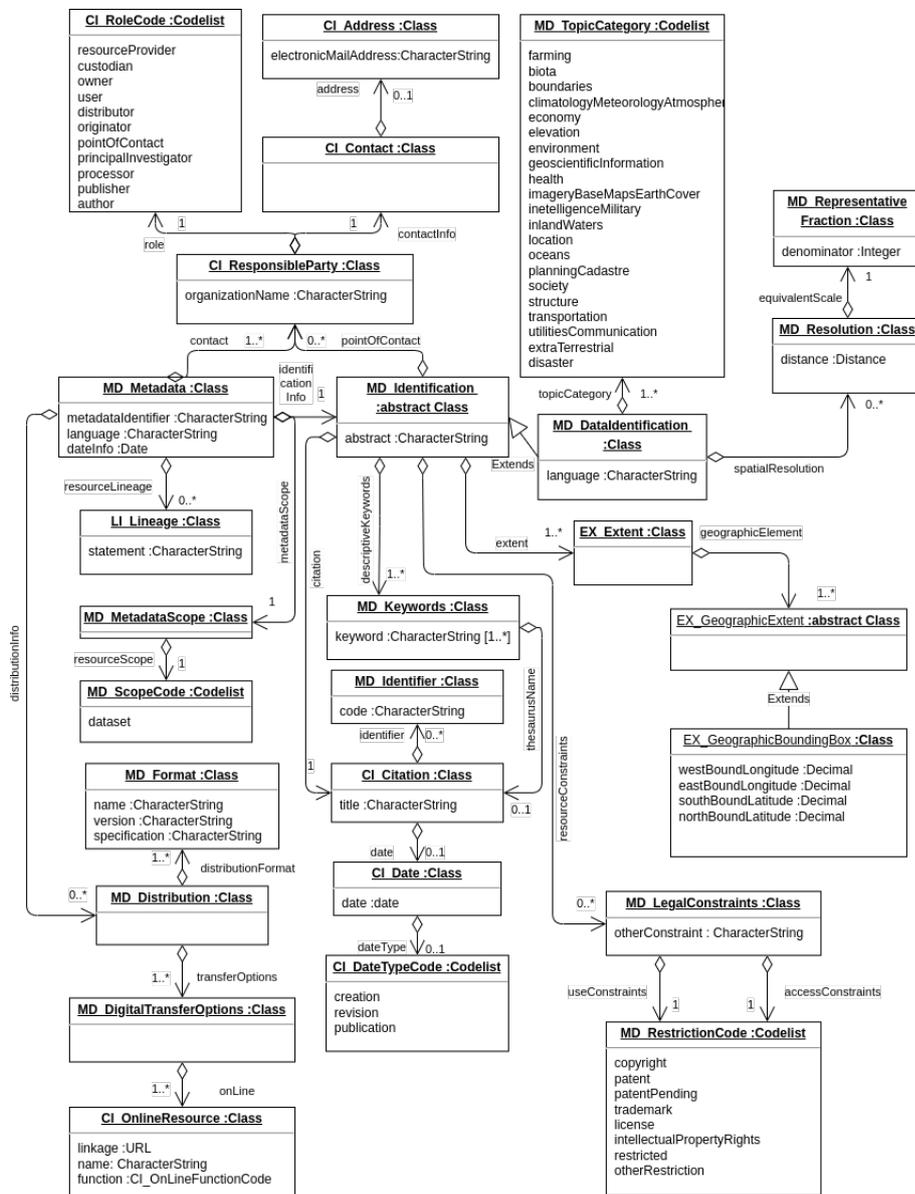


Abbildung 3.1: Übersicht über das ISO 19115 Metadatenmodell für Datensätze

Um die Auffindbarkeit der Metadaten in Metadatenkatalogen zu gewährleisten, findet man in Annex F *Discovery metadata for geographic resources* eine Auflistung von Metadatenfeldern die zwar nicht verpflichtend zu befüllen sind, jedoch eine Befüllung empfohlen

wird. Als Beispiel sei das Feld *MD\_Metadata.metadataIdentifier* genannt, das für jeden Metadatensatz eine eindeutige ID enthält. Ohne diese ID funktionieren Metadatenkataloge, welche den Catalogues Service Web (CSW) Standard implementieren nicht, da keine *GetRecordById* Aufrufe möglich sind. Die Arbeit orientiert sich für eine minimale Befüllung an den Feldern des Annex F.

Die Metadateninformation ist um die *MD\_Metadata* Klasse modelliert, die als Wurzelobjekt dient. Die *MD\_Metadata* Klasse enthält alle Informationen über die Metadaten selbst, wie Aktualität oder den *metadataIdentifier*, der den Metadatensatz identifiziert. Kontaktinformationen zu den Metadaten, Beschreibungen zur Entstehungshistorie und Bezugsmöglichkeiten sind in direkter Abhängigkeit zur Wurzelklasse modelliert. Weiters wird auch der *Scope* der Metadaten angegeben, also ob es sich um die Beschreibung eines Datensatzes oder eines Dienstes handelt. Die referenzierten Kontakte erfordern mindestens einen Organisationsnamen und eine Kontakt-Emailadresse. Die Rolle des Kontakts der Metadaten muss vom Typ *pointOfContact* sein.

Die Ressource wird über die abstrakte Klasse *MD\_IdentificationInfo* beschrieben. Je nachdem, ob es sich um einen Datensatz oder einen Dienst handelt, wird die Klasse über den Typ *MD\_DataIdentification* oder *SV\_ServiceIdentification* konkretisiert.

Die Klasse *MD\_IdentificationInfo* enthält Angaben, die sowohl für Datensätze als auch Dienste verwendet werden. Dazu gehört der Titel, eine Beschreibung, die Angaben des Datums zur Erstellung, Überarbeitung oder Publikation, die Schlüsselwörter mit der die Ressource klassifiziert werden und damit näher beschrieben werden kann und einen Identifier über den die Resource eindeutig identifiziert werden kann. Der Titel wird dabei in einer *CI\_Citation* Klasse gemeinsam mit den Datumsangaben spezifiziert. Die Datumsangaben haben dabei, wie der Kontakt, einen Typ, der über den *CI\_DateTypeCode* angegeben wird. Die Schlüsselwörter können aus einem kontrollierten Vokabular oder Thesaurus stammen. Die Angabe der Herkunft des Schlüsselwortes kann ebenfalls über eine *CI\_Citation* Referenz durchgeführt werden, die wiederum eine Datumsangabe enthält, welches üblicherweise das Publikationsdatum des Thesaurus angibt. Der Kontakt zur Ressource wird über die Relation *pointOfContact* angegeben, dabei können je nach Rolle auch mehrere Kontakte verwendet werden.

Die geografische Ausdehnung des Datensatzes wird über eine verschachtelte Relation mit einer *EX\_GeographicBoundingBox* durchgeführt, welche die WGS84 Koordinaten des umgrenzenden Rechtecks enthält.

Angaben zur Verwendung und zum Zugang zu den Daten können in den *MD\_LegalConstraints* hinterlegt werden. Für die möglichen Einschränkungen wird eine Codeliste angeboten.

Für die nähere Beschreibung des Datensatzes dient die Zuordnung eines Themas, welches mit der Liste an Themen aus der *MD\_TopicCategory* Klasse umgesetzt wird. Die Auflösung des Datensatzes bzw. der Verwendungsmaßstab wird über die *MD\_Resoultion* Klasse bereitgestellt.

### 3. METADATENSTANDARDS IN DER GEOINFORMATION

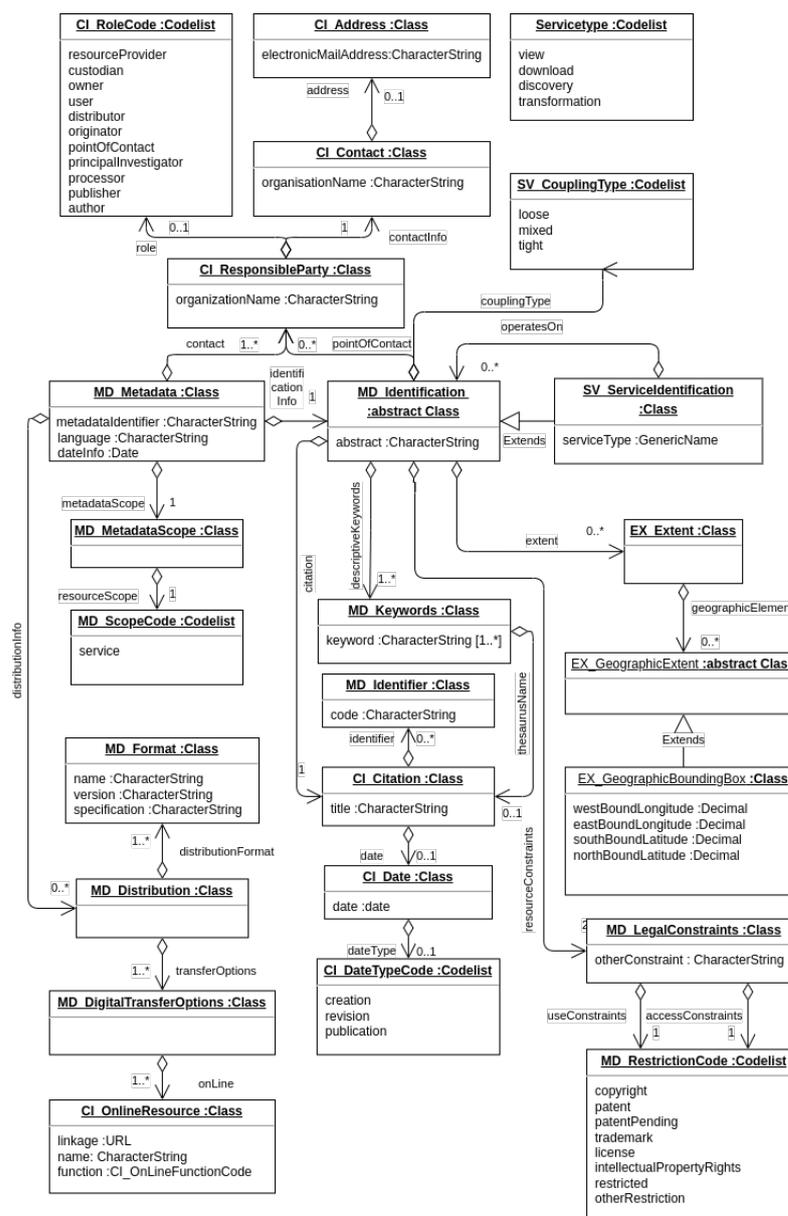


Abbildung 3.2: Übersicht über das ISO 19115 Metadatenmodell für Dienste

Die Beschreibung von Diensten wird in Abbildung 3.2 dargestellt. Im Gundschema gleicht der Aufbau eines Dienste-Metadatensatzes einem Datensatz-Metadatensatz. Der *Scope* des Metadatensatzes wird mit *service* angegeben, ansonsten finden sich die Unterschiede in der konkreten Klasse *SV\_ServiceIdentification*. Die Angabe des Service Typs, also ob es ein *Discovery*-, *Download*-, *Transformation*- oder *View Service* ist wird über den *serviceType* festgelegt und aus der Codelist *Servicetype* befüllt.

Dienste können an Datensätze oder andere Dienste gekoppelt sein. Diese Beziehung wird

über das *operatesOn* Attribut definiert, wobei hier üblicherweise über den Identifier der Ressource auf das *MD\_IdentificationInfo* eines anderen Metadatenatzes verwiesen wird. Wie stark die Kopplung zu den anderen Ressourcen ist, kann über *SV\_CouplingType* festgelegt werden.

### 3.3 Aufbau eines minimalen INSPIRE Metadatenatzes

Der INSPIRE Metadatenstandard ist vom ISO19115:2003 Standard abgeleitet und ähnelt wegen der gemeinsamen Entstehungshistorie daher sehr stark dem Aufbau des ISO19115:2014 Standard. INSPIRE definiert sogenannte *TG Requirements*, die für eine valide Implementierung verpflichtend sind und sogenannte *TG Recommendations*, die für die Umsetzung empfohlen werden, jedoch auch alternative Ansätze in der Implementierung erlauben. Die *TG Requirements* schreiben nicht nur die syntaktische und semantische Umsetzung vor, sondern definieren teilweise auch genaue Befüllungstexte von Feldern die im ISO Standard eigentlich Freitext sind (z.B.: Spezifikation der Konformität oder Definition der Einschränkungen der Ressource.)

Abbildung 3.3 gibt eine Übersicht über alle Metadatenattribute und Klassen für Datensätze in INSPIRE. Der INSPIRE-Metadatenstandard für Datensätze basiert auf den Prinzipien von ISO 19115, enthält jedoch spezifische Erweiterungen und Anpassungen, die auf die Anforderungen der EU-Richtlinie zugeschnitten sind. Wie bei ISO 19115 steht die zentrale Klasse *MD\_Metadata* im Mittelpunkt, die alle Metadateninformationen bündelt. Diese Klasse beschreibt die Metadaten selbst, darunter Aktualität, Kontaktinformationen und Beschreibungen zur Entstehung und Verfügbarkeit der Daten. Ein wesentlicher Unterschied zu ISO 19115 besteht darin, dass INSPIRE zusätzliche, für die Richtlinie relevante Informationen fordert, insbesondere in Bezug auf Konformität mit den INSPIRE-Verordnungen, die zwar im Standard nicht als Codelisten geführt werden, aber genau definierte Texte für die Freitextfelder vorsehen.

Ein weiterer grundlegender Unterschied zu ISO 19115 ist der Fokus auf rechtliche und lizenzrechtliche Aspekte. INSPIRE fordert obligatorisch die Angabe von Nutzungsbedingungen, Zugangsrechten und möglichen Beschränkungen der Weiterverwendung von Daten. Diese Informationen werden in der Klasse *MD\_LegalConstraints* festgehalten. Während ISO 19115 ebenfalls die Möglichkeit bietet, Nutzungsrechte zu beschreiben, sind diese Angaben im INSPIRE-Kontext verpflichtend und strenger definiert, um die Konformität mit den EU-Datenrichtlinien sicherzustellen. Auch hier sind ähnlich der Konformitätserklärungen, definierte Texte in den Freitextfeldern zu verwenden, ohne entsprechend Codelisten in der Modellierung vorzusehen.

Ein zusätzlicher Unterschied liegt in der Klassifizierung von Ressourcen. In ISO 19115 wird die Ressource über die Klasse *MD\_IdentificationInfo* beschrieben, die entweder durch *MD\_DataIdentification* für Datensätze oder *SV\_ServiceIdentification* für Dienste konkretisiert wird. INSPIRE übernimmt diese Struktur, fordert jedoch zusätzliche Informationen zu den INSPIRE-Themen, die in den Anhängen der Richtlinie definiert sind. Die Angabe wird als *MD\_Keyword* mit bestimmten Thesauri modelliert. Das INSPIRE

### 3. METADATENSTANDARDS IN DER GEOINFORMATION

Datenthema ist auch als erstes Schlüsselwort anzugeben. Zusätzlich müssen für das Monitoring auch *Priority Datasets* oder der *SpatialScope* als Schlüsselwort angegeben werden. Für sonstige Schlüsselwörter ist sind Begriffe aus Thesauri wie dem GEMET (General Multilingual Environmental Thesaurus) vorgeschrieben.

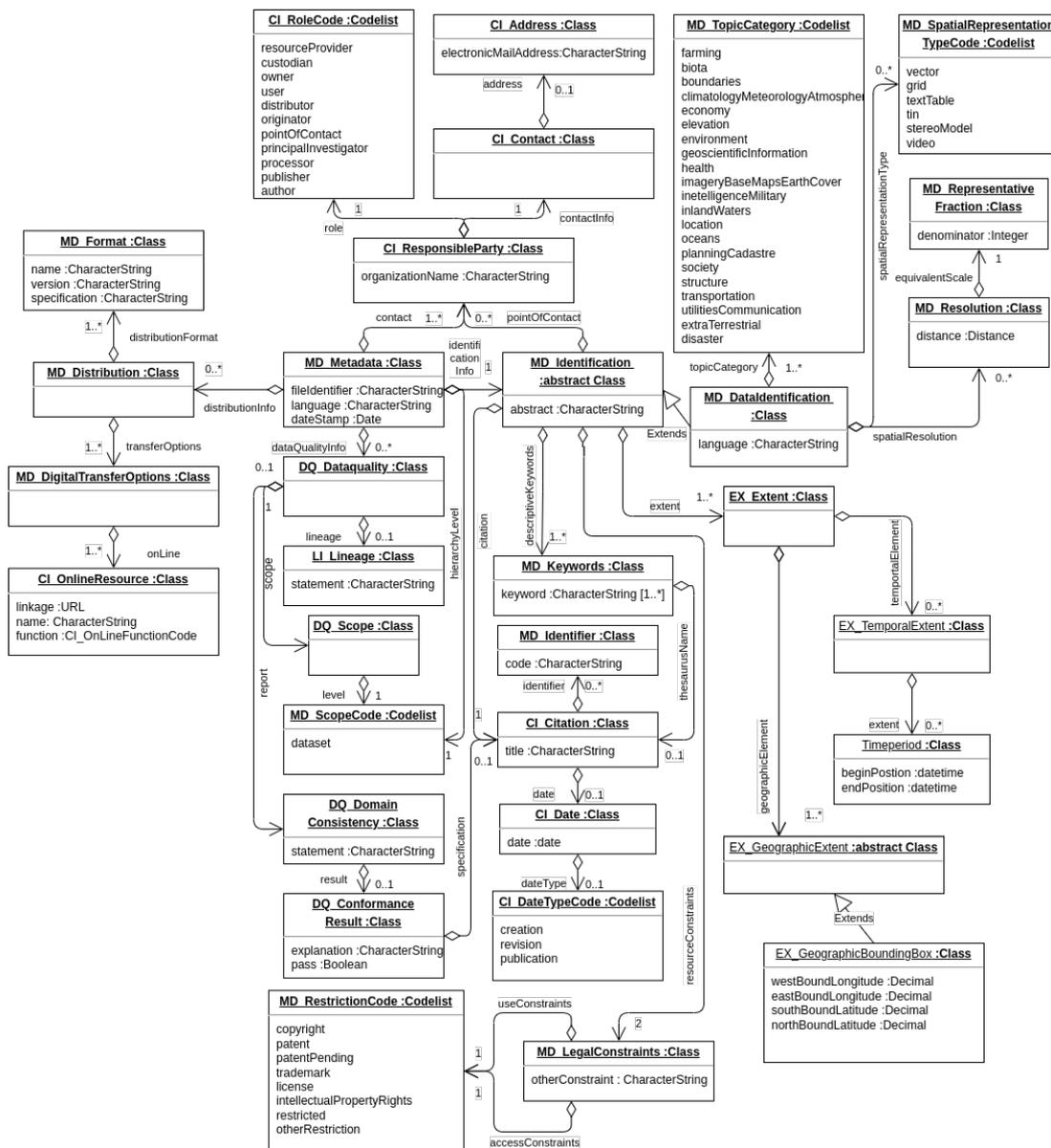


Abbildung 3.3: Übersicht über das INSPIRE Metdatenmodell für Datensätze

Eine Übersicht über das Metadatenmodell für INSPIRE-Dienste ist in Abbildung 3.4 dargestellt. Die Struktur der Metadaten für Dienste baut auf der von ISO 19115:2003

und ISO 19119:2005 festgelegten Architektur auf, ist jedoch aufgrund der spezifischen Anforderungen der INSPIRE-Richtlinie in einigen Bereichen deutlich restriktiver.

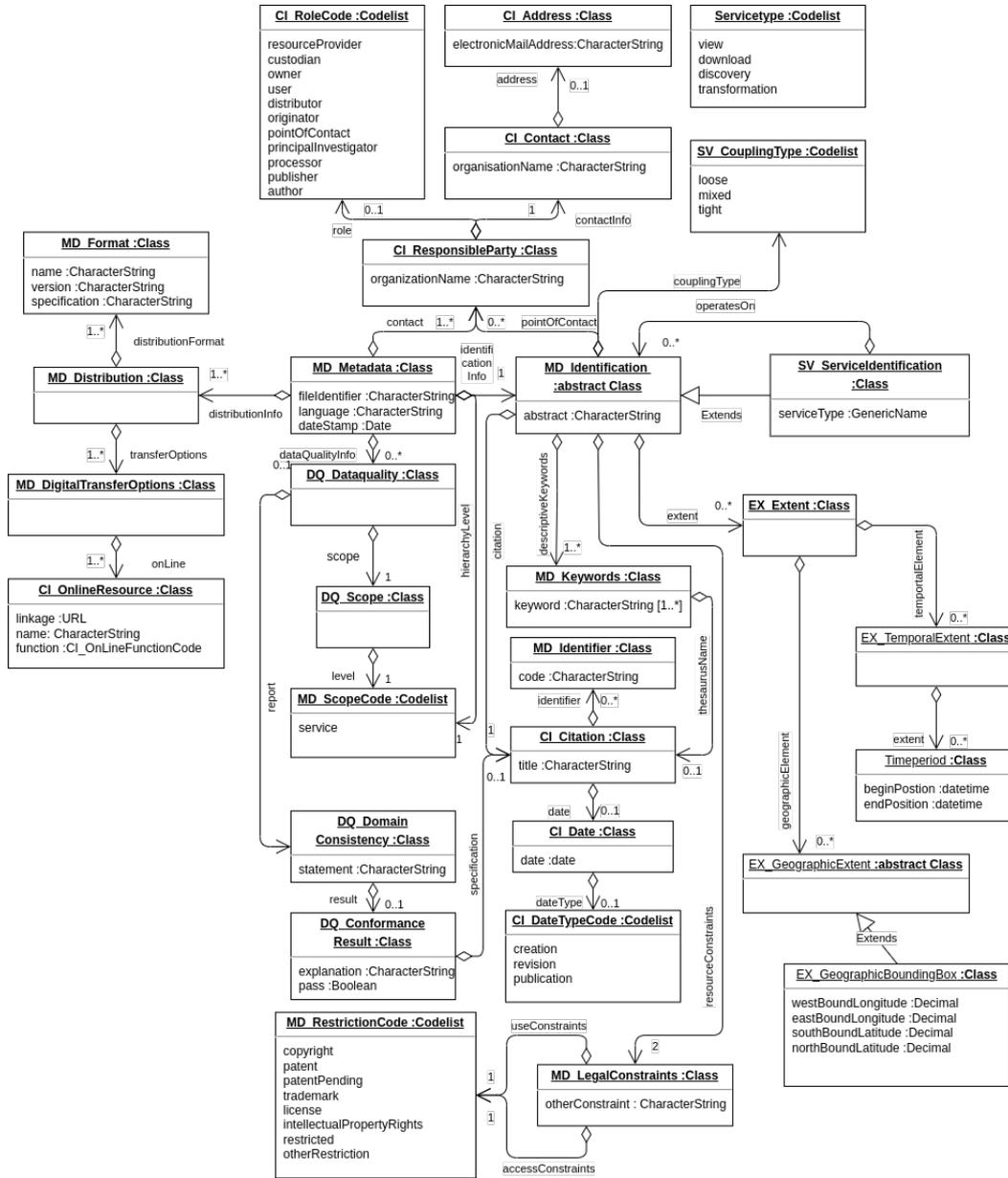


Abbildung 3.4: Übersicht über das INSPIRE Metdatenmodell für Dienste

Die Beschreibung der Konformität zu den INSPIRE-Regeln erfolgt über die Klasse *DQ\_ConformanceResult*, die innerhalb der *DQ\_Quality*-Klasse eingebettet ist. Dies stellt sicher, dass Dienste, die gemäß INSPIRE bereitgestellt werden, den rechtlichen

und technischen Vorgaben entsprechen. Im Vergleich zu ISO 19115 und ISO 19119 ist diese Konformitätsprüfung im INSPIRE-Umfeld verpflichtend und stärker formalisiert, was bedeutet, dass die Metadaten nicht nur die grundlegende Qualität der Daten oder des Dienstes beschreiben, sondern auch explizit deren Einhaltung der INSPIRE-Spezifikationen.

Ein wesentlicher Bestandteil der Dienstmetadaten ist die Klasse *SV\_ServiceIdentification*, die als zentrale Struktur zur Beschreibung der Ressource dient. Diese Klasse spezifiziert allgemeine Informationen zum Dienst, wie den Titel, die Beschreibung, den Typ des Dienstes (*serviceType*) und die Kategorisierung des Dienstes. Die Typisierung erfolgt zusätzlich auch über Schlüsselwörter die ähnlich wie bei den Datasets zwingend vorgeschrieben sind und aus einem INSPIRE Vokabular referenziert werden müssen.

Die Klasse *MD\_Distribution* wird verwendet, um den Zugang zu den Diensten zu beschreiben. INSPIRE verlangt hier nicht nur die Angabe einer URL, sondern auch die Definition des Formats der Ressource, die über das Service publiziert wird.

#### 3.4 Aufbau eines minimalen DCAT-AP Metadatensatzes

Der Zweck von DCAT-AP ist es, gesamte Datenkataloge und die in den Katalogen bereitgestellten Dokumente, Daten und Dienste, sowie deren Distribution zu beschreiben. DCAT-AP hat grundsätzlich keinen expliziten Bezug zu Geodaten, sondern hat allgemeiner Dokumente und deren Distribution im Fokus.

Der Standard unterscheidet zwischen *mandatory* Klassen und Attributen, die jedenfalls zu befüllen sind, *recommended* Attributen, die zu befüllen sind sofern die Information vorhanden ist und vor allem der Auffindbarkeit (*mandatory for discovery*) dienen und optional befüllt werden können, um eine bessere Beschreibung zu bieten oder anderwertig relevant für das Verständnis der Ressource sind. Die Anzahl der verpflichtenden Attribute ist sehr gering, dadurch soll eine hohe Akzeptanz gewährleistet werden. In der Praxis ist jedoch ein Metadatensatz, der nur eine Minimalbefüllung aufweist und nicht noch Verweise auf weiterführende Metadaten bereitstellt nur für sehr simple und selbsterklärende Daten verwendbar.

Die Wurzelklasse ist der *dcat:Catalog*, in der nur der Catalog selbst beschrieben wird. Die Eigenschaften der Ressource werden in der Klasse *dcat:Dataset* festgehalten, die eine 1..\* Relation zur Klasse *dcat:Catalog* aufweist. Kontaktinformationen werden in der Klasse *foaf:Agent* abgebildet. Die Klasse *dcat:Distribution* enthält alle Informationen die beschreiben wie man die Ressource beziehen kann. Dienste werden in der Klasse *dcat:DataService* beschrieben, wobei diese Klasse nur optional ist. Ein *dcat:Catalog* muss mindestens einen Datensatz mit der Klasse *dcat:Dataset* beschreiben. Ein Katalog, der nur Dienste beschreibt, ist nicht möglich. Eine Beschreibung der Distribution ist *recommended*, aber nicht verpflichtend. Daraus ergibt sich, dass ein minimaler Katalog, einen Datensatz und einen Kontakt enthalten muss.

### 3.4. Aufbau eines minimalen DCAT-AP Metadatensatzes

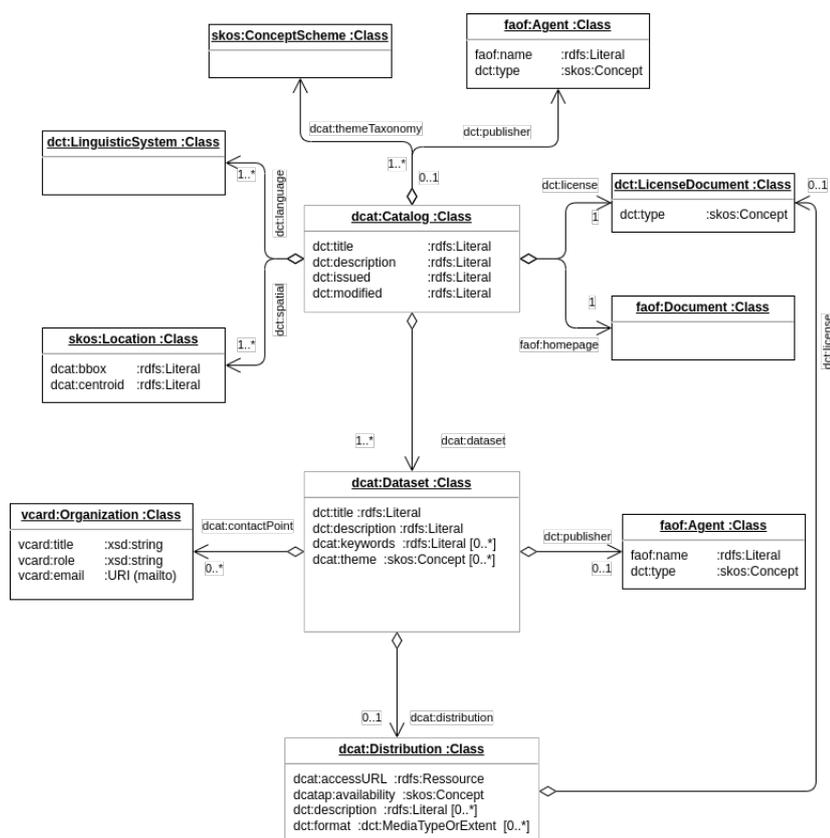


Abbildung 3.5: Übersicht über das DCAT-AP Metdatenmodell für Datensätze

GeoDCAT-AP ist eine Erweiterung von DCAT-AP und definiert sich wie folgt:

*“GeoDCAT-AP is an extension of the DCAT application profile for data portals in Europe (DCAT-AP) for describing geospatial datasets, dataset series, and services. Its basic use case is to make spatial datasets, dataset series, and services searchable on general data portals, thereby making geospatial information better findable across borders and sectors. For this purpose, GeoDCAT-AP provides an RDF vocabulary and the corresponding RDF syntax binding for the union of metadata elements of the core profile of ISO 19115:2003 and those defined in the framework of the INSPIRE Directive of the European Union.”*[Geo20]

Der Zweck der Erweiterung ist daher die Abbildung von INSPIRE Metadaten auf DCAT-AP Metdaten. Der GeoDCAT-AP Core verwendet dabei alle Klassen und Attribute von DCAT-AP ohne Erweiterung. Da nicht alle Informationen von INSPIRE in DCAT-AP abbildbar sind, stellt GeoDCAT-AP Extended eine Erweiterung bereit, die vordergründig aus Properties aus vorhandenen Vokabularen bestehen. Für Informationen die nicht in anderen Vokabularen existieren wurden eigene Properties definiert, die unter dem Namespacekürzel *geodcat* angeführt werden.

### 3. METADATENSTANDARDS IN DER GEOINFORMATION

Die Transformation von INSPIRE Metadaten zu GeoDCAT Metadaten ist nicht verlustfrei. Da die Transformation von INSPIRE Metadaten in GeoDCAT-AP jedoch nur die Auffindbarkeit und Durchsuchbarkeit in Datenkatalogen der EU vereinfachen soll, ist der dabei auftretende Informationsverlust nachrangig.

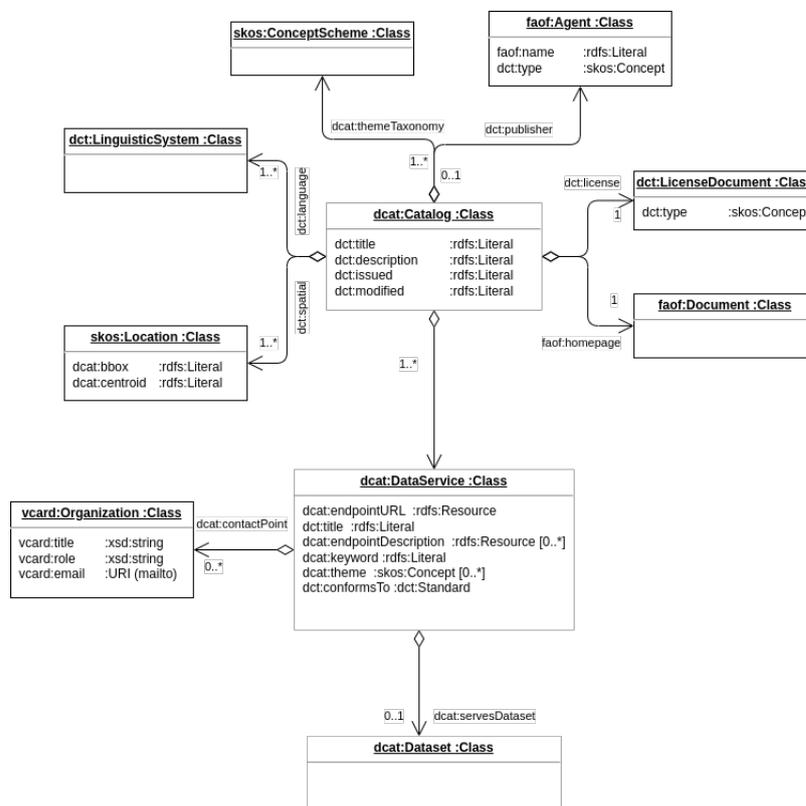


Abbildung 3.6: Übersicht über das DCAT-AP Metdatenmodell für Dienste

# Harmonisierung der Metadatenstandards

Für die Harmonisierung der Metadatenstandards wird eine Ontologie erarbeitet. Das Ziel der Ontologie ist es alle Metadaten der Metadatenstandards ISO 19115, INSPIRE und DCAT-AP, die minimal erforderlich sind, in einem Modell zu vereinigen, um mit der Befüllung eines Metadatensatzes alle angeführten Standards in einer validen Form produzieren zu können.

## 4.1 Vorgehen

Die Erarbeitung der Ontologie orientiert sich an dem Vorgehen des Ontology 101-Ansatzes:

- **Klassenidentifikation:** Bestimmung der wichtigsten Klassen (Konzepte) der Ontologie, die die relevanten Objekte der Domäne darstellen.
- **Hierarchiebildung:** Strukturierung der Klassen in einer Hierarchie, wobei übergeordnete und untergeordnete Beziehungen zwischen den Klassen festgelegt werden.
- **Eigenschaftenbestimmung (Slots):** Definition der Attribute oder Merkmale der Klassen und ihrer Instanzen.
- **Einschränkungen (Facetten):** Festlegung von Einschränkungen für die Eigenschaften, wie z.B. zulässige Werte oder Datentypen.
- **Instanzenerstellung:** Erstellung von konkreten Instanzen der definierten Klassen, die reale Objekte der Domäne repräsentieren.

- **Evaluierung und Verfeinerung:** Überprüfung und Verbesserung der Ontologie durch iterative Evaluierungen, um sicherzustellen, dass sie konsistent, korrekt und nützlich ist.

Das Vorgehen des Ontology 101-Ansatzes ist kein linearer Prozess, sondern ein iterativer Ansatz. Das bedeutet, dass die einzelnen Schritte nicht streng nacheinander durchlaufen werden, sondern dass ein stetiger Rückgriff auf vorherige Schritte erforderlich ist, um die Ontologie zu verfeinern und anzupassen. In diesem Kapitel wird nicht der Entstehungsprozess dokumentiert, sondern das Endergebnis mit den aus dem Prozess gewonnen Erkenntnissen präsentiert. Für das Vokabular werden die englischen Bezeichnungen verwendet, da die Standards nur in englischer Sprache vorliegen und so eine bessere Nachvollziehbarkeit gewährleistet wird.

Die Metadatenstandards beschreiben einen Datensatz (Dataset) oder einen Dienst (Service), wobei ein Dienst einen Datensatz bereitstellen kann. Es ist prinzipiell auch möglich eine Serie - eine Serie von Datensätzen zu beschreiben. Eine Serie bildet eine logische Klammer über mehrere Datensätze. In DCAT-AP ist die Beschreibung von Serien bis zur Veröffentlichung der Version 3.0.0 auch nicht möglich gewesen, ISO 19115 und INSPIRE unterstützen den Typ seit Beginn. In der Praxis kommen Serien kaum vor, als Beispiel werden exemplarisch Befliegungen von Orthophotos angeführt. Da Serien keine zusätzlichen Attribute benötigen, sondern nur auf andere Datensätze oder Dienste referenzieren und für eine valide Abbildung der Metainformation nicht verpflichtend sind, wird in dieser Arbeit auf eine Implementierung verzichtet.

DCAT-AP kennt eine abstrakte Klasse *Resource*, welche als Superklasse für Datensätze und Dienste dient. INSPIRE und ISO19115 modellieren das äquivalent Klasse als *MD\_Identification*, von dem *MD\_Dataidentification* für Datensätze bzw. *SV\_ServiceIdentification* für Dienste abgeleitet wird. In den INSPIRE Technical Guidelines wird auch das Vokabular *Resource title* oder *Resource format* verwendet, um die Attribute der *MD\_Dataidentification* Klasse zu beschreiben. Als Basis für alle Elemente der Geodateninfrastruktur wird daher der Begriff *Resource* verwendet.

Für die Publikation von Metadaten verwendet DCAT-AP einen Metadatenkatalog *dcat:Catalog*, der alle Ressourcen referenziert. In INSPIRE und ISO19115 ist das Pendant dazu ein *Discovery Service*, welches die Metadaten referenziert und bereitstellt. Daraus ergibt sich, dass die minimale Infrastruktur für die Bereitstellung von Metadaten einen *Discovery Service* benötigt.

In INSPIRE wird ein Downloaddienst für eine valide Infrastruktur gefordert, in ISO19115 ist ein Downloaddienst möglich aber nicht verpflichtend. Für die Distribution von Datensätzen kann in DCAT-AP als Downloaddienst optional ein *dcat:Dataservice* beschrieben werden. Somit ist für eine minimale Infrastruktur auch ein *Download Service* notwendig.

Abbildung 4.1 gibt einen Überblick über die minimale Geodateninfrastruktur die in Folge für die Erarbeitung der Ontologie als Referenz verwendet wird.

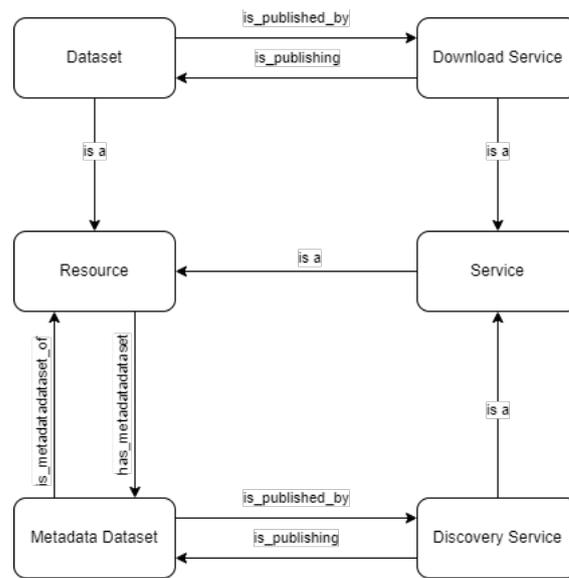


Abbildung 4.1: Architektur der Referenz-Geodateninfrastruktur

Für die Validierung der Ontologie ergibt sich daher die Anlage eines Datensatzes, eines Discovery- und eines Download-Dienstes.

## 4.2 Klassen und Klassenhierarchie

Im folgenden Abschnitt werden die Klassen der Ontologie definiert. Die Konzepte wurden aus den INSPIRE Metadata Technical Guidelines, dem ISO 19115 Metadatenstandard und dem DCAT-AP Metadatenstandard extrahiert. Die Definitionen wurden selbst formuliert. Eine Definition für das KonzeptA und KonzeptB wird folgendermaßen durchgeführt:

$\exists x (\text{KonzeptA}(x)) = \text{def. Definition von A.}$

$\exists x (\text{KonzeptB}(x)) = \text{def. Definition von B.}$

Die Klassenhierarchie beschreibt die “is\_a” Beziehung zwischen den Klassen:

$\forall x (\text{KonzeptA}(x) \Rightarrow \text{KonzeptB}(x))$

Aufgrund der Übersichtlichkeit werden die Hierarchien in Listings dargestellt, wobei eine Einrückung eine “is\_a” Beziehung definiert. Klassen auf gleicher Einrückungsebene (KonzeptA und KonzeptC) befinden sich daher auf gleicher Hierarchiestufe:

```

1 KonzeptA
2   KonzeptB
3   KonzeptC
  
```

Listing 4.1: Beispiel Klassenhierarchie als Listing

Die Beschreibung der Klassen und der Klassenhierarchie wurde in mehrere Unterabschnitte zusammengefasst. Die Aufteilung ergab sich aus der praktischen Umsetzung, hat jedoch keine inhaltliche Relevanz. Das Endergebnis ist die Summe aller Definitionen und Klassenhierarchien und wird gesamt im Listing 6.1 im Anhang dargestellt.

### 4.2.1 Referenzarchitektur

#### Definitionen

Aus der Referenz-Geodateninfrastruktur ergeben sich die Definitionen der Klassen wie folgt:

$\exists x$  (Resource( $x$ )) = def. any entity that is managed within a geodata infrastructure.

$\exists x$  (Dataset( $x$ )) = def. a collection of related data.

$\exists x$  (Service( $x$ )) = def. a functionality provided to users or systems.

$\exists x$  (MetadataDataset( $x$ )) = def. provides descriptive information about a resource.

$\exists x$  (DiscoveryService( $x$ )) = def. a service that allows users and systems to search and find resources.

$\exists x$  (DownloadService( $x$ )) = def. a service that enables users and systems to retrieve datasets.

Für die standardspezifischen Klassen der einzelnen Ressourcen ergeben sich folgende Definitionen:

$\exists x$  (InspireDataset( $x$ )) = def. a dataset resource that contains all metainformation compliant to the INSPIRE technical guidelines.

$\exists x$  (DcatApDataset( $x$ )) = def. a dataset resource that contains all metainformation compliant to the DCAT-AP specification.

$\exists x$  (ISO19115Dataset( $x$ )) = def. a dataset resource that contains all metainformation compliant to the ISO19115 specification.

$\exists x$  (DcatApCatalog( $x$ )) = def. a service that represents all metainformation to describe a Catalog class compliant to the DCAT-AP specification.

$\exists x$  (InspireDiscoveryService( $x$ )) = def. a service that represents all metainformation to describe an INSPIRE Discovery Service compliant to the INSPIRE technical guidelines.

$\exists x$  (ISO19115DiscoveryService( $x$ )) = def. a service that represents all metainformation to describe a Discovery Service compliant to the ISO19115 specification.

$\exists x$  (DcatApDataService( $x$ )) = def. a service that represents all metainformation to describe a DataService class compliant to the DCAT-AP specification.

$\exists x$  (InspireDownloadService( $x$ )) = def. a service that represents all metainformation to describe an INSPIRE Download Service compliant to the INSPIRE technical guidelines.

$\exists x$  (ISO19115DownloadService( $x$ )) = def. a service that represents all metainformation to describe a Download Service compliant to the ISO19115 specification.

### Hierarchie

Der Aufbau der der Klassenhierarchie für die Referenzarchitektur wird in Listing 4.2 definiert:

```

1  Thing
2  Resource
3  MetadataDataset
4  Dataset
5  InspireDataset
6  DcatApDataset
7  ISO19115Dataset
8  Service
9  DiscoveryService
10 DcatApCatalog
11 InspireDiscoveryService
12 ISO19115DiscoveryService
13 DownloadService
14 DcatApDataService
15 InspireDownloadService
16 ISO19115DownloadService

```

Listing 4.2: Klassenhierarchie der Referenzarchitektur

#### 4.2.2 Codelisten

INSPIRE und ISO19115 verwenden zur Beschreibung der Datensätze und Dienste Codelisten aus denen vordefinierte Attribute ausgewählt werden. Die Codelisten referenzieren entweder auf ISO Standards oder auf die INSPIRE Registry. Mit den Codelisten soll sichergestellt werden, dass im Gegensatz zu Freitext Attributen keine Seiteneffekte in der Definition entstehen und die maschinelle Verarbeitung, insbesondere die Validierung sichergestellt ist. Durch die Referenzierung auf den ISO Standard bzw. die INSPIRE Registry wird auch die Semantik der Einträge definiert. Codelisten sind nicht veränderlich und können nicht einfach um eigene Einträge erweitert werden.

### Definitionen

- $\exists x$  (Codelist( $x$ )) = def. an enumerated list of values with an anchor URL pointing to a fixed definition.
- $\exists x$  (CharacterSet( $x$ )) = def. the encoding of the resource.
- $\exists x$  (Condition( $x$ )) = def. a requirement or restriction associated with the use or access of a resource.
- $\exists x$  (AccessCondition( $x$ )) = def. a condition that specifies the requirements for accessing a resource.
- $\exists x$  (UseCondition( $x$ )) = def. a condition that specifies the requirements for using a resource.
- $\exists x$  (ContactRole( $x$ )) = def. the role of a contact person or organization related to the resource.
- $\exists x$  (CoordinateSystem( $x$ )) = def. a reference system used to represent the locations of geographic features.
- $\exists x$  (CouplingType( $x$ )) = def. the nature of the link between a service and its associated data.
- $\exists x$  (DateType( $x$ )) = def. the classification of a date, indicating its purpose (e.g., creation, publication).
- $\exists x$  (DCP( $x$ )) = def. a protocol or platform used for communication in web services (Distributed Computing Platform).
- $\exists x$  (Language( $x$ )) = def. the language in which the resource or service is provided.
- $\exists x$  (DistributionFormat( $x$ )) = def. the format of the data in the resource.
- $\exists x$  (Restriction( $x$ )) = def. a limitation or constraint on the use or access of a resource.
- $\exists x$  (Scope( $x$ )) = def. the extent or coverage of the resource or service in terms of purpose or applicability.
- $\exists x$  (ServiceType( $x$ )) = def. the classification of a service based on its function or purpose.
- $\exists x$  (SpatialRepresentationType( $x$ )) = def. the method used to represent spatial data in a dataset.
- $\exists x$  (TransferProtocol( $x$ )) = def. the protocol used for transferring data between systems or services.

## Hierarchie

Der Aufbau der Klassenhierarchie von Codelisten ist in Listing 4.3 dargestellt:

```

1  Thing
2  CodeList
3    CharacterSet
4    Condition
5      AccessCondition
6      UseCondition
7    ContactRole
8    CoordinateSystem
9    CouplingType
10   DateType
11   DCP
12   Language
13   DistributionFormat
14   Restriction
15   Scope
16   ServiceType
17   SpatialRepresentationType
18   TransferProtocol

```

Listing 4.3: Klassenhierarchie der Codelisten

### 4.2.3 Kontrollierte Vokabulare

Kontrollierte Vokabulare bieten eine reichhaltige Struktur zur Beschreibung von Inhalten, Beziehungen und Konzepten, die über einfache Codelisten hinausgeht. Thesauri wie GEMET, bieten eine hierarchische und semantisch vernetzte Struktur von Begriffen, die komplexe Suchanfragen und semantische Interoperabilität ermöglichen. INSPIRE verwendet den GEMET Thesaurus für Schlüsselwörter und INSPIRE Datenthemen. Weiters werden zusätzliche Thesauri für die Beschreibung von Priority Datasets, den SpatialScope und die Art des Spatial Data Service verwendet. Für die Befüllung der Metadatenstandards ist es nötig einerseits das kontrollierte Vokabular selbst zu beschreiben, als auch die Werte, die aus dem kontrollierten Vokabular verwendet werden.

#### Definitionen

$\exists x$  (ControlledValue( $x$ )) = def. a predefined value of a controlled vocabulary.

$\exists x$  (Keyword( $x$ )) = def. a term used to describe the content of a resource.

$\exists x$  (KeywordGemetConcept( $x$ )) = def. a keyword that corresponds to a concept defined in the GEMET thesaurus.

$\exists x$  (KeywordGemetInspireTheme( $x$ )) = def. a keyword that corresponds to a theme defined by INSPIRE within the GEMET Thesaurus.

#### 4. HARMONISIERUNG DER METADATENSTANDARDS

---

- $\exists x$  (KeywordInspirePriorityDataset( $x$ )) = def. a keyword used to identify datasets that are prioritized under INSPIRE due to their significance for environmental reporting.
- $\exists x$  (KeywordInspireSpatialDataService( $x$ )) = def. a keyword used to categorize services that provide access to spatial data within the INSPIRE framework.
- $\exists x$  (KeywordInspireSpatialScope( $x$ )) = def. a keyword that specifies the geographical or spatial scope of a dataset within the INSPIRE framework.
- $\exists x$  (QualityReportExplanation( $x$ )) = def. a textual description providing information about the quality evaluation process of a dataset or service. INSPIRE predefines texts to indicate the conformance of datasets and services with this explanation.
- $\exists x$  (QualityReportPass( $x$ )) = def. a boolean indicator that defines if a dataset or service meets the specified quality requirements.
- $\exists x$  (TopicCategory( $x$ )) = def. an ISO classification used to broadly group datasets by subject.
- $\exists x$  (ControlledVocabulary( $x$ )) = def. an organized collection of terms and definitions ensuring semantic interoperability.
- $\exists x$  (QualityReportSpecification( $x$ )) = def. a formal statement outlining the quality criteria and standards against which a dataset or service is validated.
- $\exists x$  (Thesaurus( $x$ )) = def. a structured and organized set of terms and concepts.
- $\exists x$  (ThesaurusGemetConcept( $x$ )) = def. a thesaurus that includes concepts from the GEMET vocabulary.
- $\exists x$  (ThesaurusGemetInspireTheme( $x$ )) = def. a thesaurus specifically aligned with INSPIRE themes based on the GEMET Thesaurus.
- $\exists x$  (ThesaurusInspirePriorityDataset( $x$ )) = def. a thesaurus that includes terms related to INSPIRE priority datasets.
- $\exists x$  (ThesaurusInspireSpatialDataService( $x$ )) = def. a thesaurus used to standardize terms for categorizing spatial data services within the INSPIRE framework.
- $\exists x$  (ThesaurusInspireSpatialScope( $x$ )) = def. a thesaurus that provides standardized terms for defining the spatial scope of datasets within the INSPIRE framework.

## Hierarchie

Die Struktur der Klassenhierarchie der kontrollierten Vokabulare ist in Listing 4.4 dargestellt:

```

1  Thing
2    ControlledValue
3      Keyword
4        KeywordGemetConcept
5        KeywordGemetInspireTheme
6        KeywordInspirePriorityDataset
7        KeywordInspireSpatialDataScope
8    QualityReportExplanation
9    QualityReportPass
10   TopicCategory
11   ControlledVocabulary
12   QualityReportSpecification
13   Thesaurus
14     ThesaurusGemetConcept
15     ThesaurusGemetInspireTheme
16     ThesaurusInspirePriorityDataset
17     ThesaurusInspireSpatialDataService
18     ThesaurusInspireSpatialScope

```

Listing 4.4: Klassenhierarchie der kontrollierte Vokabulare

### 4.2.4 Komplexe Klassen

Komplexe Klassen setzen sich aus ein oder mehreren Codelisteneinträgen und teilweise auch aus Freitexteinträgen zusammen. Die Attribute beschreiben jedoch immer ein gemeinsames Konstrukt, d.h. die einzelnen Eigenschaften müssen über diese Klammer zusammengeführt werden, damit sie einen gemeinsamen Sinn ergeben. Da die komplexen Klassen inhaltlich keine Gemeinsamkeiten haben werden sie auch unabhängig voneinander modelliert.

#### Definitionen

$\exists x$  ( $\text{Contact}(x)$ ) = def. an entity responsible for information or services related to a resource.

$\exists x$  ( $\text{Constraint}(x)$ ) = def. a rule defining access or use conditions of a resource.

$\exists x$  ( $\text{AccessConstraint}(x)$ ) = def. a constraint that regulates access to a resource.

$\exists x$  ( $\text{UseConstraint}(x)$ ) = def. a constraint that specifies terms of use for a resource.

$\exists x$  ( $\text{DigitalTransferOption}(x)$ ) = def. a method for the electronic transfer of data.

$\exists x$  ( $\text{GeographicExtent}(x)$ ) = def. a spatial definition of the area covered by a resource.

$\exists x$  ( $\text{QualityReport}(x)$ ) = def. a report detailing the quality of a resource.

$\exists x$  ( $\text{ServiceOperation}(x)$ ) = def. an operation provided by a service.

## Hierarchie

Die Klassenhierarchie der komplexen Klassen ist in Listing 4.5 veranschaulicht.

```

1  Thing
2    Constraint
3      AccessConstraint
4      UseConstraint
5  Contact
6  DigitalTransferOption
7  GeographicExtent
8  QualityReport
9  ServiceOperation

```

Listing 4.5: Klassenhierarchie der komplexen Klassen

## 4.3 Eigenschaften und Einschränkungen

Die ersten Modellierungsversuche der Eigenschaften wurden ähnlich dem Aufbau eines Datenmodells in einer Datenbank durchgeführt, wobei als Domäne immer die spezifischen Klassen der Subjekte verwendet wurden. Für die Benennung wurde *has\_* gefolgt von dem Typ der Klasse des Subjekts, gefolgt von dem Typ der Klasse des Objekts verwendet. Dieser Ansatz hat die Komplexität der Ontologie massiv erhöht, da die Anzahl der Properties gestiegen ist und auch die Benennung sehr lange Property-Namen produziert hat, ohne weitere Vorteile zu bringen.

Alle Objekteigenschaften und alle Dateneigenschaften werden daher in der Ontologie universell modelliert. Dazu wird als Domäne immer *Thing* verwendet, sodaß sie mit jeder Klasse verwendet werden können. Für die Range wird bei den Objekteigenschaften immer ein spezifischer Datentyp verwendet, bei den Dateneigenschaften wird der Einfachheit halber für alle simplen Datentypen ein *string* verwendet.

Der Namensaufbau beginnt mit *has\_* gefolgt mit einem Wort, welches die Semantik des Objekts beschreibt, z.B.: *has\_creationdate*. Gibt es mehrere Ausprägungen dieser Semantik, z.B.: bei der Beschreibung einer geografischen Ausdehnung wird die nähere Spezifikation mit einem weiteren Unterstrich angefügt, z.B.: *has\_extent\_west*. Prädikate die nicht auf simple Datentypen, sondern auf Klassen verweisen sind exakt gleich aufgebaut, wobei als Semantik der Typ der Klasse verwendet wird, z.B.: *has\_scope* referenziert das Objekt *Scope*.

Eigenschaften die bei einer Instanz einer Klasse nur einmal vorkommen dürfen, werden als *functional properties* markiert. Als Beispiel sei hier der *has\_title* genannt, da es keinen Sinn macht, das eine Instanz mehr als einen Titel hat. Diese Einschränkung leitet sich aus den Kardinalitäten der Objektmodelle der Metadatenstandards ab.

$$\exists x (\text{Thing}(x) \wedge \exists^=1 y (\text{has\_title}(x, y) \wedge \text{string}(y)))$$

Grundsätzlich werden in der Ontologie für Ressourcen nur Instanzen der Klassen *Dataset* und *Service* angelegt. Die spezifischen Servicetypen wurden mit Axiomen modelliert und werden so durch Reasoning automatisch abgeleitet, sobald alle für die spezifische Ressource erforderlichen Objekteigenschaften und Dateneigenschaften angelegt werden. Dadurch ist eine flexible Erweiterbarkeit der Ressourcen in der Ontologie gegeben.

$$\begin{aligned} \forall x (\text{DcatApDataset}(x) \Leftrightarrow & \\ & \text{Dataset}(x) \wedge \\ & \exists \text{title} (\text{has\_title}(x, \text{title}) \wedge \text{string}(\text{title})) \wedge \\ & \exists \text{abstract} (\text{has\_abstract}(x, \text{abstract}) \wedge \text{string}(\text{abstract})) \wedge \\ & \exists \text{contact} (\text{has\_publisher}(x, \text{contact}) \wedge \text{Contact}(\text{contact})) \wedge \\ & \exists \text{contactrole} (\text{has\_publisher\_contactrole}(x, \text{contactrole}) \wedge \text{ContactRole}(\text{contactrole}))) \end{aligned}$$

Für die Modellierung der Kontakte und Kontaktrollen für INSPIRE und ISO19115 wurde ein impliziter Ansatz gewählt. Die Relation zwischen Datensätzen und Services zu Kontakten ist eine m:n Beziehung. Auf der Beziehung ist in den beiden Standards die Rolle in der sich der Kontakt in Bezug auf das Dataset oder das Service befindet, als Eigenschaft modelliert. Für eine äquivalente Umsetzung müsste jedem Contact Individual vor der Zuweisung zur Ressource eine Rolle hinzugefügt werden. Durch die implizite Verwendung der Rolle in der Semantik der Relation *has\_publisher* und der Zuweisung eines beliebigen Kontakts, kann die *contactrole* in Folge automatisiert bestimmt werden und die Verwendung der Ontologie vereinfacht sich dadurch. INSPIRE und ISO19115 lassen über eine definierte Codeliste relativ viele Rollen für Kontakte zu. In der Arbeit werden aber nur die Rollen *pointOfContact* und *publisher* verwendet. Alle anderen Kontaktrollen lassen sich analog implementieren. Die Implementierung der Datumsangaben zu einer Ressource wurden äquivalent modelliert, da das Modell von INSPIRE und ISO19115 einen vergleichsweise komplexen Modellierungsansatz dafür gewählt haben.

Eine Übersicht über alle Objekt- und Dateneigenschaften findet sich im Anhang. Die Axiome für die Bestimmung der spezifischen Subtypen der Klassen *Dataset* und *Service* befindet sich ebenfalls im Anhang.

Mit diesem Konstrukt ist es möglich alle relevanten Informationen für DCAT-AP, ISO19115 und INSPIRE Metadaten für Ressourcen zu erfassen und die Ressourcen automatisiert zu spezifischen Typen zu klassifizieren.

## 4.4 Instanzen und Validierung

Für die Validierung wurde die erarbeitete Ontologie in Protege implementiert. Protege bietet ein komfortables User Interface, welches mit dynamischen Auswahllisten die Eingabe unterstützt.

#### 4. HARMONISIERUNG DER METADATENSTANDARDS

Nach Implementierung der Klassen und Klassenhierarchien kann mit dem OWLViz Plugin eine Visualisierung erzeugt werden. Abbildung 4.2 bietet eine grafische Übersicht über die Taxonomie der Ontologie.

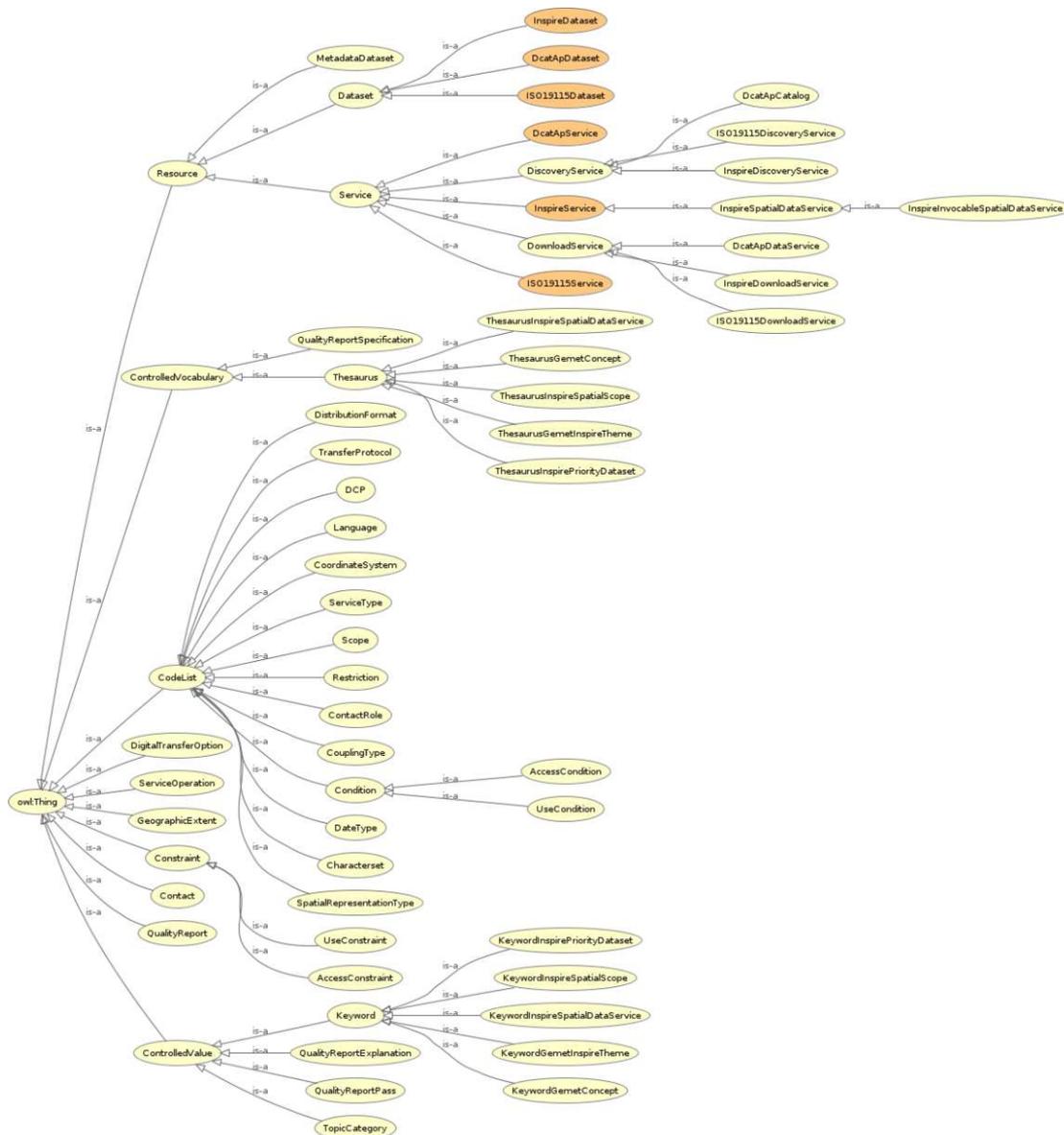


Abbildung 4.2: Grafische Darstellung der Taxonomie der Ontologie

Für alle Codelisten, kontrollierten Vokabulare und komplexen Klassen wurden manuell jene Individuals erzeugt, die für einen grundlegenden Test und die Validierung der Ontologie notwendig waren. Auf eine umfassende Implementierung aller möglichen Werte der Codelisten und kontrollierten Vokabulare wurde aufgrund der steigenden Komplexität

und der dadurch leidenden Übersicht in Protege verzichtet.

Für die Prüfung der Axiome wurde der Pellet Reasoner verwendet, da er nicht nur TBox Reasoning beherrscht sondern in Folge auch für das ABox Reasoning mit den Individuals eingesetzt werden kann.

In Abbildung 4.3 ist ein Screenshot von Protege im Zuge der Validierung zu sehen. Die Ansicht bietet eine Übersicht zur Klasse *DcatApService*. Links ist die Baumstruktur der Klassenhierarchie abgebildet. Im rechten oberen Bereich werden die Annotationen der Klasse angezeigt. Im unteren Bereich sieht man das Axiom, welches die Instanzen der allgemeinen Klasse *Service* zu der spezielleren Klasse *DcatApService* ableitet. Alle gelb markierten Informationen wurden nicht spezifiziert, sondern wurden durch Reasoning logisch geschlossen. Die angezeigten Service-Individuals erfüllen das Axiom der Klasse *DcatApService*, daher sind sie auch Individuals der Klasse und besitzen auch den Typ.

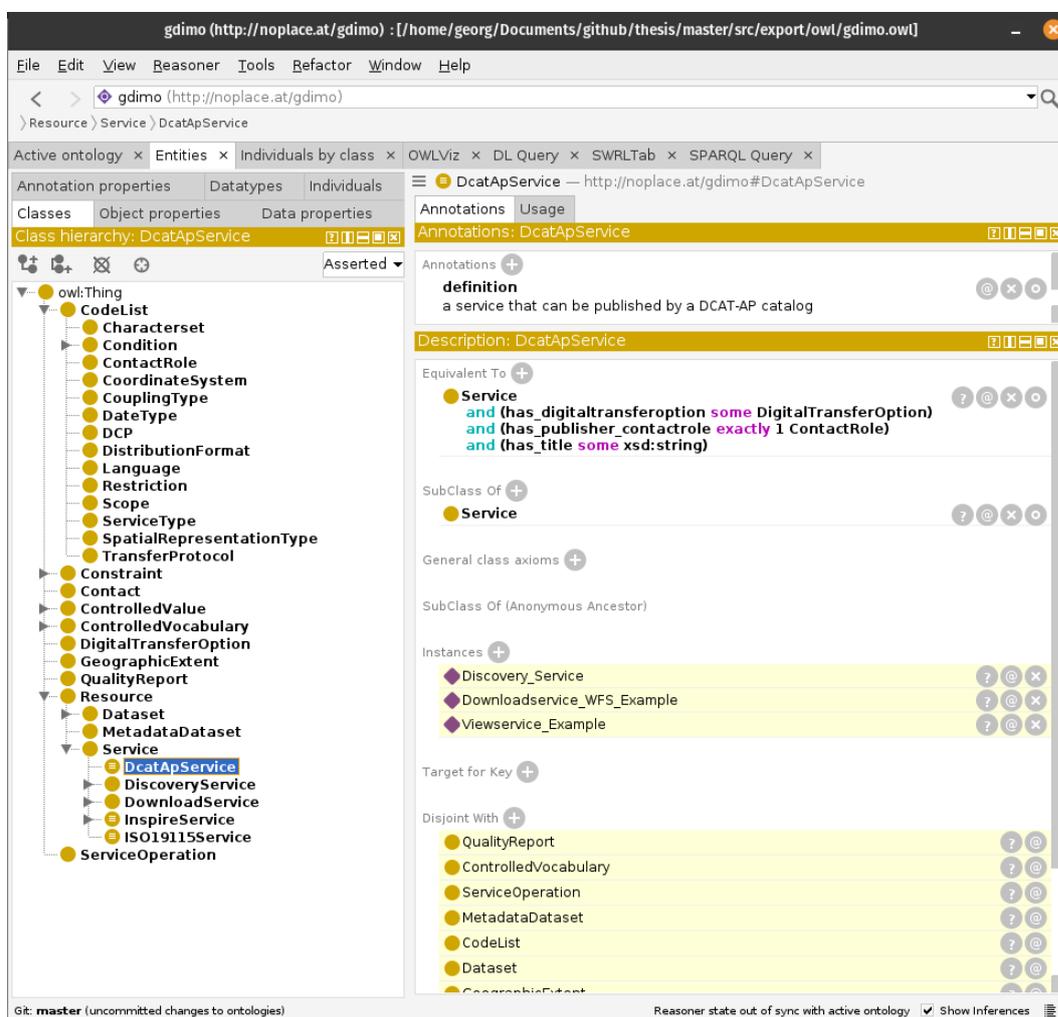


Abbildung 4.3: Klassen-Ansicht von Protege mit Reasoning

Abbildung 4.4 zeigt einen Screenshot von der Individuals Ansicht. Das ausgewählte Download Service hat neben der definierten Klasse *Service* durch Reasoning auch die Klassifizierungen *DcatApService*, *ISO19115Service* und *INSPIREService* erhalten.

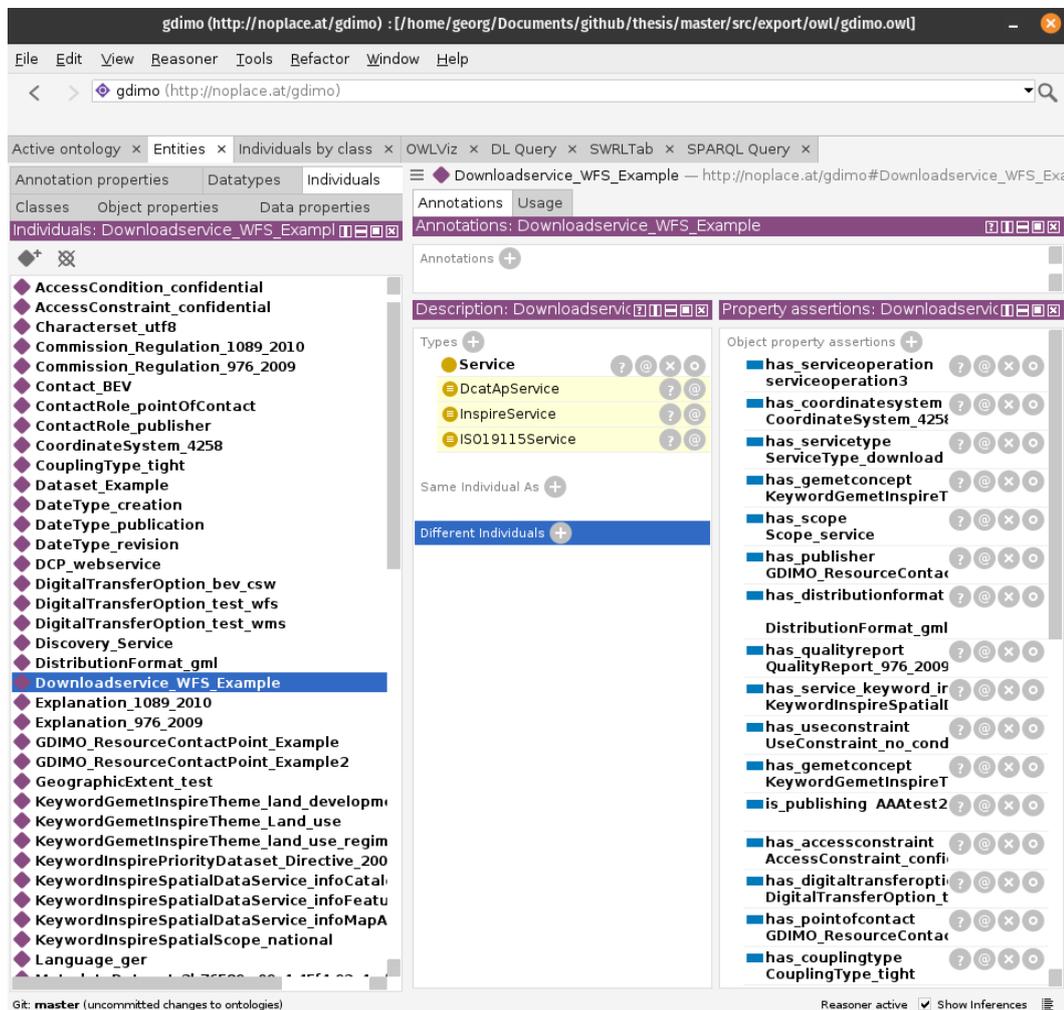


Abbildung 4.4: Individual-Ansicht von Protege mit Reasoning

## 4.5 Implementierung

Die Harmonisierung der Metadatenstandards konnte unter Zuhilfenahme von Protege erfolgreich durchgeführt und validiert werden. Der entstandene Wissensgraph modelliert alle notwendigen Informationen, um die Mindestanforderungen für die Metadatenstandards zu erfüllen. Durch die Umsetzung als Ontologie und der Einsatz eines Resoners konnte gezeigt werden, dass durch logisches Schließen automatisiert Klassifizierungen der Ressourcen durchgeführt werden konnten. Es wurde jedoch festgestellt, dass es in OWL

und SWRL nicht möglich ist automatisiert Individuals zu erstellen oder DataProperties zu manipulieren. Dadurch können keine Prozesse wie z.B.: eine Publikation einer Resource auf einem Service abgebildet werden. Die Modellierung der notwendigen Klassen und Eigenschaften der Referenzarchitektur war zwar möglich, aber für eine Verwendung benötigt es weiterführende Möglichkeiten einer Programmiersprache. Daher wurde die Entscheidung getroffen, die Ontologie owlready2 erneut implementieren und die noch fehlenden Funktionalitäten in python zu umzusetzen. Die Umsetzung in python erlaubt es auch in weiterer Folge Kodierungen der Metadaten in XML für INSPIRE bzw. ISO19115 und RDF für DCAT-AP zu erzeugen. Damit ist es möglich das harmonisierte Modell nicht nur manuell zu validieren, sondern im Internet verfügbare Validierungsapplikationen zu verwenden.

Für die Umsetzung des Publikationsprozesses wurden das allgemeine Konzept der Referenzarchitektur folgendermaßen implementiert:

$$\forall x (\text{Thing}(x) \Rightarrow \exists^1 y (\text{has\_metadatadataset}(x, y) \wedge \text{MetadataDataset}(y)))$$

$$\forall x, y (\text{has\_metadatadataset}(x, y) \Leftrightarrow \text{is\_metadatadataset\_of}(y, x))$$

$$\forall x (\text{Thing}(x) \Rightarrow \exists y (\text{is\_publishing}(x, y) \wedge \text{Resource}(y)))$$

$$\forall x, y (\text{is\_publishing}(x, y) \Leftrightarrow \text{is\_published\_by}(y, x))$$

**has\_metadatadataset:** Diese Beziehung drückt aus, dass ein Objekt vom Typ *Thing* mit einem Metadatensatz (*MetadataDataset*) verknüpft ist. Es handelt sich um eine funktionale Eigenschaft, was bedeutet, dass ein Objekt nur mit genau einem Metadatensatz verknüpft sein kann.

**is\_metadatadataset\_of:** Diese Beziehung beschreibt, dass ein Metadatensatz zu einer bestimmten Ressource (*Resource*) gehört. Es ist die inverse Beziehung zu *has\_metadatadataset*, was bedeutet, dass wenn ein Objekt einen Metadatensatz hat, dieser Metadatensatz umgekehrt zu einer Ressource gehört.

**is\_publishing:** Diese Beziehung besagt, dass ein Objekt vom Typ *Thing* eine Ressource (*Resource*) veröffentlicht.

**is\_published\_by:** Diese Beziehung gibt an, dass ein Objekt von einem bestimmten Service (*Service*) veröffentlicht wird. Es handelt sich um die inverse Beziehung zu *is\_publishing*.

Damit ist es möglich Ressourcen über andere Ressourcen zu publizieren. Der Code in Listing 4.6 definiert eine Funktion die eine Publikation auf Basis der Ontologie durchführt. Dabei wird für die Ressource die publiziert wird ein neues *MetadataDataset* angelegt, der *fileIdentifier* befüllt, der *Scope* für den Metadatenatz gesetzt und alle anderen relevanten Attribute für einen Metadatenatz erzeugt. Am Ende wird in der Ontologie definiert, dass das übergebene Service die Ressource publiziert.

```

1  def publish_resource(service, resource):
2      if not hasattr(resource, 'has_metadatadataset')
3          or not resource.has_metadatadataset:
4
5          file_identifier = str(uuid.uuid4())
6          metadata = MetadataDataset('MetadataDataset_%s' % file_identifier)
7          metadata.has_fileidentifier = file_identifier
8          if type(resource) == gdimo.Service:
9              metadata.has_scope = scope_service
10         else:
11             metadata.has_scope = scope_dataset
12
13         metadata.has_language = default_metadata_language
14         metadata.has_characterset = characterset_utf8
15         metadata.has_datestamp = datetime.now().strftime('%Y-%m-%dT%H:%M:%S')
16
17         resource.has_publicationdate = datetime.now().strftime('%Y-%m-%d')
18
19         metadata.is_metadatadataset_of = resource
20         service.is_publishing.append(resource)

```

Listing 4.6: python Funktion zur Publikation einer Ressource

Durch die Definition von SWRL-Regeln können in der Ontologie logische Ableitungen getroffen werden, die sowohl für die publizierte Ressource als auch für das Service, das diese Ressource veröffentlicht, gelten. Listing 4.7 zeigt die Regeln, die bewirken, dass ein Datensatz *?d*, der über ein Service *?s* publiziert wird, automatisch die digitalen Transferoptionen und das Distributionsformat von diesem Service übernimmt. Das bedeutet, dass, wenn ein Service spezifische Zugangsmöglichkeiten (z.B. Download-Optionen oder digitale Übertragungsmethoden) oder bestimmte Distributionsformate definiert hat, diese Informationen auch auf den Datensatz übertragen werden. Dadurch werden diese Merkmale direkt in den Metadatenatz des Datensatzes integriert, ohne dass sie separat erfasst werden müssen. Durch diesen Ansatz kann auf die Definition der digitalen Transferoptionen und das Distributionsformat bei einem Datensatz komplett verzichtet werden, da sie abhängig von dem Service über den er bereitgestellt wird automatisch gesetzt werden.

```

1  Dataset(?d)
2    ^ Service(?s)
3    ^ is_published_by(?d, ?s)
4    ^ has_digitaltransferoption(?s, ?option)
5      -> has_digitaltransferoption(?d, ?option)
6
7  Dataset(?d)
8    ^ Service(?s)
9    ^ is_published_by(?d, ?s)
10   ^ has_distributionformat(?s, ?format)
11     -> has_distributionformat(?d, ?format)

```

Listing 4.7: SWRL Regeln für publizierte Datensätze

Umgekehrt können auch Informationen des Datensatzes an das Service übergeben werden. Listing 4.8 zeigt SWRL-Regeln, die es ermöglichen, dass bestimmte Eigenschaften eines Datensatzes *?d*, der über ein Service *?s* publiziert wird, auch auf das Service übertragen werden.

Die erste Regel legt fest, dass wenn ein Datensatz eine bestimmte thematische Kategorie (*has\_topiccategory*) hat, diese auch dem Service zugewiesen wird, das den Datensatz publiziert. Ebenso wird durch die zweite Regel die geografische Ausdehnung (*has\_geographicextent*) des Datensatzes automatisch auf das Service übertragen. Die dritte Regel stellt sicher, dass, wenn ein Datensatz ein bestimmtes GEMET-Konzept (*has\_gemetconcept*) nutzt, dieses Konzept auch dem Service zugeordnet wird.

Diese Regeln gewährleisten, dass Informationen, die ursprünglich im Datensatz definiert sind, auch für das Service verfügbar gemacht werden, welches den Datensatz publiziert. Dadurch entsteht eine konsistente und bidirektionale Verbindung zwischen den Metadaten des Datensatzes und des Services, welche die Vollständigkeit der Metadaten auf beiden Ebenen sicherstellt.

```

1      Service(?s)
2      ^ Dataset(?d)
3      ^ is_published_by(?d, ?s)
4      ^ has_topiccategory(?d, ?topic)
5          -> has_topiccategory(?s, ?topic)
6
7      Service(?s)
8      ^ Dataset(?d)
9      ^ is_published_by(?d, ?s)
10     ^ has_geographicextent(?d, ?extent)
11         -> has_geographicextent(?s, ?extent)
12
13     Service(?s)
14     ^ Dataset(?d)
15     ^ is_published_by(?d, ?s)
16     ^ has_gemetconcept(?d, ?concept)
17         -> has_gemetconcept(?s, ?concept)

```

Listing 4.8: SWRL Regeln für Services die Datensätze publizieren

Die angeführten Beispiele verdeutlichen, wie durch die semantische Verknüpfung und die Implementierung geeigneter Regeln die Konsistenz der Metadaten verbessert und der Pflegeaufwand signifikant reduziert werden kann. Die vorgestellten Regeln sind exemplarisch und sollen das Potenzial dieser Methodik veranschaulichen. Eine vollständige Implementierung aller denkbaren Regeln würde jedoch den Umfang dieser Arbeit übersteigen und bleibt als weiterführende Forschungs- oder Entwicklungsaufgabe offen.

Vor einem Export wird mit owlready2 der Pellet Reasoner ausgeführt um die logischen Schlüsse abzuleiten.

Für die Implementierung der Exporte wurden python-Module entwickelt, welche die Metadaten aus der Ontologie standardkonform in XML oder RDF kodieren. Für die

Erstellung von ISO19115- und INSPIRE-konformen XML-Dokumenten kam lxml zum Einsatz, während für die RDF-Kodierung des DCAT-AP-Standards rdflib verwendet wurde.

Die Struktur der ISO19115- und INSPIRE-XML-Dokumente wird in lxml aufgebaut, wobei die benötigten Attribute mithilfe von owlready2 aus der Ontologie abgefragt und in die entsprechenden XML-Elemente und -Attribute eingefügt werden. Um die Übersichtlichkeit zu gewährleisten, wurde das Rendering der XML-Fragmente in separate Funktionen ausgelagert, die sowohl für Datensätze als auch für Dienste wiederverwendet werden können. Der Exportprozess durchläuft alle Individuen der Klasse *MetadataDataset*, und für jede Instanz wird ein entsprechendes XML-Dokument im Dateisystem abgelegt.

Der Export eines DCAT-AP-Datenkatalogs erfordert zunächst die Erstellung einer *DcatApCatalog*-Ressource, an die die Datensätze und Dienste mithilfe der zuvor genannten Prädikate veröffentlicht werden. Da der DCAT-AP-Standard nur vollständige Kataloge beschreibt, ist dieser Schritt notwendig. Für den Exportprozess wird über alle Individuen der Klasse *DcatApCatalog* iteriert. Durch die Verfolgung des Prädikats *is\_publishing* können die referenzierten Ressourcen identifiziert und ihre Attribute dem RDF-Graphen hinzugefügt werden.

Für die Validierung der erstellten Metadaten-Encodings wurden verschiedene Online-Validatoren herangezogen. Für INSPIRE wurde der INSPIRE-Validator<sup>1</sup> der Europäischen Union verwendet. Dabei ist darauf zu achten, je nach Art der Ressource, ob Datensatz oder Dienst, den entsprechenden Validierungsprozess auszuwählen. Die in dieser Arbeit erstellten INSPIRE-Metadaten erfüllen sämtliche Mindestanforderungen und validieren auch gegen optionale Kategorien vollständig.

Für die Validierung von DCAT-AP-Katalogen wurde der EU-Validator<sup>2</sup> verwendet. Die in dieser Arbeit erstellten Metadaten für DCAT-AP-Kataloge wurden erfolgreich gegen alle definierten Profile validiert.

Für ISO 19115 wurde der Validator von Geoscience Australia<sup>3</sup> eingesetzt. In diesem Fall kam das Schema Validate-Profil zur Anwendung, da die anderen Optionen, ISO19115-3 Validate und Geoscience Australia Profile Validate, nicht für die in der Arbeit erstellten Metadaten relevant sind.

---

<sup>1</sup><https://inspire.ec.europa.eu/validator/home/index.html>

<sup>2</sup><https://www.itb.ec.europa.eu/shacl/dcat-ap/upload>

<sup>3</sup><https://www.ga.gov.au/data-pubs/datastandards/cataloguestandard/metadata-profile-validator>

## Zusammenfassung und Ausblick

Die Fragestellung *“Kann eine integrierte Lösung entwickelt werden, die eine einmalige Erfassung von Metadaten ermöglicht und gleichzeitig den Anforderungen der Metadatenstandards ISO 19115, INSPIRE und DCAT-AP entspricht?”*, kann mit *ja* beantwortet werden, wobei zu berücksichtigen ist, dass durch die strikte Vorgabe von Texten in Freitextfeldern von INSPIRE und die verpflichtende Angabe von Attributen zu einem Datenkatalog in DCAT-AP keine uneingeschränkte Kompatibilität gegeben ist. Als Lösung dieser Inkompatibilitäten wurde in der Arbeit gezeigt, dass durch die Verwendung des INSPIRE Mindeststandards gemeinsam mit der zusätzlichen Beschreibung eines zugehörigen Discovery Services mit einer Einmalerfassung der Informationen alle drei Standards erzeugt werden können. Da INSPIRE die strengsten Anforderungen an die Befüllung von Metadaten stellt, sollte bei der Festlegung einer Mindestbefüllung für die Beschreibung von Geoinformationsdaten und -dienste innerhalb einer Organisation der INSPIRE Mindeststandard als Basis gewählt werden.

Zentrale Elemente der Arbeit sind die semantische Modellierung der minimalen ISO-, DCAT-AP- und INSPIRE-konformen Metadatenmodelle mit einer Ontologie, sowie die Implementierung eines prototypischen Publikationsprozesses. Durch den Einsatz von OWL-Inferenzmechanismen konnte nachgewiesen werden, dass Metadateninformationen über verschiedene Standards hinweg verwaltet und zwischen Ressourcen weitergegeben werden können. Dadurch wird nicht nur die Erfassung der Metadaten vereinheitlicht, sondern auch die Datenqualität verbessert und der manuelle Eingabeaufwand reduziert.

Die Implementierung des Publikationsprozesses erfolgte in python unter Verwendung der Bibliotheken owlready2 für die Ontologieverarbeitung, lxml zur Erstellung der XML-Formate für ISO 19115 und INSPIRE sowie rdflib für die Generierung DCAT-AP konformer RDF-Daten. Zur Validierung der erzeugten Metadatensätze wurden etablierte Online-Validatoren wie der INSPIRE Validator und der DCAT-AP SHACL Validator eingesetzt. Die Ergebnisse zeigen, dass die entwickelten Metadatensätze sowohl die verpflichtenden als auch die empfohlenen Anforderungen der jeweiligen Standards erfüllen.

Die Analyse der verwendeten Metadatenstandards zeigt, dass ISO 19115 und INSPIRE strukturelle Ähnlichkeiten aufweisen, da INSPIRE auf der Version 2003 von ISO 19115 basiert. Allerdings erschweren Unterschiede in der Namensgebung von Attributen, abweichende Mindestanforderungen hinsichtlich der Multiplizität sowie INSPIRE-spezifische Pflichtangaben mit strikten Texten in Freitextfeldern eine Harmonisierung.

Im Gegensatz dazu verfolgt DCAT-AP einen semantisch annotierten und strukturell flacheren Ansatz, der sowohl verständlicher als auch einfacher zu implementieren ist. Allerdings schränkt die Vorgabe, auch Metadaten des publizierenden Katalogs verpflichtend zu erfassen, die Interoperabilität zwischen den Standards ein. Um eine gleichwertige Abbildung von ISO 19115- oder INSPIRE-Metadaten in DCAT-AP zu ermöglichen, ist zusätzlich eine Beschreibung des zugehörigen Discovery Services erforderlich.

Bei der Erstellung von Metadaten können Inhalte über organisationsspezifische Voreinstellungen automatisch gesetzt werden. Dazu gehören beispielsweise vordefinierte Lizenzen für alle Datensätze oder die Implementierung einer zentralen Kontaktverwaltung, wodurch der manuelle Eingabeaufwand minimiert wird.

Zusammenfassend zeigt diese Arbeit, dass eine Ontologie-basierte semantische Modellierung einen wesentlichen Beitrag zur Standardisierung, Automatisierung und Effizienzsteigerung im Metadatenmanagement von Geodateninfrastrukturen leisten kann. Die entwickelten Konzepte bilden eine fundierte Basis für zukünftige Erweiterungen und können in verschiedenen Szenarien zur Verbesserung der Interoperabilität und Qualität von Geodaten-Metadaten eingesetzt werden.

### 5.1 Erreichte Ergebnisse

- Entwicklung eines Mindestmodells zur einheitlichen Erfassung von Metadaten, das die Anforderungen der Standards ISO 19115, INSPIRE und DCAT-AP erfüllt.
- Erstellung einer Taxonomie, die die Konzepte dieses Mindestmodells semantisch beschreibt.
- Implementierung einer Ontologie, die als Wissensbasis für die Verwaltung und Publikation von Geoinformationsdaten- und Diensten dient.
- Erfolgreicher Einsatz von Inferenzmechanismen, die eine automatisierte Weitergabe und Verknüpfung von beschreibenden Informationen zwischen Daten- und Dienstemetadaten ermöglichen.

### 5.2 Herausforderungen und nicht vollständig gelöste Aspekte

- Die tief verschachtelten XML-Strukturen von ISO 19115 und INSPIRE stellen eine Herausforderung für die Verarbeitung und Implementierung dar. Trotz der

erfolgreichen Generierung valider Metadaten erschweren die komplexen Hierarchien eine direkte Umwandlung in andere Formate, insbesondere in das vergleichsweise flache DCAT-AP-Modell.

- Die Transformation zwischen den Standards ist nicht verlustfrei umsetzbar, da einige Informationen in bestimmten Standards zwingend erforderlich sind, während sie in anderen optional oder gar nicht vorgesehen sind. Zusätzlich sind die Multiplizitäten in den Relationen der Objekte in den Standards unterschiedlich, was eine direkte Harmonisierung ohne das Mindestmodell für die Erfassung zu verwenden verunmöglichen kann bzw. zu Informationsverlusten führt.
- In der Arbeit wurde nur das Mindestmodell der Standards berücksichtigt. Ob das Konzept auch auf die optionalen Attribute und Objekte der Standards ausgeweitet werden kann, wurde nicht analysiert.

### 5.3 Zukünftige Potenziale und Erweiterungsmöglichkeiten

- Die semantische Beschreibung und Verknüpfung der in den Standards verwendeten Codelisten ist eine wesentliche Voraussetzung für die vollständige Abdeckung eines zentralen Metadatenmodells. In dieser Arbeit wurden die Codelisteneinträge so gewählt, dass keine Konflikte entstehen. Für den praktischen Einsatz wäre jedoch ein automatisiertes Matching inhaltlich gleicher Einträge erforderlich. Da Begriffe in den Codelisten häufig hierarchische Beziehungen aufweisen, beispielsweise als Über- oder Unterbegriffe, könnte ein eindeutiges Matching nicht in allen Fällen möglich sein. Eine weiterführende Analyse wäre notwendig, um potenzielle Überschneidungen zu identifizieren und geeignete Mechanismen zur Harmonisierung zu entwickeln.
- Die Entwicklung einer umfassenden Domain-Ontologie würde eine ganzheitliche Modellierung und Verwaltung von Geodateninfrastrukturen ermöglichen. Da eine solche Ontologie eine deutlich höhere Abstraktionsebene erfordert und eine Vielzahl domänenspezifischer Konzepte sowie deren Relationen abbilden müsste, übersteigt ihr vollständiger Entwurf den Rahmen dieser Arbeit. Die in dieser Arbeit entwickelte Applikationsontologie legt jedoch eine fundierte Grundlage für eine zukünftige Erweiterung in diese Richtung.
- Die semantische Verknüpfung der erarbeiteten Konzepte mit bestehenden Vokabularen könnte die Auffindbarkeit und Nachnutzbarkeit der Metadaten erheblich verbessern. Durch die Anbindung an etablierte Vokabulare lassen sich die Meta-informationen in offene Datenplattformen integrieren. Darüber hinaus würde der Einsatz von Inferenzmechanismen (Reasoning) weiteres Wissen generieren, indem implizite Beziehungen zwischen Konzepten automatisch erschlossen werden. Da eine umfassende semantische Verknüpfung jedoch nicht im Fokus dieser Arbeit lag, bietet dieser Aspekt Potenzial für weiterführende Forschungsarbeiten.



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar  
The approved original version of this thesis is available in print at TU Wien Bibliothek.

## 6.1 Ontologie Begriffsdefinitionen

$\exists x$  (Resource( $x$ )) = def. any entity that is managed within a geodata infrastructure.

$\exists x$  (Dataset( $x$ )) = def. a collection of related data.

$\exists x$  (Service( $x$ )) = def. a functionality provided to users or systems.

$\exists x$  (MetadataDataset( $x$ )) = def. provides descriptive information about a resource.

$\exists x$  (DiscoveryService( $x$ )) = def. a service that allows users and systems to search and find resources.

$\exists x$  (DownloadService( $x$ )) = def. a service that enables users and systems to retrieve datasets.

$\exists x$  (InspireDataset( $x$ )) = def. a dataset resource that contains all metainformation compliant to the INSPIRE technical guidelines.

$\exists x$  (DcatApDataset( $x$ )) = def. a dataset resource that contains all metainformation compliant to the DCAT-AP specification.

$\exists x$  (ISO19115Dataset( $x$ )) = def. a dataset resource that contains all metainformation compliant to the ISO19115 specification.

- $\exists x$  (DcatApCatalog( $x$ )) = def. a service that represents all metainformation to describe a Catalog class compliant to the DCAT-AP specification.
- $\exists x$  (InspireDiscoveryService( $x$ )) = def. a service that represents all metainformation to describe an INSPIRE Discovery Service compliant to the INSPIRE technical guidelines.
- $\exists x$  (ISO19115DiscoveryService( $x$ )) = def. a service that represents all metainformation to describe a Discovery Service compliant to the ISO19115 specification.
- $\exists x$  (DcatApDataService( $x$ )) = def. a service that represents all metainformation to describe a DataService class compliant to the DCAT-AP specification.
- $\exists x$  (InspireDownloadService( $x$ )) = def. a service that represents all metainformation to describe an INSPIRE Download Service compliant to the INSPIRE technical guidelines.
- $\exists x$  (ISO19115DownloadService( $x$ )) = def. a service that represents all metainformation to describe a Download Service compliant to the ISO19115 specification.
- $\exists x$  (Codelist( $x$ )) = def. an enumerated list of values with an anchor URL pointing to a fixed definition.
- $\exists x$  (CharacterSet( $x$ )) = def. the encoding of the resource.
- $\exists x$  (Condition( $x$ )) = def. a requirement or restriction associated with the use or access of a resource.
- $\exists x$  (AccessCondition( $x$ )) = def. a condition that specifies the requirements for accessing a resource.
- $\exists x$  (UseCondition( $x$ )) = def. a condition that specifies the requirements for using a resource.
- $\exists x$  (ContactRole( $x$ )) = def. the role of a contact person or organization related to the resource.
- $\exists x$  (CoordinateSystem( $x$ )) = def. a reference system used to represent the locations of geographic features.
- $\exists x$  (CouplingType( $x$ )) = def. the nature of the link between a service and its associated data.
- $\exists x$  (DateType( $x$ )) = def. the classification of a date, indicating its purpose (e.g., creation, publication).

- $\exists x$  (DCP( $x$ )) = def. a protocol or platform used for communication in web services (Distributed Computing Platform).
- $\exists x$  (Language( $x$ )) = def. the language in which the resource or service is provided.
- $\exists x$  (DistributionFormat( $x$ )) = def. the format of the data in the resource.
- $\exists x$  (Restriction( $x$ )) = def. a limitation or constraint on the use or access of a resource.
- $\exists x$  (Scope( $x$ )) = def. the extent or coverage of the resource or service in terms of purpose or applicability.
- $\exists x$  (ServiceType( $x$ )) = def. the classification of a service based on its function or purpose.
- $\exists x$  (SpatialRepresentationType( $x$ )) = def. the method used to represent spatial data in a dataset.
- $\exists x$  (TransferProtocol( $x$ )) = def. the protocol used for transferring data between systems or services.
- $\exists x$  (ControlledValue( $x$ )) = def. a predefined value of a controlled vocabulary.
- $\exists x$  (Keyword( $x$ )) = def. a term used to describe the content of a resource.
- $\exists x$  (KeywordGemetConcept( $x$ )) = def. a keyword that corresponds to a concept defined in the GEMET thesaurus.
- $\exists x$  (KeywordGemetInspireTheme( $x$ )) = def. a keyword that corresponds to a theme defined by INSPIRE within the GEMET Thesaurus.
- $\exists x$  (KeywordInspirePriorityDataset( $x$ )) = def. a keyword used to identify datasets that are prioritized under INSPIRE due to their significance for environmental reporting.
- $\exists x$  (KeywordInspireSpatialDataService( $x$ )) = def. a keyword used to categorize services that provide access to spatial data within the INSPIRE framework.
- $\exists x$  (KeywordInspireSpatialScope( $x$ )) = def. a keyword that specifies the geographical or spatial scope of a dataset within the INSPIRE framework.
- $\exists x$  (QualityReportExplanation( $x$ )) = def. a textual description providing information about the quality evaluation process of a dataset or service. INSPIRE predefines texts to indicate the conformance of datasets and services with this explanation.

- $\exists x$  (QualityReportPass( $x$ )) = def. a boolean indicator that defines if a dataset or service meets the specified quality requirements.
- $\exists x$  (TopicCategory( $x$ )) = def. an ISO classification used to broadly group datasets by subject.
- $\exists x$  (ControlledVocabulary( $x$ )) = def. an organized collection of terms and definitions ensuring semantic interoperability.
- $\exists x$  (QualityReportSpecification( $x$ )) = def. a formal statement outlining the quality criteria and standards against which a dataset or service is validated.
- $\exists x$  (Thesaurus( $x$ )) = def. a structured and organized set of terms and concepts.
- $\exists x$  (ThesaurusGemetConcept( $x$ )) = def. a thesaurus that includes concepts from the GEMET vocabulary.
- $\exists x$  (ThesaurusGemetInspireTheme( $x$ )) = def. a thesaurus specifically aligned with INSPIRE themes based on the GEMET Thesaurus.
- $\exists x$  (ThesaurusInspirePriorityDataset( $x$ )) = def. a thesaurus that includes terms related to INSPIRE priority datasets.
- $\exists x$  (ThesaurusInspireSpatialDataService( $x$ )) = def. a thesaurus used to standardize terms for categorizing spatial data services within the INSPIRE framework.
- $\exists x$  (ThesaurusInspireSpatialScope( $x$ )) = def. a thesaurus that provides standardized terms for defining the spatial scope of datasets within the INSPIRE framework.
- $\exists x$  (Contact( $x$ )) = def. an entity responsible for information or services related to a resource.
- $\exists x$  (Constraint( $x$ )) = def. a rule defining access or use conditions of a resource.
- $\exists x$  (AccessConstraint( $x$ )) = def. a constraint that regulates access to a resource.
- $\exists x$  (UseConstraint( $x$ )) = def. a constraint that specifies terms of use for a resource.
- $\exists x$  (DigitalTransferOption( $x$ )) = def. a method for the electronic transfer of data.
- $\exists x$  (GeographicExtent( $x$ )) = def. a spatial definition of the area covered by a resource.
- $\exists x$  (QualityReport( $x$ )) = def. a report detailing the quality of a resource.
- $\exists x$  (ServiceOperation( $x$ )) = def. an operation provided by a service.

## 6.2 Ontologie Klassenhierarchie

```

1  Thing
2  Resource
3  MetadataDataset
4  Dataset
5  InspireDataset
6  DcatApDataset
7  ISO19115Dataset
8  Service
9  DiscoveryService
10 DcatApCatalog
11 InspireDiscoveryService
12 ISO19115DiscoveryService
13 DownloadService
14 DcatApDataService
15 InspireDownloadService
16 ISO19115DownloadService
17 CodeList
18 Characterset
19 Condition
20 AccessCondition
21 UseCondition
22 ContactRole
23 CoordinateSystem
24 CouplingType
25 DateType
26 DCP
27 Language
28 DistributionFormat
29 Restriction
30 Scope
31 ServiceType
32 SpatialRepresentationType
33 TransferProtocol
34 ControlledValue
35 Keyword
36 KeywordGemetConcept
37 KeywordGemetInspireTheme
38 KeywordInspirePriorityDataset
39 KeywordInspireSpatialDataScope
40 QualityReportExplanation
41 QualityReportPass
42 TopicCategory
43 ControlledVocabulary
44 QualityReportSpecification
45 Thesaurus
46 ThesaurusGemetConcept
47 ThesaurusGemetInspireTheme
48 ThesaurusInspirePriorityDataset
49 ThesaurusInspireSpatialDataService
50 ThesaurusInspireSpatialScope
51 Constraint

```

## 6. ANHANG

---

52	AccessConstraint
53	UseConstraint
54	Contact
55	DigitalTransferOption
56	GeographicExtent
57	QualityReport
58	ServiceOperation

Listing 6.1: Klassenhierarchie der Ontologie

### 6.3 Ontologie Objekt- und Dateneigenschaften

- $$\forall x (\text{Thing}(x) \Rightarrow \exists^=1 y (\text{has\_value}(x, y) \wedge \text{string}(y)))$$
- $$\forall x (\text{Thing}(x) \Rightarrow \exists^=1 y (\text{has\_anchor}(x, y) \wedge \text{string}(y)))$$
- $$\forall x (\text{Thing}(x) \Rightarrow \exists^=1 y (\text{has\_version}(x, y) \wedge \text{string}(y)))$$
- $$\forall x (\text{Thing}(x) \Rightarrow \exists^=1 y (\text{has\_title}(x, y) \wedge \text{string}(y)))$$
- $$\forall x (\text{Thing}(x) \Rightarrow \exists^=1 y (\text{has\_publicationdate}(x, y) \wedge \text{string}(y)))$$
- $$\forall x (\text{Thing}(x) \Rightarrow \exists^=1 y (\text{has\_publicationdatetype}(x, y) \wedge \text{DateType}(y)))$$
- $$\forall x (\text{Thing}(x) \Rightarrow \exists^=1 y (\text{has\_vocabulary}(x, y) \wedge \text{ControlledVocabulary}(y)))$$
- $$\forall x (\text{Thing}(x) \Rightarrow \exists^=1 y (\text{has\_name}(x, y) \wedge \text{string}(y)))$$
- $$\forall x (\text{Thing}(x) \Rightarrow \exists^=1 y (\text{has\_email}(x, y) \wedge \text{string}(y)))$$
- $$\forall x (\text{Thing}(x) \Rightarrow \exists^=1 y (\text{has\_restriction}(x, y) \wedge \text{Restriction}(y)))$$
- $$\forall x (\text{Constraint}(x) \Rightarrow \exists^=1 y (\text{has\_condition}(x, y) \wedge \text{Condition}(y)))$$
- $$\forall x (\text{Thing}(x) \Rightarrow \exists^=1 y (\text{has\_extent\_west}(x, y) \wedge \text{string}(y)))$$
- $$\forall x (\text{Thing}(x) \Rightarrow \exists^=1 y (\text{has\_extent\_south}(x, y) \wedge \text{string}(y)))$$
- $$\forall x (\text{Thing}(x) \Rightarrow \exists^=1 y (\text{has\_extent\_east}(x, y) \wedge \text{string}(y)))$$
- $$\forall x (\text{Thing}(x) \Rightarrow \exists^=1 y (\text{has\_extent\_north}(x, y) \wedge \text{string}(y)))$$
- $$\forall x (\text{Thing}(x) \Rightarrow \exists^=1 y (\text{has\_scope}(x, y) \wedge \text{Scope}(y)))$$
- $$\forall x (\text{Thing}(x) \Rightarrow \exists^=1 y (\text{has\_specification}(x, y) \wedge \text{QualityReportSpecification}(y)))$$
- $$\forall x (\text{Thing}(x) \Rightarrow \exists^=1 y (\text{has\_explanation}(x, y) \wedge \text{QualityReportExplanation}(y)))$$
- $$\forall x (\text{Thing}(x) \Rightarrow \exists^=1 y (\text{has\_result}(x, y) \wedge \text{QualityReportPass}(y)))$$
- $$\forall x (\text{Thing}(x) \Rightarrow \exists^=1 y (\text{has\_transferprotocol}(x, y) \wedge \text{TransferProtocol}(y)))$$
- $$\forall x (\text{Thing}(x) \Rightarrow \exists^=1 y (\text{has\_servicetype}(x, y) \wedge \text{ServiceType}(y)))$$
- $$\forall x (\text{Thing}(x) \Rightarrow \exists^=1 y (\text{has\_linkage}(x, y) \wedge \text{string}(y)))$$
- $$\forall x (\text{Thing}(x) \Rightarrow \exists^=1 y (\text{has\_dcp}(x, y) \wedge \text{DCP}(y)))$$

- $\forall x (\text{Thing}(x) \Rightarrow \exists^{=1} y (\text{has\_abstract}(x, y) \wedge \text{string}(y)))$
- $\forall x (\text{Thing}(x) \Rightarrow \exists^{=1} y (\text{has\_creationdate}(x, y) \wedge \text{string}(y)))$
- $\forall x (\text{Thing}(x) \Rightarrow \exists^{=1} y (\text{has\_creationdatetype}(x, y) \wedge \text{DateType}(y)))$
- $\forall x (\text{Thing}(x) \Rightarrow \exists^{=1} y (\text{has\_revisiondate}(x, y) \wedge \text{string}(y)))$
- $\forall x (\text{Thing}(x) \Rightarrow \exists^{=1} y (\text{has\_revisiondatetype}(x, y) \wedge \text{DateType}(y)))$
- $\forall x (\text{Thing}(x) \Rightarrow \exists^{=1} y (\text{has\_pointofcontact}(x, y) \wedge \text{Contact}(y)))$
- $\forall x (\text{Thing}(x) \Rightarrow \exists^{=1} y (\text{has\_publisher}(x, y) \wedge \text{Contact}(y)))$
- $\forall x (\text{Thing}(x) \Rightarrow \exists^{=1} y (\text{has\_publisher\_contactrole}(x, y) \wedge \text{ContactRole}(y)))$
- $\forall x (\text{Thing}(x) \Rightarrow \exists y (\text{has\_gemetconcept}(x, y) \wedge \text{KeywordGemetConcept}(y)))$
- $\forall x (\text{Thing}(x) \Rightarrow \exists^{=1} y (\text{has\_useconstraint}(x, y) \wedge \text{UseConstraint}(y)))$
- $\forall x (\text{Thing}(x) \Rightarrow \exists^{=1} y (\text{has\_accessconstraint}(x, y) \wedge \text{AccessConstraint}(y)))$
- $\forall x (\text{Thing}(x) \Rightarrow \exists y (\text{has\_geographicextent}(x, y) \wedge \text{GeographicExtent}(y)))$
- $\forall x (\text{Thing}(x) \Rightarrow \exists^{=1} y (\text{has\_qualityreport}(x, y) \wedge \text{QualityReport}(y)))$
- $\forall x (\text{Thing}(x) \Rightarrow \exists y (\text{has\_digitaltransferoption}(x, y) \wedge \text{DigitalTransferOption}(y)))$
- $\forall x (\text{Thing}(x) \Rightarrow \exists^{=1} y (\text{has\_fileidentifier}(x, y) \wedge \text{string}(y)))$
- $\forall x (\text{Thing}(x) \Rightarrow \exists^{=1} y (\text{has\_characteraset}(x, y) \wedge \text{Characteraset}(y)))$
- $\forall x (\text{Thing}(x) \Rightarrow \exists^{=1} y (\text{has\_datestamp}(x, y) \wedge \text{string}(y)))$
- $\forall x (\text{Thing}(x) \Rightarrow \exists y (\text{has\_coordinatesystem}(x, y) \wedge \text{CoordinateSystem}(y)))$
- $\forall x (\text{Thing}(x) \Rightarrow \exists y (\text{has\_gemetinspiretheme}(x, y) \wedge \text{KeywordGemetInspireTheme}(y)))$
- $\forall x (\text{Thing}(x) \Rightarrow \exists y (\text{has\_dataset\_keyword\_inspireprioritydataset}(x, y) \wedge \text{KeywordInspirePriorityDataset}(y)))$
- $\forall x (\text{Thing}(x) \Rightarrow \exists y (\text{has\_inspirespatialscope}(x, y) \wedge \text{KeywordInspireSpatialScope}(y)))$
- $\forall x (\text{Thing}(x) \Rightarrow \exists^{=1} y (\text{has\_spatialrepresentationtype}(x, y) \wedge \text{SpatialRepresentationType}(y)))$
- $\forall x (\text{Thing}(x) \Rightarrow \exists^{=1} y (\text{has\_lineage}(x, y) \wedge \text{string}(y)))$
- $\forall x (\text{Thing}(x) \Rightarrow \exists^{=1} y (\text{has\_spatialresolution}(x, y) \wedge \text{string}(y)))$

$$\forall x (\text{Thing}(x) \Rightarrow \exists^1 y (\text{has\_identifier}(x, y) \wedge \text{string}(y)))$$

$$\forall x (\text{Thing}(x) \Rightarrow \exists y (\text{has\_topiccategory}(x, y) \wedge \text{TopicCategory}(y)))$$

$$\forall x (\text{Thing}(x) \Rightarrow \exists y (\text{has\_distributionformat}(x, y) \wedge \text{DistributionFormat}(y)))$$

$$\forall x (\text{Thing}(x) \Rightarrow \exists^1 y (\text{has\_language}(x, y) \wedge \text{Language}(y)))$$

$$\forall x (\text{Thing}(x) \Rightarrow \exists^1 y (\text{has\_couplingtype}(x, y) \wedge \text{CouplingType}(y)))$$

$$\forall x (\text{Thing}(x) \Rightarrow \exists y (\text{has\_serviceoperation}(x, y) \wedge \text{ServiceOperation}(y)))$$

$$\forall x (\text{Thing}(x) \Rightarrow \exists y (\text{has\_service\_keyword\_inspirespatialdataservice}(x, y) \wedge \text{KeywordInspireSpatialDataService}(y)))$$

$$\forall x (\text{Thing}(x) \Rightarrow \exists^1 y (\text{has\_metadatadataset}(x, y) \wedge \text{MetadataDataset}(y)))$$

$$\forall x (\text{Thing}(x) \Rightarrow \exists^1 y (\text{is\_metadatadataset\_of}(x, y) \wedge \text{Resource}(y)))$$

$$\forall x (\text{Thing}(x) \Rightarrow \exists y (\text{is\_publishing}(x, y) \wedge \text{Resource}(y)))$$

$$\forall x (\text{Thing}(x) \Rightarrow \exists y (\text{is\_published\_by}(x, y) \wedge \text{Service}(y)))$$

$$\forall x, y (\text{has\_metadatadataset}(x, y) \Leftrightarrow \text{is\_metadatadataset\_of}(y, x))$$

$$\forall x, y (\text{is\_publishing}(x, y) \Leftrightarrow \text{is\_published\_by}(y, x))$$

## 6.4 Ontologie Axiome

### 6.4.1 DCAT-AP Dataset Axiom

$$\begin{aligned} \text{DCATAPDataset}(x) &\Leftrightarrow \text{Dataset}(x) \wedge \exists y (\text{has\_title}(x, y) \wedge \text{string}(y)) \\ &\wedge \exists z (\text{has\_abstract}(x, z) \wedge \text{string}(z)) \\ &\wedge \exists w (\text{has\_publisher}(x, w) \wedge \text{Contact}(w)) \\ &\wedge \exists v (\text{has\_publisher\_contactrole}(x, v) \wedge \text{ContactRole}(v)) \end{aligned}$$

### 6.4.2 ISO19115 Dataset Axiom

$$\begin{aligned} \text{ISO19115Dataset}(x) &\Leftrightarrow \text{Dataset}(x) \wedge \exists y (\text{has\_title}(x, y) \wedge \text{string}(y)) \\ &\wedge \exists z (\text{has\_abstract}(x, z) \wedge \text{string}(z)) \\ &\wedge \exists w (\text{has\_creationdate}(x, w) \wedge \text{string}(w)) \\ &\wedge \left( \exists a (\text{has\_creationdate}(x, a) \wedge \text{string}(a)) \right. \\ &\quad \left. \wedge \exists b (\text{has\_creationdatatype}(x, b) \wedge \text{DateType}(b)) \right) \\ &\vee \exists c (\text{has\_revisiondate}(x, c) \wedge \text{string}(c)) \\ &\quad \wedge \exists d (\text{has\_revisiondatatype}(x, d) \wedge \text{DateType}(d)) \\ &\vee \exists e (\text{has\_publicationdate}(x, e) \wedge \text{string}(e)) \\ &\quad \wedge \exists f (\text{has\_publicationdatatype}(x, f) \wedge \text{DateType}(f)) \\ &\wedge \exists g (\text{has\_geographicextent}(x, g) \wedge \text{GeographicExtent}(g)) \\ &\wedge \exists h (\text{has\_pointofcontact}(x, h) \wedge \text{Contact}(h)) \\ &\wedge \exists i (\text{has\_pointofcontact\_contactrole}(x, i) \wedge \text{ContactRole}(i)) \\ &\wedge \exists j (\text{has\_topiccategory}(x, j) \wedge \text{TopicCategory}(j)) \\ &\wedge \exists k (\text{has\_identifier}(x, k) \wedge \text{string}(k)) \\ &\wedge \exists l (\text{has\_language}(x, l) \wedge \text{Language}(l)) \\ &\wedge \exists m (\text{has\_spatialresolution}(x, m) \wedge \text{string}(m)) \\ &\wedge \exists n (\text{has\_scope}(x, n) \wedge \text{Scope}(n)) \\ &\wedge \exists o (\text{has\_digitaltransferoption}(x, o) \wedge \text{DigitalTransferOption}(o)) \\ &\wedge \exists p (\text{has\_gemetconcept}(x, p) \wedge \text{KeywordGemetConcept}(p)) \\ &\wedge \exists q (\text{has\_accessconstraint}(x, q) \wedge \text{AccessConstraint}(q)) \\ &\wedge \exists r (\text{has\_useconstraint}(x, r) \wedge \text{UseConstraint}(r)) \\ &\wedge \exists s (\text{has\_lineage}(x, s) \wedge \text{string}(s)) \end{aligned}$$

### 6.4.3 INSPIRE Dataset Axiom

$$\begin{aligned}
& \text{InspireDataset}(x) \Leftrightarrow \text{Dataset}(x) \wedge \exists y (\text{has\_title}(x, y) \wedge \text{string}(y)) \\
& \wedge \exists z (\text{has\_abstract}(x, z) \wedge \text{string}(z)) \\
& \wedge \left( \exists a (\text{has\_creationdate}(x, a) \wedge \text{string}(a)) \right. \\
& \quad \left. \wedge \exists b (\text{has\_creationdatatype}(x, b) \wedge \text{DateType}(b)) \right) \\
& \vee \exists c (\text{has\_revisiondate}(x, c) \wedge \text{string}(c)) \\
& \quad \wedge \exists d (\text{has\_revisiondatatype}(x, d) \wedge \text{DateType}(d)) \\
& \vee \exists e (\text{has\_publicationdate}(x, e) \wedge \text{string}(e)) \\
& \quad \wedge \exists f (\text{has\_publicationdatatype}(x, f) \wedge \text{DateType}(f)) \\
& \wedge \exists g (\text{has\_identifier}(x, g) \wedge \text{string}(g)) \\
& \wedge \exists h (\text{has\_pointofcontact}(x, h) \wedge \text{Contact}(h)) \\
& \wedge \exists i (\text{has\_pointofcontact\_contactrole}(x, i) \wedge \text{ContactRole}(i)) \\
& \wedge \exists j (\text{has\_gemetconcept}(x, j) \wedge \text{KeywordGemetConcept}(j)) \\
& \wedge \exists k (\text{has\_dataset\_keyword\_inspireprioritydataset}(x, k) \\
& \quad \wedge \text{KeywordInspirePriorityDataset}(k)) \\
& \wedge \exists l (\text{has\_inspirespatialscope}(x, l) \wedge \text{KeywordInspireSpatialScope}(l)) \\
& \wedge \exists m (\text{has\_gemetinspiretheme}(x, m) \wedge \text{KeywordGemetInspireTheme}(m)) \\
& \wedge \exists n (\text{has\_accessconstraint}(x, n) \wedge \text{AccessConstraint}(n)) \\
& \wedge \exists o (\text{has\_useconstraint}(x, o) \wedge \text{UseConstraint}(o)) \\
& \wedge \exists p (\text{has\_spatialrepresentationtype}(x, p) \wedge \text{SpatialRepresentationType}(p)) \\
& \wedge \exists q (\text{has\_spatialresolution}(x, q) \wedge \text{string}(q)) \\
& \wedge \exists r (\text{has\_language}(x, r) \wedge \text{Language}(r)) \\
& \wedge \exists s (\text{has\_topiccategory}(x, s) \wedge \text{TopicCategory}(s)) \\
& \wedge \exists t (\text{has\_geographicextent}(x, t) \wedge \text{GeographicExtent}(t)) \\
& \wedge \exists u (\text{has\_distributionformat}(x, u) \wedge \text{DistributionFormat}(u)) \\
& \wedge \exists v (\text{has\_digitaltransferoption}(x, v) \wedge \text{DigitalTransferOption}(v)) \\
& \wedge \exists w (\text{has\_qualityreport}(x, w) \wedge \text{QualityReport}(w)) \\
& \wedge \exists x (\text{has\_scope}(x, x) \wedge \text{Scope}(x)) \\
& \wedge \exists y (\text{has\_lineage}(x, y) \wedge \text{string}(y))
\end{aligned}$$

#### 6.4.4 DCAT-AP Service Axiom

$$\begin{aligned} \text{DcatApService}(x) &\Leftrightarrow \text{Service}(x) \wedge \exists y (\text{has\_title}(x, y) \wedge \text{string}(y)) \\ &\wedge \exists z (\text{has\_publisher\_contactrole}(x, z) \wedge \text{ContactRole}(z) \wedge \text{cardinality}(z, 1)) \\ &\wedge \exists w (\text{has\_digitaltransferoption}(x, w) \wedge \text{DigitalTransferOption}(w)) \end{aligned}$$

#### 6.4.5 ISO19115 Service Axiom

$$\begin{aligned} \text{ISO19115Service}(x) &\Leftrightarrow \text{Service}(x) \wedge \exists y (\text{has\_title}(x, y) \wedge \text{string}(y)) \\ &\wedge \left( \exists a (\text{has\_creationdate}(x, a) \wedge \text{string}(a)) \right. \\ &\quad \left. \wedge \exists b (\text{has\_creationdatatype}(x, b) \wedge \text{DateType}(b)) \right) \\ &\vee \exists c (\text{has\_revisiondate}(x, c) \wedge \text{string}(c)) \\ &\quad \wedge \exists d (\text{has\_revisiondatatype}(x, d) \wedge \text{DateType}(d)) \\ &\vee \exists e (\text{has\_publicationdate}(x, e) \wedge \text{string}(e)) \\ &\quad \wedge \exists f (\text{has\_publicationdatatype}(x, f) \wedge \text{DateType}(f)) \\ &\wedge \exists g (\text{has\_abstract}(x, g) \wedge \text{string}(g)) \\ &\wedge \left( \exists h (\text{has\_pointofcontact}(x, h) \wedge \text{Contact}(h)) \right. \\ &\quad \left. \wedge \exists i (\text{has\_pointofcontact\_contactrole}(x, i) \wedge \text{ContactRole}(i)) \right) \\ &\vee \exists j (\text{has\_publisher}(x, j) \wedge \text{Contact}(j)) \\ &\quad \wedge \exists k (\text{has\_publisher\_contactrole}(x, k) \wedge \text{ContactRole}(k)) \\ &\wedge \exists l (\text{has\_gemetconcept}(x, l) \wedge \text{KeywordGemetConcept}(l)) \\ &\wedge \exists m (\text{has\_useconstraint}(x, m) \wedge \text{UseConstraint}(m)) \\ &\wedge \exists n (\text{has\_accessconstraint}(x, n) \wedge \text{AccessConstraint}(n)) \\ &\wedge \exists o (\text{has\_servicetype}(x, o) \wedge \text{ServiceType}(o)) \\ &\wedge \exists p (\text{has\_couplingtype}(x, p) \wedge \text{CouplingType}(p)) \\ &\wedge \exists q (\text{has\_digitaltransferoption}(x, q) \wedge \text{DigitalTransferOption}(q)) \\ &\wedge \exists r (\text{has\_geographicextent}(x, r) \wedge \text{GeographicExtent}(r)) \\ &\wedge \exists s (\text{has\_topiccategory}(x, s) \wedge \text{TopicCategory}(s)) \\ &\wedge \exists t (\text{has\_scope}(x, t) \wedge \text{Scope}(t)) \end{aligned}$$

### 6.4.6 INSPIRE Service Axiom

$$\begin{aligned}
& \text{InspireService}(x) \Leftrightarrow \text{Service}(x) \wedge \exists y (\text{has\_title}(x, y) \wedge \text{string}(y)) \\
& \wedge \left( \exists a (\text{has\_creationdate}(x, a) \wedge \text{string}(a)) \right. \\
& \quad \left. \wedge \exists b (\text{has\_creationdatatype}(x, b) \wedge \text{DateType}(b)) \right) \\
& \vee \exists c (\text{has\_revisiondate}(x, c) \wedge \text{string}(c)) \\
& \quad \wedge \exists d (\text{has\_revisiondatatype}(x, d) \wedge \text{DateType}(d)) \\
& \vee \exists e (\text{has\_publicationdate}(x, e) \wedge \text{string}(e)) \\
& \quad \wedge \exists f (\text{has\_publicationdatatype}(x, f) \wedge \text{DateType}(f)) \\
& \wedge \exists g (\text{has\_abstract}(x, g) \wedge \text{string}(g)) \\
& \wedge \left( \exists h (\text{has\_pointofcontact}(x, h) \wedge \text{Contact}(h)) \right. \\
& \quad \left. \wedge \exists i (\text{has\_pointofcontact\_contactrole}(x, i) \wedge \text{ContactRole}(i)) \right) \\
& \vee \exists j (\text{has\_publisher}(x, j) \wedge \text{Contact}(j)) \\
& \quad \wedge \exists k (\text{has\_publisher\_contactrole}(x, k) \wedge \text{ContactRole}(k)) \\
& \wedge \exists l (\text{has\_gemetconcept}(x, l) \wedge \text{KeywordGemetConcept}(l)) \\
& \wedge \exists m (\text{has\_service\_keyword\_inspirespatialdataservice}(x, m)) \\
& \quad \wedge \text{KeywordInspireSpatialDataService}(m) \\
& \wedge \exists n (\text{has\_useconstraint}(x, n) \wedge \text{UseConstraint}(n)) \\
& \wedge \exists o (\text{has\_accessconstraint}(x, o) \wedge \text{AccessConstraint}(o)) \\
& \wedge \exists p (\text{has\_servicetype}(x, p) \wedge \text{ServiceType}(p)) \\
& \wedge \exists q (\text{has\_couplingtype}(x, q) \wedge \text{CouplingType}(q)) \\
& \wedge \exists r (\text{has\_serviceoperation}(x, r) \wedge \text{ServiceOperation}(r)) \\
& \wedge \exists s (\text{has\_digitaltransferoption}(x, s) \wedge \text{DigitalTransferOption}(s)) \\
& \wedge \exists t (\text{has\_qualityreport}(x, t) \wedge \text{QualityReport}(t)) \\
& \wedge \exists u (\text{has\_scope}(x, u) \wedge \text{Scope}(u))
\end{aligned}$$



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar  
The approved original version of this thesis is available in print at TU Wien Bibliothek.

# Abbildungsverzeichnis

3.1	Übersicht über das ISO 19115 Metadatenmodell für Datensätze . . . . .	38
3.2	Übersicht über das ISO 19115 Metadatenmodell für Dienste . . . . .	40
3.3	Übersicht über das INSPIRE Metdatenmodell für Datensätze . . . . .	42
3.4	Übersicht über das INSPIRE Metdatenmodell für Dienste . . . . .	43
3.5	Übersicht über das DCAT-AP Metdatenmodell für Datensätze . . . . .	45
3.6	Übersicht über das DCAT-AP Metdatenmodell für Dienste . . . . .	46
4.1	Architektur der Referenz-Geodateninfrastruktur . . . . .	49
4.2	Grafische Darstellung der Taxonomie der Ontologie . . . . .	58
4.3	Klassen-Ansicht von Protege mit Reasoning . . . . .	59
4.4	Individual-Ansicht von Protege mit Reasoning . . . . .	60



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar  
The approved original version of this thesis is available in print at TU Wien Bibliothek.

# Listings

2.1	Beispiel SPARQL-Abfrage . . . . .	18
2.2	Beispiel SHACL-Shape zur Validierung einer Ontologie . . . . .	19
2.3	Beispiel SWRL-Regel . . . . .	19
4.1	Beispiel Klassenhierarchie als Listing . . . . .	49
4.2	Klassenhierarchie der Referenzarchitektur . . . . .	51
4.3	Klassenhierarchie der Codelisten . . . . .	53
4.4	Klassenhierarchie der kontrollierte Vokabulare . . . . .	55
4.5	Klassenhierarchie der komplexen Klassen . . . . .	56
4.6	python Funktion zur Publikation einer Ressource . . . . .	62
4.7	SWRL Regeln für publizierte Datensätze . . . . .	62
4.8	SWRL Regeln für Services die Datensätze publizieren . . . . .	63
6.1	Klassenhierarchie der Ontologie . . . . .	73



# Literaturverzeichnis

- [ASS15] Robert Arp, Barry Smith, and Andrew D Spear. *Building ontologies with basic formal ontology*. Mit Press, 2015.
- [BCD<sup>+</sup>19] Jean Brodeur, Serena Coetzee, David Danko, Stéphane Garcia, and Jan Hjelmager. Geographic information metadata—an outlook from the international standardization perspective. *ISPRS International Journal of Geo-Information*, 8(6):280, 2019.
- [Bha15] Subhra Bikash Bhattacharyya. *Introduction to SNOMED CT*. Springer, 2015.
- [CC05] Matteo Cristani and Roberta Cuel. A survey on ontology creation methodologies. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 1(2):49–69, 2005.
- [Com07] European Commission. 2011/833/eu: Commission decision of 12 december 2011 on the reuse of commission documents. <https://eur-lex.europa.eu/eli/dec/2011/833/oj>, 03 2007.
- [Con04] Gene Ontology Consortium. The gene ontology (go) database and informatics resource. *Nucleic acids research*, 32(suppl\_1):D258–D261, 2004.
- [DCA14] Data Catalog Vocabulary (DCAT). Recommendation, World Wide Web Consortium (W3C), 2014.
- [DCA20] DCAT Application Profile for data portals in Europe Version 2.0.1. Standard, European Commission Joint Research Centre, June 2020.
- [FLGPJ97] Mariano Fernández-López, Asunción Gómez-Pérez, and Natalia Juristo. Methontology: from ontological art towards ontological engineering. 1997.
- [FPO<sup>+</sup>16] Cristiano Fugazza, Monica Pepe, Alessandro Oggioni, Paolo Tagliolato, Fabio Pavesi, and Paola Carrara. Describing geospatial assets in the web of data: A metadata management scenario. *ISPRS International Journal of Geo-Information*, 5(12):229, 2016.

- [FPO<sup>+</sup>18] Cristiano Fugazza, Monica Pepe, Alessandro Oggioni, Paolo Tagliolato, and Paola Carrara. Raising semantics-awareness in geospatial metadata management. *ISPRS International Journal of Geo-Information*, 7(9):370, 2018.
- [GEO10] Geodateninfrastrukturgesetz - bundesgesetz über eine umweltrelevante geodateninfrastruktur des bundes (geodateninfrastrukturgesetz – geodig). <https://www.ris.bka.gv.at/GeltendeFassung.wxe?Abfrage=Bundesnormen&Gesetzesnummer=20006708>, 2010.
- [Geo20] GeoDCAT-AP - Version 2.0.0. Standard, GeoDCAT-AP Working Group, December 2020.
- [GHM<sup>+</sup>14] Birte Glimm, Ian Horrocks, Boris Motik, Giorgos Stoilos, and Zhe Wang. Hermit: an owl 2 reasoner. *Journal of automated reasoning*, 53:245–269, 2014.
- [IMM<sup>+</sup>13] Rizwan Iqbal, Masrah Azrifah Azmi Murad, Aida Mustapha, Nurfadhli-na Mohd Sharef, et al. An analysis of ontology engineering methodologies: A literature review. *Research journal of applied sciences, engineering and technology*, 6(16):2993–3000, 2013.
- [INS14] INSPIRE Generic Conceptual Model. Standard, European Commission Joint Research Centre, Italy, 04 2014.
- [INS24] Technical Guidance for the implementation of INSPIRE dataset and service metadata based on ISO/TS 19139:2007. Standard, European Commission Joint Research Centre, Italy, January 2024.
- [ISO03a] ISO 19115 - Geographic information - Metadata. Standard, International Organization for Standardization, Geneva, CH, 2003.
- [ISO03b] ISO 19115:2003/Cor 1:2006 - Geographic information - Metadata. Standard, International Organization for Standardization, Geneva, CH, 2003.
- [ISO05] ISO 19116 - Geographic information - Services. Standard, International Organization for Standardization, Geneva, CH, 2005.
- [ISO07] ISO/TS 19139 - Geographic information - Metadata - XML schema implementation. Standard, International Organization for Standardization, Geneva, CH, 2007.
- [ISO08] ISO 19119:2005/Amd 1:2008 Geographic information - Services - Amendment 1: Extensions of the service metadata model. Standard, International Organization for Standardization, Geneva, CH, 2008.

- [ISO09] ISO 19115 - Geographic information - Metadata - Part 2: Extensions for imagery and gridded data. Standard, International Organization for Standardization, Geneva, CH, 2009.
- [ISO13] ISO/TS 19157 - Geographic information - Data quality. Standard, International Organization for Standardization, Geneva, CH, 2013.
- [ISO14] ISO 19115 - Geographic information - Metadata - Part 1: Fundamentals. Standard, International Organization for Standardization, Geneva, CH, 2014.
- [ISO16] ISO 19116 - Geographic information - Services. Standard, International Organization for Standardization, Geneva, CH, 2016.
- [ISO18a] ISO 19115 - Geographic information - Metadata - Part 1: Fundamentals - Amendment 1. Standard, International Organization for Standardization, Geneva, CH, 2018.
- [ISO18b] ISO 19157:2013/Amd 1:2018 - Geographic Information - Data quality - Amendment 1: Describing data quality using coverages. Standard, International Organization for Standardization, Geneva, CH, 2018.
- [ISO19a] ISO/TS 19139-1:2019 - Geographic Information - XML schema implementation - Part 1: Encoding rules. Standard, International Organization for Standardization, Geneva, CH, 2019.
- [ISO19b] ISO 19115-2:2019 Geographic information - Metadata - Part 2: Extensions for acquisition and processing. Standard, International Organization for Standardization, Geneva, CH, 2019.
- [ISO20] ISO 19115 - Geographic information - Metadata - Part 1: Fundamentals - Amendment 2. Standard, International Organization for Standardization, Geneva, CH, 2020.
- [ISO22] ISO 19115-2:2019/Amd 1:2022 Geographic information - Metadata - Part 2: Extensions for acquisition and processing - Amendment 1. Standard, International Organization for Standardization, Geneva, CH, 2022.
- [ISO23] ISO 19157-1:2023 - Geographic Information - Data quality - Part 1: General requirements. Standard, International Organization for Standardization, Geneva, CH, 2023.
- [IWG15] Bundesgesetz über die weiterverwendung von informationen öffentlicher stellen (informationsweiterwendungsgesetz – iwg). <https://www.ris.bka.gv.at/GeltendeFassung.wxe?Abfrage=Bundesnormen&Gesetzesnummer=20004375>, 2015.

- [KKS12] Yevgeny Kazakov, Markus Krötzsch, and František Simančík. Elk: a reasoner for owl el ontologies. *System Description*, 2012.
- [Noy01] N Noy. Ontology development 101: A guide to creating your first ontology, 2001.
- [NP01] Ian Niles and Adam Pease. Towards a standard upper ontology. In *Proceedings of the international conference on Formal Ontology in Information Systems- Volume 2001*, pages 2–9, 2001.
- [OGD12] Rahmenbedingungen für Open Government Data Plattformen / Open Government Data - 1.1.0. White paper, Projektgruppe Cooperation Open Government Data Österreich, Austria, 2012.
- [OGD17] OGD Metadaten – 2.4. White paper, Arbeitsgruppe Metadaten der Cooperation OGD Österreich, Austria, 2017.
- [PC07] European Parliament and Council. Directive 2007/2/ec of the european parliament and of the council establishing an infrastructure for spatial information in the european community (inspire). <http://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32007L0002&from=EN>, 03 2007.
- [PC08] European Parliament and Council. Commission regulation (ec) no 1205/2008 implementing directive 2007/2/ec of the european parliament and of the council as regards metadata. <http://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32008R1205&from=EN>, 12 2008.
- [PC13a] European Parliament and Council. Directive 2003/98/ec of the european parliament and of the council on the re-use of public sector information. <http://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32003L0098&from=EN>, 06 2013.
- [PC13b] European Parliament and Council. Directive 2013/37/eu of the european parliament and of the council amending directive 2003/98/ec on the re-use of public sector information. <http://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32013L0037&from=EN>, 06 2013.
- [PSI14] Erläuterung der XML-Spezifikation für die Metadaten Schnittstelle / metadaten-md 1.0.0. White paper, e-government Bund Länder Städte Gemeinden (BLSG) PG UPSIR / BLSG AG-II, Austria, 2014.
- [PSI15a] Klassifikation von Informationen für PSI-Umsetzung / psi-klassifikation 1.0.0. White paper, e-government Bund Länder Städte Gemeinden (BLSG) PG UPSIR, Austria, 2015.
- [PSI15b] Technisch/organisatorische Umsetzung der PSI Richtlinie / upsir-1.0.0. White paper, e-government Bund Länder Städte Gemeinden (BLSG) PG UPSIR, Austria, 2015.

- [SFGPFL11] Mari Carmen Suárez-Figueroa, Asunción Gómez-Pérez, and Mariano Fernández-López. The neon methodology for ontology engineering. In *Ontology engineering in a networked world*, pages 9–34. Springer, 2011.
- [SPB20] Adrian Stadnicki, Filip Filip Pietroní, and Patryk Burek. Towards a modern ontology development environment. *Procedia Computer Science*, 176:753–762, 2020.
- [SPG<sup>+</sup>07] Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, and Yarden Katz. Pellet: A practical owl-dl reasoner. *Journal of Web Semantics*, 5(2):51–53, 2007.