



Hochschule Neubrandenburg  
University of Applied Sciences

**Hochschule Neubrandenburg**  
**Studiengang Geoinformatik**

Entwicklung einer Ontologie für die wissensbasierte Erschließung des  
ISDC-Repository und die Visualisierung kontextrelevanter  
semantischer Zusammenhänge

**Masterarbeit**

vorgelegt von: Sabine Pfeiffer

Zum Erlangen des akademischen Grades  
**„Master of Engineering“ (M.Eng.)**

Erstprüfer: Prof. Dr.-Ing. Ernst Heil

Zweitprüfer: Dipl.-Phys. Bernd Ritschel

Bearbeitungszeitraum: 03.05.2010 - 03.11.2010

## **Eidesstattliche Erklärung**

Hiermit versichere ich, die vorliegende Masterarbeit ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus den Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Neubrandenburg, den

## Danksagung

An dieser Stelle möchte ich allen danken, die mich ,fachlich und persönlich, während meiner Studienzeit und im Speziellen bei der Erstellung meiner Masterarbeit unterstützt haben.

Für die Betreuung meiner Masterarbeit danke ich insbesondere Herrn Prof. Dr.-Ing. Ernst Heil, sowie Herrn Dipl.-Phys. Bernd Ritschel. Ihre konstruktiven Ratschläge und Hinweise haben stets zur Verbesserung der Arbeit geführt.

Besonderer Dank geht an meine Familie und Freunde, die mich während des Studiums stets unterstützt haben.

## Zusammenfassung

In der heutigen Zeit sind Informationen jeglicher Art über das World Wide Web (WWW) für eine breite Bevölkerungsschicht zugänglich. Dabei ist es jedoch schwierig die existierenden Dokumente auch so aufzubereiten, dass die Inhalte für Maschinen inhaltlich interpretierbar sind. Das Semantic Web, eine Weiterentwicklung des WWWs, möchte dies ändern, indem es Webinhalte in maschinenverständlichen Formaten anbietet. Dadurch können Automatisierungsprozesse für die Suchanfragenoptimierung und für die Wissensbasenvernetzung eingesetzt werden. Die Web Ontology Language (OWL) ist eine mögliche Sprache, in der Wissen beschrieben und gespeichert werden kann (siehe Kapitel 4 OWL). Das Softwareprodukt Protégé unterstützt den Standard OWL, weshalb ein Großteil der Modellierungsarbeiten in Protégé durchgeführt wurde.

Momentan erhält der Nutzer in den meisten Fällen bei der Informationsfindung im Internet lediglich Unterstützung durch eine von Suchmaschinenbetreibern vorgenommene Verschlagwortung des Dokumentinhaltes, d.h. Dokumente können nur nach einem bestimmten Wort oder einer bestimmten Wortgruppe durchsucht werden. Die Ausgabeliste der Suchergebnisse muss dann durch den Nutzer selbst gesichtet und nach Relevanz geordnet werden. Das kann ein sehr zeit- und arbeitsintensiver Prozess sein. Genau hier kann das Semantic Web einen erheblichen Beitrag in der Informationsaufbereitung für den Nutzer leisten, da die Ausgabe der Suchergebnisse bereits einer semantischen Überprüfung und Verknüpfung unterliegt. Deshalb fallen hier nicht relevante Informationsquellen von vornherein bei der Ausgabe heraus, was das Finden von gesuchten Dokumenten und Informationen in einem bestimmten Wissensbereich beschleunigt.

Um die Vernetzung von Daten, Informationen und Wissen im WWW zu verbessern, werden verschiedene Ansätze verfolgt. Neben dem Semantic Web mit seinen verschiedenen Ausprägungen gibt es auch andere Ideen und Konzepte, welche die Verknüpfung von Wissen unterstützen. Foren, soziale Netzwerke und Wikis sind eine Möglichkeit des Wissensaustau-

ches. In Wikis wird Wissen in Form von Artikeln gebündelt, um es so einer breiten Masse zur Verfügung zu stellen. Hier angebotene Informationen sollten jedoch kritisch hinterfragt werden, da die Autoren der Artikel in den meisten Fällen keine Verantwortung für die dort veröffentlichten Inhalte übernehmen müssen. Ein anderer Weg Wissen zu vernetzen bietet das Web of Linked Data. Hierbei werden strukturierte Daten des WWWs durch Verweise auf andere Datenquellen miteinander verbunden. Der Nutzer wird so im Zuge der Suche auf themenverwandte und verlinkte Datenquellen verwiesen.

Die geowissenschaftlichen Metadaten mit ihren Inhalten und Beziehungen untereinander, die beim GFZ unter anderem im Information System and Data Center (ISDC) gespeichert sind, sollen als Ontologie in dieser Arbeit mit den Sprachkonstrukten von OWL modelliert werden. Diese Ontologie soll die Repräsentation und Suche von ISDC-spezifischem Domänenwissen durch die semantische Vernetzung persistenter ISDC-Metadaten entscheidend verbessern.

Die in dieser Arbeit aufgezeigten Modellierungsmöglichkeiten, zunächst mit der Extensible Markup Language (XML) und später mit OWL, bilden die existierenden Metadatenbestände auf einer semantischen Ebene ab (siehe Abbildung 2). Durch die definierte Nutzung der Semantik, die in OWL vorhanden ist, kann mittels Maschinen ein Mehrwert aus den Metadaten gewonnen und dem Nutzer zur Verfügung gestellt werden. Geowissenschaftliche Informationen, Daten und Wissen können in semantische Zusammenhänge gebracht und verständlich repräsentiert werden. Unterstützende Informationen können ebenfalls problemlos in die Ontologie eingebunden werden. Dazu gehören z.B. Bilder zu den im ISDC gespeicherten Instrumenten, Plattformen oder Personen. Suchanfragen bezüglich geowissenschaftlicher Phänomene können auch ohne Expertenwissen über Zusammenhänge und Begriffe gestellt und beantwortet werden. Die Informationsrecherche und -aufbereitung gewinnt an Qualität und nutzt die existierenden Ressourcen im vollen Umfang.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Helmholtz-Zentrum Potsdam Deutsches GeoForschungsZentrum GFZ . . . . .	2
1.2	Information System and Data Center . . . . .	3
<b>2</b>	<b>Grundlagen für das Semantic Web</b>	<b>4</b>
2.1	Schichten des Semantic Web . . . . .	5
2.2	Tripel-Syntax . . . . .	9
<b>3</b>	<b>Ontologie</b>	<b>12</b>
3.1	Erstellung einer Ontologie . . . . .	13
3.2	Beispiele aus der Praxis . . . . .	16
<b>4</b>	<b>OWL</b>	<b>18</b>
4.1	Aufbau einer OWL-Datei . . . . .	19
4.2	Ontologiesprachen im Vergleich . . . . .	21
4.3	Zugrunde liegende Logikformalismen . . . . .	23
<b>5</b>	<b>Verwendete Software</b>	<b>26</b>
5.1	Begriffsklärung . . . . .	26
5.2	Cmap Tools Ontology Editor . . . . .	29
5.3	Protégé . . . . .	31
5.3.1	Grundlegender Aufbau des OWL Plugins . . . . .	32
5.4	Pellet . . . . .	34
5.5	XMLSpy . . . . .	36
<b>6</b>	<b>Untersuchung der Informationsquellen</b>	<b>37</b>
6.1	Externe Quellen . . . . .	37
6.2	ISDC . . . . .	38

6.3	Analyse der ISDC-Metadatendokumente . . . . .	40
6.3.1	Analyse des existierenden Modells . . . . .	42
6.3.2	Zu beantwortende Anwendungsfälle . . . . .	47
6.3.3	Grafische Modellierung der Beziehungen der ISDC-Klassen . . . . .	49
<b>7</b>	<b>Modellierung der Ontologie in Protégé-OWL</b>	<b>52</b>
7.1	Umsetzung der Anwendungsfälle . . . . .	70
7.2	Nächste Schritte . . . . .	75
<b>8</b>	<b>Fazit</b>	<b>80</b>
	<b>Anhänge</b>	<b>I</b>
<b>A</b>	<b>Abkürzungsverzeichnis</b>	<b>I</b>
<b>B</b>	<b>ISDC-Metadatenbestand</b>	<b>III</b>
<b>C</b>	<b>Elemente der Klassen</b>	<b>XVI</b>
<b>D</b>	<b>Konkordanzen</b>	<b>XXI</b>
<b>E</b>	<b>Zeichen der Prädikatenlogik</b>	<b>XXIII</b>

## Abbildungsverzeichnis

1	Webseite des ISDC . . . . .	3
2	Schichtenmodell des Semantic Web [32] . . . . .	5
3	RDF-Aussage [32] . . . . .	7
4	OWL-Teilsprachen [32] . . . . .	18
5	Concept Map . . . . .	29
6	Metadata-Tab . . . . .	33
7	Hauptkomponenten des Pellet Reasoners . . . . .	36
8	Existierende ISDC-Kategorien . . . . .	43
9	Konkordanz am Beispiel Institution . . . . .	44
10	Neue ISDC-Kategorien . . . . .	46
11	Anwendungsfall . . . . .	47
12	Beziehungen zwischen den ISDC-Klassen . . . . .	50
13	OWL Classes-Tab . . . . .	53
14	Individuals-Tab . . . . .	54
15	Properties-Tab; Objekt-Property . . . . .	55
16	Transitive Eigenschaft(Personell, Institution, Country) . . . . .	57
17	Transitive Eigenschaft(Producttype, Instrument, Platform, Project) . . . . .	57
18	Properties-Tab; Datatype-Property . . . . .	58
19	Existenzquantor . . . . .	60
20	Existenzquantor am Beispiel der Klasse Instrument-Aux . . . . .	60
21	Allquantor . . . . .	61
22	hasValue-Restriktion . . . . .	62
23	Bedingungen der Klasse Project-TOR . . . . .	62
24	Notwendige Bedingungen . . . . .	63
25	Notwendige und hinreichende Bedingungen . . . . .	63
26	Project-Institution-GFZ als Beispiel für eine definierte Klasse . . . . .	64
27	Asserted und Inferred Hierarchy . . . . .	66



---

Abbildungsverzeichnis

---

28	Klasse Project (Asserted Hierarchy) in OWLViz . . . . .	67
29	OWL-Datei im WWW . . . . .	68
30	Klasse Project (Inferred Hierarchy) in OWLViz . . . . .	69
31	Bedingung für Anwendungsfall 1 . . . . .	70
32	Ergebnis für Anwendungsfall 1 . . . . .	71
33	Bedingung für Anwendungsfall 2 . . . . .	72
34	Ergebnis für Anwendungsfall 2 . . . . .	72
35	Bedingung für Anwendungsfall 3 . . . . .	73
36	Bedingung für Anwendungsfall 4 . . . . .	74
37	Ergebnis für Anwendungsfall 4 . . . . .	74
38	Bedingung für Anwendungsfall 5 . . . . .	75
39	Dublin Core Import . . . . .	76
40	Queries-Tab . . . . .	77
41	Bild zum Instrument SG-GWR-SG-056 . . . . .	78
42	Integration der Datenprodukte . . . . .	79
43	Elemente der Klasse Projekt . . . . .	XVI
44	Elemente der Klasse Produkttyp . . . . .	XVII
45	Elemente der Klasse Plattform . . . . .	XVIII
46	Elemente der Klasse Instrument . . . . .	XIX
47	Elemente der Klasse Institution . . . . .	XX
48	Konkordanz: Instrument - Plattform . . . . .	XXI
49	Konkordanz: Instrument - Projekt . . . . .	XXI
50	Konkordanz: Instrument - Produkttyp . . . . .	XXI
51	Konkordanz: Produkttyp - Plattform . . . . .	XXII
52	Konkordanz: Projekt - Plattform . . . . .	XXII
53	Konkordanz: Projekt - Produkttyp . . . . .	XXII

## Tabellenverzeichnis

1	OWL vs. RDFS . . . . .	22
2	Festlegung von Begriffen . . . . .	27
3	Begriffe . . . . .	28
4	Beziehungen der Metadatenkokumente . . . . .	41
5	Product_type . . . . .	III
6	Project . . . . .	VIII
7	Platform . . . . .	IX
8	Instrument . . . . .	XII
9	Institution . . . . .	XIII
10	Zeichen der Prädikatenlogik . . . . .	XXIII

## Quelltextverzeichnis

1	N3-Schreibweise von Abbildung 3 . . . . .	9
2	Verwendung von Qnames . . . . .	10
3	Turtle-Schreibweise (Teil 1) . . . . .	11
4	Turtle-Schreibweise (Teil 2) . . . . .	11
5	OWL-Datei . . . . .	20

## 1 Einleitung

Aufgabe des ISDC-Portals des Helmholtz-Zentrum Potsdam Deutsches GeoForschungsZentrum GFZ ist, die dort verwalteten Informationen und Daten in bestmöglicher Form für den Nutzer zugänglich zu machen. Das Semantic Web und die damit verbundenen Techniken erweitern und verbessern die Präsentation und Selektion von relevanten Informationen eines spezifischen Wissensgebietes, ohne dass der Nutzer dabei immer über tiefgehendes Wissen über die inneren Zusammenhänge und Begriffe des jeweiligen Wissensgebietes verfügen muss. Dieser Ansatz soll auch für die Wissenspräsentation und Informationssuche im ISDC angewendet werden.

Bevor die Umsetzung der ISDC-Ontologie erfolgen kann, ist die Analyse der vorhandenen Metadatenbestände unabdingbar. Vor allem die existierenden XML-Dateien müssen in die Betrachtungen einbezogen werden. Im Zuge des Analyse- und Modellierungsprozesses werden Softwaretools aus den Bereichen Wissensmodellierung und Semantic Web zum Einsatz kommen. Ziel dieser Arbeit ist es, eine Domänen-Ontologie aufzubauen, welche die aufgestellten Anwendungsfälle (siehe Punkt 6.3.2) beantworten kann. Beispielsweise soll die Selektion relevanter Produkttypen, die von einem bestimmten Instrument stammen, möglich sein.

## 1.1 Helmholtz-Zentrum Potsdam Deutsches GeoForschungsZentrum GFZ

Das GFZ erforscht als nationales deutsches Geoforschungszentrum unseren Heimatplaneten, die Erde. Die Untersuchung verschiedener geowissenschaftlicher Phänomene wird durchgeführt, um Erkenntnisse über Prozesse zu erlangen, die sowohl im Inneren, als auch auf der Erdoberfläche und der Atmosphäre stattfinden. Es werden geologische, physikalische, chemische und biologische Prozesse erforscht. Dies schlägt sich in den verschiedenen Fachabteilungen nieder [19]. Das GFZ strukturiert sich in folgende Departments:

- Department 1: Geodäsie und Fernerkundung
- Department 2: Physik der Erde
- Department 3: Geodynamik und Geomaterialien
- Department 4: Chemie und Stoffkreisläufe der Erde
- Department 5: Prozesse der Erdoberfläche.

Darüber hinaus existieren diverse Fachabteilungen, die sich mit wissenschaftlicher Infrastruktur und Plattformen beschäftigen, unter Berücksichtigung geoinformationstechnischer Entwicklungen. Zu ihren Aufgaben gehören unter anderem der Aufbau, die Pflege und die Bereitstellung von geowissenschaftlichen Datenbeständen und den daraus resultierenden Forschungsergebnissen [18].

Ziel dieser interdisziplinären Forschung ist es, Vorgänge des Systems Erde besser verstehen zu können. Dazu ist es notwendig, vielfältige Wechselwirkungen, wie sie z.B. zwischen der belebten Welt und der festen Erde vorkommen, zu berücksichtigen und zu analysieren.

## 1.2 Information System and Data Center

Das Information System and Data Center (ISDC) ist ein Onlinedienst des GFZ, der geowissenschaftliche Datenprodukte verschiedener Projekte, wie z.B. CHAMP, GRACE, GNSS oder TSX-TOR, verwaltet und für Wissenschaftler auf der ganzen Welt zur Verfügung stellt. Des Weiteren gehören zu den einzelnen Datenprodukten auch beschreibende und weiterführende Informationsquellen, wie beispielsweise Metadaten, Publikationen, PowerPoint-Präsentationen oder Newsletter. Die Hauptaufgaben des ISDC-Portals sind die Langzeitspeicherung und die Online-Bereitstellung von Messdaten, bzw. die aus Messdaten gewonnenen Datenprodukte, für verschiedene nationale und internationale Forschungsarbeiten. Datenprodukte, die innerhalb bestimmter Projekte generiert werden, müssen durch ISDC-Services angenommen und archiviert, sowie nach Recherche auf Anfrage wieder bereit gestellt werden. Das ISDC-Portal stellt somit für Datenproduzenten und Datennutzer ein wichtiges Werkzeug für die Informations- und Produktsuche dar. Darüber hinaus obliegt dem ISDC auch die Nutzerverwaltung, die den Zugriff auf gewünschte Daten und Informationen steuert [25].

The screenshot displays the ISDC website interface. At the top, the GFZ logo and 'INFORMATION SYSTEMS AND DATA CENTER' are visible. A navigation menu on the left includes sections for 'Information', 'Data Access', and 'Collaboration'. The main content area features a 'Welcome to the Information System and Data Center for geoscientific data' message, followed by several announcements for 'GRACE Monthly Report' and 'GRACE Technical Note #05'. A right-hand sidebar contains a 'Forum' section with 'LAST FORUM POSTS' and 'FORUMS' lists, a 'Personal Block' indicating the user is not logged in, and a 'Data Flow (last 60d)' graph showing input and output trends. At the bottom right, a 'Storage Statistics' box reports 14.78 TB of data and 24.37 Mio products.

Abbildung 1: Webseite des ISDC

## 2 Grundlagen für das Semantic Web

Das WWW bietet in seiner heutigen Form universell verfügbare Informationen, die von einer breiten Bevölkerungsschicht kostengünstig abgerufen werden können. Die Präsentation der Inhalte ist dabei größtenteils auf den Menschen ausgerichtet, d.h. es obliegt dem Nutzer selbst die Informationen zu interpretieren. Bei der Informationssuche erhält man hauptsächlich Hilfestellungen durch Suchmaschinen. Diese durchsuchen das WWW nach Zeichenketten und geben anschließend eine Ergebnisliste mit passenden Webseiten aus. Die Ergebnismenge muss dann vom Anwender analysiert und interpretiert werden, um an benötigte Daten und Informationen zu gelangen [31, Seiten 9-12]. Der Inhalt des Internets setzt sich also aus unzähligen Webseiten in einem verteilten Netzwerk zusammen, die überwiegend auf den Menschen als Endnutzer ausgerichtet sind.

Maschinen können den Inhalt der dort hinterlegten Informationen nicht erfassen. Dies möchte das Semantic Web ändern. Dazu ist es allerdings notwendig, Daten so aufzubereiten, dass sie für Maschinen lesbar sind. Zunächst einmal bedeutet dies zusätzliche Arbeit, denn Datenbestände müssen mit neuen, für das Semantic Web kompatiblen Standards abgebildet werden. Sind diese Transformationen der Inhalte abgeschlossen, erschließen sich neue Wege der Datensuche in einer einheitlichen Infrastruktur. Durch Semantic Web basierte Applikationen erhält der Nutzer ein Werkzeug, um möglichst schnell die richtigen Daten am richtigen Ort zu suchen und zu finden [41, Seite 1-5].

Sowohl das WWW, als auch das Semantic Web, beruhen auf dem Konzept von Tim Berners-Lee. Das Semantic Web stellt eine Erweiterung des aktuellen WWWs dar und ist nicht als Ersatz für das WWW gedacht. In der Zeitschrift „Scientific American“ ist es wie folgt definiert:

„The Semantic Web is an extension of the current web in which information is given a well-defined meaning, better enabling computers and people to work in cooperation.“ [39]

Ziel dieser Bestrebungen ist es, die Menschen vor einem Überangebot an irrelevanten Informationen und Daten zu bewahren. Der Nutzer soll nur das Ergebnis bekommen, das er auch tatsächlich gesucht hat. Wird beispielsweise nach dem Begriff „Jaguar“ gesucht, erhält man Webseiten, die sich mit der Automarke und dem Tier Jaguar beschäftigen. Entwicklungen des Semantic Webs unterstützen die Ausgabe kontextrelevanter Informationen, d.h. bewegt sich der Nutzer in der Biologie, wird er auch nur Webseiten bekommen, die sich mit dem Tier befassen. Dieses implizite Wissen wird aus den mit Semantik angereicherten Dokumenten durch Ableitungsregeln und Schlussfolgerungen hergeleitet.

## 2.1 Schichten des Semantic Web

Im folgenden Abschnitt wird auf verschiedene, vom World Wide Web Consortium (W3C) empfohlene, Grundlagen und Standards für das Semantic Web eingegangen. Abbildung 2 zeigt ein Modell, in dem die Schichten abgebildet sind, die zur Umsetzung des Semantic Webs benötigt werden. Je höher die Schicht im Schichtenmodell liegt, desto näher kommt man der vollständigen Realisierung des Semantic Web.

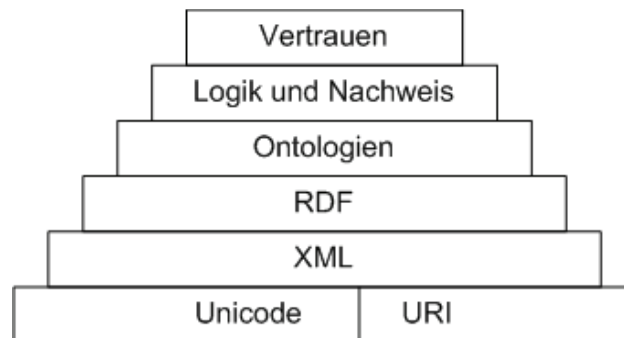


Abbildung 2: Schichtenmodell des Semantic Web [32]

Die unterste Schicht setzt sich aus dem Unicode und dem Uniform Resource Identifier (URI) zusammen. Sie bilden eine gemeinsame Basis für weitere Arbeiten. Der Unicode ist in der ISO Norm 10646 verankert und definiert die Kodierung von internationalen Schriftzeichen. Schriftzeichen werden in einer 2 Byte langen Zahl abgelegt. Somit ist Platz für 65 536 Zeichen. Dadurch ist es möglich, Schriftzeichen aus asiatischen und anderen Spra-

chen, sowie weitere Sonderzeichen, mit in das Kodierungssystem aufzunehmen. Es werden permanent neue Zeichen in den Unicode aufgenommen, d.h. er unterliegt der ständigen Weiterentwicklung [30].

URIs sind standardisierte Bezeichner und dienen der eindeutigen Identifizierung und Adressierung von Objekten im WWW. Für die Lokalisierung von Ressourcen im Netzwerk benötigt man Server- und Pfadangaben sowie den jeweiligen Dateinamen. Ressourcen, die mit einer URI versehen werden können, sind z.B. Dokumente, Bilder oder Webseiten mit dazugehörigen Diensten. All diese Ressourcen sind über URIs im Web erreichbar [31, Seiten 25-29].

XML ist die Grundlage für eine große Sprachfamilie, genauer gesagt von Auszeichnungssprachen. XML ist als das Grundgerüst für das Semantic Web anzusehen, denn auf ihm bauen alle anderen Sprachen auf. Beispiele hierfür sind RDF oder OWL, was Abbildung 2 zu entnehmen ist. Alle Mitglieder dieser Sprachfamilie benutzen eine XML-basierte Syntax zum Dokumentenaufbau. Auf diese wird im Folgenden näher eingegangen. Markup-Sprachen strukturieren Inhalte mit sogenannten Tags. In XML gibt es keine Vorgaben für die Erstellung von Tags. Durch die Markups erhalten Inhalte zusätzliche Informationen in Form von Metadaten, welche die logische Struktur festlegen. Es gibt zwei Möglichkeiten die Strukturvorgaben zu definieren. Zum einen geschieht dies durch Document Type Definitions (DTDs), zum anderen durch ein XML-Schemata. Ein XML Schema ist eine eigene XML-basierte Sprache, die geforderte Strukturen für Dokumente definiert. XML-Strukturen ermöglichen die Kommunikation von Maschine zu Maschine in einem standardisierten System. Maschinen können jedoch die Bedeutungen der Inhalte noch nicht interpretieren [22]. In XML werden Namensräume verwendet, um die Eindeutigkeit von Namen zu gewährleisten und Namenskonflikte auszuschließen. Namensräume werden durch URIs repräsentiert [44]. Das Präfix `owl` identifiziert beispielsweise OWL-konforme Konstrukte. Unter Punkt 4.1 Aufbau einer OWL-Datei ist eine Namensraum-Deklaration für ein OWL-Dokument aufgeführt.



Das Resource Description Framework (RDF) ist die nächst höhere Ausdrucksform im Schichtenmodell (siehe Abbildung 2) und ist gleichzeitig das grundlegende Darstellungsformat des Semantic Webs, das explizit strukturierte Informationen beschreibt. Informationen werden durch gerichtete Graphen ausgedrückt. Die Tripel eines solchen Graphen bestehen aus Subjekt, Prädikat und Objekt und werden jeweils mittels URI referenziert. Folglich ist die eindeutige Namensgebung garantiert. Subjekt und Objekt stellen dabei Knoten und das Prädikat eine Kante dar. Abbildung 3 zeigt die Struktur eines solchen Tripels. Die Interpretation dieses Graphen sagt aus, dass ein Produkttyp beim ISDC recherchierbar ist.

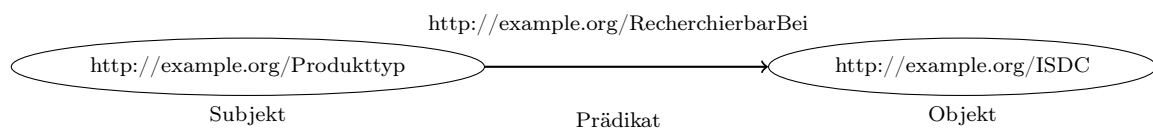


Abbildung 3: RDF-Aussage [32]

RDF Schema (RDFS) ist ein erweitertes RDF-Dokument, das aus einem Satz definierter Bezeichner für Individuen, Beziehungen und Klassen besteht. Die Gesamtheit dieses Satzes wird als Vokabular bezeichnet. Mithilfe von RDFS ist es möglich Zusammenhänge und Strukturen über Informationen auszudrücken. Mit RDFS haben Entwickler ein Werkzeug, um sogenannte leichtgewichtige Ontologien abzubilden [31, Seiten 66-68]. Leichtgewichtige Ontologien beinhalten noch keine Axiome oder Beschränkungen. Die dargestellten Beziehungen und Klassenzugehörigkeiten lassen einfache Schlussfolgerungen zu. Zwar sind die Ausdrucksmöglichkeiten gegenüber OWL-basierten Ontologien noch beschränkt, aber es ist ein weiterer Schritt in Richtung Semantic Web [41, Seiten 91-102].

Der Begriff Ontologie setzt sich aus den griechischen Worten *on* und *logos* zusammen und bezeichnet in der Philosophie die „Lehre vom Sein“. In der Informatik wird eine Ontologie, z.B. von Thomas R. Gruber, wie folgt definiert:

„ An ontology is an explicit specification of a conceptualization.“ [21]

Eine Ontologie ist demnach eine explizite Spezifikation der Darstellung eines Konzepts. Es kann jedes Phänomen der realen Welt durch ein abstraktes Modell dargestellt werden. Eine Ontologie beschreibt eine Wissensbasis mit einem Vokabular, das sich aus Klassen, Relationen und Regeln zusammensetzt. Ontologien können Kommunikationsprozesse, sowie das Verwenden und Wiederverwenden von Wissen fördern und erleichtern. Darüber hinaus ermöglichen sie Maschinen, durch definierte Regeln, das automatische Schlussfolgern. Die Entwicklungen des Semantic Webs verstärken auch das Interesse an Ontologien in der Informatik, da sie eine Möglichkeit darstellen, Wissen maschinenverständlich aufzubereiten [15]. Weitere Definitionen und Ausführungen befinden sich in Kapitel 3 Ontologie.

Nachdem die Modellierung einer Ontologie abgeschlossen ist, folgt der nächste Schritt des Schichtenmodells, nämlich das automatische Schlussfolgern. Dies wird im Allgemeinen durch die Prädikatenlogik der ersten Ordnung realisiert. Die zu verwendende Logik muss entscheidbar sein, damit eine Schlussfolgerung auch lösbar ist und von einem Reasoning Tool bearbeitet werden kann.

Vertrauen auf die im WWW gespeicherten und durch Dienste gelieferten Informationen ist momentan nicht garantiert, denn jeder kann alles über jedes Themengebiet behaupten und veröffentlichen. Um diese Schicht adäquat umsetzen zu können, bedarf es der weiteren Grundlagenforschung [22]. Derzeit gibt es verschiedene Ansätze um die Glaubwürdigkeit von Quellen zu prüfen. Das W3C-Konsortium konzentriert sich dabei hauptsächlich auf technische Standards, die Vertrauen garantieren sollen. Dazu zählen unter anderem Verschlüsselungsverfahren oder digitale Signaturen. Forscher der Stanford University in Kalifornien verfolgen ein anderes Verfahren. Sie beschäftigen sich mit Vertrauensnetzwer-

ken, die auf sozialen Prinzipien beruhen. Den Menschen oder Ressourcen, die wir kennen und bewerten können, wird eher Vertrauen geschenkt, als denen die wir nicht kennen. Mit dem offenen Bewertungssystem wird eine weitere Methode zur Bewertung des Vertrauens in Betracht gezogen. Dabei erhalten Quellen bzw. Personen eine Bewertung auf ihre Informationen bezüglich ihres vermuteten Wahrheitsgehaltes. Es ist auch ein hybrider Ansatz möglich, der sowohl technische als auch soziale Bestandteile enthält [36].

## 2.2 Tripel-Syntax

Im Zuge der Semantic Web-Bestrebungen haben sich verschiedene Ausgabeformate entwickelt. Auch diesem Themenbereich widmete sich Tim Berners-Lee. Er hat erstmals die Umsetzung von Informationen des WWWs in Tripeln aufgegriffen. Dies ist in der Notation 3 (N3) beschrieben. N3 kann auch komplexere Ausdrücke, wie z.B. Pfade und Regeln, beinhalten. Eine einfachere Umsetzung von N3 ist die Syntax N-Triple. N-Triple wurde zu einem späteren Zeitpunkt zur Syntax Turtle erweitert, um unter anderem eine praktische Kurzschreibweise zu integrieren. N-Triple und Turtle sind im Wesentlichen Teile von N3, d.h. sie sind Abwandlungen dieser Syntax [31, Seiten 40-42]. Dabei ist N-Triple eine Untermenge von Turtle. Das kann auch in Logikschreibweise ausgedrückt werden:  $N\text{-Triple} \subseteq \text{Turtle} \subseteq N3$ .

N3 setzt sich aus einem Subjekt, einem Prädikat und einem Objekt zusammen. Die einzelnen Bestandteile sind durch ein Leerzeichen getrennt. Vollständig qualifizierte URIs sind von spitzen Klammern umgeben. Das Ende eines Tripels wird durch einen Punkt angezeigt. Der Inhalt der Abbildung 3 sieht in N3 beispielsweise so aus [12]:

Quelltext 1: N3-Schreibweise von Abbildung 3

```
1 <http://example.org/Produkttyp> <http://example.org/RecherchierbarBei>
   <http://example.org/ISDC> .
```

Es ist hier zu beachten, dass sich Subjekt, Prädikat und Objekt aus Platzgründen nicht in einer Zeile abbilden lassen. Deshalb wurde, zur Gewährleistung der Lesbarkeit, die Ausgabe um einen Zeilenumbruch ergänzt.

Anstatt des vollen URI können auch Qualified names (Qnames) verwendet werden. Sie ersetzen den vollen URI durch ein vordefiniertes Namensraumpräfix. Das Präfix und der lokale Bezeichner sind durch einen Doppelpunkt getrennt [20]:

Quelltext 2: Verwendung von Qnames

```
1 xmlns:ex="http://example.org#"
2 ex:Datenprodukte ex:RecherchierbarBei ex:ISDC .
```

Zeile 1 ist ein Beispiel für einen Qname, welcher in Zeile 2 zum Einsatz kommt. Mit dem Quelltext 2 wird ebenfalls Abbildung 3 dargestellt, aber die Schreibweise wurde erheblich verkürzt. N-Triples wurde entwickelt, um ein einfaches Format für Analysezwecke zur Verfügung zu stellen. Die N-Triple-Syntax ist für Skripte optimiert, weswegen der Inhalt für den Menschen schwer zu lesen ist. Sie enthalten keine Namensräume [5].

Turtle steht für Terse RDF Triple Language und wurde speziell für RDF geschaffen, um die Lesbarkeit für den Menschen zu verbessern. Die Turtle-Syntax beinhaltet Kurzschreibweisen, was die Interpretation für Programmierer vereinfacht. Sind viele Tripel enthalten, die sich auf dasselbe Subjekt beziehen, kann ein Semikolon gesetzt werden. Quelltext 3 enthält zwei Tripel, in denen das Subjekt `Produkttyp` gleich ist. Wertzuweisungen werden in Anführungszeichen gesetzt. In diesem Fall wird der Titel durch einen Wert in Form eines Textes ausgedrückt. Im Quelltext 3 kommt zum Ausdruck, dass der Produkttyp den Namen „CH-AI-3-ATM“ hat, und dass der Titel des Produkttypen „CHAMP profiles of vertical atmospheric parameter“ ist.

### Quelltext 3: Turtle-Schreibweise (Teil 1)

```
1 @prefix ex: <http://example.org/>
2 ex:Produkttyp ex:hatName ex:CH-AI-3-ATM;
3           ex:hatTitle "CHAMP profiles of vertical atmospheric parameter".
```

Darüber hinaus können Subjekt und Prädikat für das nächste Tripel übernommen werden, indem ein Komma gesetzt wird [45]. So können mehrere Produkttypnamen gleichzeitig zugewiesen werden. Im Fall von Quelltext 4 erhält der Produkttyp die Namen **CH-AI-3-ATM** und **TSX-ORB-3PSA**.

### Quelltext 4: Turtle-Schreibweise (Teil 2)

```
1 ex:Produkttyp ex:hatName ex:CH-AI-3-ATM, ex:TSX-ORB-3-PSA .
```

Detaillierte Informationen zu den verschiedenen Ausgabeformaten, sind der von Berners-Lee verfassten Website [39] zu entnehmen.

## 3 Ontologie

Neben der Definition von Gruber (siehe Abschnitt 2.1 Schichten des Semantic Web) existieren noch andere Definitionen des Begriffes **Ontologie**. Die folgenden Beschreibungen stammen aus Online-Lexika und geben Auskunft über die Bedeutung des Begriffs **Ontologie**.

„Defines concepts, providing a way to move towards consistency in vocabulary. It provides a working model of the entities and interactions of a particular topic.“[26]

„In the context of Digital Asset Management, an ontology shows the relationships, properties and functions between terms or concepts which can express a wider range of relationships between attributes or terms than can a simple hierarchy. This can be very useful when attempting to represent complex or multi-faceted relationships.“[2]

Ontologien haben die Aufgabe Informationen zu strukturieren, um bereits vorhandenes Wissen zusammenzuführen und die Suche nach relevanten Daten zu erleichtern. Die Zusammenführung verschiedener Wissensbasen, die verschiedene Abbildungsschwerpunkte haben, birgt allerdings Probleme in sich. Diese Problematiken bezüglich der Abbildung müssen in der Ontologie ausreichend berücksichtigt werden, da es auch zu einem späteren Zeitpunkt möglich sein muss, Manipulationen am Datenbestand vornehmen zu können. Dies ist notwendig, um Prozesse der realen Welt adäquat abbilden zu können. Bei der Modellierung ist das ausreichend zu berücksichtigen. Die momentan gängigste Darstellungsform zur Beschreibung von realen Objekten und ihren Beziehungen ist die Ontologie. Sie hat in der Informationsverarbeitung die Aufgabe Objekte und deren Beziehungen zu formalisieren, um eine Verbindung zwischen der informationstechnischen Darstellung und den Erfahrungen des Menschen herzustellen. Dabei ist es schwierig Homonyme richtig zu zuordnen. Dies kann am Begriff Bank verdeutlicht werden, denn eine Bank kann für eine Sitzgelegenheit, oder aber auch für ein Geldinstitut stehen. Die richtige Bedeutung kann nur anhand des

Kontextes ermittelt werden. Reale Objekte der Welt können in verschiedene Arten (konkrete und abstrakte Objekte) unterteilt werden. Konkrete Objekte stehen beispielsweise für Personen, abstrakte Objekte z.B. für Organisationen.

Ontologien können durch ihren Zweck, wie sie Wissen repräsentieren, unterschieden werden. Zum einen gibt es Top-Level Ontologien, die Wissen allgemeiner Natur darstellen, wie z.B. Raum oder Zeit unabhängig von einer Domäne. Zum anderen bilden Domänen-Ontologien einen speziellen Wissensbereich ab. Die für das ISDC zu entwickelnde Ontologie ist der Domänen-Ontologie zuzuordnen. Eine weitere Unterteilung kann aufgrund der Rolle, welche die Ontologie spielt, vorgenommen werden. Eine Aufgabenontologie enthält Definitionen, mit deren Hilfe dann eine allgemeine Aktivität beschrieben werden kann. Die Anwendungsontologie beschreibt spezielle Aspekte der Anwendung und ist meist auf eine konkrete Aufgabe zugeschnitten. Domänen-Ontologien und Aufgabenontologien spezialisieren Wissen von einer Top-Level Ontologie. Eine Anwendungsontologie differenziert eine Domänen- oder eine Aufgabenontologie. Egal welcher Ontologietyp zur Repräsentation von Wissen verwendet wird, die Funktion bleibt die gleiche, nämlich die korrekte Beschreibung der Realität. Darüber hinaus soll die wahrheitsgemäße Interpretation eines gegebenen Datenbestandes unterstützt werden. [37, Seiten 5-8, 55-57, 97]

#### **3.1 Erstellung einer Ontologie**

Ontologien haben die Aufgabe Datenbestände zu strukturieren, verschiedene Wissensbasen zusammenzuführen und neues Wissen aus dem Datenbestand abzuleiten. Zur Umsetzung werden Sprachstandards benutzt, wie z.B. OWL (siehe Kapitel 4 OWL). Sprachstandards bieten den Entwicklern von Ontologien ein formales Gerüst für die Modellierungsarbeiten. Definierte Standards bieten jedoch noch keine Antwort auf die Frage, wie detailliert eine gute und korrekte Ontologie aussieht. Die Frage welche Objekte und Beziehungen zu welchem Zweck abgebildet werden sollen, ist eines der Hauptprobleme bei der Modellierung. Beim Entwurf gilt es zwei Ebenen zu berücksichtigen, die eben beschriebene formale und die konzeptuelle Ebene. Die konzeptuelle Ebene beschäftigt sich mit der Struktur und der

Terminologie der beschriebenen Domäne. In diesem Fall ist die zu bearbeitende Domäne das ISDC-Portal und die dort verfügbaren Inhalte. Bei der Umsetzung der Ontologie in die gewählte Sprache kann es zu Konflikten zwischen den Ontologie-Sprachkonstrukten und den Anforderungen der Wissensbasis kommen. Denn nicht alle Inhalte und semantischen Beziehungen lassen sich in der gewählten Sprache ausdrücken. In den meisten Fällen ist es nicht möglich, die vollständige Semantik zu formulieren. Ein weiteres Problem stellt die Anwendung der Sprachkonstrukte dar, weil viele Lösungsmöglichkeiten für die Abbildung von Semantik existieren. Es gibt also nicht „den einen Weg“ ein Wissensgebiet zu modellieren. Es muss je nach Anforderung die geeignetste Form gewählt werden. Die Ontologieerstellung ist ein iterativer Prozess mit einer schrittweisen Formalisierung. Das wiederum macht den Aufbau einer Ontologie zu einem sehr zeit- und arbeitsintensiven Vorgang. [37, Seiten 157-160]

In der Informationsverarbeitung findet sich ein wichtiges Anwendungsgebiet für Ontologien. Sie sollen die semantische Datenintegration gewährleisten, d.h. Daten aus heterogenen Quellen sollen für einen gemeinsamen Zweck nutzbar sein. Ontologien bieten eine neutrale Struktur, die für ein globales Datenmodell notwendig ist. Durch Formalismen können implizite Annahmen durch Inferenzmethoden abgeleitet werden. Das hilft unter anderem Inkonsistenzen im definierten Modell aufzudecken. [37, Seiten 215, 225, 226]

Bevor eine Ontologie neu entwickelt wird, sollte nach bereits vorhandenen und verwendbaren Ontologie-Modellen gesucht werden, um die anfallenden Modellierungsarbeiten und die damit verbundenen Kosten so gering wie möglich zu halten. Gibt es keine Ontologie die in abgeänderter Form verwertbar ist, muss ein Neuentwurf erfolgen. Existieren schon Modelle und sollen diese in die jeweilige Modellentwicklung integriert werden, dann sind Inkonsistenzen der Ontologie-Vokabulare zu verhindern. In einem Vokabular sind die Begrifflichkeiten der zugrunde liegenden Konzepte mit ihren Relationen hinterlegt. Bei Verwendung von Informationen aus verschiedenen Modellansätzen können nützliche Strukturen und Daten, die zur Beantwortung einer Aufgabenstellung hilfreich sind, themenübergreifend zur



Verfügung gestellt werden. Im Entwicklungsprozess ist Augenmerk auf das Gleichgewicht zwischen Ausdrucksstärke und effizientem Schlussfolgern zu legen. Darüber hinaus sind andere Standards und deren Kompatibilität zu beachten, auch um die Interoperabilität und Universalität zu gewährleisten [41, Seiten 271, 272].

Vor der Entwicklung einer Ontologie sollte man sich mit verschiedenen Fragen auseinandersetzen, die zur Einordnung der Wissensbasis dienen. Zum einen ist hier die Domäne festzulegen. Der geowissenschaftliche Metadatenbestand des ISDCs mit seinem Inhalt und seinen Beziehungen ist die Domäne dieser Arbeit. Des Weiteren ist der genaue Zweck der Ontologie festzulegen. Dadurch steht fest welche Fragestellungen beantwortet werden sollen. Die Ontologie soll die Integration des ISDC-Wissens unterstützen und bei der Informationsrecherche helfen. Die Visualisierung von semantischen Zusammenhängen soll die Repräsentation des Wissens erleichtern und Vernetzung von Informationen sichtbar machen. Sogenannte Competency Questions (siehe Abschnitt 6.3.2 Zu beantwortende Anwendungsfälle) geben Auskunft darüber, welche wissenschaftlichen Fragestellungen durch die Ontologie beantwortbar sein sollen. Während der Modellierungsphase kann durch die Anwendungsfälle immer wieder geprüft werden, ob die Modellierung korrekt ist, oder ob noch Änderungen notwendig sind. Außerdem ist zu prüfen, wer die Ontologie nutzen und pflegen wird. Als Nutzer kommen Wissenschaftler, Studenten, Lehrer, aber auch interessierte Laien in Frage. Die Pflege der Ontologie und der dazugehörigen Datenbestände muss durch Mitarbeiter des ISDCs sichergestellt werden. Nachdem ein umfassendes Bild über die Wissensbasis existiert, können relevante Begriffe und Zusammenhänge identifiziert und in das Modell, in Form von Klassen, Individuen und Relationen, übernommen werden [27].

Um die Komplexität von Ontologien beherrschbar zu machen, gibt es diverse Tools für die Erstellung von Ontologien. Sie erleichtern Modellierungsarbeiten z.B. durch grafische Oberflächen und Konsistenzprüfungen, oder sie helfen bei der Fehlersuche. Um einen Gesamtüberblick über das Modell zu bekommen, eignen sich vor allem Visualisierungen. Entwurfsmuster vereinfachen zusätzlich die Modellierungsarbeiten. Des Weiteren bieten einige

Ontologie-Editoren Methoden des logischen Schließens, was die Arbeiten erheblich erleichtert [37, Seiten 187, 188]. Für diese Masterarbeit wurden die Editoren Cmap Tools Ontology Editor (COE) und Protégé eingesetzt. Auf diese wird in Kapitel 5 Verwendete Software noch genauer eingegangen.

### 3.2 Beispiele aus der Praxis

Das Konzept Friend of a Friend (FOAF) repräsentiert personenbezogene Informationen durch die Vernetzung von sozialen Netzwerkinformationen. Darüber hinaus werden auch Beziehungen zwischen Personen untereinander abgebildet. Zu den Personendaten zählen neben Name, Geschlecht, E-Mail-Adresse, Telefonnummer und Anschrift, auch Angaben zu Projekten und Organisationen, denen Personen angehören können, sowie verfasste Dokumente oder Fotos. FOAF schafft Rahmenbedingungen für die Nutzung von verteilten Personendaten im WWW, welche für die maschinelle Verarbeitung zusammengetragen werden. Es handelt sich hierbei um eine praktische Umsetzung von Semantic Web-Technologien [46].

Das Simple Knowledge Organisation System (SKOS) organisiert, basierend auf RDF und RDFS, Thesauri, Taxonomien, Folksonomien oder anderes kontrolliertes Wortgut. In einem Thesaurus sind definierte Begriffe enthalten, die durch Relationen miteinander verbunden sind. Durch die Relationen wird auf weitere Begriffe zu einem Suchbegriff verwiesen. Lautet der Suchbegriff beispielsweise Pflanzen, dann wird mittels Relationen auf Grünpflanzen und Blütenpflanzen gezeigt. Anhand von Taxonomien können Objekte bzw. Begriffe klassifiziert werden, d.h. es erfolgt eine Zuordnung zu einer bestimmten Kategorie oder Klasse. Ein Objekt kann beispielsweise der Klasse belebt oder unbelebt zugeordnet sein. Folksonomien sammeln frei vergebene Schlagwörter, die vornehmlich im Social Tagging zum Einsatz kommen. SKOS ist zur Repräsentation von Wissenssystemen angelegt. Erweiterungen sind berücksichtigt worden, d.h. jeder gewünschte Thesaurus ist integrierbar [1].

Die Dublin Core Metadata Initiative (DCMI) entwickelt den Dublin Core Standard, um umfassende Meta-Angaben an Objekte im WWW anbringen zu können. Solche Objekte

können beispielsweise aus dem bibliothekarischen Bereich stammen. Mit Hilfe der 15 Kernelemente dieses Standards, wie z.B. Creator, Title oder Description, werden zusätzliche beschreibende Informationen (Metadaten) zur Verfügung gestellt. Diese sind sehr hilfreich für die Beschreibung von Dokumenten und die Beantwortung von Suchanfragen. Deswegen unterstützen viele große Suchmaschinen den Dublin Core Standard, um eine exaktere Ergebnisausgabe zu erreichen [9].

## 4 OWL

Das Akronym OWL bedeutet Web Ontology Language. OWL wurde entwickelt, um Webdokumente und -applikationen mittels Klassen, Relationen und Instanzen abzubilden. So können unter anderem Beziehungen vieler Daten durch automatische Agenten analysiert und ausgewertet werden. Das automatische Schlussfolgern wird durch Logik-Formalismen realisiert. Wie bereits erwähnt, bedient sich OWL hierfür der Prädikatenlogik erster Stufe. Für OWL existieren verschiedene Untersprachen, die unterschiedliche Ausdrucksstärken besitzen. Sie heißen OWL Lite, OWL DL und OWL Full. Die logische Schreibweise  $\text{OWL Lite} \subseteq \text{OWL DL} \subseteq \text{OWL Full}$  zeigt, dass OWL Full aus den Teilsprachen OWL DL und OWL Lite besteht. Dies wird ebenfalls in Abbildung 4 zum Ausdruck gebracht.

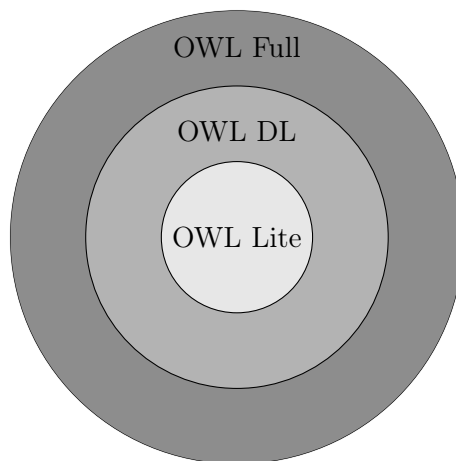


Abbildung 4: OWL-Teilsprachen [32]

OWL Lite stellt in erster Linie eine Klassifikationshierarchie und einfache Beschränkungsmöglichkeiten von Relationen zur Verfügung. Einsatzgebiet hierfür ist beispielsweise die Abbildung von Taxonomien. Das DL bei OWL DL steht für Description Logics und verfolgt das Ziel, eine entscheidbare Beschreibungslogik zu implementieren. Diese Teilsprache bietet viele Ausdrucksmöglichkeiten über Wissensbasen. OWL DL unterstützt zusätzlich den Einsatz von Inferenzmaschinen, da Daten in endlicher Zeit auf Zusammenhänge geprüft werden können. OWL Full besitzt die maximale Ausdrucksstärke. Jedoch gewährleistet

die syntaktische Freiheit nicht mehr die vollständige Entscheidbarkeit. Zum momentanen Zeitpunkt gibt es noch keine vollständige Softwarelösung für diese Teilsprache. Bei der Entwicklung einer Ontologie muss der Entwickler selbst entscheiden, welche OWL-Sprache den Anforderungen gerecht wird [43].

#### 4.1 Aufbau einer OWL-Datei

Der folgende Quelltext 5 zeigt den Grundaufbau einer OWL-Datei. Das XML-basierte Dokument beginnt mit dem Prolog, der die Versionsnummer, sowie die optionale Angabe des Kodierungsformates beinhaltet. Im Beispiel wird UTF-8 verwendet, die am weitesten verbreitete Kodierung für Unicode-Zeichen (siehe Zeile 1). Nach dem Prolog folgt das Wurzelement `rdf:RDF`, welches die Inhalte der OWL-Datei einschließt (siehe Zeile 2 und 25). Die Inhalte sind in Kopf- und Dokumententeil aufgeteilt. Im Kopf einer OWL-Ontologie finden Angaben zu Namensräumen (siehe Zeilen 3-7) und weiteren Metainformationen, wie Kommentierung (siehe Zeile 10), Versionskontrolle (siehe Zeile 11) und der Import anderer Ontologien (siehe Zeile 12) Platz. Die zusätzlichen Metainformation werden unter dem Tag `owl:Ontology` eingruppiert (siehe Zeilen 9-13) [31, Seiten 128, 129]. Der anschließende Dokumentteil (siehe Zeile 15-24) beinhaltet die Hauptinformationen der Ontologie. In unserem Fall wird eine Klasse Produkttyp angelegt, die mit einer Restriktion belegt ist. Die angegebene Kardinalität sagt aus, dass ein Produkttyp genau einen eindeutigen Bezeichner besitzen muss.

## Quelltext 5: OWL-Datei

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <rdf:RDF
3   xmlns="http://example.org/"
4   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
5   xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
6   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
7   xmlns:owl="http://www.w3.org/2002/07/owl#"
8
9   <owl:Ontology rdf:about="http://example.org/Produkttyp">
10     <rdfs:comment> Beispielontologie </rdfs:comment>
11     <owl:versionInfo> v0.2 </owl:versionInfo>
12     <owl:imports rdf:resource="http://isdc.gfz-potsdam.de/eineAndere#">
13   </owl:Ontology>
14
15   <owl:Class rdf:about:"http://example./org#Produkttyp">
16     <rdfs:subClassOf>
17       <owl:Restriction>
18         <owl:onProperty rdf:resource="hatEindeutigenBezeichner" />
19         <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">
20           1
21         </owl:cardinality>
22       </owl:Restriction>
23     </rdfs:subClassOf>
24   </owl:Class>
25 </rdf:RDF>
```

## 4.2 Ontologiesprachen im Vergleich

RDF-Tripel ermöglichen es, Beziehungen durch gerichtete Graphen eindeutig auszudrücken. Subjekt, Prädikat und Objekt werden durch Ressourcen, also durch URIs, dargestellt. Ab RDFS sind einfache Ontologien umsetzbar. Dabei werden die Ausdrücke von RDF erweitert. So ist z.B. die Definition von Klassen erlaubt. Daher ist es möglich, Hierarchien durch Unter- und Oberklassen, sowie durch Unter- und Oberproperties, abzubilden. Darüber hinaus können Typinformation sowie Angaben zu Domain und Range einer Property gemacht werden. Darauf aufbauend ist die Sprachfamilie OWL (siehe Abbildung 4) aus den Ontologiesprachen DARPA Agent Markup Language - Ontology (DAML-ONT) und Ontology Inference Layer(OIL) entstanden. OWL besitzt weitere Ausdrucksmöglichkeiten bezüglich Logik, was erweiterte maschinelle Inferenzmechanismen zulässt.

Tabelle 1 stellt die Fähigkeiten von RDFS und OWL gegenüber. Es ist eindeutig sichtbar, dass OWL vielfältigere Ausdrucksmöglichkeiten aufweist. Wird eine detaillierte Wissensrepräsentation angestrebt, dann ist die Verwendung einer dafür geeigneten Ontologiesprache unumgänglich. In diesem Fall ist die Wahl auf OWL gefallen. Im Folgenden werden die wichtigsten Relationseigenschaften und Mengenoperationen von OWL erläutert.

Eine symmetrische Beziehung sagt aus, dass wenn A in Beziehung zu B steht, steht auch B in Beziehung zu A. Das kann an der Property `hatZwilling` verdeutlicht werden. Wenn Person A einen Zwilling Person B hat, dann ist auch Person B der Zwilling von Person A. Transitivität drückt aus, dass wenn A zu B und B zu C in Bezug stehen, es auch eine Beziehung zwischen A und C gibt. Dies lässt sich am Besten durch Verwandtschaftsverhältnisse erklären. Ist Person A mit Person B, und Person B mit Person C verwandt, dann steht auch Person A mit Person C in einem Verwandtschaftsverhältnis. Mit einer funktionalen Beziehung wird ausgesagt, dass wenn A zu B und A zu C in Relation stehen, B und C identisch sind. Wenn eine Person mit Namen Maximilian einen Personalausweis A hat und eine Person mit Namen Max ebenfalls den Personalausweis A besitzt, dann sind Maximilian und Max ein und dieselbe Person. Bei einer inversen Eigenschaft existieren

Eigenschaften	RDFS	OWL	Sprachkonstrukte
Klassenbildung	zur Darstellung v. hierarchischen Beziehungen	Darstellung v. differenzierten hierarchischen Beziehungen	rdfs:Class, owl:Class, rdfs:subClassOf, rdfs:subPropertyOf, owl:ObjectProperty, owl:DatatypeProperty
Maschinenverarbeitbare Semantik	ja	ja	–
Inferenz	ja	ja	–
Symmetrie	nein	ja	owl:SymmetricProperty
Transitivität	nein	ja	owl:TransitiveProperty
Funktionalität	nein	ja	owl:FunctionalProperty
Inverse	nein	ja	owl:inverseOf
Kardinalitäten	nein	ja	owl:cardinality, owl:maxCardinality, owl:minCardinality
Äquivalenz	nein	ja	owl:equivalentClass, owl:equivalentProperty
Disjunktion	nein	ja	owl:unionOf
Konjunktion	nein	ja	owl:intersectionOf
Negation	nein	ja	owl:complementOf

Tabelle 1: OWL vs. RDFS



entgegengesetzte Beziehungen. `Gemessen` und `gemessenVon` ist so eine inverse Relation. Zu den Mengenoperationen zählen die Gesamtmenge, die Schnittmenge und die Ergänzungsmenge. OWL erkennt eine Ressource als ein und dieselbe an, auch wenn sie in zwei Tripeln mit unterschiedlichen Namen benannt ist. Das geht mit RDFS nicht, denn dort muss Namensgleichheit bestehen, damit Ressourcen als identisch erkannt werden [35].

### 4.3 Zugrunde liegende Logikformalismen

Bereits Aristoteles befasste sich mit dem Wesen der Argumentation und dem logischen Schließen. Im Laufe der Zeit haben sich verschiedene Betrachtungsweisen durchgesetzt. Logik kann philosophisch, linguistisch, mathematisch oder im informatischen Sinne betrachtet werden. Allgemein formuliert beschäftigt sich die Lehre der Logik mit dem Erkennen und Erfassen von Mustern der realen Welt, um die dargestellten Zusammenhänge in Gesetzmäßigkeiten abzubilden. In Formeln abgebildetes Wissen ermöglicht mit Hilfe geeigneter Schlussregeln das Ableiten von neuem Wissen [38]. Eine zu modellierende Ontologie ist hauptsächlich durch formale Logik definiert, um das menschliche Denken und Schlussfolgern beschreiben zu können. Sprachkonstrukte dieser Logik beschreiben einen konkret definierten Zustand der realen Welt. Aus einer bestehenden Formel, die einen Zustand beschreibt, kann eine syntaktische neue Formel abgeleitet werden [37, Seiten 36, 37].

Logik und Ontologien haben eine Gemeinsamkeit. Sie beschäftigen sich beide mit der formalisierten Beschreibung der Welt. Ontologien konzentrieren sich dabei auf die inhaltliche Komponente. Dagegen stellt die formale Logik die Sprache zur Abstraktion von realen Gegebenheiten bereit. Schlussfolgerungsregeln ermöglichen das Ableiten von implizitem Ontologiewissen über die abgebildeten Inhalte. Darüber hinaus kann durch Logik geprüft werden, ob ein bestimmter Zustand der Realität dem Wissen der Ontologie entspricht.

Logikbasierte Ontologien eignen sich dazu komplexere Zusammenhänge zu modellieren. Um Logik in geeigneter Form abbilden zu können, bietet sich die Prädikatenlogik der ersten Stufe an. Sie bedient sich der logischen Operatoren, auch Junktoren genannt, und der Beschreibungslogik *Attributive Language with Complement (ALC)*. Zu den Junktoren gehören unter anderem die Negation, die Konjunktion, die Disjunktion oder die Äquivalenz. Zusätzlich stehen hier Quantoren als Ausdrucksmittel zur Verfügung. Der Existenzquantor ( $\exists$ ) bedeutet in natürlicher Sprache, dass mindestens ein Objekt einer Klasse vorhanden sein muss. Beim Allquantor ( $\forall$ ) wird ausgedrückt, dass eine bestimmte Aussage für jedes Element einer Klasse zutrifft. Logische Konstanten stellen Objekte und konkrete Werte dar. Dagegen repräsentieren logische Prädikate Kategorien und Relationen. Kategorien werden durch einstellige und Relationen durch zweistellige Prädikate dargestellt. Die Zugehörigkeit zu einer Klasse wird durch die Anwendung eines einstelligen Prädikates ausgedrückt. Das könnte z.B.  $(\text{Produkttyp}(\text{CH-AI-1-HR}))$  so aussehen, dass das Element CH-AI-1-HR der Kategorie Produkttyp zugewiesen wird. Eine Implikation, repräsentiert durch dieses Symbol  $\rightarrow$ , stellt eine Subklassenbeziehung zwischen einstelligen Prädikaten dar [37, Seiten 37-53]. Im Anhang E Zeichen der Prädikatenlogik befindet sich eine Zusammenfassung der gängigsten Zeichen der Prädikatenlogik.

OWL DL entspricht der sehr ausdrucksstarken aber dennoch entscheidbaren Beschreibungslogik SHOIN(D). Es wird eine Typentrennung verlangt, d.h. es muss klar definiert sein, was eine Klasse, ein Individuum oder eine Property ist. Das jeweilige Objekt kann nicht gleichzeitig zwei Typen bedienen. Die einzelnen Buchstaben von SHOIN(D) stehen für:

S = ALC plus Rollentransitivität

H = Unterrollenbeziehung

O = abgeschlossene Klassen

I = inverse Rollen

N = Zahlenrestriktionen und

D = Datentypen.

Darüber hinaus lassen sich in SHOIN(D) Gleichheit und Ungleichheit zwischen zwei Individuen ausdrücken. Ebenso sind transitive und funktionale Rollen darstellbar [31, Seiten 164-172]. Die Description Logic wird beispielsweise von der Entwicklungssoftware Protégé und dem Reasoner Pellet unterstützt (siehe Kapitel 5 Verwendete Software).

In OWL Full dürfen alle OWL- und RDFS-Sprachkonstrukte uneingeschränkt genutzt werden. Dadurch eröffnet OWL Full einen hohen Grad an Modellierungsfreiheit. Durch die hier gestattete Metamodellierung können Klassen einer Ontologie auch als Individuen verwendet werden. Genau diese Metamodellierung macht OWL Full unentscheidbar. Deshalb gibt es nur eine bedingte Unterstützung von OWL Full bei aktuellen OWL-Inferenzmaschinen [31, Seiten 151, 152].

## 5 Verwendete Software

Im Rahmen dieser Arbeit kommen verschiedene Entwicklungswerkzeuge für die Analyse und Darstellung des bereits vorhandenen Metadatenbestands sowie für die praktische Modellierung der Ontologie zum Einsatz. Bevor diese Tools kurz beschrieben werden, ist eine Erläuterung der verschiedenen Begriffe und deren Bedeutungen notwendig (siehe Tabelle 2).

### 5.1 Begriffsklärung

Objekte der realen Welt, ihre Kategorien und die charakteristischen Eigenschaften werden in verschiedenen Disziplinen und Abbildungssystemen unterschiedlich bezeichnet. Das führt unter Umständen zu Verwirrungen. Zum Verständnis folgt deshalb eine Einführung und Einordnung der verschiedenen Bezeichnungen. Sogenannte Kategorien bündeln eine Gruppe von Objekten der realen Welt und ermöglichen so eine Zuordnung. Der Begriff Konzept ist unter anderem bei der künstlichen Intelligenz und der Begriff Klasse ist in der Informatik anzutreffen. Kategorie, Konzept oder aber Klasse stellen dieselben Inhalte zur Verfügung. Objekte der realen Welt können zum einen durch Entitäten, Instanzen oder aber durch Elemente repräsentiert werden. Der Begriff Entität stammt hierbei von den Categoriesystemen der Philosophie ab. In der Informatik und der formalen Logik zeigen Instanzen die Zuordnung eines Objektes zu einer bestimmten Kategorie an. Elemente werden als eine Menge von Objekten einer Kategorie aufgefasst. Charakteristische Eigenschaften von Objekten werden durch die Begriffe Eigenschaften, Attribute, Rollen, Relationen, Slots oder Properties dargestellt. Slots finden vorwiegend bei Beschreibungslogiken und semantischen Netzen Anwendung. Der Begriff Rollen entstammt der deskriptiven Logik (DL). UML bezeichnet Eigenschaften als Relationen. Das Semantic Web spricht vornehmlich von Properties, die dann wiederum in Objekt- und Datatype-Properties gesplittet sind. [37, Seiten 23, 24] Objekt-Properties verbinden zwei Individuen miteinander. Datatype-Properties stellen dagegen eine Verbindung zwischen einem Individuum und einem Literal her. Ein Literal weist einen Wert, z.B. Zahlen oder Texte, direkt zu.

Objekte der realen Welt	Kategorien	Charakteristische Eigenschaften von Objekten
Entitäten	Konzepte	Eigenschaften
Instanzen	Klassen	Attribute
Elemente		Slots
		Properties
		Rollen
		Relationen

Tabelle 2: Festlegung von Begriffen

In der folgenden Tabelle 3 sind die Bezeichnungen für COE und Protégé aufgelistet. Beide Softwareprodukte nutzen Klassen und Individuen. COE beschreibt Beziehungen durch Objekt- und durch Datatype-Properties. In Protégé werden Relationen im frame-basierten Ansatz durch Slots und im OWL Plugin durch Properties repräsentiert. In beiden Fällen findet eine Unterscheidung zwischen Objekt- und Datatype-Properties statt. Mit beiden Systemen können Beziehungen zwischen Individuen und Klassen sowie Klassenhierarchien abgebildet werden. Protégé ermöglicht es darüber hinaus, Super- und Subklassenbeziehungen zu beschreiben. Beide Anwendungen sind dafür ausgelegt, differenzierte Property-Beziehungen mit ihren Eigenschaften darzustellen. Dazu zählen unter anderem Domain und Range, symmetrische, transitive oder inverse Beziehungen. Obwohl COE auch alle Funktionalitäten dazu bieten sollte, können im vorliegenden Fall nicht alle Möglichkeiten aufgrund von fehlender oder fehlerhafter Funktionalität ausgedrückt werden. Es kann z.B. vorkommen, dass die inverse Beziehung nicht abbildbar ist. Der Fehler ist bekannt und in der Anleitung dokumentiert. Bereits COE bietet die Möglichkeit Kardinalitäten anzubringen. Sollen darüber hinaus Quantoren zur Spezifizierung der Relation verwendet werden, ist die Nutzung von Protégé notwendig.

<b>COE</b>	<b>Protégé</b>
URIs und Namensräume	URIs und Namensräume
Klassen	Klassen oder Konzepte
Individuen	Individuen
Literale	
Objekt-Property	Properties/Slots: Objekt-Property
Datatype-Property	Properties/Slots: Datatype-Property
	Annotation-Property
Beziehung zwischen Individuum u. Klasse	Beziehung zwischen Individuum u. Klasse
Aufbau einer Klassenhierarchie	Aufbau einer Klassenhierarchie (Super-Subklassen-Hierarchie)
Darstellung v. differenzierten Property-Beziehungen	Darstellung v. differenzierten Property-Beziehungen
Restriktionen: Kardinalitäten	Restriktionen: Kardinalitäten, Quantifier-Restriktion (Gewichtung)

Tabelle 3: Begriffe

## 5.2 Cmap Tools Ontology Editor

Der Cmap Tools Ontology Editor (COE) ist eine Entwicklung des Institute For Human And Machine Cognition und basiert auf Cmap Tools, einem Werkzeug zur reinen Concept Map-Generierung. In Concept Maps wird Domänen-Wissen grafisch in Form von Strukturdiagrammen modelliert und abgebildet. Alle Funktionen von Cmap Tools sind in COE enthalten. In COE kommen zusätzliche Ausdrucksmöglichkeiten für semantische Zusammenhänge hinzu.

Den grundlegenden Aufbau einer Concept Map zeigt Abbildung 5 auf. Eine Concept Map ist ein Graph, der sich aus verschiedenen Knoten und Verbindungslinien zusammensetzt. Ein Knoten stellt ein Konzept dar, welches eine Klasse, ein Individuum oder ein Literal sein kann. Die Verbindungspfeile zeigen die Beziehung zweier Knoten an. Zur Erläuterung der einzelnen Beziehungen erhalten die Verbindungslinien Beschriftungen.

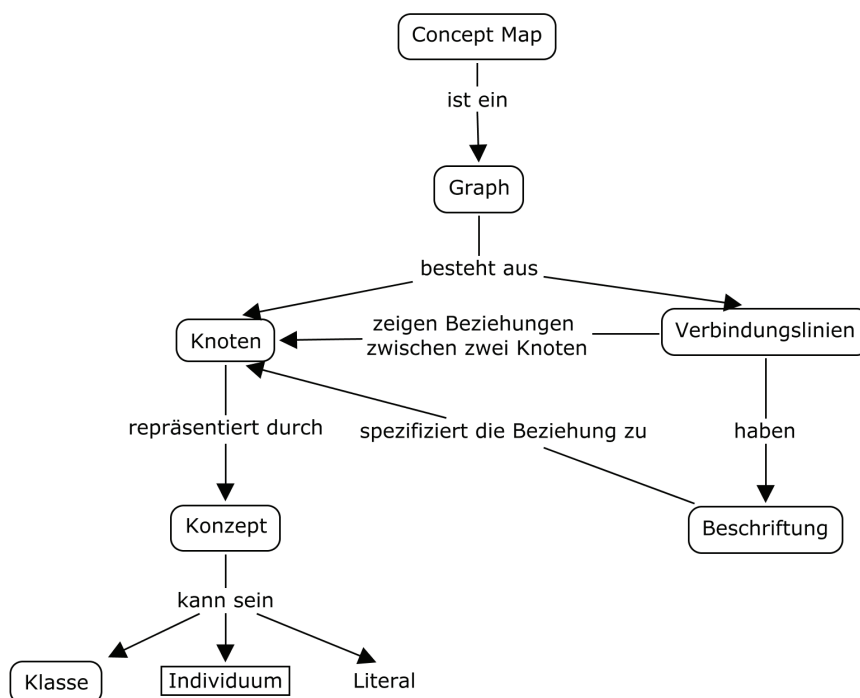


Abbildung 5: Concept Map

Vor der Erstellung einer Concept Map sollte mindestens ein Anwendungsfall definiert sein. Anhand dieses Falles kann immer wieder die Korrektheit der Modellierung überprüft werden. Beim Entwurf ist darauf zu achten, dass hierarchische Strukturen einzuhalten sind, d.h. allgemeinere Konzepte befinden sich an der Spitze und die spezifischeren Konzepte darunter. Die modellierten Zusammenhänge können unter Umständen unentdeckte Querverbindungen aufzeigen, was zu neuen Erkenntnissen bzw. zur Aufdeckung von Widersprüchen führen kann [40]. Nach Abschluss der Modellierungsarbeiten bietet COE die Möglichkeit des Exports. Zur Auswahl stehen u. a. die Formate OWL, N3-Triple oder TURTLE.

Die mit dem Cmap Tools Ontology Editor produzierbaren Wissenszusammenhänge entsprechen einem Semantischen Netz. Denn auch sie verwenden Graphen zur Beschreibung von Begriffen und deren Relationen. Eine graphenbasierende Informationsdarstellung hat den Vorteil, dass sich Datenstrukturen gut darstellen lassen. Außerdem sind Algorithmen bekannt, die diese Graphen verarbeiten können. Knoten stellen Begriffe dar. Relationen geben Auskunft über die Beziehungsstrukturen [37, Seiten 28-35]. Ein Beispiel für ein semantisches Netz, erzeugt durch COE, zeigen die Abbildungen 5 und 12.

COE kam in dieser Arbeit vorwiegend zur visuellen Erfassung des bereits existierenden Metadatenbestandes zum Einsatz. Anhand der erarbeiteten Grafik konnten fehlende Elemente der Dokumente erkannt und ergänzt werden (siehe Punkt 6.3). Die spätere Modellierung der Ontologie erfolgte mit dem Tool Protégé, das weit mächtiger als COE ist.



### 5.3 Protégé

Protégé wurde vom Stanford Center for Biomedical Informatics Research der Stanford University entwickelt. Das Open-Source-Tool beinhaltet einen Editor, der in der Informatik zur Modellierung von Ontologien genutzt werden kann. Die Umsetzung der Software erfolgte in Java, d.h. dass eine Plattformunabhängigkeit gegeben ist. Protégé ermöglicht die Generierung einer formalen und maschinenlesbaren Wissensrepräsentation. Die Software unterstützt die Modellierung von Strukturen eines bestimmten Wissensgebietes, die Erstellung und Visualisierung, sowie die Manipulation von Ontologien. Je nach Modellierungsaufgabe können verschiedene Plugins als Erweiterungen aufgenommen werden. Folglich sind individuelle Anpassungen möglich.

Zur Wissensmodellierung stehen grundsätzlich zwei Arten zur Verfügung. Diese sind der ursprüngliche frame-basierte Ansatz und das W3C-konforme Protégé-OWL Plugin. Protégé-frames bildet Domänenwissen in einer hierarchischen Struktur ab und nutzt dafür Klassen, Slots und Individuen. Protégé-OWL beruht auf dem Semantic Web-Standard OWL. Der entscheidende Unterschied liegt hier in zusätzlichen Logikmechanismen, die dafür konzipiert sind, Wissen aus mehreren Ontologien von unterschiedlichen Quellen für das Schlussfolgern zu bündeln. [34]

Für die zu erstellende ISDC-Ontologie wird die Protégé-Version 3.4.4 in Verbindung mit dem Protégé-OWL Ansatz genutzt. Hier wird die Einbindung von RDF- und RDFS-Dateien unterstützt. Die Entscheidung fiel deshalb auf die Version 3.4.4, weil bereits existierende Beispiele aus der Praxis (siehe Punkt 3.2) mit eingebunden bzw. bei der Modellierung berücksichtigt werden sollen. Diese externen Quellen beruhen unter anderem auf RDF und RDFS. Darüber hinaus existiert für diese Version eine Vielzahl von Tools in Form von Plugins, die von einer großen Nutzergemeinschaft gepflegt werden. Außerdem stehen umfangreiche Funktionen für die Semantic Web Rule Language (SWRL) zur Verfügung. Abschließend ist anzumerken, dass sich diese Version durch seine Stabilität auszeichnet, was ebenfalls für Protégé 3.4.4 spricht [33].

### 5.3.1 Grundlegender Aufbau des OWL Plugins

Das OWL Plugin setzt auf das ursprüngliche frame-basierte Protégé auf. Grundsätzlich wird zwischen dem internen Modell, also der internen Repräsentation der Ontologie oder der Wissensbasis, und der Ansicht, die der Nutzer sieht unterschieden. Das interne System basiert auf einem objektorientierten Metamodell. Hier werden Klassen, Properties bzw. Slots, Facetten und Instanzen verwaltet. Eine Facette ist eine Eigenschaft, die Properties beschreibt. Das Metamodell selbst ist eine Protégé-Ontologie. Über das Java Application Programming Interface (API) können Anfragen und Änderungen am Model vorgenommen werden. Die Benutzeroberfläche bietet dem Nutzer die Möglichkeit, das Model zu entwickeln und zu visualisieren. Dazu können Klassen, Properties, Restriktionen und Individuen genutzt werden. Dafür stehen jeweils eigene Editoren als Tabs zur Verfügung. Mit dem Forms-Editor kann der Nutzer individuelle Einstellungen für die Ausgabe gewünschter Inhalte vornehmen. Generell bietet Protégé die Möglichkeit verschiedene Formate, wie z.B. RDF, XML oder relationale Datenbanken zu laden, zu editieren und zu speichern. Protégé ist hoch skalierbar, d.h. es können Ontologien mit tausenden Elementen erstellt werden. Es existiert ebenfalls eine Java-API für OWL-Ontologien. Dafür wurde die Kern-API speziell für die Bedürfnisse von OWL erweitert.

Wird ein neues Projekt geöffnet, sind standardmäßig fünf Tabs sichtbar. Dazu zählen der Metadata-Tab, der OWL Classes-Tab, der Properties-Tab, der Individuals-Tab und der Forms-Tab. Jeder Tab steht für verschiedene Aspekte der Ontologie in unterschiedlichen Sichten. Die Namensgebung der Tabs gibt Aufschluss darüber, ob Klassen, Individuen oder Properties verwaltet werden können. Unter dem Metadata-Tab können Namensräume und Importe verwaltet und gesetzt werden. Abbildung 6 zeigt, wie sich ein neues Projekt standardmäßig öffnet.[13] Die meisten einzubindenden Tabs enthalten auf der linken Seite eine Baumstruktur, die das Modell repräsentiert. Auf der rechten Seite können Details des selektierten Elementes abgelesen und manipuliert werden. Die Eingabe erfolgt über sogenannte Widgets, welche im Forms-Editor in Größe und in Platzierung angepasst werden können. Werden die standardmäßigen Widgets nicht benötigt, besteht die Möglichkeit sie

auszuschalten. Im oberen Bereich können Metadaten gesetzt werden. Dazu gehören unter anderem der Name des Objektes sowie Kommentare. Im darunter liegenden Widget können Annotations-Properties angebracht werden. Bei den Annotationen wird auch die Nutzung von anderen Ontologien, z.B. Dublin Core, ermöglicht, denn hier können externe Ressourcen aufgerufen und mit Daten belegt werden.[14]

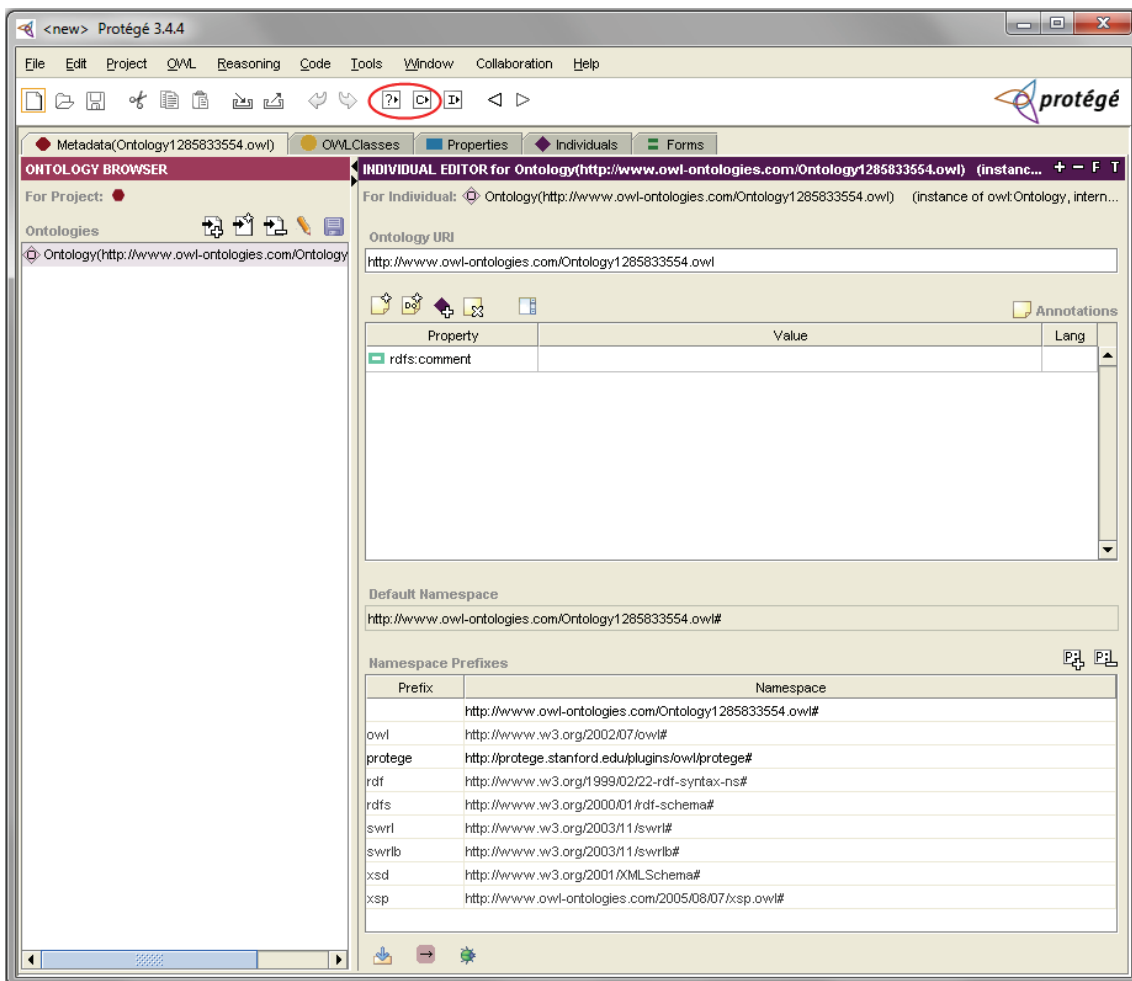


Abbildung 6: Metadata-Tab

Eingegebene Inhalte können mittels eines Reasoners, der in Protégé integrierbar ist, ausgewertet werden. Dazu stehen verschiedene Möglichkeiten zur Verfügung, die über die rot eingekreisten Button der Abbildung 6 zu erreichen sind. Der erste Button führt einen Konsistenzcheck durch, d.h. es wird getestet, ob jeder Klasse Individuen zugewiesen werden können. Dieser Schritt kann entweder für alle Klassen oder eine ausgewählte Klasse durchgeführt werden. Die Klassifikation wird mit dem zweiten Button angestoßen. Durch die Ausführung des Klassifikationsmechanismus wird ein geschlussfolgert Baum, beruhend auf den angelegten Klassendefinitionen, erstellt. Vor der Klassifikation wird die der Ontologie zugrunde liegende OWL-Sprache ermittelt. Sollten Sprachkonstrukte von OWL Full verwendet worden sein, wird die Ontologie temporär in OWL DL konvertiert. Danach kann der Konsistenzcheck und die anschließende Klassifikation durchgeführt werden. Die Ausgabe der geschlussfolgerten Baumstruktur erfolgt in einem separaten Fenster. Das OWL Plugin unterstützt einige Features von OWL Full. Neben der Protégé-Datei (ontology.pprj) wird zusätzlich eine OWL-Datei zum Projekt angelegt. In dieser Datei ist eine Zusammenfassung aller vom Nutzer getätigten Angaben gespeichert.[13]

### 5.4 Pellet

Pellet ist ein in Java implementierter Open Source OWL Reasoner. Er kann mit der OWL Lite- und der OWL DL-Syntax arbeiten, also Sprachen die auf der Description Logic Shoin (D) basieren. OWL DL kann komplett interpretiert und ausgewertet werden. Die nächste OWL-Entwicklungsstufe OWL Full ist dagegen nicht vollständig prüfbar. Pellet stellt Funktionen für die Konsistenzüberprüfung, die Überprüfung der Erfüllbarkeit der Aussagen, sowie Klassifikationsdienste zur Verfügung. Um diese bereitstellen zu können, hat der Reasoner besondere Anforderungen zu erfüllen [10]. Zum einen muss sichergestellt sein, dass alle geschlussfolgerten Aussagen korrekt sind. Das heißt die Klassifikation, also die Darstellung der gesamten Klassenhierarchie, muss vollständig sein. Zum anderen sollte in endlicher Zeit vom Reasoner feststellbar sein, ob eine Aussage gültig ist. Von der gewählten Sprache hängt die Erfüllbarkeit dieser Anforderungen ab. Je einfacher die Repräsentationssprache gebaut ist, desto schneller und effizienter kann ein Reasoner arbeiten. Ein Reasoner

sollte mit der wachsenden Informationsflut des Semantic Webs umgehen können, d.h. auch bei einer großen Wissensbasis sollte eine effiziente Arbeitsweise garantiert sein. Die ständigen Änderungen im Web und die dort eventuell fehlerhaft vorhandenen Informationen stellen ebenfalls eine große Herausforderung an den Reasoner dar. Deshalb muss er eine gewisse Robustheit aufweisen, um mit fehlerhaften Daten, wie z.B. widersprüchlichen Formeln oder Aussagen, umgehen zu können. Ferner ist die Integration von verteiltem Wissen zu unterstützen, d.h. Wissen aus unterschiedlichen Ontologien muss kombinierbar sein [11].

Abbildung 7 zeigt die grundlegenden Komponenten des Pellet Reasoners. Zunächst wird überprüft, ob die Ontologie OWL DL oder OWL Full angehört. Danach folgt die Konvertierung und Ladung der Ontologieaussagen mittels eines Parsers. Die abgebildeten Inhalte gehören nun der Wissensbasis an. Pellet selbst besitzt keinen eigenen RDF/OWL-Parser. Aber es besteht die Möglichkeit andere Softwaretools über eine Schnittstelle einzubinden. Dabei müssen die unterschiedlichen Strukturen der Systeme, z.B. die Triple-orientierte Sicht, erkannt und berücksichtigt werden. In der T-Box befinden sich Aussagen über Klassen und deren Eigenschaften. Hier können allgemeingültige Beziehungen, die die Klassen betreffen, ausgedrückt werden. Die A-Box enthält direkte Aussagen über Individuen der jeweiligen Klassen, also den konkreten Datenbestand. Aussagen der T-Box und der A-Box bilden die Wissensbasis. Der Tableaux-Reasoner hat die Funktion der Konsistenzüberprüfung. Eine Ontologie ist konsistent, wenn alle Regeln des Tableaux-Reasonings beachtet und erfüllt wurden. Der XSD-Reasoner überprüft die Korrektheit der Datentypen, da im XML Schema schon viele standardmäßige Datentypen definiert sind, die auch in OWL-Dokumenten zum Einsatz kommen. Die Schnittstelle der Wissensbasis führt einen Konsistenzcheck der A-Box aus, d.h. es wird überprüft, ob alle Individuen zu Klassen zugehörig sind. Außerdem wird die Beziehung der Individuen zu den Klassen auf Konsistenz geprüft. Intern erfolgt der Daten- und Wissensabgleich durch eine baumartige Struktur. Die Abläufe darüber bleiben dem Nutzer verborgen. Die A-Box Query Engine unterstützt beispielsweise Abfragen der Sprache SPARQL Protocol and RDF Query Language (SPARQL). [10]

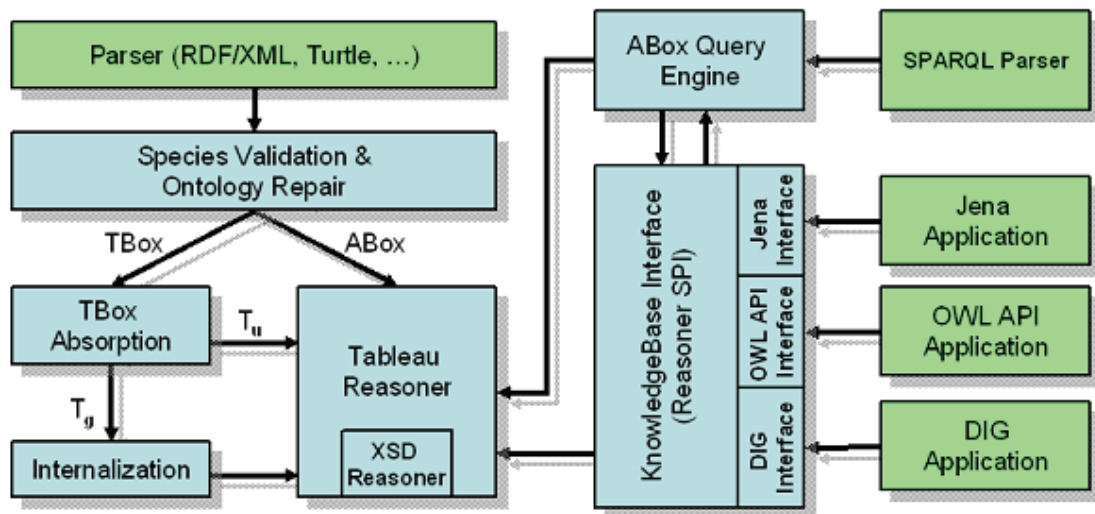


Abbildung 7: Hauptkomponenten des Pellet Reasoners [32]

Protégé unterstützt verschiedene Reasoner, unter anderem auch den Pellet Reasoner. Er ist bereits Bestandteil des Protégé-Frameworks und muss lediglich ausgewählt werden. Für die Nutzung von Pellet spricht, dass er in vielen Projekten bereits erfolgreich zum Einsatz kommt. Deshalb wird er auch im Rahmen dieser Arbeit verwendet.

## 5.5 XMLSpy

Das Werkzeug XMLSpy der Firma Altova ist eine integrierte Entwicklungsumgebung für XML-basierte Systeme. Unter anderem bietet XMLSpy die Möglichkeit, XML-Schemata zu erstellen bzw. anzupassen. Funktionen, wie z.B. die Validierung von Dokumenten, die Kenntlichmachung von Fehlern und die verschiedenen Ansichten von der Dokumentstruktur und den Inhalten im Editor, erleichtern die Arbeit von Entwicklern [3].

## 6 Untersuchung der Informationsquellen

In diesem Kapitel wird auf Informationsquellen im Allgemeinen und speziell auf die zur Verfügung stehenden Informationen des ISDCs eingegangen, die zur Beantwortung von wissenschaftlichen Fragestellungen genutzt werden können. Insbesondere werden die einzelnen ISDC-Metadatendokumente analysiert und den Anforderungen entsprechend angepasst.

### 6.1 Externe Quellen

Neben dem kontrollierten Wortgut vom GCMD, was der Beschreibung von geowissenschaftlichen Daten dient, existiert auch anderes Wortgut, das bereits zum Einsatz kommt. Beispielsweise findet das NERC Data Grid (NDG) beim SeaDataNet Anwendung. Das kontrollierte Wortgut dazu wird vom British Oceanographic Data Centre (BODC) bereitgestellt. Es ist speziell für den maritimen Bereich entwickelt worden [6]. Kontrolliertes Wortgut ist als eine vertrauenswürdige Quelle anzusehen, mit der es möglich ist, Inhalte einer speziellen Domäne zu charakterisieren und zu beschreiben. Es bietet eine Sammlung von Bezeichnern, denen wiederum eindeutige Begriffe einschließlich Interpretationen zugeordnet sind.

In sogenannten Wikis wird Nutzerwissen online gesammelt, aufbereitet und für die Allgemeinheit in Form von verlinkten Artikeln zur Verfügung gestellt. Wikis sind aus der Web 2.0-Bewegung heraus entstanden und können eine wichtige Informationsquelle zur Beantwortung von Suchanfragen darstellen. Wikipedia ist der bekannteste Vertreter dieser Wissenssammlungen. Jeder kann als Autor tätig werden und Artikel veröffentlichen bzw. ergänzen. Die Anonymität der Autoren ist jedoch als problematisch anzusehen, da für veröffentlichte Inhalte keine Verantwortung übernommen werden muss. Nutzer die sich der Inhalte von Wikipedia bedienen, müssen die Informationen kritisch hinterfragen, um sich kein falsches Wissen anzueignen. Um dies zu vermeiden, gingen die Entwicklungen weiter zu Citizendium, einem möglichen Nachfolger von Wikipedia. Citizendium erlaubt keine

anonymen Beiträge, d.h. der Autor muss für veröffentlichte Inhalte bürgen. Fachlich kompetente Lektoren, die die Autoren betreuen, sollen ein hohes Niveau und die Richtigkeit der Inhalte gewährleisten [7].

Die freie Verschlagwortung des Web 2.0 stellt ebenfalls viele Informationen für Suchanfragen zur Verfügung. Internetnutzer können eigenes Wissen und Erfahrungen auf Internetseiten zugänglich machen, z.B. in Form von Produktbewertungen. Der potentielle Käufer kann so beispielsweise in Erfahrung bringen, wie andere Käufer ein Produkt bewerten. So kann die kollektive Intelligenz aller Internetnutzer für eine breite Masse zur Verfügung gestellt werden. Als Nachteil ist der eventuelle Qualitätsverlust anzusehen, denn auch falsche und irreführende Informationen können so ins WWW eingespeist werden [42].

## 6.2 ISDC

Die vom ISDC-Portal bereitgestellten ISDC-Datenprodukte, kategorisiert nach Produkttypen, setzen sich aus den jeweiligen Datenfiles und den dazugehörigen Metadatenfiles zusammen und werden in relationalen und hierarchischen Datenstrukturen gespeichert. In den Metadatendokumenten zu den Produkttypen sind viele Informationen gespeichert und abrufbar, d.h. sie stellen eine wichtige semantische Informationsquelle dar. ISDC-Datenprodukte sind beim Portal für angemeldete Nutzer recherchier- und abrufbar. Weiterhin stellt das ISDC neben geowissenschaftlichen Datenprodukten auch weiterführende Informationen, in Form von Büchern, Artikeln, Postern oder PowerPoint Präsentationen, zur Verfügung. Beschreibende Schlagworte die zur Kategorisierung von wissenschaftlichen Artikeln genutzt werden, dienen der Vernetzung von bereits vorhandenem Wissen [32].

Bei der Entwicklung der ISDC-Metadatenklassen hat sich das ISDC-Team am Global Change Masterdirectory (GCMD) von der NASA orientiert. Hier wird nicht nur ein umfangreiches kontrolliertes geowissenschaftliches Wortgut, sondern auch eine Dokumentation für deren Verwendung angeboten [23]. Mit Hilfe des Directory Interchange Format (DIF)



Writer's Guide [17] wurde das Grundgerüst für den Produkttyp geschaffen. Der Produkttyp war die erste Metadatenklasse beim ISDC. Der DIF-Standard hat sich im Laufe der Zeit den wachsenden Anforderungen an Metadaten angepasst. Der heutige DIF-Standard basiert auf XML und bildet somit die Inhalte in XML-konformen Strukturen ab. Jeder Produkttyp wird durch ein ISDC-DIF-Metadatendokument, in früheren Arbeiten auch Parent-DIF genannt, beschrieben. Angaben die das Datenprodukt betreffen, befinden sich im Metadatendokument des Datenfiles (Data-Child-DIF). Das Schema der beiden Metadatendateien ist beim ISDC als `base-dif.xsd` hinterlegt.

Zusätzlich zum Produkttyp sollen zusätzliche Metadatenklassen für Instrumente, Projekte, Plattformen und Institutionen hinzukommen. Auch sie entstehen in erweiterter Form nach dem Vorbild des GCMD. Beschreibungen dazu sind dem Ancillary Description Writer's Guide [16] zu entnehmen. Einzige Ausnahme stellt das `Institution`-Dokument dar, welches das `Data Center`-Dokument des GCMD erweitert. Es enthält ausführlichere Angaben zu den Institutionen und deren Beschäftigten. Da diese Metadatendokumente lediglich in ihren Grundstrukturen vorhanden sind, werden im Analyseprozess in dieser Arbeit noch Anpassungen speziell für die Bedürfnisse des ISDCs vorgenommen. Die einzelnen bereits vorhandenen und neuen Elemente sind der folgenden Analyse zu entnehmen.

### 6.3 Analyse der ISDC-Metadatendokumente

Die Tabellen 5, 6, 7, 8 und 9 (siehe Anhang B) zeigen den Inhalt der verschiedenen ISDC-Klassen mit ihren zugehörigen Metadatenelementen. Da das ISDC Wissenschaftler auf der ganzen Welt mit Daten bedient, sind sämtliche Inhalte in Englisch verfasst. Die Neustrukturierung und die Erweiterung der Inhalte, also das Auslagern und insbesondere das Hinzufügen von Informationen in die neu geschaffenen Metadatenklassen, dienen einem größeren Informationsgehalt und einer besseren Übersichtlichkeit. Aus diesem Grunde wurden die Metadatenklassen Institution, Instrument, Projekt und Plattform definiert und modelliert. In einem ersten Analyseschritt wurden zunächst die einzelnen Inhalte erfasst, welche im Anhang B ISDC-Metadatenbestand zu finden sind.

Bevor der Analyseprozess weiter geführt werden kann, ist zuerst die Beschreibung der Inhalte der einzelnen Metadatenklassen zu definieren. Die Metadatenklasse Produkttyp besteht aus vielen einzelnen Elementen, die Detailinformationen zu den Daten liefern. Acht Elemente sind obligatorisch. Dazu gehören unter anderem Entry\_ID, Entry\_Title, Parameters, Summary oder Data Center. Bei den sehr empfohlenen Elementen sind beispielsweise Project, Quality, Platform, Instrument, Personnel und Location zu nennen. Empfehlenswerte Felder, wie z.B. Keyword oder Originating Center, runden den möglichen Informationsgehalt ab [17]. Die Institution enthält u.a. Informationen zur Entry\_ID, zum Namen, zu Plattformen, Instrumenten, Projekten, Produkttypen und zum Personal. Die Person, die einer Institution zugeordnet ist, wird mit ihrem Namen und den Kontaktdaten erfasst. Das Instrument enthält neben den standardmäßigen Einträgen, wie z.B. Entry\_ID, Name, Parameters, Keyword und Summary, zusätzliche instrumentenspezifische Informationen. Dazu zählen beispielsweise Instrument\_Identification, Instrument\_Logistics und Associated\_Platform. Die Plattform ist u.a. durch eine Entry\_ID, Name, Parameters, Keywords, Summary und Instrument charakterisiert. Hier sind alle Informationen enthalten, die zur Identifikation einer Plattform dienen. Das Projekt beinhaltet ebenfalls Entry\_ID, Name, Summary, Parameters oder Keywords. Darüber hinaus stehen Elemente zur Verfügung, um

die Beziehung zu den jeweils anderen Kategorien herzustellen. Jedes Metadatendokument enthält eine ID, die zur eindeutigen Identifizierung dient [16].

Nach der Bestandsaufnahme wurden in Absprache mit dem ISDC-Team fehlende Elemente in den Metadatendokumenten ergänzt, um semantische Abfragen realisieren zu können. Tabelle 4 zeigt alle Klassen des ISDCs. Soll eine Verbindung zwischen Klassen hergestellt werden, so muss ein Verweis auf die zugehörigen Metadatendokumente erfolgen. Beispielsweise muss Institution Elemente für Instrument, Plattform, Projekt und Produkttyp in das Schema integrieren, um solch eine Verbindung abbilden zu können.

	<b>Product type</b>	<b>Institution</b>	<b>Instrument</b>	<b>Project</b>	<b>Platform</b>
<b>Product type</b>		X	X	X	X
<b>Institution</b>	X		X	X	X
<b>Instrument</b>	X	X		X	X
<b>Project</b>	X	X	X		X
<b>Platform</b>	X	X	X	X	

Tabelle 4: Beziehungen der Metadatenkokumente

Außerdem war zu überprüfen, ob weitere notwendige Elemente für eine spätere semantische Abfrage fehlen. Da es sich hier um geowissenschaftliche Datenbestände handelt, müssen in jedem Fall räumliche sowie zeitliche Komponenten definiert und abrufbar sein. Weiterhin soll der Nutzer die Möglichkeit erhalten mit Hilfe von Schlüsselwörtern zu suchen. Diese können aus einem kontrollierten Wortgut oder aus einer freien Verschlagwortung stammen. Das kontrollierte Wortgut des ISDCs stammt vom GCMD und ist bereits teilweise implementiert. Ferner ist noch die Möglichkeit gegeben, freie Schlagwörter zu vergeben.

### 6.3.1 Analyse des existierenden Modells

Die noch zu tätigen Ergänzungen in den XML-Schemata-Dateien sind in Abbildung 8 dargestellt. Hier sind sowohl bereits bestehende als auch neue Inhalte enthalten, die zur semantischen Verknüpfung notwendig sind. Alle Elemente der Abbildung 8, die mit einem Pfeil versehen sind, besitzen noch Unterelemente. Im Anhang C Elemente der Klassen befinden sich die einzelnen Klassen der Abbildung 8 noch einmal in detaillierter Form. Hier sind die weiteren Unterelemente sichtbar. Bei der Nachmodellierung wurde auf Namensgleichheit der eingeführten neuen Elemente geachtet. Das wiederum erleichtert das Finden von Konkordanzen zwischen den einzelnen Metadatendokumenten. Klassen werden durch die blauen abgerundeten Vierecke dargestellt. Die Rechtecke symbolisieren die einzelnen Elemente. Bereits vorhandene Bestandteile werden in blau dargestellt. Neue noch fehlende Elemente sind gelb hinterlegt. Beim Produkttyp-Schema sind die wenigsten Ergänzungen notwendig, da hier viele Informationen schon vorhanden sind. Anders als beim GCMD soll das ISDC auch Daten über Personen und deren Aufgabe in der Institution anbieten. Die weiteren Metadatenklassen, wie Project, Platform und Instrument, bestanden im Grundgerüst schon. Sie wurden lediglich um fehlende Bestandteile ergänzt. Das Element Trust wird eingeführt, um zu einem späteren Zeitpunkt die Metadatendokumente um einen Vertrauenswert zu ergänzen. Der Trust-Wert soll Aussagen zur Qualität der Daten geben.

Am Beispiel Institution wird nun gezeigt, wo die Konkordanzen, also die Überschneidungspunkte, im Dokument zu finden sind. Die folgenden Gegenüberstellungen in Abbildung 9 stellen die erforderlichen Verbindungen zwischen Institution und den anderen Klassen her. Das Institution-Dokument stellt seine Entry\_ID als Verbindungselement zur Verfügung. Des Weiteren müssen in die Institution die IDs der anderen Metadatenklassen aufgenommen werden. Der Vollständigkeit halber wurden zusätzlich noch der Short\_Name und der Long\_Name ergänzt. Damit ist sichergestellt, dass die einzelnen Klassen bzw. deren Instanzen auch ohne Kenntnis der weiteren Dokumente gut lesbar sind. Für die anderen Klassen mussten diese Verknüpfungspunkte ebenfalls geschaffen werden. Die Abbildungen dazu sind im Anhang D Konkordanzen zu finden.

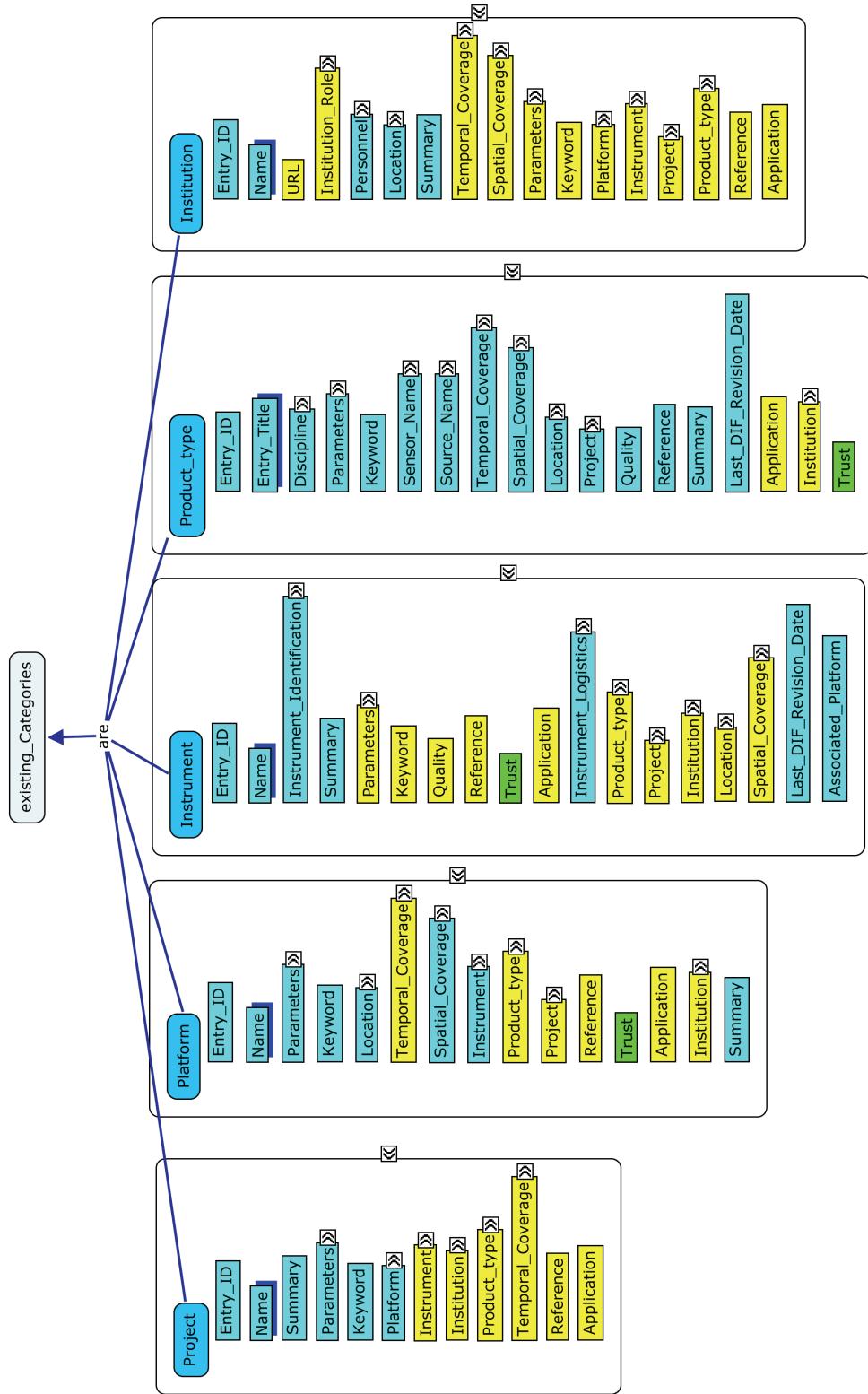


Abbildung 8: Existierende ISDC-Kategorien

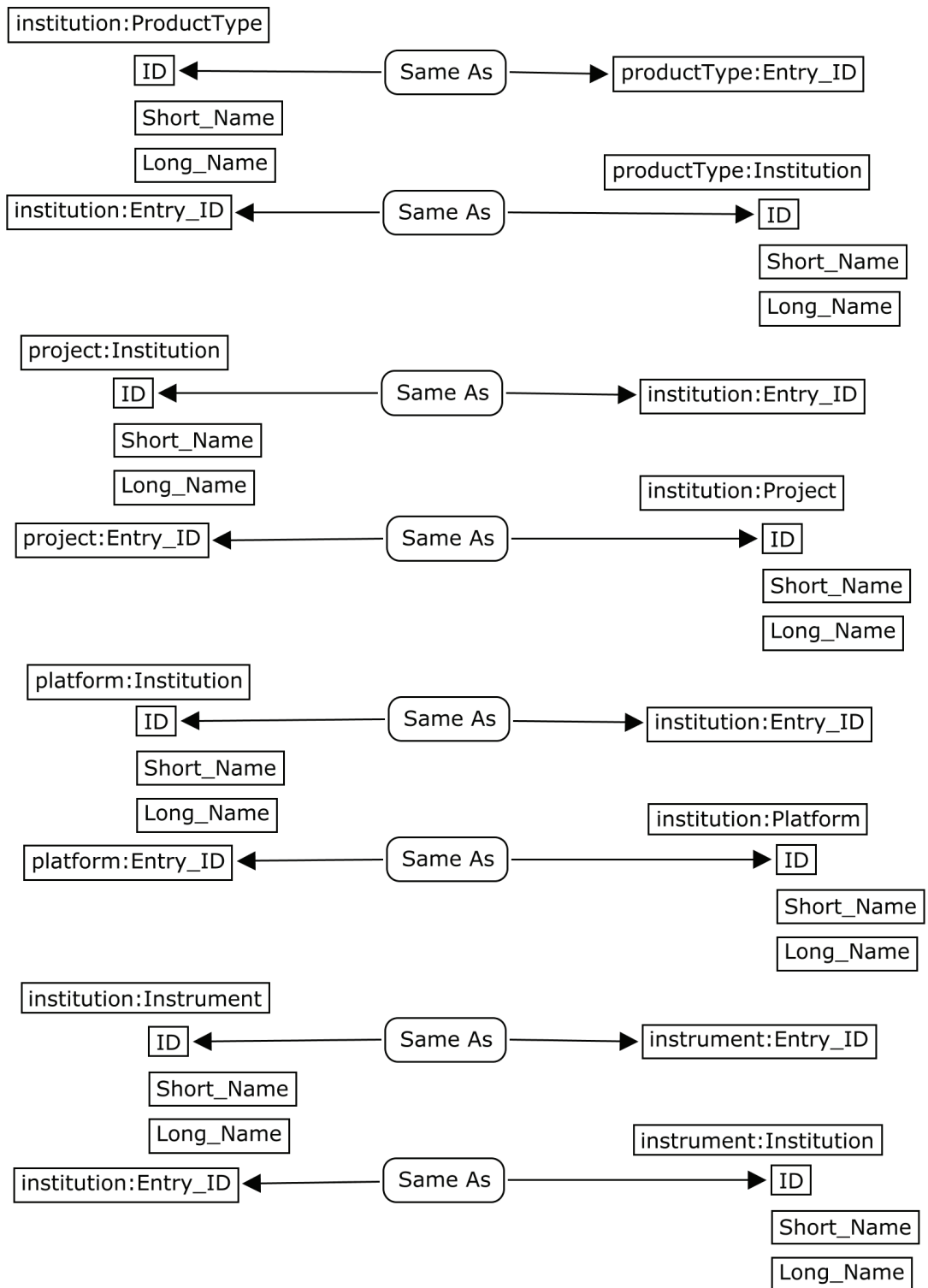


Abbildung 9: Konkordanz am Beispiel Institution

Da Suchanfragen auch Antworten zu Raum, Zeit und Schlagwörtern liefern sollen, werden noch drei Suchkategorien eingeführt. Die dazu benötigten Informationen stammen aus den Metadatenklassen, d.h. es sind keine weiteren Dokumente zu erstellen. Die Inhalte von `Temporal_Coverage` und `Instrument_Logistics` liefern die zeitliche Komponente. `Location` und `Spatial_Coverage` decken den 3D-Raumbezug ab. Die Schlagwörter werden durch `Keyword`, `Parameters` und `Discipline` in den Metadatenklassen repräsentiert (siehe Abbildung 10). `Temporal_Coverage` ist in `Projekt`, `Plattform`, `Produkttyp` und `Institution` zu finden. Ausnahme ist die Metadatenklasse `Instrument`. Hier wird die Zeit durch `Instrument_Logistics` abgebildet, was dem GCMD-Standard geschuldet ist. Im Metadokument `Projekt` wurde der Raumbezug nicht dargestellt, weil dies unnötig ist. Die Schlagwörter sind in allen fünf Metadatenklassen enthalten. Das kontrollierte Wortgut des GCMD kommt bei `Parameters` und `Discipline` zur Anwendung. Die Ergänzungen in den Schemata wurden mit dem Werkzeug XMLSpy (siehe Abschnitt 5.5) durchgeführt.

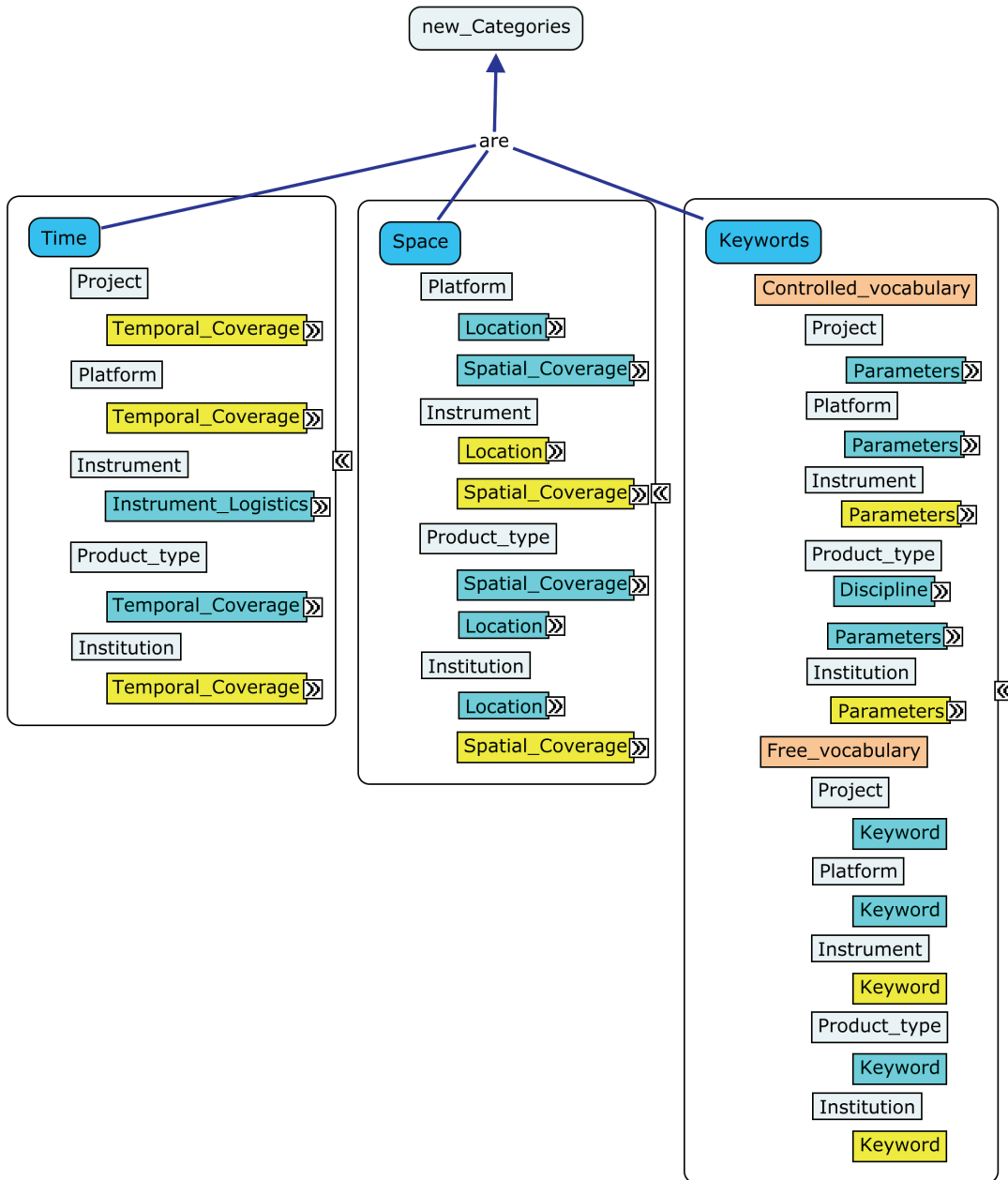


Abbildung 10: Neue ISDC-Kategorien



### 6.3.2 Zu beantwortende Anwendungsfälle

Die durchgeführten Änderungen an den ISDC-Metadatenklassen ermöglichen die geplanten semantischen Verknüpfungen im ISDC-Portal. Diese Weiterentwicklung ermöglicht das Auffinden und Präsentieren von relevanten geowissenschaftlichen Datenbeständen für die verschiedenen Nutzergruppen.

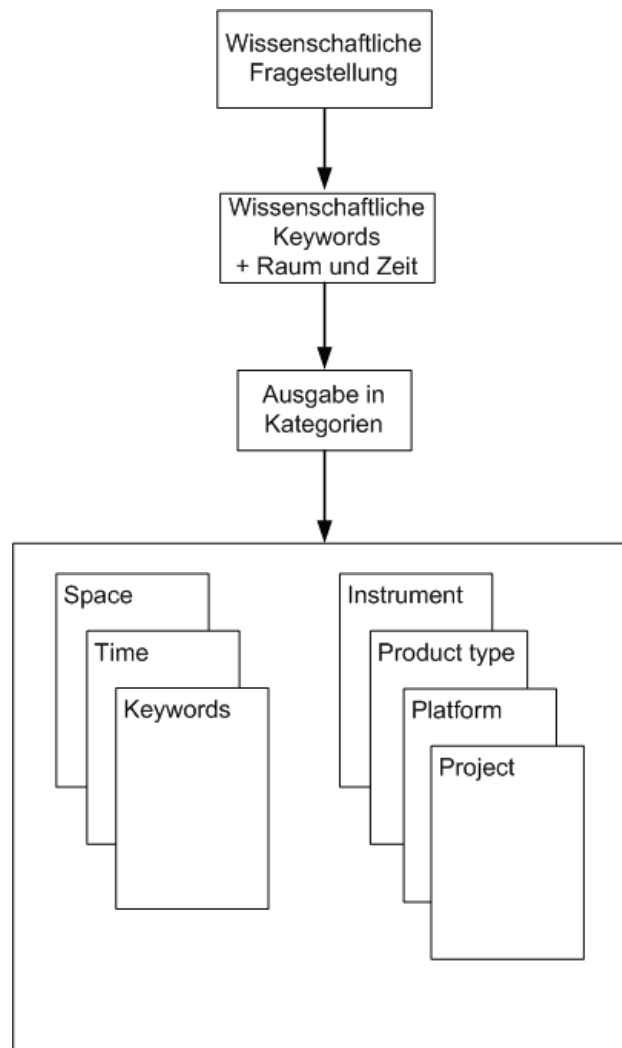


Abbildung 11: Anwendungsfall

Abbildung 11 zeigt den schematischen Ablauf für den allgemeinsten Anwendungsfall, der gestartet wird, wenn eine Suchanfrage gestellt ist. Zunächst einmal muss eine wissenschaft-

liche Fragestellung formuliert sein. Diese wird dann mit Hilfe wissenschaftlicher Keywords beantwortet. Dabei werden natürlich räumliche und zeitliche Aspekte berücksichtigt. Die Ergebnisausgabe der relevanten Informationen erfolgt innerhalb der Kategorien. Es wird zwischen den Kategorien Instrument, Produkttyp, Plattform, Projekt, Institution, Raum, Zeit und Keywords unterschieden. Für wissenschaftliche Fragestellungen ist der räumliche und der zeitliche Bezug von erheblicher Bedeutung, um abschätzen zu können, ob bestimmte Daten notwendig zur Beantwortung einer Anfrage sind. Nur so ist gewährleistet, dass eine gestellte Frage bestmöglich beantwortet wird. Die Quelle der Schlüsselwörter, die bei der Suche Anwendung finden, sind in Abbildung 8 und Abbildung 10 zu finden. In Abbildung 8 agieren nur die mit einem blauen Schatten hinterlegten Rechtecke als Schlüsselbegriffe. Bei Projekt, Plattform, Instrument und Institution ist das Keyword der Inhalt des Elements `Name` und beim Produkttyp ist das Keyword der `Entry_Title`. Dagegen enthält Abbildung 10 ausschließlich Keywords, die zur Verwendung zur Verfügung stehen. Sie setzen sich aus dem kontrollierten Wortgut des GCMD und der freien Verschlagwortung zusammen. Zum kontrollierten Wortgut zählen hier `Location`, `Parameters` und `Discipline`. Konkrete Anwendungsfälle (Competency Questions), die in dieser Arbeit mit der zu erstellenden ISDC-Ontologie beantwortet werden sollen, sind der folgenden Liste zu entnehmen.

- 1 Zeige alle Produkttypen an, die auch die Schlüsselwörter vom Geophänomen „Climate Change“ besitzen!
- 2 Zeige alle Personen an, die in Institutionen in den USA arbeiten und das GGP-Projekt betreuen!
- 3 Zeige alle Produkttypen an, die zum Projekt GRACE, zur Plattform GRACE-A, zur Institution GFZ und zum Instrument Star Camera System gehören!
- 4 Zeige alle Produkttypen, Projekte, Plattformen, Institutionen und Instrumente an, die mittels der Schlüsselwörter (Keywords) Satellite, Earth Science und Gravity beschrieben sind!
- 5 Zeige alle Produkttypen, Projekte, Plattformen, Institutionen und Instrumente an, die in Beziehung zum Geophänomen „Climate Change“ stehen!

### 6.3.3 Grafische Modellierung der Beziehungen der ISDC-Klassen

In Abbildung 12 sind die gerichteten Beziehungen der ISDC-Metadatenklassen untereinander dargestellt. Da die Beziehungen wechselseitig zwischen zwei Klassen sind, können diese zu einem späteren Zeitpunkt als invers deklariert werden. Hier werden sie noch getrennt aufgeführt, um die gegenläufigen Beziehungen explizit auszudrücken. Darüber hinaus sind in Abbildung 12 Restriktionen in Form von Kardinalitäten zu sehen. **Must be at least** drückt die Mindestanzahl eines Knotens aus. Die exakte Anzahl wird durch **must be exactly** dargestellt. Nachfolgend werden die dargestellten Beziehungen in Worten ausgedrückt:

- 1 Ein Produkttyp wird durch mindestens ein Projekt definiert.
- 2 Ein Projekt definiert mindestens einen Produkttyp.
- 3 Eine Institution erzeugt mindestens Null Produkttypen.
- 4 Ein Produkttyp stammt von mindestens einer Institution.
- 5 Eine Institution beaufsichtigt mindestens Null Projekte.
- 6 Ein Projekt wird durch mindestens eine Institution beaufsichtigt.
- 7 Eine Institution betreibt mindestens Null Instrumente.
- 8 Ein Instrument wird durch mindestens eine Institution betrieben.
- 9 Ein Instrument misst Daten für mindestens einen Produkttypen.
- 10 Ein Produkttyp wird von mindestens einem Instrument abgeleitet.
- 11 Eine Plattform wird für mindestens ein Projekt bereitgestellt.
- 12 Ein Projekt stellt mindestens eine Plattform bereit.
- 13 Eine Plattform trägt mindestens ein Instrument.
- 14 Ein Instrument wird von genau einer Plattform getragen.
- 15 Ein Projekt nutzt mindestens ein Instrument.

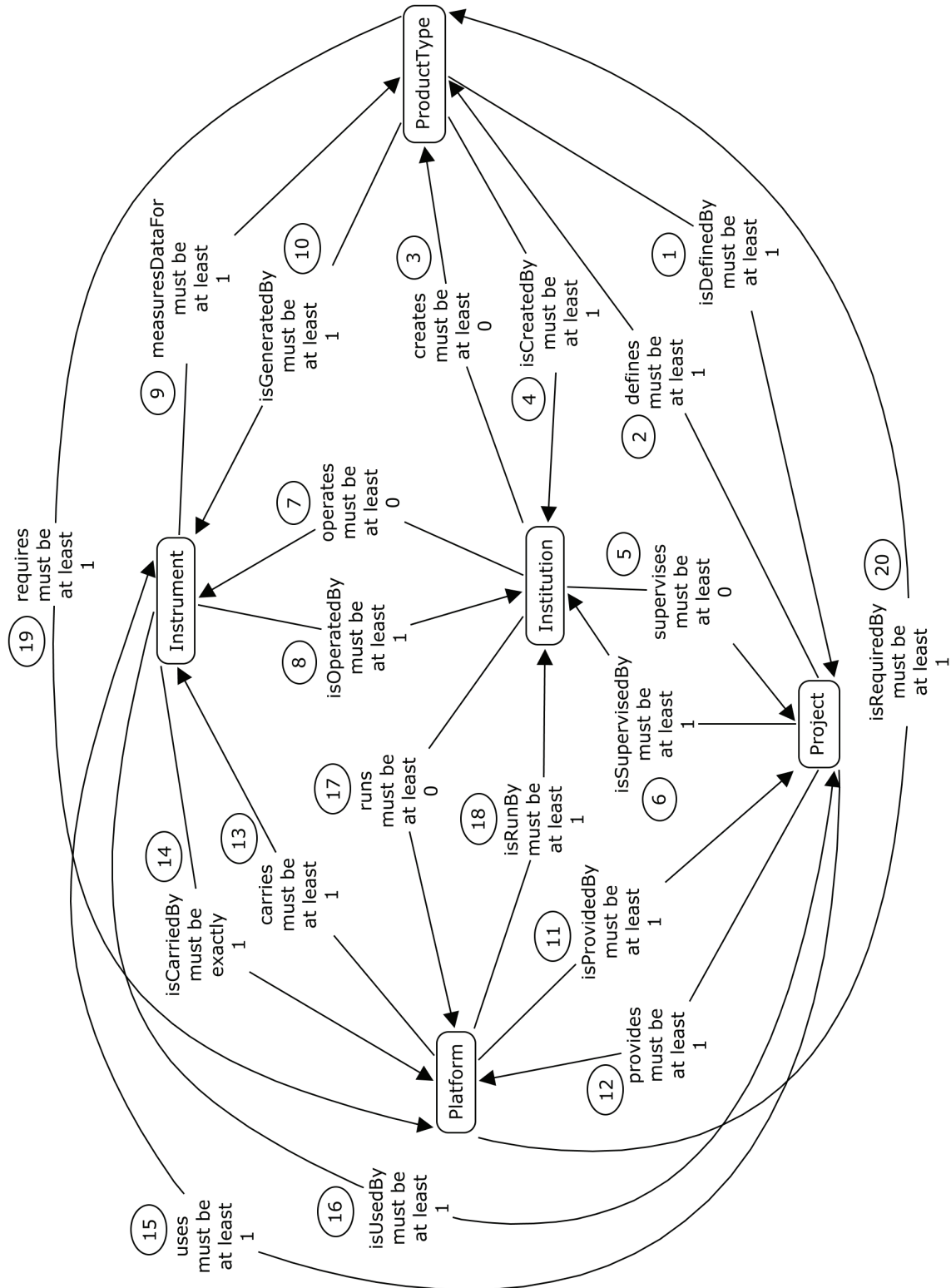


Abbildung 12: Beziehungen zwischen den ISDC-Klassen

- 16 Ein Instrument wird von mindestens einem Projekt genutzt.
- 17 Eine Institution betreibt mindestens Null Plattformen.
- 18 Eine Plattform wird von mindestens einer Institution betrieben.
- 19 Ein Produkttyp benötigt mindestens eine Plattform.
- 20 Eine Plattform wird von mindestens einem Produkttypen benötigt.

Die Institution nimmt hier eine Sonderstellung ein, denn sie kann, muss aber nicht, ein Instrument, ein Projekt, eine Plattform oder einen Produkttyp betreuen.

## 7 Modellierung der Ontologie in Protégé-OWL

Für die Modellierung der Ontologie wird ein vorher definierter Metadatenbestand vom ISDC genutzt, um die Ausgabeergebnisse auch prüfen zu können. Die Inhalte der Excel-Tabelle `Ontologie_Datengrundlage.xlsx` (siehe beigelegte CD) beinhalten diesen Bestand, der in der Ontologie abgebildet ist. Zusätzlich werden auch Beziehungen, die nicht in den XML-Metadokumenten enthalten sind modelliert, um eine umfassende und soweit wie möglich vollständige Wissensbasis zu erhalten.

Nachdem ein neues Projekt angelegt wurde, ist unter dem Metadata-Tab die Ontologie-URI zu setzen. Für die ISDC-Ontologie lautet diese `http://isdc.gfz-potsdam.de/ontology/isdc_1.0.owl`. Hinweise zur Bildung eines URI sind dem Standard von Tim Berners-Lee entnommen [4]. Darüber hinaus kann ein URI auch mit einem Präfix belegt werden, um die Übersichtlichkeit zu steigern, d.h. es muss nicht mehr der vollständige URI angegeben werden. Das gewählte Präfix für unser Beispiel lautet `isdc` und wird vor allen ISDC-Objekten und Properties erscheinen, die im Zuge dieser Arbeit angelegt werden. Dieses Präfix stellt die Eindeutigkeit sicher, wenn andere Ontologien mit ins Modell aufgenommen werden. Andere Ontologien werden durch ein anderes Präfix repräsentiert. Dublin Core nutzt beispielsweise das Präfix `dc`. Somit ist zu jedem Zeitpunkt klar, aus welcher Wissensbasis ein Element stammt. Für die weiteren schriftlichen Ausführungen dieses Abschnittes wird auf das Präfix verzichtet, da sich hier explizit auf die ISDC-Ontologie bezogen wird. Nachdem die grundlegenden Voreinstellungen durchgeführt sind, werden im nächsten Schritt die grobe ISDC-Klassenstruktur mit den dazu gehörigen Individuen, sowie die benötigten Properties angelegt. Im Zuge der Arbeiten können immer wieder Anpassungen bzw. Ergänzungen an dem bereits definierten Datenbestand vorgenommen werden. Klassen sind immer dann anzulegen, wenn noch weitere Unterklassifizierungen möglich sind. Ist keine weitere Klassifizierung mehr durchführbar, sind die Inhalte als Individuen festzulegen. Darüber hinaus ist darauf zu achten, dass Klassen der gleichen Ebene disjunkt sind. Damit wird ausgedrückt, dass wenn ein Objekt zur Klasse A gehört, es nicht gleichzeitig zu Klasse

B gehören kann. In der ISDC-Ontologie sind beispielsweise alle Unterklassen vom ISDC disjunkt. Diese Eigenschaft wird im unteren rechten Bereich des OWL Classes-Tab gesetzt (siehe Abbildung 13, Disjoints-Widget). Auch angelegte Individuen einer Klasse sind disjunkt voneinander anzulegen.

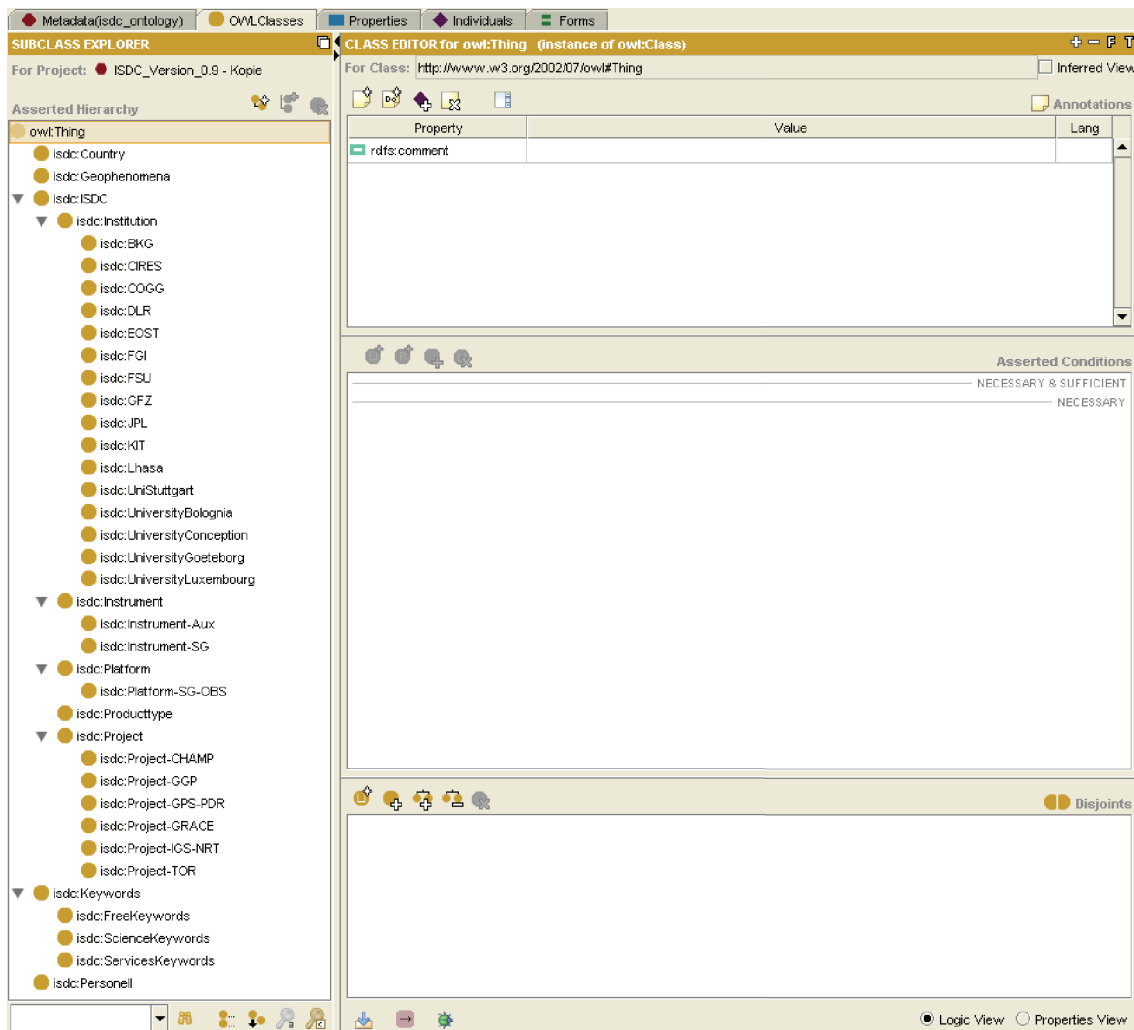


Abbildung 13: OWL Classes-Tab

Abbildung 13 zeigt die Klassenstruktur der ISDC-Ontologie, die unter dem OWL Classes-Tab von Protégé zu finden ist. Das Wurzelement des Baumes ist `owl:Thing`, d.h. alle

darunter angelegten Klassen sind Unterklassen von owl:Thing. Unter der Superklasse ISDC liegen die Unterklassen Institution, Instrument, Platform, Producttype und Project. Denn diese Klassen stehen für die Hauptinhalte, die beim ISDC verfügbar sind. Die einzelnen Institutionen sind als Klasse zu definieren, denn es könnten dort noch differenzierte Unterkategorien zum Einsatz kommen, wie z.B. Abteilungen. In der Abbildung 13 ist nur ein Teil der tatsächlich verwendeten Institution abgebildet, um die Übersichtlichkeit zu steigern. Im eigentlichen Protégé-Projekt sind alle verwendeten Institutionen vorzufinden. Ähnlich verhält es sich bei der Modellierung von Project. Auch hier ist es denkbar, noch Unterprojekte zu definieren. Aus diesem Grund wurden auch hier die Projekte als Subklassen angelegt. Bei den Klassen Instrument und Platform werden Unterklassen zur weiteren Strukturierung der Individuen genutzt. Die einzelnen Instrumente, Plattformen und Produkttypen enthalten Individuen, welche die realen Objekte repräsentieren. Individuen können unter dem Individuals-Tab erzeugt werden. Die Eingabe kann zum einen einzeln oder mit Hilfe einer Eingabemaske gruppenweise erfolgen. Abbildung 14 zeigt Individuen der Klasse Producttype.

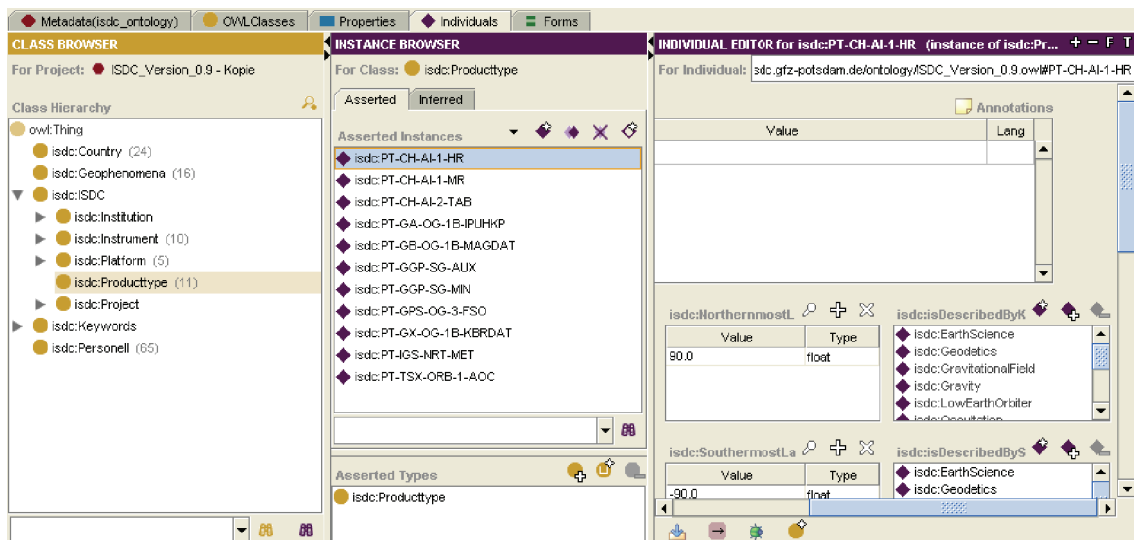


Abbildung 14: Individuals-Tab



Die Klassen Country, Geophenomena, Keywords und Personell sind nicht Unterklassen des ISDCs, weil sie universellen Charakter besitzen. Diese Klassen können auch für andere Ontologien und Wissensbasen stehen, bzw. können sie ihre Inhalte auch von ISDC-fremden Wissensbasen beziehen.

Die Bezeichnungen für die Objekt-Properties sind der Abbildung 12 entnommen. Des Weiteren mussten zusätzliche Properties angelegt werden, um alle Zusammenhänge realisieren zu können. Neue Properties werden z.B. dafür benötigt, um Keywords zu den einzelnen Klassen zu zuordnen, oder aber die Länder zu Personen bzw. zu Institutionen zu zuweisen. Die bisher angelegten Properties sind Objekt-Properties, d.h. durch sie werden immer zwei

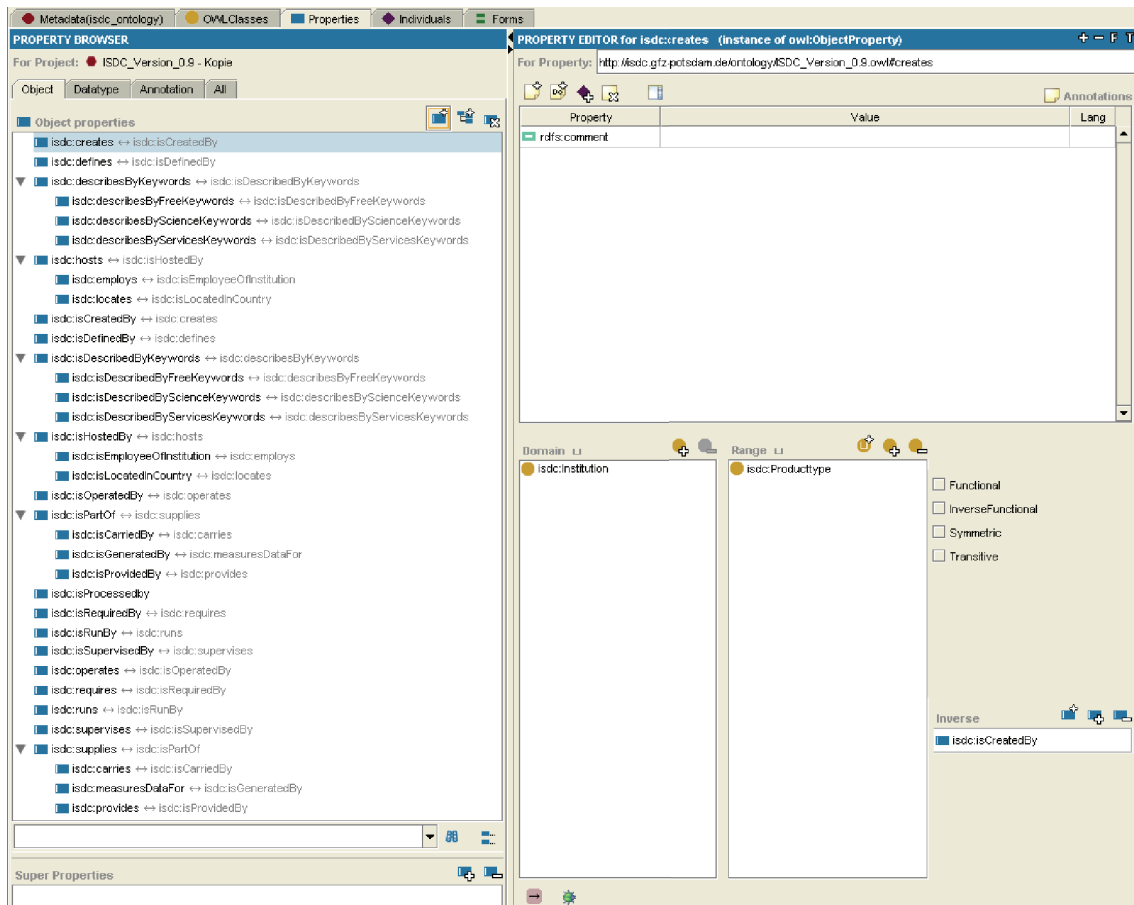


Abbildung 15: Properties-Tab; Objekt-Property

reale Objekte verbunden, welche durch Individuen und Klassen repräsentiert sind. Jede dieser Objekt-Properties konnte durch eine inverse Property ergänzt werden. Alle verwendeten Objekt-Properties sind der Abbildung 15 zu entnehmen.

Teile der ISDC-Ontologie nutzen die Transitivitätseigenschaft, um einen Mehrwert aus den verfügbaren Informationen zu erhalten. Zum einen kommt sie bei den Beziehungen zwischen Personell, Institution und Country zum Einsatz. Die Oberproperties `hosts` und `isHostedBy` beinhalten Unterproperties, die den direkten Bezug zu allen beteiligten Klassen herstellen. So können Informationen, die den Unterproperties zugewiesen wurden, auch über die Oberproperty abgefragt werden. Personen wird eine Institution, in der sie arbeiten zugewiesen. Die Institutionen haben einen Bezug zu einem Land, in dem sie niedergelassen sind. Beide Zuweisungen werden ebenfalls invers modelliert, d.h. den Institutionen werden Personen und Ländern werden Institutionen zugeordnet. Aufgrund der transitiven Eigenschaft, die im Properties-Tab gesetzt werden muss (siehe Abbildung 15), ist es möglich, Abfragen wie z.B. „In welchem Land arbeitet Person A?“ zu beantworten. Die transitive Eigenschaft wird durch Aktivierung des Häkchens von `Transitive` gesetzt. Dabei ist es nicht notwendig, die Beziehung zwischen Person und Land explizit zu modellieren. Aufgrund der Zugehörigkeit der Person zu einer Institution, die wiederum einem Land angehört, kann geschlussfolgert werden, dass sich Person A auch in dem Land der Institution aufhält. Darüber hinaus wurde mit Domain und Range gearbeitet. Eine Property verbindet Objekte einer Domain zu Objekten des Range. Die Property `employs` hat beispielsweise die Klasse `Institution` als Domain und die Klasse `Personell` als Range. Mit Domain und Range ist eine eindeutige Zuordnung von Subjekten bzw. Objekten eines Tripels möglich. Das gerichtete Tripel heißt in diesem Fall `Institution employs Personell`. Die Property agiert dabei immer als Prädikat eines Tripels. Die hier dargelegten Ausführungen sind in der Abbildung 16 nochmals grafisch aufbereitet.

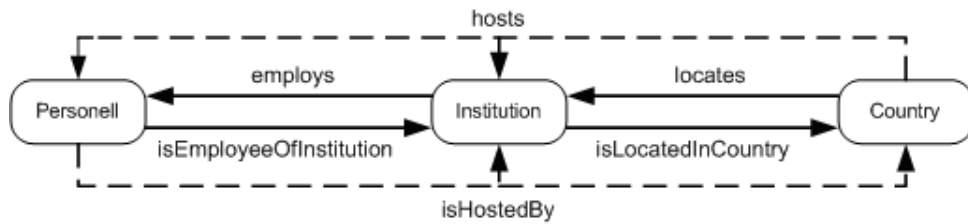


Abbildung 16: Transitive Eigenschaft (Personell, Institution, Country)

Die zweite Verwendung einer transitiven Eigenschaft kommt bei den vier Hauptklassen des ISDCs zum Einsatz, nämlich bei Producttype, Instrument, Platform und Project. Auch hier werden wieder Oberproperties gebildet, die eine Verbindung zu allen beteiligten Klassen herstellen können. Die beiden Oberproperties heißen `isPartOf` und `supplies`. Die weiteren beteiligten Properties sind der Abbildung 17 zu entnehmen. Nachdem alle Beziehungen mit Inhalten gefüllt sind, können danach Abfragen wie z.B. „Zeige alle Produkttypen, die von allen Instrumenten und deren Plattformen, die auch für das Projekt CHAMP benutzt werden, an!“ beantwortet werden. Damit ist es möglich, einen groben Überblick über komplexe Beziehungen der ISDC-Klassen zu erhalten. Für Abfragen ist es somit nicht zwingend notwendig, die jeweiligen Unterproperties zu nutzen, sondern es kann auf die Oberproperty zurückgegriffen werden. Weitere Angaben zu den verschiedenen Beziehungstypen, wie z.B. transitiv oder symmetrisch und deren Funktionsweisen, sind Abschnitt 4.2 Ontologiesprachen im Vergleich zu entnehmen.

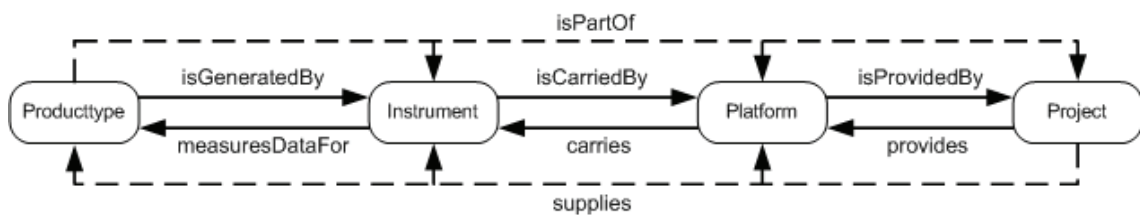


Abbildung 17: Transitive Eigenschaft (Producttype, Instrument, Platform, Project)

Datatype-Properties verhalten sich ähnlich wie Objekt-Properties. Sie verbinden Klassen oder Individuen mit konkreten Werten, z.B. in Form von Strings oder Integer-Werten.

Ein großer Unterschied zur Objekt-Property besteht jedoch darin, dass keine inversen Properties angelegt werden können, d.h. Abfragen sind nur in eine Richtung modellierbar. Außerdem besteht nur die Möglichkeit eine funktionale Eigenschaft einer Property zuzuweisen. Funktional bedeutet, dass nur ein Wert zugewiesen werden kann [28]. Über Datatype-Properties wurden den Produkttyp-Individuen IDs und Titel mit den Properties `hasID` und `hasTitle` zugewiesen. Darüber hinaus konnten z.B. den Projekten über `hasTemporalCoverage` Projektlaufzeiten zugeordnet werden. Die Individuen der Subklasse `Platform-SG-OBS` konnten mit räumlichen Angaben über die Datatype-Property `hasSpatialCoverage` versehen werden. Die Angaben zu Raum und Zeit sind noch nicht vollständig erfasst, jedoch sind sie in der Ontologie schon vorgesehen und teilweise beispielhaft eingetragen. Die Personen-Individuen wurden ebenfalls durch die Datatype-Properties `PersonellFirstName` und `PersonellName` mit Zusatzinformationen versehen, soweit die dafür notwendigen Daten verfügbar waren. Abbildung 18 enthält alle verwendeten Datatype-Properties.

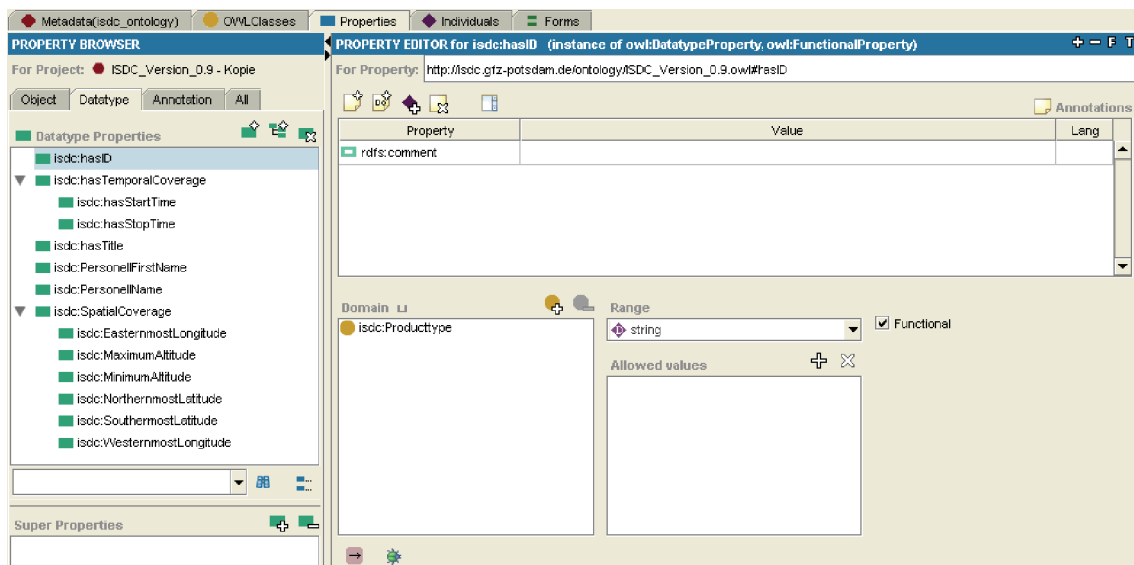


Abbildung 18: Properties-Tab; Datatype-Property

`Keywords` wurden als Klassen mit dazugehörigen Individuen angelegt. Für die Beschreibung von Objekten mit Keywörtern stehen die Objekt-Properties `describesByKeywords`,

`isDescribedByKeywords` und deren Unterproperties zur Verfügung. Wie bereits dargelegt, bieten Objekt-Properties die Möglichkeit Inverse anzulegen, d.h. Abfragen sind in beide Richtungen möglich. Es können beispielsweise die Keywörter eines Instrumentes ermittelt werden, aber auch alle Instrumente, die zur Beschreibung ein oder mehrere bestimmte Keywörter nutzen. Damit diese Abfragen sichergestellt sind, kommen Objekt-Properties mit ihren Eigenschaften zum Einsatz. Durch die Nutzung von Datatype-Properties wären die Abfragemöglichkeiten eingeschränkt. Deshalb wurde auf ihre Verwendung in diesem Fall verzichtet.

Über den Individuals-Tab werden auch die Informationen zu einem speziellen Individuum definiert. Das Individuum `PT-CH-AI-1-HR` (siehe Abbildung 14) erhält hier unter anderem seine speziellen Keywords. Dazu wird die benötigte Property gewählt und die beschreibenden Informationen sind dann in das Widget der Property einzutragen bzw. aus dem eingegebenen Datenpool auszuwählen. Ein bestimmter Produkttyp wird beispielsweise durch `Earth Science`, `Geodetics` oder `Gravity` beschrieben. Weitere Properties mit deren Inhalten können durch scrollen sichtbar gemacht werden.

Die bereits angelegten Properties werden für die späteren Modellierungsschritte benötigt, um Restriktionen ausdrücken zu können. Quantoren, Kardinalitäten oder die `hasValue`-Restriktion können genutzt werden, um Einschränkungen zu formulieren. Bei den Quantoren wird zwischen dem Existenzquantor und dem Allquantor unterschieden. Der Existenzquantor, auch als `someValuesFrom`-Restriktion bezeichnet, beschreibt eine Menge von Individuen, welche eine anonyme Klasse darstellen, die mindestens eine spezielle Property zu einer definierten Klasse aufweisen. Alle Individuen der anonymen Klasse, in diesem Fall die Individuen `A1`, `A2` und `A3`, weisen mindestens eine Beziehung zu den einzelnen Individuen der Klasse auf. Sie können jedoch auch noch andere Beziehungen zu Individuen außerhalb der Klasse haben. Diese sonstigen Beziehungen zu den Individuen `C1` und `C2` sind durch gestrichelte Linien dargestellt. In Abbildung 19 ist die Funktionsweise des Existenzquantors grafisch aufbereitet.

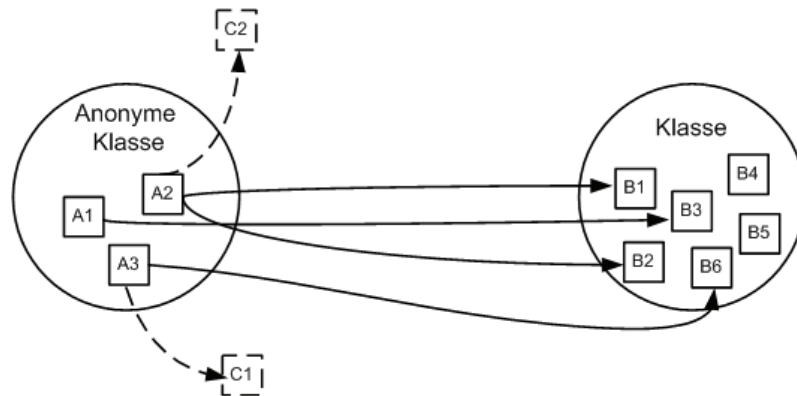
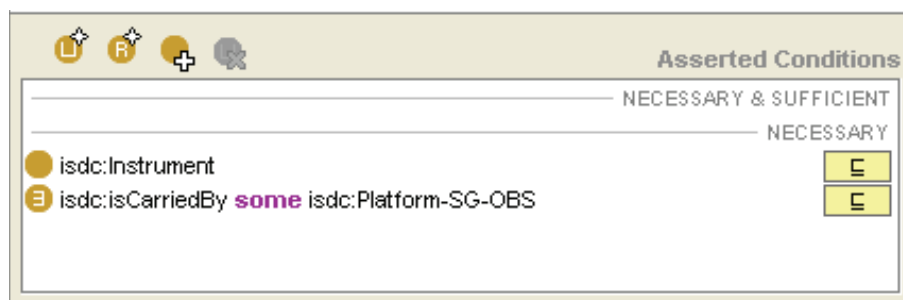


Abbildung 19: Existenzquantor

In der ISDC-Ontologie wurde die `someValuesFrom`-Restriktion unter anderem bei der Klasse `Instrument-Aux` angewendet, um auszudrücken, dass die Instrumente dieser Klasse sich auf einer der Plattformen der Klasse `Platform-SG-OBS` befinden. Der verwendete logische Ausdruck dafür lautet wie folgt:  $\exists$ : `Instrument isCarriedBy some Platform-SG-OBS`. In Protégé erfolgt die Eingabe der Restriktion im Asserted Conditions-Widget (siehe Abbildung 20). Die `someValuesFrom`-Restriktion kommt in der ISDC-Ontologie häufig zum Einsatz. Das hier angeführte Beispiel soll lediglich die konkrete Verwendung der Restriktion darlegen.

Abbildung 20: Existenzquantor am Beispiel der Klasse `Instrument-Aux`

Eine andere Form der Restriktion bietet der Allquantor, auch `allValuesFrom`-Restriktion genannt. Mit dem Allquantor wird eine Menge von Individuen beschrieben, die für eine festgelegte Property nur Beziehungen zu Individuen aufweisen, die zu einer speziellen Klasse gehören (siehe Abbildung 21). Diese Restriktion ist jedoch keine Garantie dafür, dass diese Property existiert. Durch sie wird nur sichergestellt, dass wenn die Property für die Individuen der anonymen Klasse (A1 - A3) existiert, muss sie mit einem Individuum der Klasse (B1 - B6) belegt sein. Diese Restriktion ist in der ISDC-Ontologie nicht verwendet worden, sie wird jedoch der Vollständigkeit halber erwähnt.

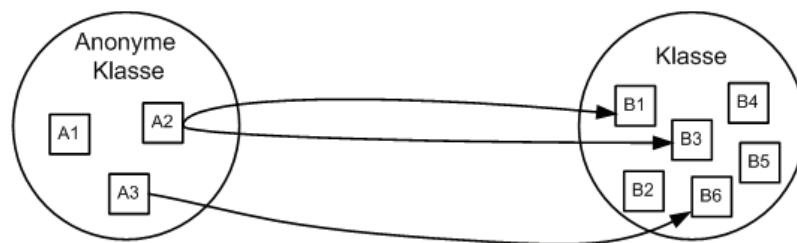


Abbildung 21: Allquantor

Die `hasValue`-Restriktion ist ein weiteres Modellierungsmittel. Sie beschreibt eine anonyme Klasse von Individuen, die eine definierte Beziehung zu einem speziellen Individuum haben. Im Gegensatz zu den Quantifizierungen ist hier nicht eine Menge von Individuen, die in einer Klasse zusammengefasst sind, sondern ein einzelnes, spezielles Individuum auswählbar. Alle Individuen der anonymen Klasse (A1 - A3) stehen in Beziehung zum einzelnen Individuum B2 (siehe Abbildung 22).

Die Klasse `Project-TOR` soll hierfür den praktischen Einsatz in der ISDC-Ontologie verdeutlichen. Deshalb wird die Abbildung 23 nachfolgend exemplarisch erläutert. `Project-TOR` ist eine Unterklasse von `Project`. Die Klasse `Project-TOR` definiert das Individuum `PT-TSX-ORB-1-AOC` der Klasse `Producttype`. Der Datatype-Property `hasStartTime` wird ein konkreter Wert in Form eines Strings zugewiesen. Der Datentyp String ist in Hochkommas zu setzen. Die Klasse `Project-TOR` ist durch diverse Keywords beschrieben, die auch als

Individuen existieren. AOCs Quaternions und Spacecraft Attitude Data sind freie Keywörter. Earth Science, Satellite Orbits, Geodetics, Gravity und Solid Earth gehören zu den Science Keywords und entstammen dem kontrollierten Wortgut des GCMD. Klassen, wie z.B. Institutionen, können auch als Individuen agieren. Dazu muss bei der Eingabe die Klassenstruktur gewählt werden. So lassen sich die Zuweisungen der Institutionen GFZ und DLR erklären. Als letzte Zuweisung kommt das Individuum Platform-TSX hinzu. Alle hier dargestellten Individuen beschreiben die Klasse Project-TOR in ihrer Gesamtheit.

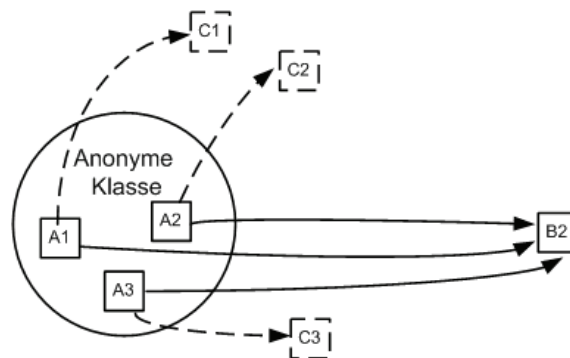


Abbildung 22: hasValue-Restriktion

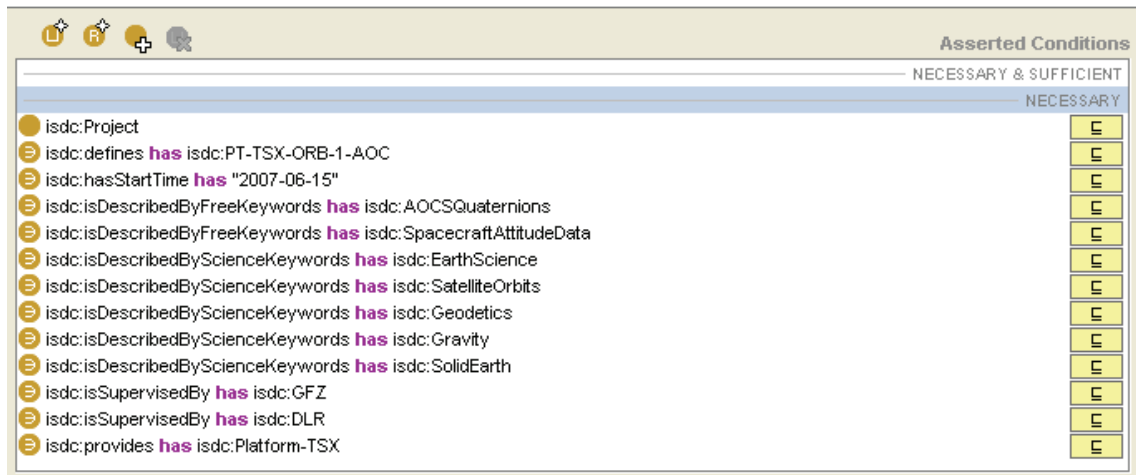


Abbildung 23: Bedingungen der Klasse Project-TOR



Wie aus den Bildern 23 und 20 ersichtlich ist, wird zwischen notwendigen (necessary) und notwendigen und hinreichenden (necessary and sufficient) Bedingungen unterschieden. Eine notwendige Bedingung bedeutet, dass wenn ein Individuum Mitglied einer Klasse ist, müssen auch die definierten Bedingungen erfüllt sein. Mit notwendigen Bedingungen können primitive Klassen gebildet werden. Primitive Klassen lassen noch keine Schlussfolgerungen in Form von gegenseitigen Beziehungen zu. Abbildung 24 stellt diese Aussage schematisch dar. Bisher angelegte Klassen sind den primitiven Klassen zu zuordnen. Mit ihnen können Aussagen wie, wenn ein Individuum Mitglied dieser Klasse ist, dann ist es notwendig die Bedingungen zu erfüllen, formuliert werden.

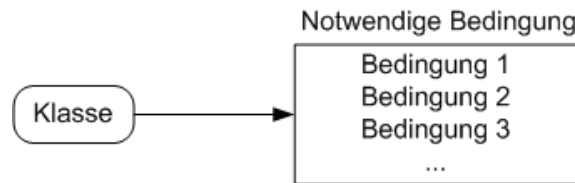


Abbildung 24: Notwendige Bedingungen

Eine definierte Klasse nutzt hingegen mindestens eine notwendige und hinreichende Bedingung. Notwendige Bedingungen müssen dazu in notwendige und hinreichende Bedingungen erweitert werden. Die Angabe von notwendigen und hinreichenden Bedingungen ermöglicht nun das Erkennen von gegenseitigen Beziehungen. Eine Klasse hat demzufolge die definierten Bedingungen zu erfüllen und die aufgestellten Bedingungen implizieren die Zugehörigkeit zu dieser Klasse (siehe Abbildung 25). Das Erstellen von definierten Klassen macht erst die Klassifizierung durch einen Reasoner (siehe Abschnitt 5.4 Pellet) möglich.

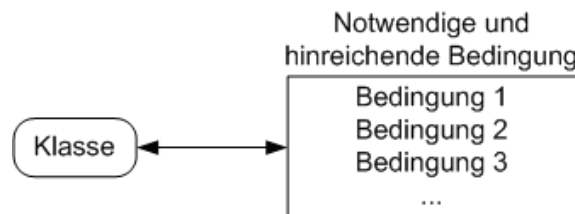


Abbildung 25: Notwendige und hinreichende Bedingungen

**Project-Institution-GFZ** ist ein Beispiel für eine definierte Klasse aus der ISDC-Ontology. Dadurch, dass die gesetzte Bedingung  $\exists$ : **isSupervisedBy some GFZ** durch eine notwendige und hinreichende Bedingung repräsentiert wird, ist die Klassifizierung und Abfrage aller Projekte möglich, die vom GFZ beaufsichtigt werden. Abbildung 26 zeigt die in Protégé formulierte Bedingung im Asserted Conditions-Widget der Klasse **Project-Institution-GFZ**. Nach dem Reasoning können die ermittelten Projekte für diese Fragestellung aus dem Inferred Hierarchy-Fenster entnommen werden.



Abbildung 26: Project-Institution-GFZ als Beispiel für eine definierte Klasse

Weitere Informationen der Wissensbasis sind analog zu den hier gezeigten Beispielen in die Ontologie eingearbeitet worden. Die angeführten Beispiele für die Restriktionen, die notwendigen und die notwendigen und hinreichenden Bedingungen sind weitestgehend der Quelle [29] entnommen und wurden auf die Inhalte des ISDCs angewendet. Die Schreibweise, die der Restriktion-Editor nutzt, heißt Manchester-OWL-Syntax. Diese Syntax wurde durch die OWL- und durch die DL-Syntax beeinflusst. Abbildung 20 zeigt z.B. eine vom Restriktion-Editor erzeugte Bedingung, welche die Manchester-Syntax nutzt. Das **some** steht für das DL-Symbol  $\exists$  bzw. für die OWL Restriktion `someValuesFrom`. Die Ontologie kann auch in dieser Syntax ausgegeben werden. Detaillierte Informationen zur Manchester-OWL-Syntax sind der Quelle [8] zu entnehmen.

Die in Protégé eingegebenen Ontologiedaten können zu jedem Zeitpunkt auf Konsistenz geprüft werden. Auch die Klassifikation ist für die Kontrolle der Eingaben verwendbar. Die

Klassifikation erzeugt die Inferred Hierarchy. In der Inferred Hierarchy werden die angegebenen Restriktionen genutzt, um die geschlussfolgerte Klassenstruktur zu bilden. Hier kann geprüft werden, ob die bisher getätigten Eingaben das gewünschte Ergebnis liefern. Abbildung 27 stellt die Asserted Hierarchy, also den Baum, der durch die direkten Eingaben entsteht, und die Inferred Hierarchy, gegenüber. Weitere Erläuterungen dazu beziehen sich nun insbesondere auf die Klasse Project. Im Asserted Hierarchy-Fenster sind alle primitiven und definierten Klassen sichtbar. Zu den primitiven Klassen gehören Project-CHAMP, Project-GGP, Project-GPS-PDR, Project-GRACE, Project-IGS-NRT und Project-TOR. Zu den definierten Klassen gehören Project-Institution-GFZ, Project-Keywords-EarthScience und Project-KW-GP-ClimateChange. Definierte Klassen sind durch eine andere Symbolik (drei Striche im Kreis) kenntlich gemacht. Nach dem Reasoning erscheint das Inferred Hierarchy-Fenster. Hier können dann die zur Klasse gehörenden Objekte ermittelt werden. Die Klasse Project-Institution-GFZ enthält alle angelegten Projekte, außer Project-TOR. Diese Klasse enthält also alle Projekte, die vom GFZ betreut werden. Die Klassen Project-KW-GP-ClimateChange und Project-Keywords-EarthScience enthalten alle angelegten Projekte. Da die Ausgabemenge in diesen Fällen gleich ist, wird nur eine Ausgabe benötigt. Deshalb wird die Klasse Project-Keywords-EarthScience als Unterklasse von Project-KW-GP-ClimateChange angelegt.

Bei der Modellierung der ISDC-Ontologie wurde auch auf die Möglichkeiten der Metamodellierung zurückgegriffen. Die Metamodellierung wird nur von OWL Full zur Verfügung gestellt. Beispielsweise sind die einzelnen Institutionen als Klassen angelegt. Diese Klassen werden aber auch als Individuen angesehen und dem entsprechend behandelt. Abbildung 23 verwendet die Institutionen GFZ und DLR als Individuen, was durch die hasValue-Restriktion ausgedrückt wird. Beim Reasoning werden die Inhalte der ISDC-Ontologie temporär in OWL DL konvertiert, um die anschließende Klassifikation durchführen zu können.

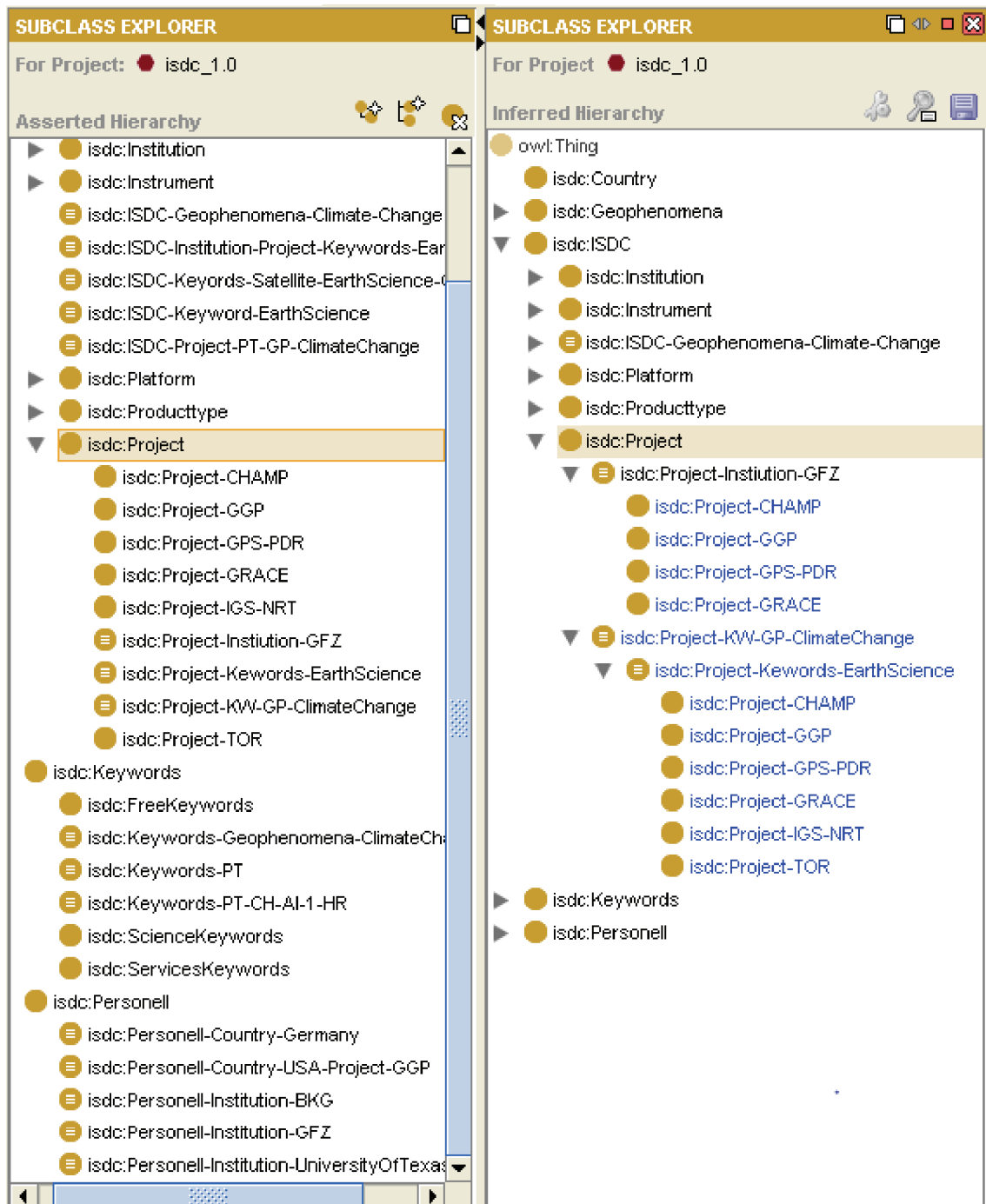


Abbildung 27: Asserted und Inferred Hierarchy

Das Plugin OWLViz ist als zusätzlicher Tab wählbar und ermöglicht es, Klassenhierarchien in Form von Graphen zu betrachten. Außerdem wird eine schrittweise Navigation in der Ontologie unterstützt. Das eröffnet eine andere Sichtweise auf das Informationsmodell und den zugrunde liegenden Datenbestand. Auch hier wird zwischen primitiven (gelb) und definierten Klassen (orange) unterschieden. Die Ausgaben des Asserted und des Inferred Hierarchy-Fensters können als Grafik exportiert und ausgewertet werden [24]. Als Beispiel für die Funktionsweise dient wieder die Klasse Project. Abbildung 28 stellt den Graphen der Asserted Hierarchy dar. Die Klasse Project sowie ihre Unterklassen gehören zur Klasse ISDC. Die Klasse ISDC ist wiederum eine Unterklasse von owl:Thing. Die Farbgebung ordnet die Unterklassen von Project den primitiven und definierten Klassen zu.

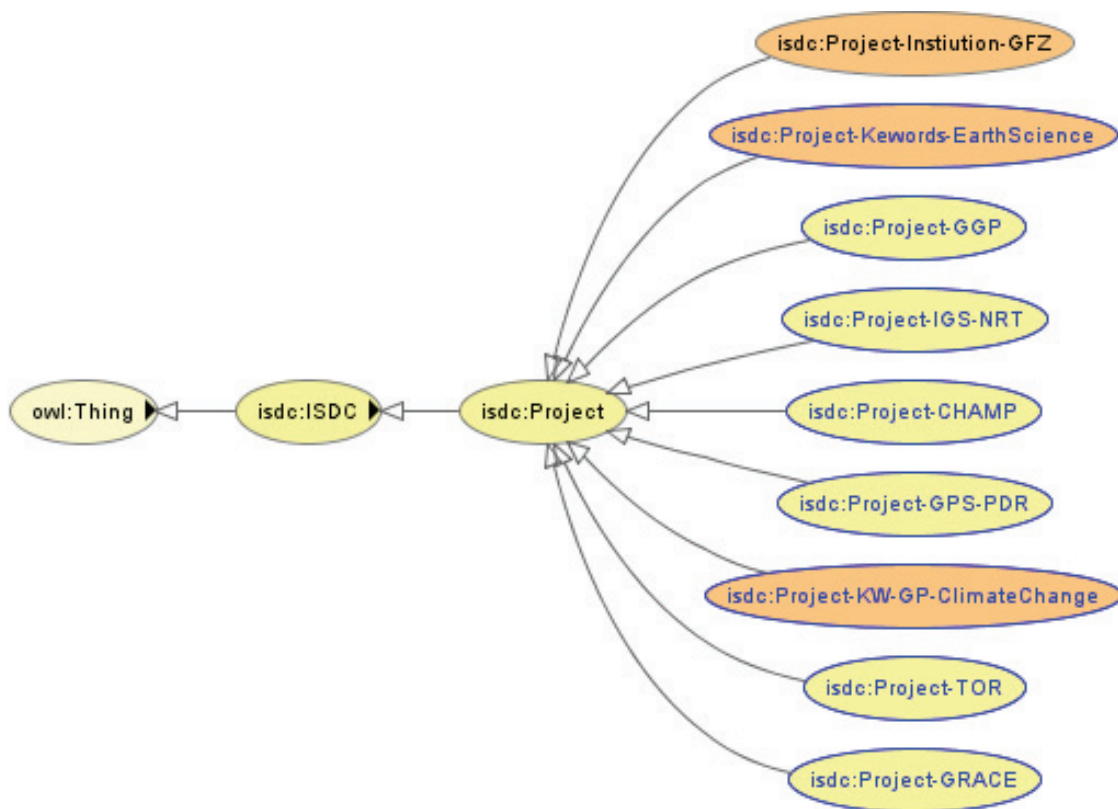


Abbildung 28: Klasse Project (Asserted Hierarchy) in OWLViz

Abbildung 30 stellt die Inferred Hierarchy dar. Hier sind die Zusammenhänge, die zwischen den einzelnen Klassen bestehen sichtbar gemacht. Die primitiven Klassen sind den dazugehörigen definierten Klassen zugeordnet. Es können mehrere Zugehörigkeiten bestehen. Das zeigen die Klassen Project-CHAMP, Project-GPS-PDR, Project-GRACE und Project-GGP. Sie sind den Klassen Project-Institution-GFZ und Project-Keywords-EarthScience zugewiesen. Der Graph der Inferred Hierarchy stellt Vererbungen und daraus resultierende Klassifizierungen deutlich heraus und kann sehr zum Verständnis des Systems beitragen.

Die ISDC-Ontologie muss im WWW zur Verfügung stehen, damit sie auch für weitere Nutzer und Anwendungen zur Verfügung stehen kann, was dem Grundgedanken des Semantic Webs entspricht. Im Internet ist die OWL-Datei unter [http://isdc.gfz-potsdam.de/ontology/isdc\\_1.0.owl](http://isdc.gfz-potsdam.de/ontology/isdc_1.0.owl) verfügbar (siehe Abbildung 29).

```

<?xml version="1.0" ?>
- <rdf:RDF xmlns:xsp="http://www.owl-ontologies.com/2005/08/07/xsp.owl#" xmlns:swrlb="http://www.w3.org/2003/11/swrlb#"
  xmlns:isdc="http://isdc.gfz-potsdam.de/ontology/isdc_1.0.owl#" xmlns:swrl="http://www.w3.org/2003/11/swrl#"
  xmlns:protege="http://protege.stanford.edu/plugins/owl/protege#" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#" xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#" xmlns:assert="http://www.owl-ontologies.com/assert.owl#" xml:base="http://isdc.gfz-
  potsdam.de/ontology/isdc_1.0">
- <owl:Ontology rdf:about="" />
- <owl:Class rdf:about="http://isdc.gfz-potsdam.de/ontology/isdc_1.0.owl#NAOM">
  - <owl:disjointWith>
    <owl:Class rdf:about="http://isdc.gfz-potsdam.de/ontology/isdc_1.0.owl#UniversityOfTexas" />
  </owl:disjointWith>
- <dfs:subClassOf>
  - <owl:Restriction>
    - <owl:onProperty>
      <owl:ObjectProperty rdf:about="http://isdc.gfz-potsdam.de/ontology/isdc_1.0.owl#isDescribedByScienceKeywords" />
    </owl:onProperty>
    - <owl:hasValue>
      - <isdc:ScienceKeywords rdf:about="http://isdc.gfz-potsdam.de/ontology/isdc_1.0.owl#TerrestrialHydrosphere">
        - <isdc:describesByScienceKeywords>
          - <isdc:Platform-SG-OBS rdf:about="http://isdc.gfz-potsdam.de/ontology/isdc_1.0.owl#Platform-BKG-SG-OBS-WE">
            - <isdc:isDescribedByScienceKeywords>
              - <isdc:ScienceKeywords rdf:about="http://isdc.gfz-potsdam.de/ontology/isdc_1.0.owl#Aquifers">
                - <isdc:describesByScienceKeywords>
                  - <isdc:Platform-SG-OBS rdf:about="http://isdc.gfz-potsdam.de/ontology/isdc_1.0.owl#Platform-EOST-SG-OBS-
                  ST">
                    - <isdc:isDescribedByScienceKeywords>

```

Abbildung 29: OWL-Datei im WWW

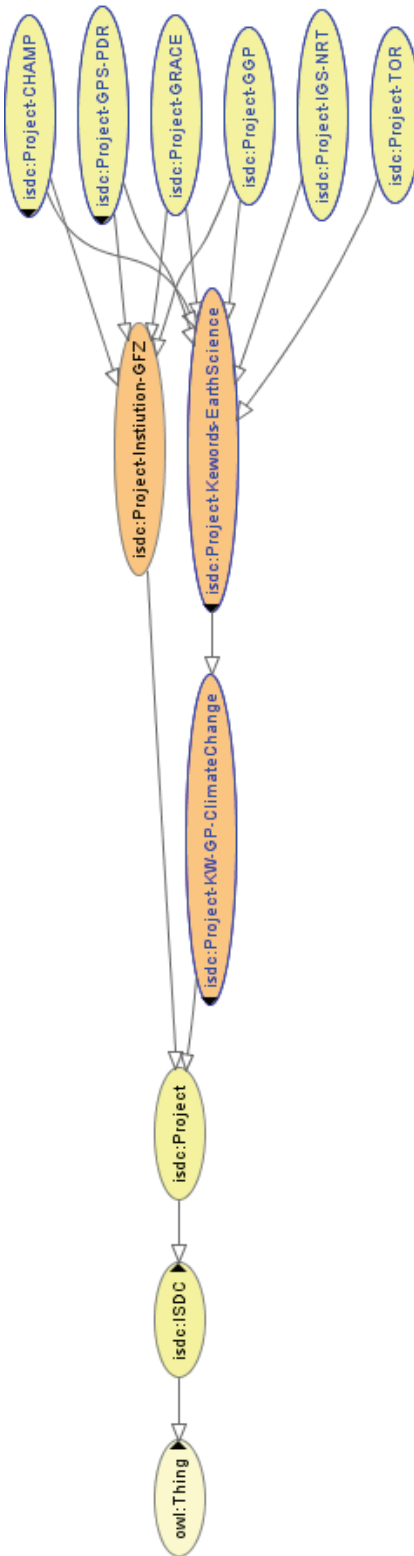


Abbildung 30: Klasse Project (Inferred Hierarchy) in OWLViz

---

## 7.1 Umsetzung der Anwendungsfälle

Im folgenden Abschnitt wird näher auf die definierten Anwendungsfälle und deren Umsetzung eingegangen.

**Anwendungsfall 1:** Zeige alle Produkttypen an, die auch die Schlüsselwörter vom Geophänomen „ClimateChange“ besitzen!

Um diesen Anwendungsfall beantworten zu können, müssen zunächst die für das Geophänomen Climate Change verwendeten Keywörter ermittelt werden. Dazu wurde unter der Klasse Keywords die Unterklasse Keywords-Geophenomena-Cli`m`ateChange angelegt. Mit dem Ausdruck  $\exists$ : Keywords describesByKeywords has ClimateChange können die für Climate Change verwendeten Keywords ermittelt und für die weitere Verwendung zur Verfügung gestellt werden. Die Klasse PT-KW-GP-Cli`m`ateChange, eine Unterklasse von Producttype, liefert durch die in Abbildung 31 dargestellte Bedingung die gesuchten Produkttypen. Jede aufgestellte Bedingung repräsentiert eine Menge von Individuen, die als ein eigenes Konzept (anonyme Klasse) zu betrachten sind.

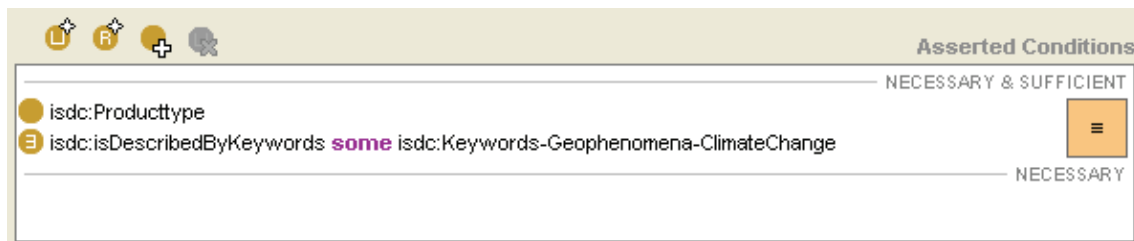


Abbildung 31: Bedingung für Anwendungsfall 1

Zur Ausgabe der beteiligten Produkttypen muss der Prozess „Compute individuals belonging to class“ angestoßen werden. Die Ausgabemenge der gesuchten Produkttypen in Form von Individuen für den Anwendungsfall 1 ist in Abbildung 32 zu sehen.



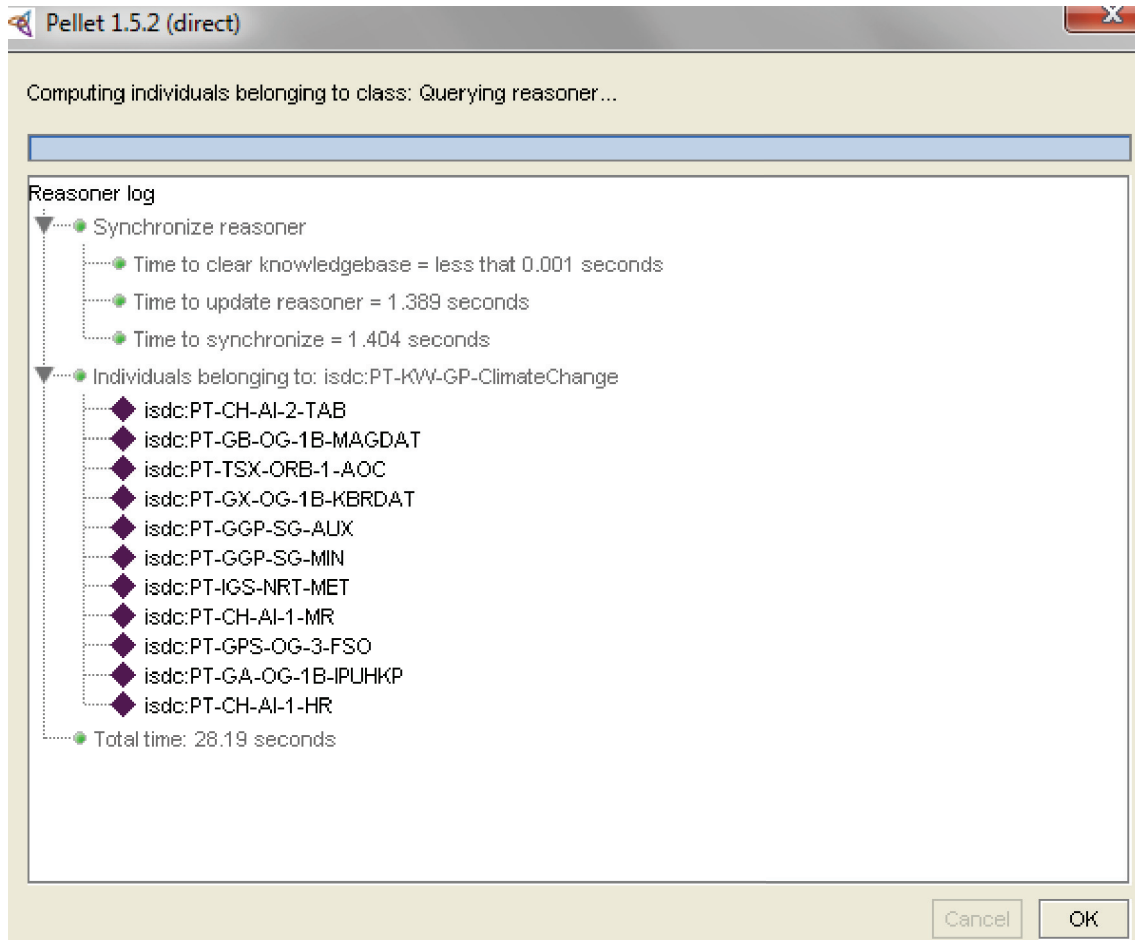


Abbildung 32: Ergebnis für Anwendungsfall 1

**Anwendungsfall 2:** Zeige alle Personen an, die in Institutionen in den USA arbeiten und das GGP-Projekt betreuen!

Um diesen Anwendungsfall beantworten zu können, müssen zwei weitere Hilfsklassen zur Filterung der Daten angelegt werden. Die Klasse `Institution-Project-GGP` gibt alle Institutionen aus, die am GGP-Projekt beteiligt sind. Dies wird mittels des Ausdrucks  $\exists$ : `Institution supervises some Project-GGP` realisiert. Nachdem die beteiligten Institutionen feststehen, können nun die bei den Institutionen beschäftigten Personen ermittelt

werden. Die Personenermittlung erfolgt in der Klasse `Personell-Project-GGP` durch den Ausdruck  $\exists$ : `Personell isEmployeeOfInstitution some Institution-Project-GGP`. Die Klasse `Personell-Country-USA-Project-GGP` beantwortet abschließend Anwendungsfall 2 durch die in Abbildung 33 definierte Bedingung. Der ermittelte Personenkreis in `Personell-Project-GGP` wird durch die Bedingung, dass die Person aus den USA stammen muss, eingeschränkt. Die Angabe des Landes erfolgt über die transitive Property `isHostedBy`. Den Personen wird indirekt ein Land zugewiesen, da sie sich in dem Land befinden, in dem sich auch die Institution befindet, in der sie arbeiten (siehe Abbildung 16). Beide Mengen werden durch das Symbol  $\sqcap$  (DL-Syntax), bzw. durch `intersectionOf` (OWL-Syntax) oder durch `and` (Manchester-Syntax) verbunden. Das bedeutet das Individuen der Klasse `Personell-Country-USA-Project-GGP` beide Aussagen erfüllen müssen, um in der Ausgabemenge zu erscheinen.

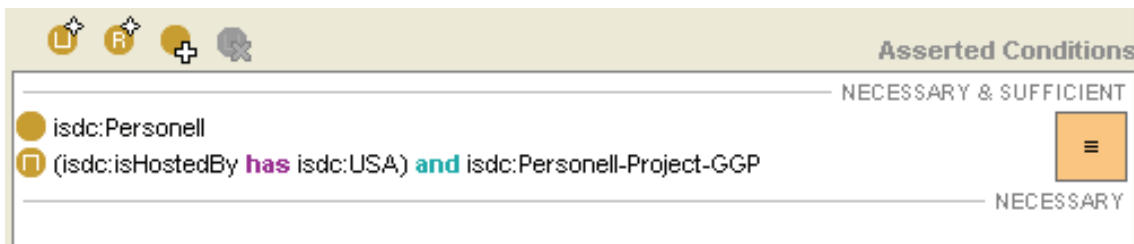


Abbildung 33: Bedingung für Anwendungsfall 2

Die Ergebnismenge für diesen Anwendungsfall ist Abbildung 34 zu entnehmen.

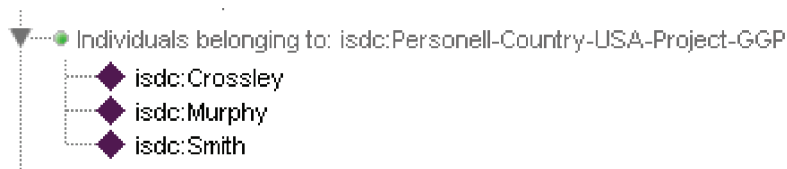


Abbildung 34: Ergebnis für Anwendungsfall 2

**Anwendungsfall 3:** Zeige alle Produkttypen an, die zum Projekt GRACE, zur Plattform GRACE-A, zur Institution GFZ und zum Instrument Star Camera System gehören!

Diese Fragestellung ist sehr detailliert und könnte beispielsweise von einem Wissenschaftler mit speziellem Hintergrundwissen gestellt werden. Die Klasse `PT-ProjectGRACE-PlatformGRACEA-InstitutionGFZ-InstrumentSCA` beantwortet diesen Anwendungsfall. In Abbildung 35 sind die dazu benötigten Bedingungen für die Institution, das Projekt, das Instrument und die Plattform zu sehen. Als Ergebnis wird diesmal ein einziger Produkttyp, nämlich `PT-GA-OG-1B-IPUHKP`, ausgegeben.

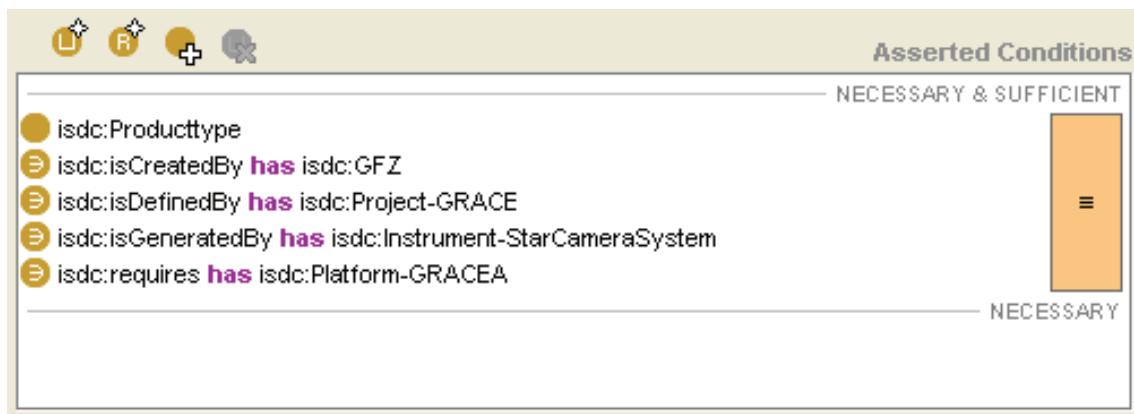


Abbildung 35: Bedingung für Anwendungsfall 3

**Anwendungsfall 4:** Zeige alle Produkttypen, Projekte, Plattformen, Institutionen und Instrumente an, die mittels der Schlüsselwörter (Keywords) Satellite, Earth Science und Gravity beschrieben sind!

Abbildung 36 enthält die Bedingungen für die Klasse `ISDC-Keywords-Satellite-Earth-Science-Gravity`. Diese Klasse gibt alle passenden Objekte aus, welche die Kombination dieser drei Keywörter enthalten.

Die Institutionen und die Projekte der Ausgabemenge sind der Inferred Hierarchy zu entnehmen. Die Individuen für Instrumente, Plattformen und Produkttypen müssen gesondert ermittelt werden, da sie nach der Klassifikation nicht automatisch in der Inferred Hierarchy erscheinen. Abbildung 37 zeigt alle im Datenbestand ermittelten Ergebnisse für diese Fragestellung. Die linke Seite stellt das Ergebnis der Inferred Hierarchy und die rechte Seite die ermittelten Individuen dar.

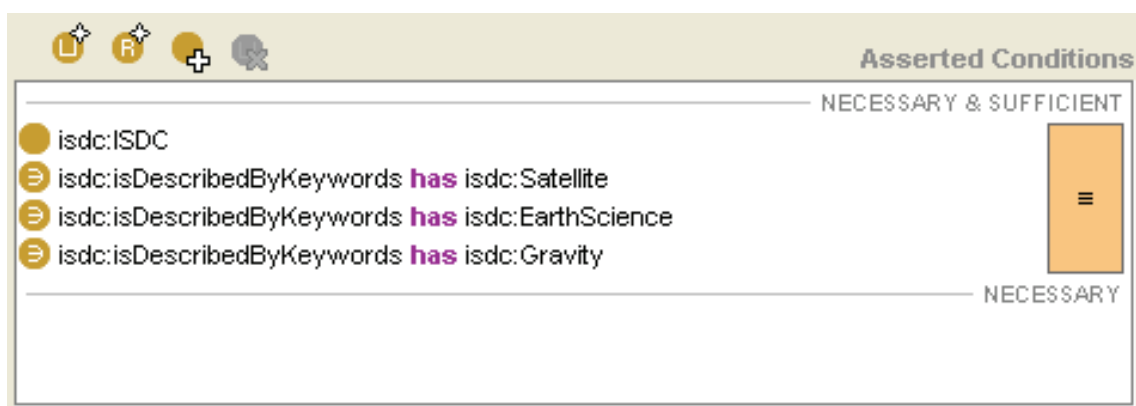


Abbildung 36: Bedingung für Anwendungsfall 4

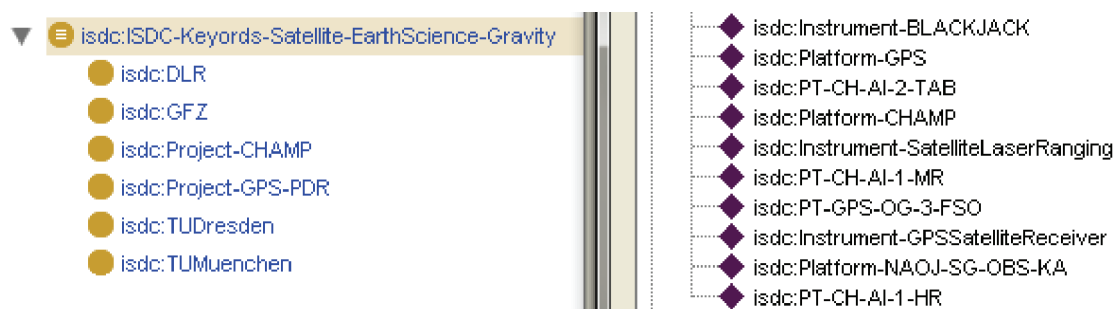


Abbildung 37: Ergebnis für Anwendungsfall 4

**Anwendungsfall 5:** Zeige alle Produkttypen, Projekte, Plattformen, Institutionen und Instrumente an, die in Beziehung zum Geophänomen „Climate Change“ stehen!

Diese Fragestellung lehnt sich an Anwendungsfall 1 an. Deshalb werden auch hier zunächst die Keywörter von Climate Change ermittelt, um sie dann weiterzuverwenden. Abbildung 38 zeigt die formulierte Bedingung der Klasse ISDC-Geophenomena-ClimateChange. Das Ausgabeergebnis ist nicht ausschließlich auf Produkttypen, wie in Anwendungsfall 1, beschränkt. Zusätzlich erhält der Nutzer Angaben zu Institutionen, Projekten, Instrumenten und Plattformen. Dadurch kann ein umfangreicher Überblick über den in Frage kommenden Datenbestand erlangt werden. Auf die Auflistung der Ergebnisse wird in diesem Fall verzichtet, da es sich hier um eine relativ große Ausgabemenge handelt.

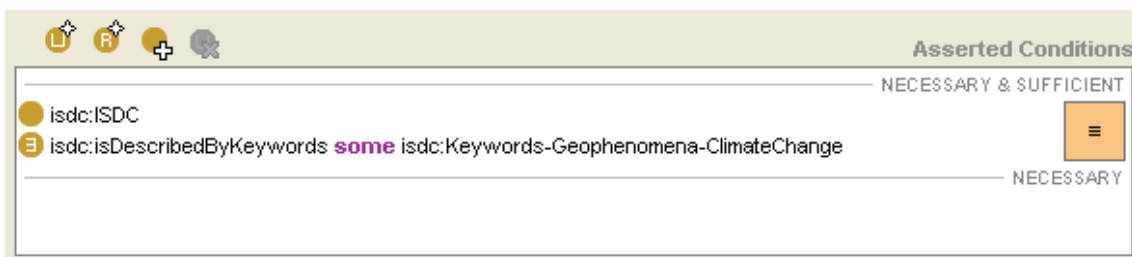


Abbildung 38: Bedingung für Anwendungsfall 5

## 7.2 Nächste Schritte

Die in dieser Arbeit erstellte Ontologie ist eine exzellente Grundlage für die nächsten Schritte. Eine Ontologie sollte falls möglich nicht ausschließlich auf einer einzigen Domäne beruhen. Existieren andere nutzbare Wissensquellen, sollten diese mit ins Modell aufgenommen werden. Nur so kann das bereits vorhandene und das neu hinzukommende Wissen optimal vernetzt und genutzt werden. Findet die Verwendung von existierenden Quellen statt, wird auch das wiederholte Abspeichern von bereits abrufbaren Daten eingeschränkt und der Generierung von Fehlern entgegengewirkt. Um dies beispielhaft für die

ISDC-Ontologie umzusetzen, wurde der Dublin Core Standard in die Ontologie integriert. Dazu muss der Metadata-Tab gewählt sein. Im Ontology Browser-Widget können Importe gesetzt werden. Dazu wird der URI der externen Quelle benötigt. Dieser ist entweder bekannt, oder es kann auch eine Auswahlliste verschiedener Wissensquellen zur Hilfe genommen werden. Nachdem der Import realisiert wurde, erscheint er im Ontology Browser. Zusätzlich werden die benötigten Namensräume und deren Präfixe gesetzt. Abbildung 39 zeigt den Metadata-Tab nach dem Import von Dublin Core.

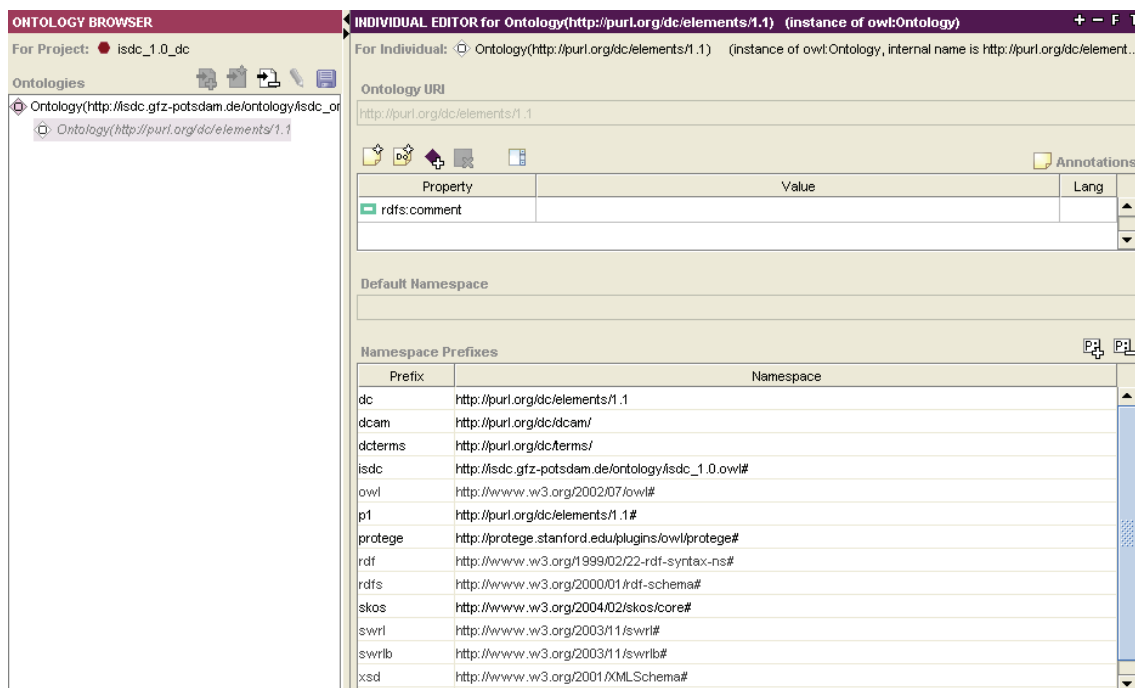


Abbildung 39: Dublin Core Import

Die nun zur Verfügung stehenden Elemente sollten mit in der Ontologie verwendet werden. Dublin Core hat z.B. das Element `title`, was z.B. die beim ISDC angelegte Datatype-Property `hasTitle` ersetzen könnte. Da die Modellierung einer Ontologie eine sehr komplexe Aufgabe ist, wurde bisher auf die Nutzung externer Quellen und deren Elemente verzichtet. Um eine gute Vernetzung von Wissen zu gewährleisten, sollte dies aber noch geändert werden.

Die Datatype-Properties für Zeit und Raum sind bereits ins Modell integriert und unter den Namen `hasSpatialCoverage` und `TemporalCoverage` angelegt. Sie kamen bisher nur exemplarisch zum Einsatz. Fehlende Werte müssen recherchiert und im Nachhinein ergänzt werden, um auch räumliche und zeitliche Abfragen beantworten zu können. Außerdem gilt es, die Abfragemechanismen für Datatype-Properties zu optimieren, ähnlich dem Prinzip des Queries-Tab. Der Queries-Tab bietet die Möglichkeit, Abfragen zu stellen, z.B. „Zeige alle Produkttypen an, die im Titel CHAMP verwenden!“. Als Ergebnis liefert diese Anfrage alle im Datenbestand vorhandenen CHAMP-Produkttypen (siehe Abbildung 40).

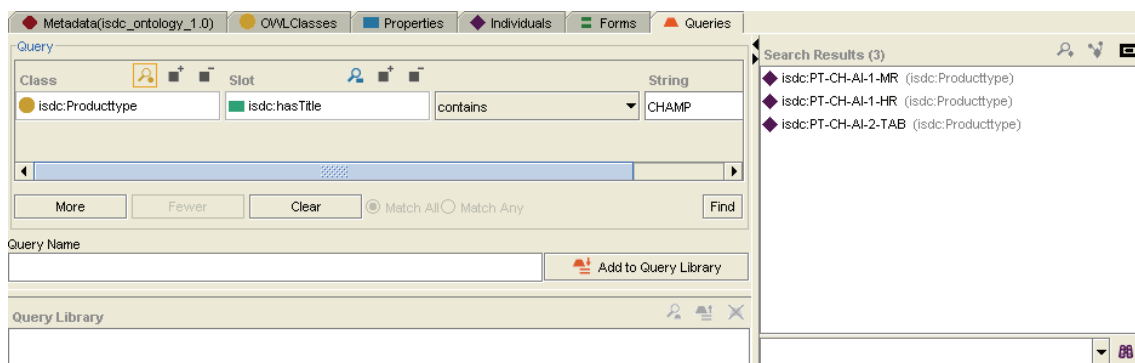


Abbildung 40: Queries-Tab

Protégé bietet darüber hinaus die Möglichkeit, Bilder in Ontologien zu integrieren. Das kann auch für die ISDC-Ontologie genutzt werden, um beispielsweise Instrumente, Plattformen, Institutionen oder Personen mit grafischen Informationen anzureichern. Zur Einbindung wird eine Datatype-Property benötigt. Für die exemplarische Umsetzung wurde die Datatype-Property `image` angelegt. Dem Instrument `SG-GWR-SG-056` wird über diese Property das entsprechende Bild zugeordnet. Dafür muss das `image`-Widget im Forms-Tab gesetzt sein. Nachdem das `image`-Widget vorhanden ist, kann dort der URI vom Bild eingetragen werden. Danach ist dann das Bild in Protégé sichtbar (siehe Abbildung 41).



Abbildung 41: Bild zum Instrument SG-GWR-SG-056

Bei der Modellierung der ISDC-Ontologie wurden bisher nur die Metadatendokumente berücksichtigt. Die einzelnen Datenprodukte der Produkttypen enthalten ebenfalls Metadaten, die auch noch ins Modell integriert werden sollten. Dazu müsste das Modell insofern geändert werden, dass die einzelnen Produkttypen in Klassen gewandelt werden. Die



Klasse Producttype hätte dann weitere Unterklassen für die jeweiligen Produkttypen. Die Unterklassen beinhalten dann das Individuum Produkttyp, sowie die dazugehörigen Datenprodukte. Um die Menge der einzelnen Datenprodukte zu bündeln, ist die Einführung einer Datenprodukt-Klasse sinnvoll. Protégé bietet die Funktionalität Individuen in Klassen zu konvertieren, so dass diese Ergänzung ohne weiteres durchführbar ist. Abbildung 42 zeigt schematisch die strukturellen Veränderungen am Beispiel des Produkttypen PT-CH-AI-1-HR. Unter der bereits existierenden Klasse Producttype wird eine Unterklasse PT-CH-AI-1-HR angelegt. Darunter liegt das eigentliche Individuum Produkttyp, was als Rechteck repräsentiert wird. Die Unterklasse DataProduct beinhaltet wiederum die Individuen der einzelnen Datenprodukte von der Klasse PT-CH-AI-1-HR.

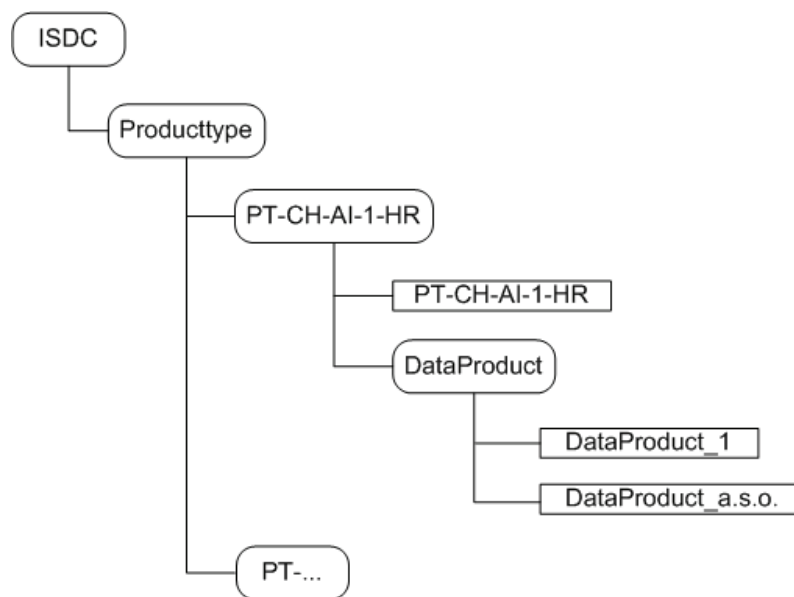


Abbildung 42: Integration der Datenprodukte

Abschließend muss aufbauend auf diese Ontologie eine Applikation entwickelt werden, welche die Inhalte der Ontologie verarbeiten und dem Nutzer in geeigneter Form zur Verfügung stellen kann. Im Sinne des Semantic Webs muss diese Applikation und deren Inhalte im WWW abrufbar und auch für andere Applikation nutzbar sein.

## 8 Fazit

Ziel dieser Arbeit war die Entwicklung einer Ontologie für das ISDC-Portal. Das ISDC verbindet mit dieser Weiterentwicklung eine bessere Informationsverwaltung und Wissensbereitstellung für die Nutzer. Bevor die eigentlichen Modellierungsarbeiten beginnen konnten, musste eine Bestandsaufnahme des vorliegenden Metadatenbestandes durchgeführt werden. Während der Bestandsanalyse konnten Widersprüche bzw. Defizite aufgedeckt werden. Ermittelte zusätzliche Elemente wurden in den Metadateien ergänzt. Nachdem die semantischen Zusammenhänge deutlich waren, konnten die eigentlichen Modellierungsarbeiten der Ontologie mit der Software Protégé beginnen. Protégé ist ein mächtiges Modellierungstool, was von einer großen Community gepflegt und erweitert wird. Im Rahmen dieser Masterarbeit konnten nicht alle beim ISDC vorkommenden Metadateien ins Modell aufgenommen werden. Die Modellierungsarbeiten beschränkten sich auf eine vorher definierte Auswahl von Dateien, da der Datenbestand des ISDCs für eine prototypische Umsetzung zu umfangreich ist.

Die festgelegten Anwendungsfälle geben nicht nur Aufschluss über die gewünschten Funktionalitäten der Domänen-Ontologie, sie können auch zur Überprüfung des Modells herangezogen werden. Als Endergebnis konnte mittels Protégé eine OWL-Datei der Version 1.0 erarbeitet werden. In dieser Datei ist das bisher vorhandene Wissen abgebildet. Das Wissen stammt aus dem ISDC-Metadatenbestand und wurde durch andere Quellen ergänzt, welche die Wissensbasis vervollständigen. Dazu zählt auch das Wissen der ISDC-Mitarbeiter. Darüber hinaus wurde der Dublin Core-Standard in die Ontologie importiert, um dort definierte Elemente nutzen zu können. Die Verwendung und Verknüpfung von verschiedenen Konzepten und Wissensbasen ist ein Grundsatz des Semantic Webs und muss daher berücksichtigt werden. Neben Dublin Core existieren diverse andere Beispiele, z.B. FOAF, die ebenfalls integriert werden können. Die vollständige Analyse von nutzbaren Elementen und deren praktische Anwendung waren im Rahmen dieser Arbeit nicht möglich, sollten aber im nächsten Schritt noch berücksichtigt und integriert werden.

Zur Visualisierung von Instrumenten, Plattformen, Institutionen und Personen können auch Bilder mit ins Modell aufgenommen werden. Das wurde exemplarisch durchgeführt. Fehlende Bilder sollten ebenfalls ergänzt werden. Der bis dato erarbeitete Wissenstand kann im WWW über die URI [http://isdc.gfz-potsdam.de/ontology/isdc\\_1.0.owl](http://isdc.gfz-potsdam.de/ontology/isdc_1.0.owl) abgerufen werden. Somit steht der Datenbestand auch für andere Verwendungszwecke zur Verfügung. Um die Ontologie und deren Inhalt vollständig nutzen zu können, muss eine geeignete Applikation entwickelt werden. Als Design-Vorbild kann die Wissensmaschine eyePloer dienen. Das eyePloer-System realisiert eine grafische Informationssuche und -darstellung. Suchergebnisse werden in verschiedene Kategorien klassifiziert ausgegeben. Des Weiteren können bei der Informationsausgabe Assoziationen zwischen den einzelnen Begriffen abgeleitet werden. Der Nutzer erhält dadurch einen umfangreichen Überblick über den gesuchten Begriff, in seinen verschiedenen Domänen und Funktionen. Die Bearbeitung des bisherigen OWL-Bestandes sollte in der Zukunft fortgeführt und ausgebaut werden, so dass ein zeitnahe Einsatz in Verbindung mit einer noch zu entwickelnden Applikation möglich ist.

## Literatur

- [1] W3C Semantic Web Activity. SKOS Simple Knowledge Organization System - Home Page. <http://www.w3.org/2004/02/skos/>, 27. Mai 2010.
- [2] Stock Artists Alliance. Glossary of Terms - Ontology. <http://www.photometadata.org/node/53#NAA>, 11. August 2010.
- [3] Altova. XMLSPY - XML Technologies. <http://www.altova.com/>, 21. Juni 2010.
- [4] Tim Berners-Lee. Uniform Resource Identifier (URI): Generic Syntax. <http://tools.ietf.org/html/rfc3986#section-3.1>, 01. Oktober 2010.
- [5] W3C Tim Berners-Lee. Notation 3. <http://www.w3.org/DesignIssues/Notation3.html>, 25. Mai 2010.
- [6] BODC. Welcome to British Oceanographic Data Centre. <http://www.bodc.ac.uk/>, 17. Juni 2010.
- [7] Citizendium. Welcome to Citizendium. <http://en.citizendium.org/>, 17. Juni 2010.
- [8] CO-ODE. The Manchester OWL Syntax. [http://www.co-ode.org/resources/reference/manchester\\_syntax/](http://www.co-ode.org/resources/reference/manchester_syntax/), 06. Oktober 2010.
- [9] DCMI. The DublinCore Metadata Initiative. <http://dublincore.org/>, 26. Juni 2010.
- [10] B.C. Grau A. Kalyanpur E. Sirin, B. Parsia and Y. Katz. Pellet: A Practical OWL-DL Reasoner. [www.mindswap.org/papers/PelletJWS.pdf](http://www.mindswap.org/papers/PelletJWS.pdf), 24. August 2010.
- [11] Erik Endres. Reasoners for the Semantic Web. [www.dfki.de/~kipp/seminar\\_ws0607/.../EricEndres-Reasoner.pdf](http://www.dfki.de/~kipp/seminar_ws0607/.../EricEndres-Reasoner.pdf), 24. August 2010.
- [12] etutorials. Misc - Practical Resource Description Framework (RDF) - Chapter 2. RDF: Heart and Soul - 2.5 RDF Serialization: N3 and N-triples. <http://etutorials.org/>, 25. Mai 2010.

- [13] Stanford Center for Biomedical Informatics Research. Documentation - Introductory Papers - The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications. <http://protege.stanford.edu/>, 30. September 2010.
- [14] Stanford Center for Biomedical Informatics Research. Documentation - Introductory Papers - Weaving the Biomedical Semantic Web with the Protégé OWL Plugin. <http://protege.stanford.edu/>, 30. September 2010.
- [15] Gesellschaft für Informatik e.V. Service - Informatiklexikon - Ontologie(n). <http://www.gi-ev.de>, 19. Mai 2010.
- [16] GCMD. Ancillary Description Writer's Guide. <http://gcmd.nasa.gov/User/suppguide/>, 17. Juni 2010.
- [17] GCMD. Directory Interchange Format (DIF) Writer's Guide. <http://gcmd.gsfc.nasa.gov/User/difguide/difman.html>, 17. Juni 2010.
- [18] GFZ. Struktur - GFZ-Organigramm. <http://www.gfz-potsdam.de/>, 06. Mai 2010.
- [19] GFZ. Willkommen am Helmholtz-Zentrum Potsdam Deutsches Geoforschungszentrum GFZ. <http://www.gfz-potsdam.de/>, 06. Mai 2010.
- [20] Open Healthcare Group. RDF - qnamequagmire.html - XML Schema's Use of Qnames. <http://www.openhealth.org/>, 25. Mai 2010.
- [21] Thomas.R. Gruber. Technical reports - Listing of Reports by KSL Report Number - KSL-92-71 - A Translation Approach to Portable Ontology specifications (1993). <http://www-ksl.stanford.edu/>, 19. Mai 2010.
- [22] Tim Hartmann. Das Semantic Web. Technical report, WS.
- [23] Monica Holland. Global Change Master Directory - Discover Earth Science Data and Services. <http://gcmd.nasa.gov/>, 17. Juni 2010.
- [24] Matthew Horridge. OWLviz. [http://protegewiki.stanford.edu/wiki/Main\\_Page](http://protegewiki.stanford.edu/wiki/Main_Page), 07. Oktober 2010.

- [25] ISDC-Team. Welcome to The Information System and Data Center for Geoscientific Data. <http://isdc.gfz-potsdam.de/>, 06. Mai 2010.
- [26] Microsoft. Search Glossary - Ontology. <http://www.microsoft.com/enterprisesearch/en/us/search-glossary.aspx#O>, 11. August 2010.
- [27] N.F. Noy and D.L. McGuinness. Ontology Development 101: A Guide to Creating Your First Ontology. [http://protege.stanford.edu/publications/ontology\\_development/ontology101.pdf](http://protege.stanford.edu/publications/ontology_development/ontology101.pdf), 12. August 2010.
- [28] University of Sussex. Data Properties and Individuals in Protégé. <http://www.informatics.sussex.ac.uk/courses/araip/summer08/dataTypesProtege.pdf>, 01. Oktober 2010.
- [29] University of Sussex. Documentation - Protégé-OWL - Edition 1.0, For Protégé 3.x - v1.0 - A Practical Guide to Building OWL Ontologies Using the Protégé-OWL Plugin and CO-ODE Tools Edition 1.0. <http://protege.stanford.edu/>, 02. Oktober 2010.
- [30] pagina GmbH. XML-Hintergründe - Pagina-Das Kompendium - themenkomplex ii - Einheitliche Zeichenkodierung-Grundlage der Systemunabhängigkeit - unicode (iso 10646). <http://www.pagina-online.de>, 08. Mai 2010.
- [31] Sebastian Rudolph und York Sure Pascal Hitzler, Markus Krötzsch. *Semantic Web*. Springer-Verlag Berlin Heidelberg, 2008. ISBN: 978-3-540-33993-9.
- [32] Sabine Pfeiffer. Verknüpfung von geowissenschaftlichen ISDC-Produkten mit Semantic-Web-Techniken zur Erschließung inhaltlicher Zusammenhänge. Technical report, September.
- [33] Protege. General Documentation - Which Version of Protege Should I Use? Protege 3 or Protege 4? - Choosing Between Versions of Protege. [http://protegewiki.stanford.edu/wiki/Main\\_Page](http://protegewiki.stanford.edu/wiki/Main_Page), 18. August 2010.

- [34] Protégé. Welcome to Protégé. <http://protege.stanford.edu/>, 26. Mai 2010.
- [35] Marcel Raisingier. Vergleich RDF/RDFS und OWL. Technical report.
- [36] Semantisches Web. Technologien - Beweise, Vertrauen, Digitale Signaturen. <http://semantisches-web.net/>, 16. Juni 2010.
- [37] Heiner Stuckenschmidt. *Ontologien - Konzepte, Technologien und Anwendungen*. Springer-Verlag Berlin Heidelberg, 2009. ISBN: 978-3-540-79330-4.
- [38] Uni Stuttgart. Geschichte der Logik. [www.fmi.uni-stuttgart.de/szs/teaching/.../10-einfuehrung-4.pdf](http://www.fmi.uni-stuttgart.de/szs/teaching/.../10-einfuehrung-4.pdf), 20. August 2010.
- [39] James Hendler und Ora Lassila Tim Berners-Lee. The Semantic Web. Technical report, Mai.
- [40] J. D. Novak und A. J. Canas. Documented - Research Publications Associated with Cmaptools - The Theory Underlying Concept Maps and How to Construct Them. <http://cmap.ihmc.us/>, 24. Mai 2010.
- [41] Dean Allemang und Jim Hendler. *Semantic Web for the Working Ontologist*. Wiley Publishing, Inc., 2009. ISBN: 978-0-470-41801-7.
- [42] Jens Behrendt und Prof. Dr. Klaus Zeppenfeld. *Informatik im Fokus - Web 2.0*. Springer-Verlag Berlin Heidelberg, 2008. ISBN: 978-3-540-73120-7.
- [43] W3C. ... or view all - OWL Web Ontology Language Overview. <http://www.w3.org/>, 20. Mai 2010.
- [44] W3C. ... or view all - XML - Namespaces in XML 1.0 (Third Edition) - W3C Recommendation 8 December 2009. <http://www.w3.org/>, 08. Mai 2010.
- [45] W3C. Turtle - Terse RDF Triple Language. <http://www.w3.org/TeamSubmission/turtle/>, 25. Mai 2010.
- [46] Semantic Web Wiki. FOAF. <http://semanticweb.org/>, 27. Mai 2010.

## A Abkürzungsverzeichnis

**ALC** Attributive Language with Complement

**API** Application Programming Interface

**BODC** British Oceanographic Data Centre

**CHAMP** CHallenging Mini-Satellite Payload

**COE** CmapTools Ontology Editor

**DAML-ONT** DARPA Agent Markup Language - Ontology

**DCMI** Dublin Core Metadata Initiative

**DIF** Directory Interchange Format

**DL** Deskriptive Logik (Description Logic)

**DRZ** Daten- und Rechenzentrum

**DTD** Document Type Definition

**GCMD** Global Change Master Directory

**GFZ** Helmholtz-Zentrum Potsdam, Deutsches GeoForschungsZentrum

**GGP** Global Geodynamic Project

**GNSS** Global Navigation Satellite System

**GRACE** Gravity Recover And Climate Experiment

**IHMC** Institute For Human And Machine Cognition

**ISDC** Information System and Data Center

**ISO** International Organization for Standardization



- NASA** National Aeronautics and Space Administration
- NDG** Nerc Data Grid
- NERC** Natural Environment Research Council
- N3** Notation 3
- OIL** Ontology Inference Layer
- OWL** Web Ontology Language
- Qname** Qualified name
- RDF** Resource Description Framework
- RDFS** RDF Schema
- SPARQL** SPARQL Protocol and RDF Query Language
- SWOOP** Semantic Web Ontology Editor
- SWRL** Semantic Web Rule Language
- TSX-TOR** TerraSAR-X - Tracking, Occultation and Ranging
- URI** Uniform Resource Identifier
- UTF-8** 8-Bit Unicode Transformation Format
- WWW** World Wide Web
- W3C** World Wide Web Consortium
- XML** Extensible Markup Language

## B ISDC-Metadatenbestand

Tabelle 5: Product\_type

<b>Product_type</b>		
Entry_ID		
Entry_Title		
Data_Set_Citation		
	Dataset_Creator	
	Dataset_Title	
	Dataset_Series_Name	
	Dataset_Release_Date	
	Dataset_Release_Place	
	Dataset_Publisher	
	Version	
	Issue_Identification	
	Data_Presentation_Form	
	Other_Citation_Details	
	Online_Resource	
Personnel		
	Role	
		DATA CENTER CONTACT
		INVESTIGATOR
		TECHNICAL CONTACT
		DIF AUTHOR

B ISDC-Metadatenbestand

---

	First_Name	
	Middle_Name	
	Last_Name	
	Email	
	Phone	
	Fax	
	Contact_Address	
		Address
		City
		Province_or_State
		Postal_Code
		Country
Discipline		
	Discipline_Name	
	Subdiscipline	
	Detailed_Subdiscipline	
Parameters		
	Category	
	Topic	
	Term	
	Variable_Level_1	
	Variable_Level_2	
	Variable_Level_3	
	Detailed_Variable	
ISO_Topic_Category		
Keyword		
Sensor_Name		

B ISDC-Metadatenbestand

---

	Short_Name	
	Long_Name	
Source_Name		
	Short_Name	
	Long_Name	
Temporal_Coverage		
	Start_Date	
	Stop_Date	
Paleo_Temporal_Coverage		
	Paleo_Start_Date	
	Paleo_Stop_Date	
	Chronostratigraphic_Unit	
		Eon
		Era
		Period
		Epoch
		Stage
		Detailed_Classification
Data_Set_Progress		
	PLANNED	
	IN WORK	
	COMPLETE	
Spatial_Coverage		
	Southernmost_Latitude	
	Northernmost_Latitude	
	Westernmost_Latitude	
	Easternmost_Latitude	

B ISDC-Metadatenbestand

---

	Minimum_Altitude	
	Maximum_Altitude	
	Minimum_Depth	
	Maximum_Depth	
Location		
	Location_Category	
	Location_Type	
	Location_Subregion1	
	Location_Subregion2	
	Location_Subregion3	
	Detailed_Location	
Data_Resolution		
	Latitude_Resolution	
	Longitude_Resolution	
	Horizontal_Resolution_Range	
	Vertical_Resolution	
	Vertical_Resolution_Range	
	Temporal_Resolution	
	Temporal_Resolution_Range	
Project		
	Short_Name	
	Long_Name	
Quality		
Access_Constraints		
Use_Constraints		
Data_Set_Language		
Originating_Center		

B ISDC-Metadatenbestand

---

Data_Center		
	Data_Center_Name	
		Short_Name
		Long_Name
	Data_Center_URL	
	Data_Set_ID	
	Personnel	
Distribution		
	Distribution_Media	
	Distribution_Size	
	Distribution_Format	
	Fees	
Multimedia_Sample		
	File	
	URL	
	Format	
	Caption	
	Description	
Reference		
Summary		
Related_URL		
	URL_Content_Type	
		Type
		Subtype
	URL	
	Description	
Parent_DIF		

B ISDC-Metadatenbestand

---

IDN_Node		
	Short_Name	
	Long_Name	
Originating_Metadata_Node		
Metadata_Name		
Metadata_Version		
DIF_Creation_Date		
Last_DIF_Revision_Date		
DIF_Revision_History		
Future_DIF_Review_Date		
Private		

Tabelle 6: Project

<b>Project</b>		
Entry_ID		
Name		
Summary		
Data_Center		
	ID	
	Short_Name	
Metadata_Name		
Metadata_Version		
Parameters		
	Category	
	Topic	
	Term	

B ISDC-Metadatenbestand

---

	Variable_Level_1	
	Variable_Level_2	
	Variable_Level_3	
	Detailed_Variable	
Keyword		
Platform		
	ID	
	Short_Name	
	Long_Name	
	Location	
		Location_Category
		Location_Type
		Location_Subregion1
		Location_Subregion2
		Location_Subregion3
		Detailed_Location
	Institution	
		ID
		Short_Name
		Long_Name
		URL

Tabelle 7: Platform

<b>Platform</b>	
Entry_ID	
Name	



---

B ISDC-Metadatenbestand

---

Data_Center	
	ID
	Short_Name
	Long_Name
	URL
Summary	
Metadata_Name	
Metadata_Version	
Platform_Identification	
	Platform_Category
	Platform_Series
	Status
Synonymous_Platform_Name	
Parameters	
	Category
	Topic
	Term
	Variable_Level_1
	Variable_Level_2
	Variable_Level_3
	Detailed_Variable
Keyword	
Location	
	Location_Category
	Location_Type
	Location_Subregion1
	Location_Subregion2

---

B ISDC-Metadatenbestand

---

	Location_Subregion3
	Detailed_Location
Spatial_Coverage	
	Southernmost_Latitude
	Northernmost_Latitude
	Westernmost_Latitude
	Easternmost_Latitude
	Minimum_Altitude
	Maximum_Altitude
	Minimum_Depth
	Maximum_Depth
Instrument	
	ID
	Short_Name
	Long_Name
Institution	
	ID
	Short_Name
	Long_Name
	URL
Related_URL	
	URL
	Description
IDN_Node	
	Short_Name
	Long_Name
DIF_Creation_Date	

Last_DIF_Revision_Date	
------------------------	--

Tabelle 8: Instrument

<b>Instrument</b>	
Entry_ID	
Name	
Summary	
Sample_Image	
Instrument_Associated_Product_Type	
Data_Center	
	ID
	Short_Name
	Long_Name
	URL
Institution	
	ID
	Short_Name
	Long_Name
	URL
Metadata_Name	
Metadata_Version	
Instrument_Identification	
	Instrument_Category
	Instrument_Class
	Short_Name
	more

B ISDC-Metadatenbestand

---

Integrated_Sensor	
Associated_Platform	
Frequency_Information	
	Wavelength_Keyword
	Number_Channels
	Frequency_Coverage
	Frequency_Resolution
Instrument_Logistic	
	Data_Rate
	Instrument_Start_Date
	Instrument_Stop_Date
	Instrument_Owner
Related_URL	
	URL
	Description
IDN_Node	
	Short_Name
	Long_Name
DIF_Creation_Date	
Last_DIF_Revision_Date	

Tabelle 9: Institution

<b>Institution</b>		
Entry_ID		
Name		
Summary		

B ISDC-Metadatenbestand

---

Metadata_Name		
Metadata_Version		
Personnel		
	Role	
	First_Name	
	Middle_Name	
	Last_Name	
	Email	
	Phone	
	Fax	
	Contact_Address	
		Address
		City
		Province_or_State
		Postal_Code
		Country
Location		
	Location_Category	
	Location_Type	
	Location_Subregion1	
	Location_Subregion2	
	Location_Subregion3	
	Detailed_Location	
Related_URL		
	URL	
	Description	
IDN_Node		

---

B ISDC-Metadatenbestand

---

	Short_Name	
	Long_Name	
DIF_Creation_Date		
Last_DIF_Revision_Date		

## C Elemente der Klassen

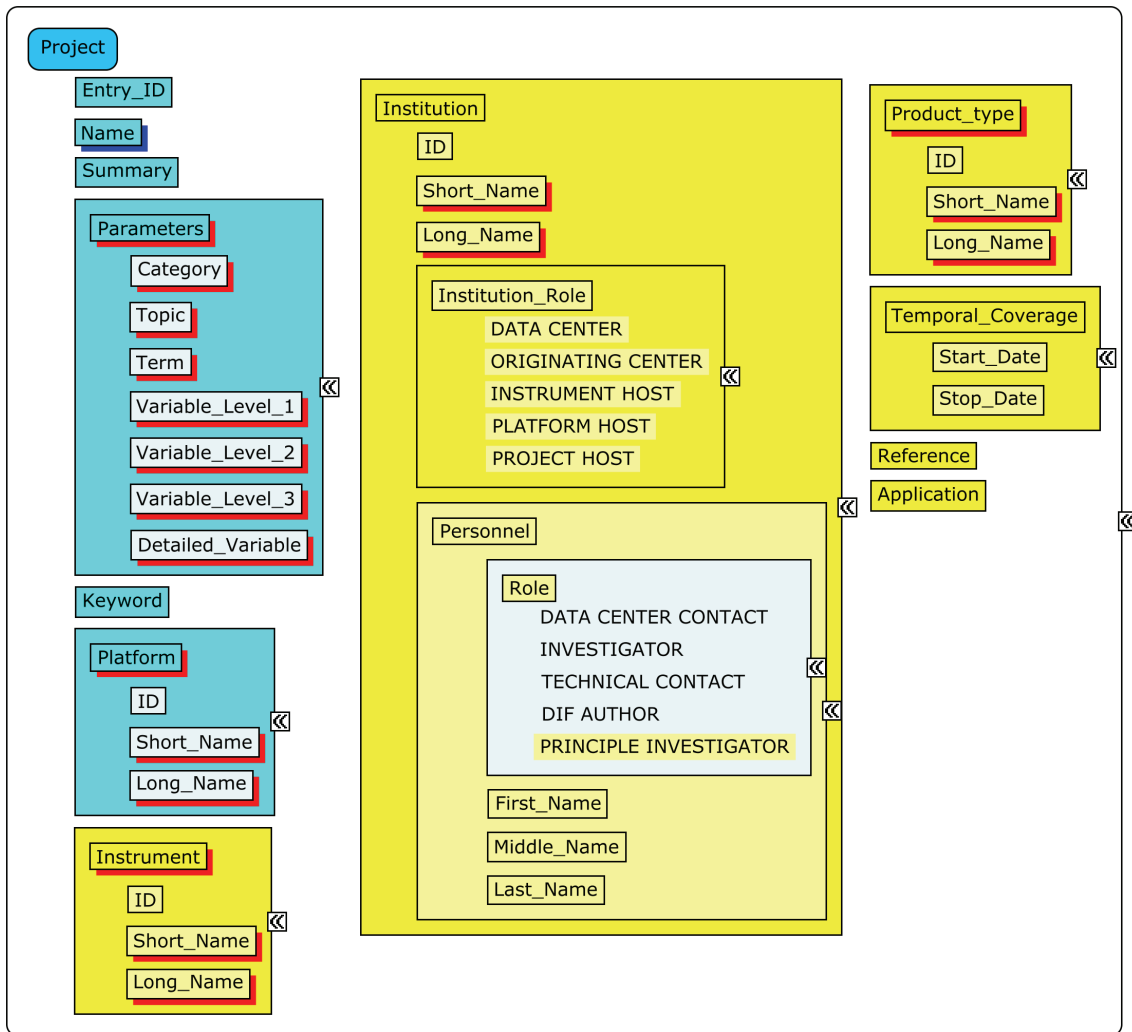


Abbildung 43: Elemente der Klasse Projekt

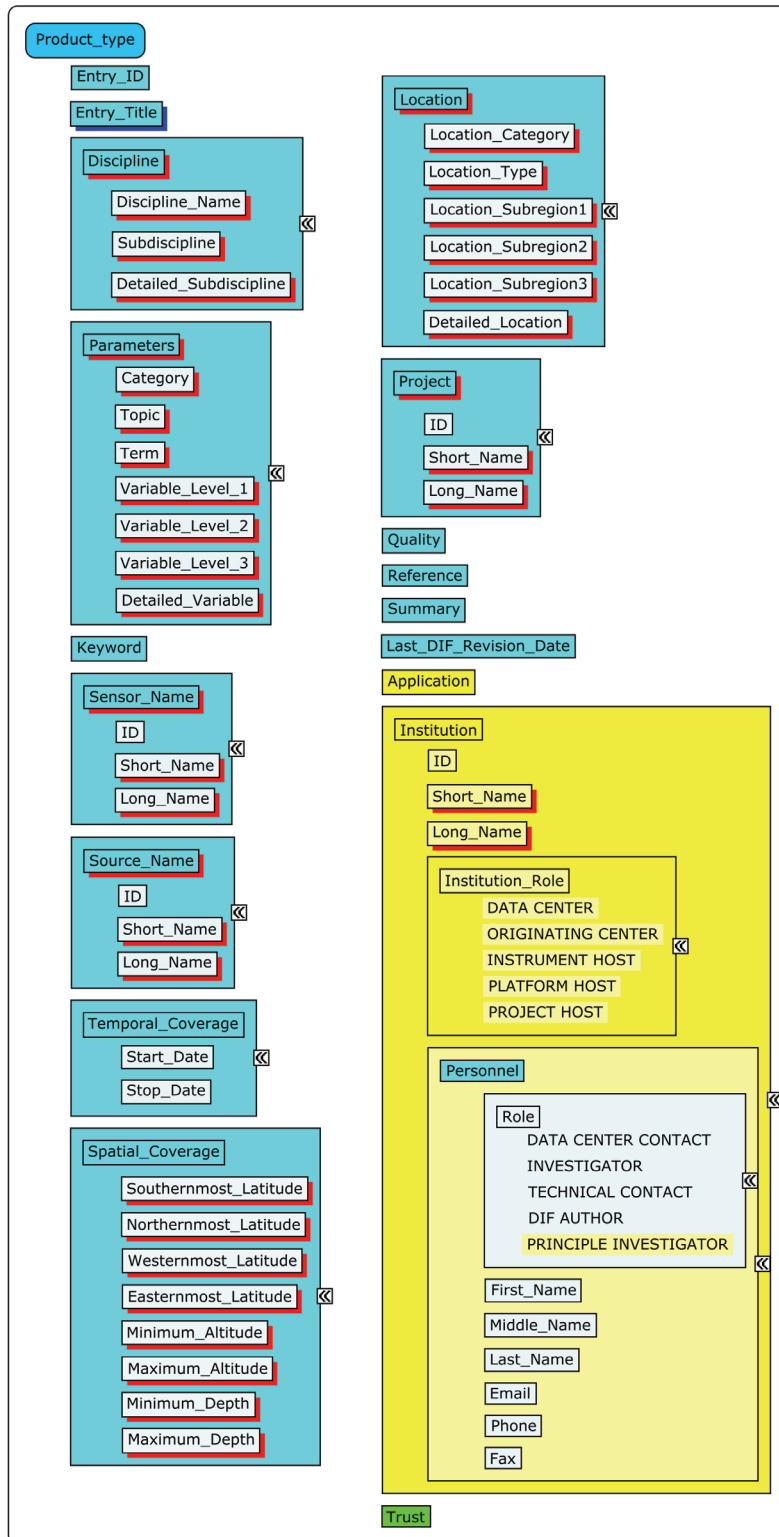


Abbildung 44: Elemente der Klasse Produkttyp



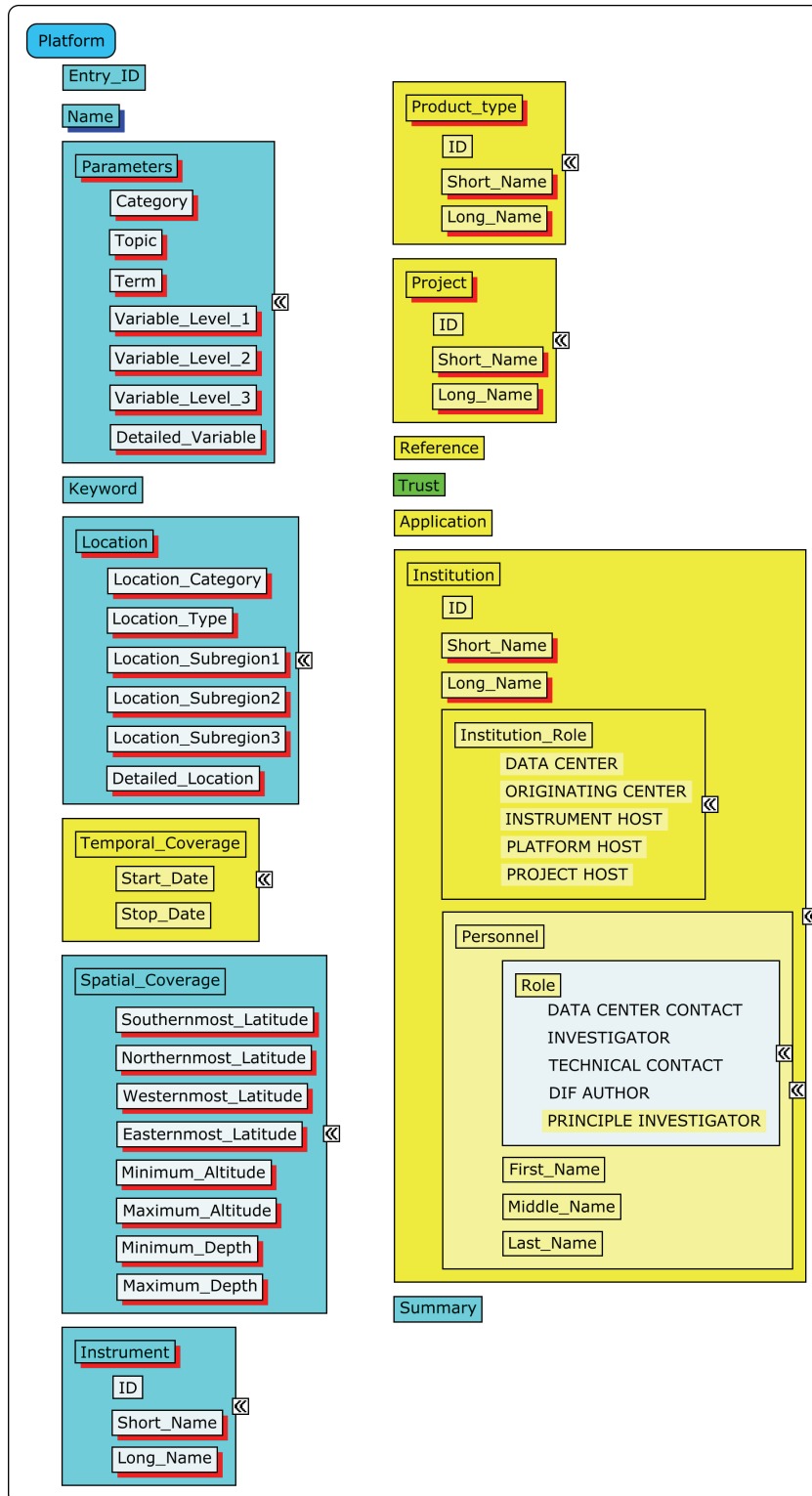


Abbildung 45: Elemente der Klasse Plattform

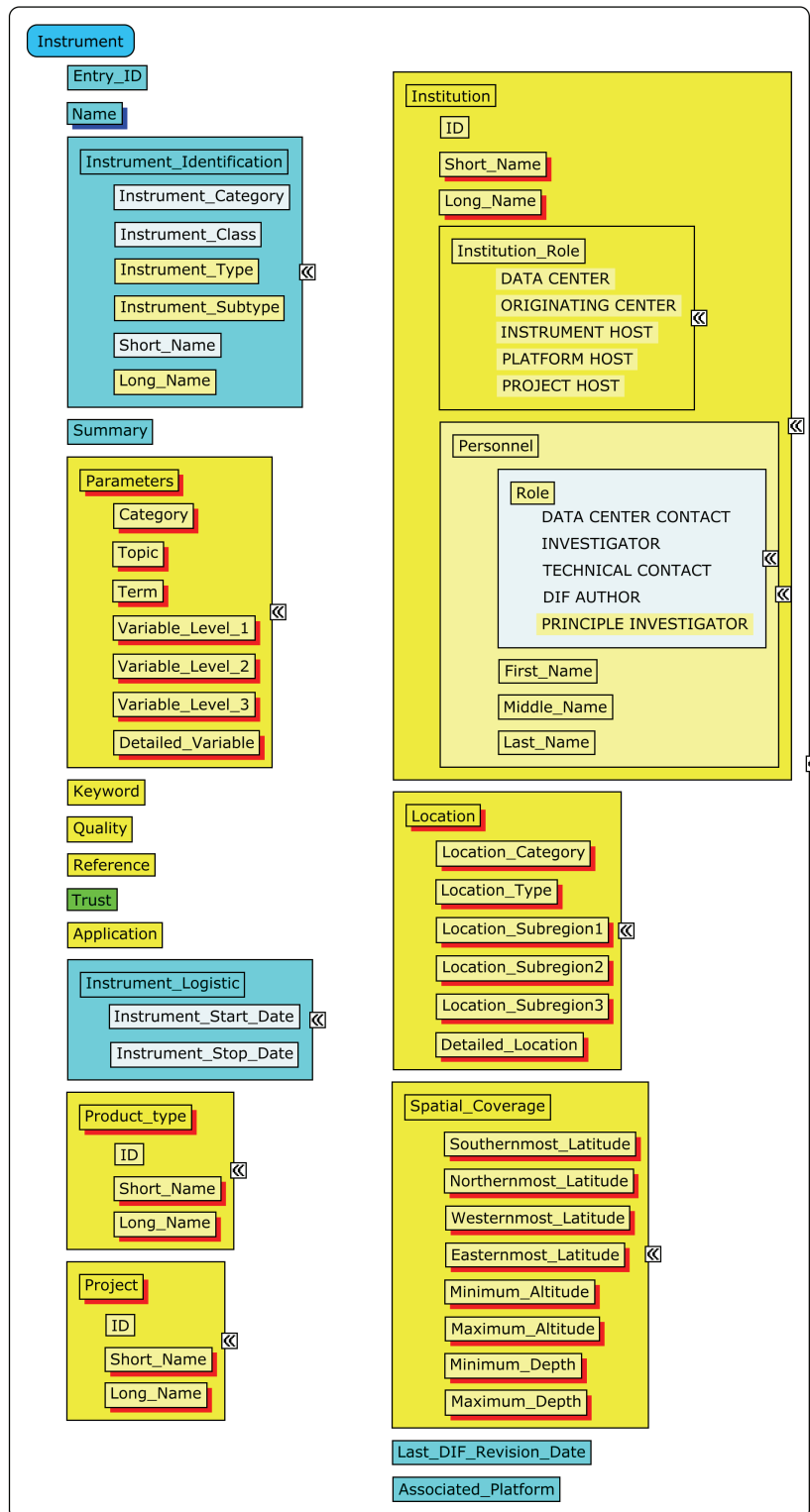


Abbildung 46: Elemente der Klasse Instrument

## C Elemente der Klassen

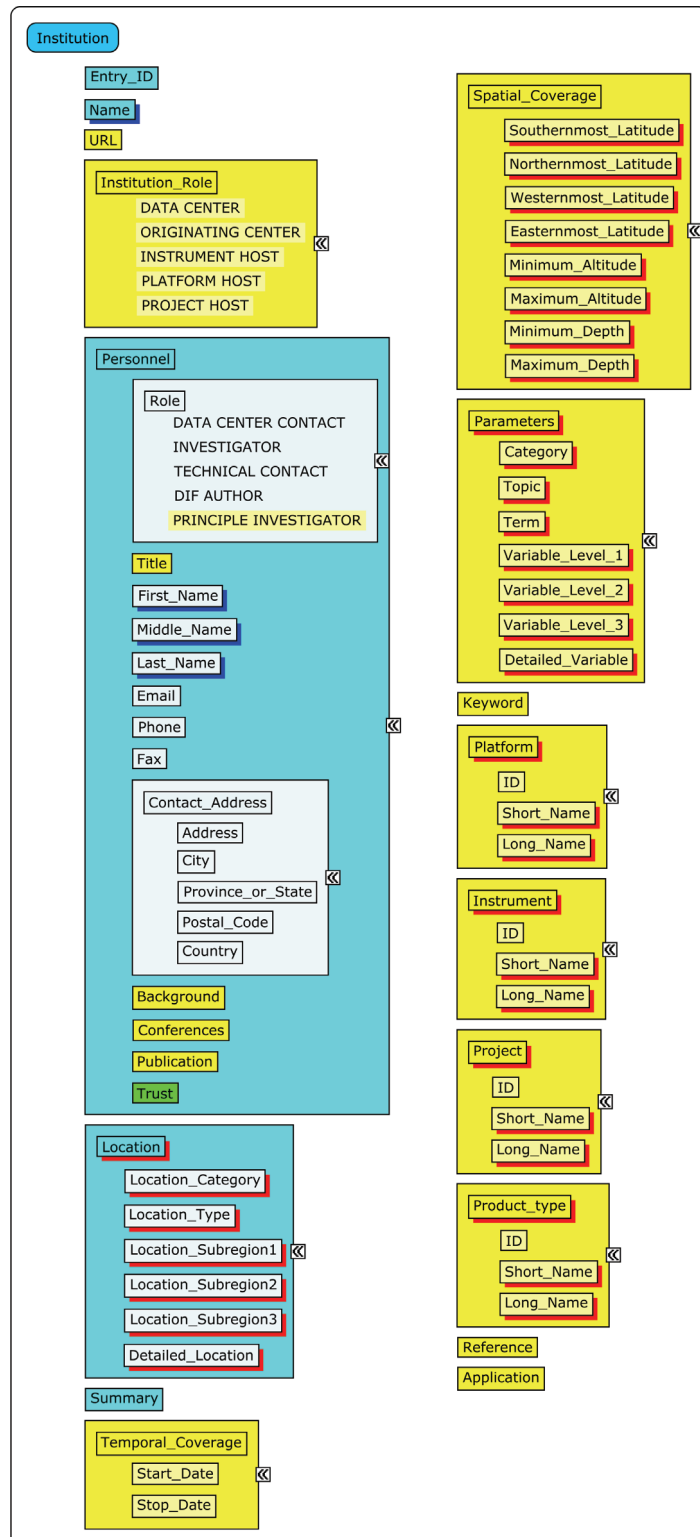


Abbildung 47: Elemente der Klasse Institution

## D Konkordanzen

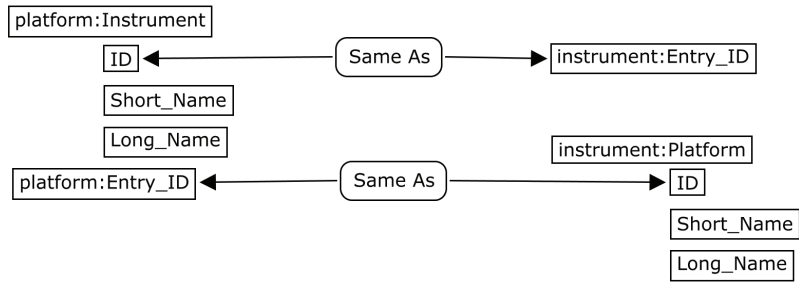


Abbildung 48: Konkordanz: Instrument - Plattform

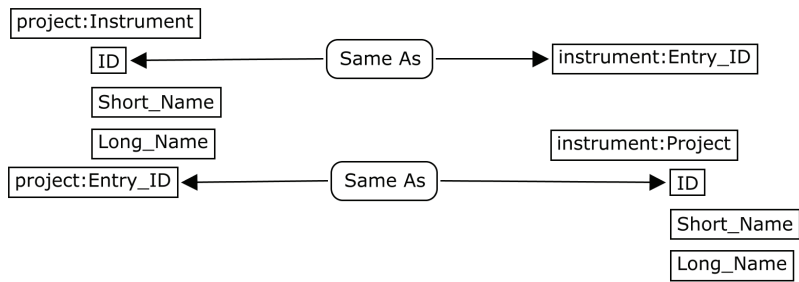


Abbildung 49: Konkordanz: Instrument - Projekt

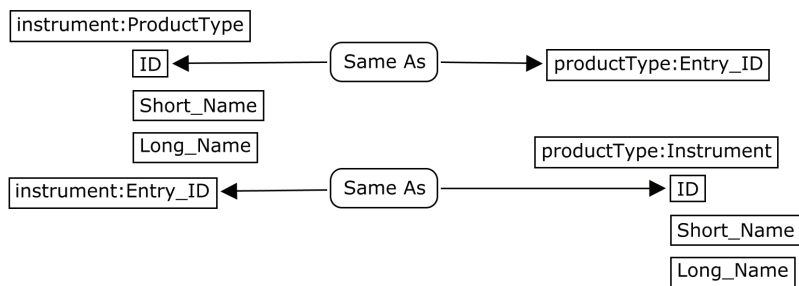


Abbildung 50: Konkordanz: Instrument - Produkttyp

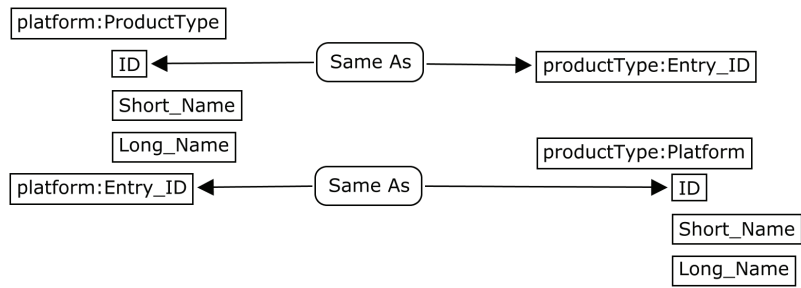


Abbildung 51: Konkordanz: Produkttyp - Plattform

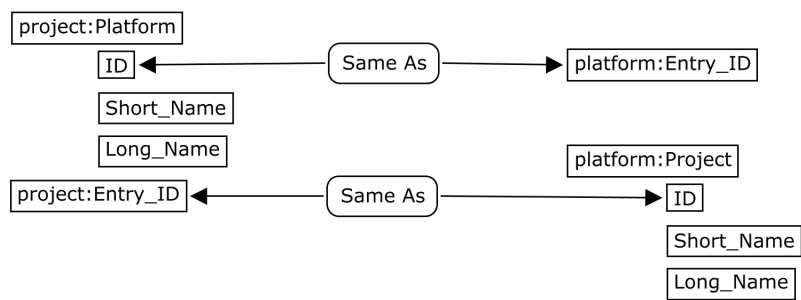


Abbildung 52: Konkordanz: Projekt - Plattform

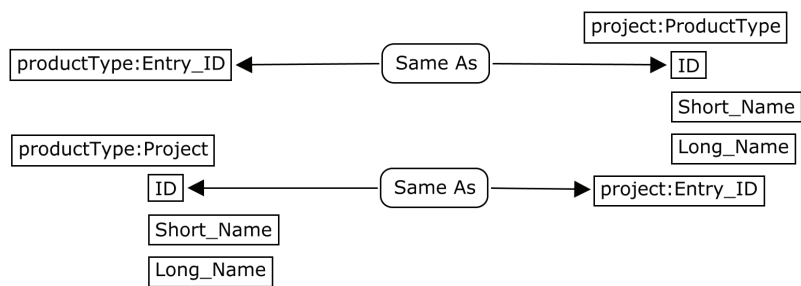


Abbildung 53: Konkordanz: Projekt - Produkttyp

## E Zeichen der Prädikatenlogik

Bezeichnung	Symbol	Bemerkung
Negation	$\neg$	Junktor
Disjunktion	$\vee$	Junktor
Konjunktion	$\wedge$	Junktor
Äquivalenz	$\equiv$	Junktor
Implikation	$\rightarrow$	Junktor
Allquantor	$\forall$	
Existenzquantor	$\exists$	
Logische Folgerung	$\models$	

Tabelle 10: Zeichen der Prädikatenlogik