NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS

SCHOOL OF SCIENCE
DEPARTMENT OF INFORMATICS AND TELECOMMUNICATIONS

POSTGRADUATE STUDIES
“INFORMATION AND DATA MANAGEMENT”

MASTER THESIS

Encoding and Validation of Earth Observation Metadata using Schema.org and SHACL

Despina - Athanasia I. Pantazi

Supervisor: **Manolis Koubarakis,** Professor

ATHENS

November 2018

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
"ΔΙΑΧΕΙΡΙΣΗ ΠΛΗΡΟΦΟΡΙΑΣ ΚΑΙ ΔΕΔΟΜΕΝΩΝ"

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Κωδικοποίηση και επαλήθευση μεταδεδομένων τηλεπισκόπισης με τη χρήση του λεξιλογίου Schema.org και της γλώσσας SHACL

Δέσποινα - Αθανασία Ι. Πανταζή

Επιβλέπων: **Μανόλης Κουμπαράκης**, Καθηγητής

ΑΘΗΝΑ

Νοέμβριος 2018

MASTER THESIS

Encoding and Validation of Earth Observation Metadata using Schema.org and SHACL

Despina - Athanasia I. Pantazi
R.N.: M1527

SUPERVISOR: **Manolis Koubarakis,** Professor

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Κωδικοποίηση και επαλήθευση μεταδεδομένων τηλεπισκόπισης με τη χρήση του λεξιλογίου Schema.org και της γλώσσας SHACL

Δέσποινα - Αθανασία Ι. Πανταζή
Α.Μ.: M1527

ΕΠΙΒΛΕΠΩΝ: **Μανόλης Κουμπαράκης,** Καθηγητής

ABSTRACT

The current thesis presents a schema.org vocabulary extension for encoding Earth observation (EO) datasets and their properties. It is based on the vocabulary defined in OGC 17-003 specification, which describes a GeoJSON and JSON-LD encoding of Earth observation metadata for datasets. We updated this vocabulary in order to make it simpler, as schema.org principals demand, without excluding any information provided for the EO datasets. We also used Shapes Constraint Language (SHACL) in order to create a shapes graph for our schema.org extension. This shapes graph includes constraints regarding the properties of our vocabulary, so that we can model and validate RDF graphs constructed by EO data. We conclude by providing detailed examples for annotating and validating EO datasets based on our schema.org vocabulary extension.

SUBJECT AREA: Semantic Web

KEYWORDS: Earth observation, linked data, schema.org, metadata, datasets, SHACL, semantic web

ΠΕΡΙΛΗΨΗ

Στην παρούσα διπλωματική εργασία παρουσιάζουμε μία επέκταση του λεξιλογίου schema.org για την κωδικοποίηση συνόλων δεδομένων και των χαρακτηριστικών τους, που αφορούν τη τηλεπισκόπιση. Η επέκταση αυτή είναι βασισμένη στο έγγραφο – οδηγία OGC 17-003, στο οποίο περιγράφεται η κωδικοποίηση μεταδεδομένων που αφορούν τη γεωσκόπηση, με τη χρήση των προτύπων GeoJSON και JSON-LD. Ανανεώσαμε αυτό το λεξιλόγιο απλουστεύοντας τη δομή του, έτσι ώστε να συμβαδίζει με τις απαιτήσεις του τύπου μικροδεδομένων schema.org, χρησιμοποιώντας όμως όλη την πληροφορία που δίνεται για τα γεωχωρικά δεδομένα. Επιπλέον, χρησιμοποιήσαμε τη γλώσσα περιορισμών SHACL για να δημιουργήσουμε γράφους περιορισμών για το λεξιλόγιό μας. Οι γράφοι περιορισμών στοχεύουν στη μοντελοποίηση και επικύρωση των γράφων δεδομένων τύπου RDF που δημιουργούνται από τα δεδομένα γεωσκόπησης. Καταλήγοντας, παραθέτουμε ένα σύνολο λεπτομερών παραδειγμάτων για να κατανοηθεί ο τρόπος που εφαρμόζεται και επικυρώνεται η επέκταση του λεξιλογίου μας σε σύνολα δεδομένων που αφορούν τη τηλεπισκόπιση.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Σημασιολογικός Ιστός

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: τηλεσκόπιση, διασυνδεδεμένα δεδομένα, μεταδεδομένα, σύνολα δεδομένων, σημασιολογικός ιστός

To my family

ACKNOWLEDGEMENTS

Firstly, I would like to thank professor Manolis Koubarakis for giving me the opportunity to be part of Knowledge Representation, Reasoning and Analytics research group (KR&R) and work on this project. I would also like to thank Yves Coene for the valuable feedback he provided. This thesis was funded by the H2020 project Copernicus App Lab, Grand Agreement number 730124.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF LISTINGS

1. INTRODUCTION

Datasets produced by Earth observation satellites are one of the most important assets we have nowadays. This data help us to secure our environment and understand our planet better, as it provides detailed information about planet Earth's physical, chemical and biological systems. There are many different kinds of Earth observations, including altimeter or seismograph photographs, radar and sonar images and analyses of water or soil samples[1]. All these various kinds of EO data, extracted by satellite images, help scientists to understand and protect our planet in more efficient ways. In addition, the applications the scientists can develop by using this data supply international relief agencies with warnings and possible solutions in case of emergency environmental operations and natural disasters, such as floods, hurricanes, tornadoes, volcanic eruptions, earthquakes, tsunamis, and other geologic processes[2].

All these different kinds of EO data are produced by many thousands of scientific observation instruments. It is crucial to make this data available on the Web as linked data in order to increase their use by developers that might not be experts in EO. In this way, great amounts of data that are generated fast, can be made "interoperable" and more valuable when they are linked together.

Search engines like Google and Yahoo have progressed dramatically from being able to find documents containing user keywords and order them according to importance using algorithms such as PageRank [3], to being able to understand that a user query such as "Alan Rickman movies" is about a real-word entity (the actor Alan Rickman) and an attribute of this entity (the set of movies he has played in). As a result, if we pose this query to Google, we will get a list of images/links to the films of Alan Rickman, an infobox giving structured information about Alan Rickman, and an ordered list of links to more information about Alan Rickman and his filmography. This ability to understand the semantics of a user query has been aided by the availability of large knowledge bases such as Google's Knowledge Graph (KG) and their use in search algorithms.

An important class of Web resources that are not currently queried with the same success as actors, movies, etc. is public datasets, although there are probably millions of such datasets on the Web. Recognizing this, Google researchers have recently issued a "call to arms" and a set of guidelines that are aimed at enabling the structured markup and hence the effective discovery of public datasets[3]. Once a dataset is discovered, Google researchers also suggest that it might be useful to also query it, in the same way that one can now query a document containing arithmetic calculations using Explore for Google Sheets. In a similar spirit, it is very important to make EO datasets available on the Web as linked geospatial data to increase their use by developers that might not be experts in EO.

The objective of this diploma thesis is to enable the publication of EO datasets on the Web and their effective discovery by modern search engines like Google. Currently, EO datasets are hidden in the archives of ESA, NASA, etc. and they are only available through specialized search interfaces. We would like to make search engines able to discover EO datasets in the same way that they can discover information about actors, movies, etc. today. To achieve this goal, we extend the schema.org vocabulary with classes and

[1] https://www.earthobservations.org/g_faq.html
[2] https://en.wikipedia.org/wiki/Natural_disaster
[3] https://research.googleblog.com/2017/01/facilitating-discovery-of-public.html

properties, in order to be able to annotate EO datasets. This extension is based on the standard OGC 17-003 [4]. Moreover, we model and validate the annotated EO datasets using the programming language SHACL.

The report has the following organization: in chapter 2 we introduce the essential concepts of the semantic web, the vocabulary schema.org, the concept of an Earth observation, and the beta version of Google Dataset Search. Chapter 3 presents the vocabulary OGC 17-003 for describing EO dataset metadata, and our updated improved approach of this vocabulary. In chapter 4 we explain how we can annotate EO datasets using the current version of schema.org and our extended version of it. Chapter 5 introduces the programming language SHACL, and explains how we use it in order to model and validate RDF EO data graphs. Last but not least, in chapter 6 we summarize our contributions and discuss future work.

2. BASIC CONCEPTS AND RELATED WORK

This chapter introduces the essential concepts of the semantic web, the schema.org vocabulary and which are the main concepts of an Earth observation. It also explains how Google can find datasets on the web, and it discusses the related work which has already been carried out.

2.1 The Semantic Web

Every day, we use large amounts of data that are not part of the web. For example, we can see our videos, our receipts from some on-line purchases on the web, our appointments in a calendar. But can we see our videos in the calendar, to remember what we were doing when they were filmed? Can we see the receipts of the on-line purchases in the calender as well, in order to keep a track of the items we bought during a month? The answer is no, and it is the outcome of the nonexistence of a web of data, as the data is controlled only by the application in which they were created, and they are not shared among other applications.

The *Semantic Web* provides a common framework which allows data to be shared and reused across application, enterprise, and community boundaries. It is a joint effort of *World Wide Web Consortium (W3C)* and a great number of researchers and partners from the industry. The relations among data on the Web can be defined between any two resources. Moreover, each relationship is named, so that the user can understand how the two resources are connected.

In order to describe and model the information that is included in web resources, *Resource Description Framework (RDF)* data model is used. The RDF model is based on the concept of creating relations between web resources based on *triples*, which is the form *subject - predicate - object*, where the *subject* indicates the resource, and the *predicate* indicates a connection between the *subject* and the *object*.

Example 2.1.1 A way to represent the notion *"The duck has the colour yellow"* in RDF is as the triple:
Subject: *"The duck"* Predicate: *"has the colour"* Object: *"yellow"*

A collection of RDF statements represents a *directed, labeled multi-graph*. The RDF data are usually stored in *Triplestores*, which are databases for the storage and retrieval of triples through semantic queries, or relational databases.

In order to perform the semantic queries, the *SPARQL Protocol and RDF Query Language - SPARQL*[1] is used. SPARQL allows queries that consist of triple patterns, conjunctions, disjunctions, and optional patterns. It is recognized as one of the key technologies of the semantic web, as it was made a standard by the *RDF Data Access Working Group (DAWG)* of the *W3C*.

To achieve the main goal of the semantic web, which is to make the web of data a reality, it is essential to have large amounts of data available on the web in a standard format. This data has to be reachable and manageable by the semantic web tools, such as the RDF

[1] https://www.w3.org/TR/sparql11-query/

and SPARQL which are described in the subsection 2.1. Moreover, relationships among data should be available as well. The collection of the datasets that include the data and the relationships between this data, which is on the web can also be referred to as *Linked Data*.

One of the most famous linked datasets is DBPedia[2]. DBPedia is a project that aims to provide the content of Wikipedia[3] as RDF. In addition, it incorporates links to other datasets included on the web, such as GeoNames[4]. The users of DBPedia are able to semantically query properties and relationships located in the resources of Wikipedia, which is quite important if we consider the large amounts of information it includes.

In 2010, Tim Berners-Lee, founder of the World Wide Web, proposed a rating system for open data. This *5-star*[5] *deployment scheme* allows the users to score the maximum five stars if their data follow these five steps:

1 - star : If the data is available on the web, in any format, but with an open license which will indicate this data is open data.

2 - stars : If the data is available as machine-readable structured data. For example, if a scanned image of a table is available in an excel spreadsheet.

3 - stars : If the data is available in a non-proprietary format. For instant, the excel spreadsheet of the 2-stars step should be available as a CSV file.

4 - stars : If the data is published using open standards from the W3C, such as RDF and SPARQL.

5 - stars : If the published data is linked to other people's data, in order to provide context.

The 5-star linked data system is cumulative. If the data meets the criteria of the previous step(s) an extra star is added.

2.2 Earth Observation

Earth observation (EO) is the procedure of gathering data about Earth's physical, chemical and biological systems. This procedure is completed by using satellite remote sensing technologies supplemented by Earth surveying techniques. These techniques allow the collection, analysis and representation of the gathered data. Earth observation is used in order to monitor and access the status and the changes happening in natural and built environments. Its results are very important for the improvement of the social and economic level of the modern human civilization, as this kind of observations can point out actions which cause negative effects, such as the global warming, so that they can be minimized.

Figure 1 includes the basic concepts of an Earth observation. The satellite is the major source of information during an Earth observation, and it carries an amount of instruments,

[2] https://wiki.dbpedia.org/
[3] https://www.wikipedia.org/
[4] http://www.geonames.org/
[5] https://5stardata.info/en/

called a platform. Each instrument may have one or more detectors (sensors), which are typically cameras, sounders, or radiometers. Any observations created by the instruments of the satellite may be referred to us Earth observation series, which contain an amount of EO datasets (EO products). In the context of this diploma thesis, we are creating an extension of the schema.org vocabulary, in order to annotate EO product metadata.

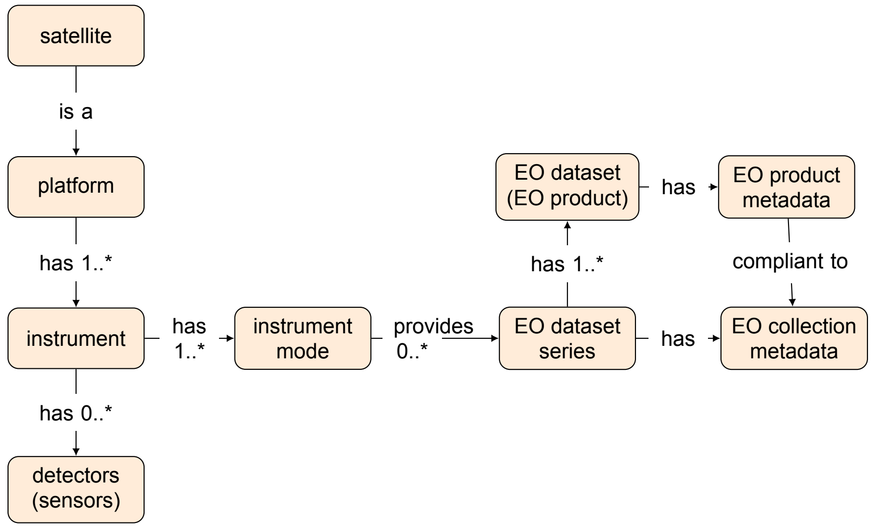


Figure 1: Basic concepts of an Earth observation

Currently, the world's biggest Earth observation program is *Copernicus*[6]. Copernicus is a European program for monitoring the Earth. It consists of a set of complex systems that collect data from satellites and in-situ sensors (e.g. air quality monitoring networks, ocean buoys and ground based weather stations), which provide integrated information, while they validate the data provided from the satellites. After the collection of the data is complete, the systems process it and allow users to have up-to-date and reliable information on a range of environmental and security issues. The EO satellites that provide the data of the Copernicus program are the six different families of *Sentinels*[7], and the contributing missions, which are operated by national, European or international organizations. [2]

Copernicus App Lab[8] is a two year project (November 2016 to October 2018) funded by the European Commission under the H2020 program. The main objective of the project is to make Earth observation data produced by the Copernicus program available on the Web as *linked data*. In this way, users who might not be Earth observation experts can take advantage of them easily.

One of the main goals of the Copernicus App Lab project is the ability to enable search engines like Google to treat datasets produced by Copernicus as *entities* in their own right and store knowledge about them in their internal knowledge graph. If this ability is available, search engines will be able to answer questions provided by the users, that involve these datasets, such as the question: "Is there a land use dataset of Greece produced by the Sentinel-1 of Copernicus?". Answering this kind of questions is beyond the capabilities of modern search engines.

The vocabulary we based the extension we implemented for the scope of this thesis is

[6]http://copernicus.eu/
[7]http://copernicus.eu/main/sentinels
[8]https://www.app-lab.eu/

OGC 17-003 - Earth Observation Dataset Metadata Vocabulary [4]. OGC 17-003 is a specification that defines a *GeoJSON(-LD)* encoding of Earth observation metadata for datasets. The vocabulary is described in detail in section 3.

2.3 Schema.org

Most webmasters use HTML tags on their pages. The HTML tags inform the web browser how the information the tags surround will be displayed. For instance, *<h2>Flight</h2>* informs the web browser that the text string *"Flight"* is displayed in a heading 2 format. However, the HTML tags do not inform the user about the meaning of the string - *"Flight"*, whether it refers to the successful movie of 2012, or it could refer to a specific scheduled flight. This drawback makes it very difficult for the search engines to propose relevant and quality content to the users.

Schema.org[9] vocabulary was created by the major search engines Bing, Google, Yahoo, and Yandex. It can be used on web pages which are written in any language. The goal was to provide a unique structured data markup schema which would include a great amount of topics, including people, places, products, events, etc [7]. The on-page markup allows search engines to understand information included in web pages, while it provides rich search features for users.

Schema.org provides multiple syntaxes for the webmasters to choose in order to annotate their web pages. Some of the most popular ones are *JavaScript Object Notation for Linked Data* (*JSON-LD*)[10], and Resource Description Framework in Attributes (*RDFa*)[11]. A newer syntax promoted by schema.org is *Microdata*[12] as part of *HTML5*[13], which was created to decrease the complexity of *RDFa*.

Schema.org provides a core, basic vocabulary which includes the description of the entities the majority of webmasters use. However, there is often the need to annotate websites that include more specialized and deeper vocabularies, that are based on the core vocabulary. To achieve this task, schema.org provides extension mechanisms[14] to allow the creation of additional vocabularies. Based on [4], our goal is to create a schema.org extension for the Earth observation dataset metadata Vocabulary, as described in OGC 17-003.

2.4 Google Dataset Search

Google has recently activated the beta version of its dataset search, where the datasets that are indexed using *schema.org*, as proposed by Google, show up. Dataset Search enables users to find datasets stored across the web by doing a simple keyword search. The tool surfaces information about datasets hosted in thousands of repositories across the web, making these datasets universally accessible and useful[15]. The goal of this project is to create a sharing ecosystem. In this way, publishers are encouraged to follow

[9] https://schema.org/
[10] https://json-ld.org/
[11] https://rdfa.info/
[12] https://www.w3.org/TR/microdata/
[13] https://www.w3.org/TR/html52/
[14] https://schema.org/docs/extension.html
[15] https://toolbox.google.com/datasetsearch

best practices for storage and publication of their data. Moreover, scientists have the opportunity to publish their work and show the impact of it in the scientific field they belong, by having their produced datasets cited.

Google can understand structured data in web pages about datasets, using *schema.org* Dataset markup[16] in order to have datasets show up in Google search results. The supported mark up formats are: *JSON-LD, Microdata* and *RDFa*. For now, we have followed these guidelines and annotated all the datasets of Copernicus App Lab by using the markup format *JSON-LD*. All these datasets can be searched and found in the Google Dataset Search.

The *schema.org* extension for Earth observation dataset metadata we created in the context of this thesis followed the described guidelines provided by Google. It improves the current situation by extending the schema.org class *Dataset* with subclasses and properties which cover the EO dataset metadata defined in OGC 17-003. In this way, users can search for EO datasets based on the instrument the observation was made, or by using all the other available metadata, which could not be used as keywords so far. In the following sections, we provide examples of the datasets we annotated based on *schema.org*, which are available at the link: `http://kr.di.uoa.gr/#datasets`.

2.5 EO Dataset Metadata Standards

2.5.1 OGC 10-157r4

The OGC 10-157r4 - Earth Observation Profile of Observations and Measurements (O&M) defines a profile of Observation and Measurements for describing EO products [5]. The goal is to provide a standard schema for encoding EO product metadata in order to describe and catalogue products from the sensors of EO satellites.

According to the OGC 10-157r4 specification, EO data products are managed within logical collections, which usually contain data items produced by sensors that belong to a satellite, or a series of satellites. Each EO product can be recognized in an EO collection based on a number of characteristics. These characteristics include the date of the acquisition, the location, and the more specific characteristics of the sensors. Some important characteristics of the image produced by the sensors may include the presence of clouds or other ground or atmospheric phenomena. Moreover, The quantity to be measured can be a complex quantity, such as a coverage, or it may be a simple quantity, such as a single temperature.

The characteristics described above are the common metadata used to differentiate the EO products, and they are defined in OGC 10-157r4 specification. In addition, EO product metadata annotated based on this specification are encoded as XML documents.

2.5.2 UMM-G

NASA's Common Metadata Repository (CMR) is a high-performance, high-quality repository for earth science metadata records that is designed to handle metadata at the Concept

[16] http://schema.org/Dataset

level[17]. The UMM is an extensible metadata model which provides a cross-walk for mapping between CMR-supported metadata standards.

The document that was used in OGC 17-003, which will be explained in section 3 describes the Unified Metadata Model for Granules (UMM-G) [1]. It includes the Granule metadata model itself, element descriptions with examples, and ISO 19115-1 and 19115-2 mappings. Values of granule metadata apply to all of the data in that one granule. Typical metadata in this category describe spatial and temporal extent of the data as well as the quality and lineage of the data. The ISO 191115-2 mapping paths and snippets used in this document are derived from ECHO to ISO 19115-2 translation, which is based on the NASA Best Practices for ISO. This translation resulted from efforts by the group assembled for the Metadata Evolution for NASA Data Systems (MENDS).

2.6 Summary

In this chapter we discuss about the basic concepts of the semantic web, the vocabulary schema.org, and we explain what an Earth observation is. Moreover, we talk about two very important EO dataset metadata standards, and explain how Google Dataset Search works.

[17] https://earthdata.nasa.gov/about/science-system-description/eosdis-components/

3. VOCABULARIES FOR ANNOTATING THE METADATA OF EARTH OBSERVATION DATASETS

This chapter introduces OGC 17-003, a vocabulary for describing Earth observation dataset metadata. In section 3.2 we present the updated version of this vocabulary.

3.1 OGC 17-003 approach

The OGC 17-003 specification[4] defines a GeoJSON(-LD) encoding of Earth observation metadata for datasets. The implementation included in this vocabulary is derived from the conceptual models defined in the Earth Observation Metadata Profile of Observations and Measurements (O&M) OGC 10-157r4[5], and the Unified Metadata Model for Granules (UMM-G)[1], as described in section 2. OGC 17-003 specification reuses pre-existing standardized property names from the OGC 10-157r4 and UMM-G documents. Moreover, it is simpler than these two previous standards, based on review comments of Committee on Earth Observation Satellites (CEOS).

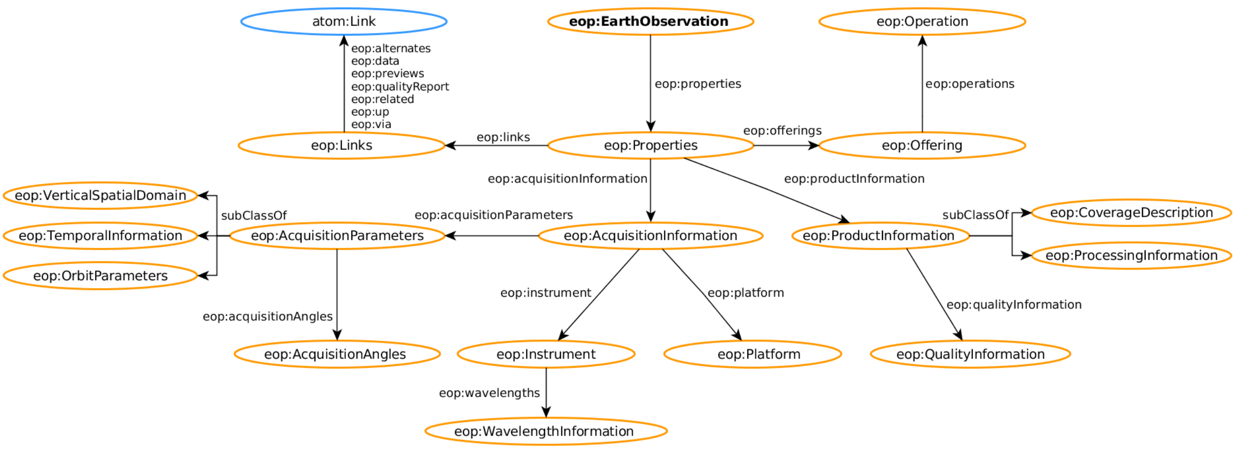


Figure 2: Classes and properties used in OGC 17-003

In Figure 2, the class *EarthObservation* is the main class of the OGC 17-003 specification, that defines an Earth observation entity according to OGC 10-157r4. It is connected to the *Properties* class with the object property *properties*. The *Properties* class is connected to the following classes:

1. The class *AcquisitionInformation*, with the object property *aquisitionInformation*. This class provides information about the Earth observation it refers to, and it is connected to the following classes:

    (a) The class *AcquisitionParameters*, with the object property *acquisitionParameters*. This class contains properties that are related to the acquisition of the data. It is a subclass of the classes *VerticalSpatialDomain*, *TemporalInformation* and *OrbitParameters*, that provide information related to the spatial extent in the vertical dimension, the start and end time of the acquisition of the data, and the orbit, respectively. In order to provide information about the acquisition angles of the entity, it connects to the class *AcquisitionAngles* with the object property *acquisitionAngles*.

(b) The class *Instrument*, with the object property *instrument*. This class contains the properties that are related to the instrument that was used to perform the observation. It is connected to the class *WavelengthInformation* with the object property *wavelengths*, which includes information related to the wavelengths properties of the instrument.

(c) The class *Platform*, with the object property *platform*. This class contains properties about the platform (satellite) which was used to perform the earth observation.

2. The class *Links*, with the object property *links*. This class contains references to related resources as hypermedia links such as to quicklooks, data download links or alternative representations of the metadata, and it inherits properties defined by OGC 14-055r2[6]. Moreover, the *Links* class is connected to the Atom class, defined in the document *RFC 4287 - The Atom Syndication Format*[1], which is an XML-based document format that describes lists of related information known as ”feeds”. Feeds include a set of items, known as ”entries”, and each one of them has an extensible set of attached metadata.

3. The class *Offering*, with the object property *offerings*. It provides information about the service or the inline content offering for an earth observation product, which will be consumed by the OGC-compliant clients. In addition, it is connected to the class *Operation*, with the object property *operations*. This class defines the operation that is used to either get the information or to get the capabilities of the offering. Both of these classes are defined by OGC 14-055r2.

4. The class *ProductInformation*, with the object property *productInformation*. This class provides information about the earth observation product, based on the OGC 17-003 specification. It is connected to the class *QualityInformation* with the object property *qualityInformation*, which includes information related to the quality of the product. Moreover, class *ProductInformation* is a subclass of the classes *CoverageInformation* and *ProcessingInformation*, that provide information related to the coverage and the processing of the data, respectively.

3.2 Our improved approach

The proposed extension to describe Earth observation metadata within schema.org has been published at `https://eop-sch.appspot.com/EarthObservation` .

In figure 3, we can see the classes that are included in the proposed Earth observation extension schema.

[1] https://tools.ietf.org/html/rfc4287

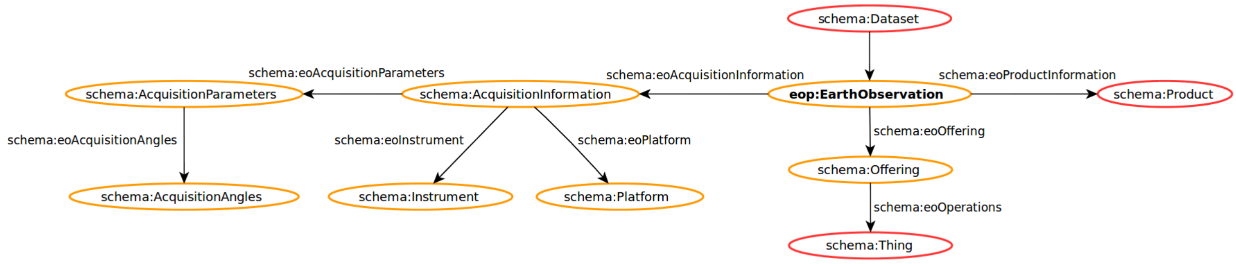


Figure 3: The classes included in the proposed Earth observation extension

As described in Section 3.1, the main class is *EarthObservation*. According to the vocabulary OGC 17-003, this class is connected to the class *Properties* with the object property *properties*, which is connected to four other classes: *AcquisitionInformation*, *Link*, *Offering*, and *ProductInformation*.

We decided not to include the *Properties* class and to include the properties of the four above classes in the new schema.org class *EarthObservation*, in order to keep the schema extension as simple as possible, without excluding any knowledge that is provided in the OGC 17-003 vocabulary. In more detail, our proposed extension is based on the following changes of the previous approach from section 3.1:

1. OGC 17-003 class *AcquisitionInformation* provides information which can be defined as the new property *eoAcquisitionProperty*, which has as its expected type the newly defined schema.org class *AcquisitionInformation*. We decided to create a new class because there was not another similar one in the existing version of the core schema[2] of schema.org, and the extended ones[3]. It includes the following properties:

   (a) *eoAcquisitionParameters* property has as its expected type the newly defined schema.org class *AcquisitionParameters*. We decided to create a new class as there was not another similar class in the existing schemas of schema.org. It includes all the data properties and object properties of the OGC 17-003 class *AcquisitionParameters*[4]:

   - *Data properties*: acquisitionStation, acquisitionSubType, cycleNumber, completionTimeFromAscendingNode, frame, groundTrackUncertainty, relativePassNumber, startTimeFromAscendingNode, tileId, track
   - *Object properties*: acquisitionAngles, acquisitionType, antennaLookDirection

   For the object property *acquisitionAngles*, a new property was created, the *eoAcquisitionAngles* property. It has as its expected type the newly defined schema.org class *AcquisitionAngles*. It includes all the properties of the OGC 17-003 class *AcquisitionAngles*[5]:

   - *Data properties*: acrossTrackIncidenceAngle, alongTrackIncidenceAngle, illuminationAzimuthAngle, illuminationElevationAngle, illuminationZenithAngle, incidenceAngle, incidenceAngleVariation, instrumentAzimuthAngle, instrumentElevationAngle, instrumentZenithAngle, maximumIncidenceAngle, minimumIncidenceAngle, pitch, roll, yaw

[2] https://schema.org/Thing
[3] https://schema.org/docs/extension.html
[4] http://geo.spacebel.be/opensearch/myDocumentation/doc/index-en.html#AcquisitionParameters
[5] http://geo.spacebel.be/opensearch/myDocumentation/doc/index-en.html#AcquisitionAngles

For every other property *AcquisitionParameters* and *AcquisitionAngles* classes have, a newly defined property is introduced, according to the provided specifications of the OGC 17-003 vocabulary, as shown in figures 4 and 5.

AcquisitionParameters

Canonical URL: http://schema.org/AcquisitionParameters

Thing > Product > AcquisitionParameters

Contains the properties related to the acquisition of the data.

Property	Expected Type	Description
Properties from AcquisitionParameters		
acquisitionStation	Text	Acquisition / receiving station code. Possible values are mission specific and should be retrieved using codespace.
acquisitionSubType	Text	The broad value defined by the acquisitionType is too restrictive, so mission specific type definition should refer to mission/ground segment dedicated codeSpace.
acquisitionType	Text	AcquisitionType can be one of: NOMINAL, CALIBRATION, OTHER.
antennaLookDirection	Text	AntennaLookDirection can be one of: left, right.
ascendingNodeDate	DateTime	Acquisition and date time. dateTime in ISO 8601 format (CCYY-MM-DDThh:mm[:ss[.cc]]Z) .
ascendingNodeLongitude	Float	Longitude at ascending node of orbit. Should be expressed in degrees.
beginningDateTime	DateTime	Acquisition start date time. dateTime in ISO 8601 format (CCYY-MM-DDThh:mm[:ss[.cc]]Z) .
completionTimeFromAscendingNode	Integer	Stop time of acquisition in milliseconds from ascending node date.
cycleNumber	Integer	Number of Cycles.
endingDateTime	DateTime	Acquisition end date time. dateTime in ISO 8601 format (CCYY-MM-DDThh:mm[:ss[.cc]]Z) .
eoAcquisitionAngles	AcquisitionAngles	Acquisition angles.
eoTrack	Text	Neutral wrsLongitudeGrid equivalent to track in track/frame, K in K/J, etc.
frame	Text	Neutral wrsLatitudeGrid equivalent to frame in track/frame, J in K/J, etc.
groundTrackUncertainty	Float	Measure of the uncertainty of the ground track. Sometimes known as deadband e.g. 1Km deadband.
highestLocation	Float	Lower bound of measurements in vertical dimension.
lastOrbitNumber	Integer	Acquisition last orbit number.
lowestLocation	Float	Upper bound of measurements in vertical dimension.
orbitDuration	Integer	Actual orbit duration in milliseconds.
orbitNumber	Integer	Acquisition orbit number.
relativePassNumber	Integer	Pass number since start of cycle.
startTimeFromAscendingNode	Integer	Start time of acquisition in milliseconds from ascending node date.
tileId	Text	While track/frame can be used to represent the first part of an MGRS coordinate (i.g. grid zone), the tileId identifies e.g. the second part of an MGRS coordinate (square identification), e.g. in case of Sentinel. Used when the world reference system coordinates can not be expressed in X/Y (Track/Frame) terms, such has for UTM tiles. (used for Sentinel-2 L1C granules).

Figure 4: The class AcquisitionParameters as defined in our new schema.org extension

AcquisitionAngles

Canonical URL: http://schema.org/AcquisitionAngles

Thing > Product > AcquisitionAngles

Contains the properties related to the acquisition angles.

Property	Expected Type	Description
Properties from AcquisitionAngles		
acrossTrackIncidenceAngle	Float	Acquisition across track incidence angle given in degrees. (i.e. uom='deg').
alongTrackIncidenceAngle	Float	Acquisition along track incidence angle given in degrees. (i.e. uom='deg').
illuminationAzimuthAngle	Float	Mean illumination/solar azimuth angle given in degrees. (i.e. uom='deg').
illuminationElevationAngle	Float	Mean illumination/solar elevation angle given in degrees. (i.e. uom='deg').
illuminationZenithAngle	Float	Mean illumination/solar zenith angle given in degrees. (i.e. uom='deg').
incidenceAngle	Float	Acquisition global incidence angle given in degrees (i.e. uom='deg').
incidenceAngleVariation	Float	Incidence angle variation.
instrumentAzimuthAngle	Float	Mean instrument azimuth angle given in degrees. (i.e. uom='deg').
instrumentElevationAngle	Float	Mean instrument elevation angle given in degrees. (i.e. uom='deg').
instrumentZenithAngle	Float	Mean instrument zenith angle given in degrees. (i.e. uom='deg').
maximumIncidenceAngle	Float	Maximum incidence angle.
minimumIncidenceAngle	Float	Minimum incidence angle.
pitch	Float	Satellite pitch angle given in degrees (i.e. uom='deg').
roll	Float	Satellite roll angle given in degrees (i.e. uom='deg').
yaw	Float	Satellite yaw angle given in degrees (i.e. uom='deg').

Figure 5: The class AcquisitionAngles as defined in our new schema.org extension

(b) *eoInstrument* property has as its expected type the newly defined schema.org class *Instrument*. We decided to create a new class as there was not a similar one in the existing version of the core schema, and the extended ones. It includes the properties of the OGC 17-003 class *Instrument*, [6]:

- *Data properties*: description, dopplerFrequency, instrumentShortName, operationalMode, resolution, samplingRates, swathIdentifier, verticalResolution
- *Object properties*: measurementType, polarisationChannels, polarisationMode, sensorType, wavelengths

For every property *Instrument* class has, a newly defined property is introduced, according to the provided specifications of the OGC 17-003 vocabulary, as shown in figure 6.

[6]http://geo.spacebel.be/opensearch/myDocumentation/doc/index-en.html#Instrument

Instrument

Canonical URL: http://schema.org/Instrument

Thing > Instrument

Contains the properties of the instrument that was used to perform the observation. Use as "id" the URI defined by GCMD to identify the instrument. A list of URI can be downloaded from https://gcmdservices.gsfc.nasa.gov/static/kms/platforms/instruments.rdf .

Property	Expected Type	Description
Properties from Instrument		
discreteWavelengths	Text	List of discrete wavelengths observed in the product.
dopplerFrequency	Text	Doppler Frequency of acquisition.
endWavelength	Text	End of the observed wavelength range.
instrumentShortName	Text	Instrument (Sensor) name.
measurementType	Text	MeasurementType can be one of absorption or emission.
operationalMode	Text	Sensor mode. Possible values are mission specific.
polarisationChannels	Text	PolarisationChannels can be one of hh or hhhv or hhhvhvv or hhvh or hhvv or hv or hvvh or undefined or vhc or vhhv or vhvv or vv or vvhv or vvvh.
polarisationMode	Text	PolarisationMode can be one of d or q or s or t or undefined.
resolution	Float	Sensor resolution. Unit of measure (m) is SI base unit (m) without prefix.
samplingRates	Float	Rate at which samples are provided in product. Some products may contain more than one sampling rate, e.g. 1kHz and 20kHz. Cardinality is therefore zero to many. Unit of measure (Hz) is SI base unit or derived unit without prefix.
sensorType	Text	SensorType can be one of altimetric or atmospheric or limb or optical or radar.
spectralRange	Text	SpectralRange can be one of infrared or nearinfrared or other or uv or visible.
startWavelength	Text	Start of the observed wavelength range.
swathIdentifier	Text	Swath identifier (e.g. Envisat ASAR has 7 distinct swaths (I1,I2,I3...I7) that correspond to precise incidence angles for the sensor).
verticalResolution	Text	lmb: Vertical spacing of data (if regular) atm: Full width at half maximum of the rows of the vertical averaging kernel matrix Unit of measure (m) is SI base unit (m) without prefix.
wavelengthResolution	Text	Spacing between consecutive wavelengths.

Figure 6: The class Instrument as defined in our new schema.org extension

(c) *eoPlatform* property has as its expected type the newly defined schema.org class *Platform*. We decided to create a new class as there was not a similar one in the existing versions of the schemas of schema.org. It includes the properties of the OGC 17-003 class *Platform*, [7]:

- *Data properties*: platformSerialIdentifier, platformShortName
- *Object properties*: orbitType

For every property *Platform* class has, a newly defined property is introduced, according to the provided specifications of the OGC 17-003 vocabulary, as shown in figure 7.

Platform

Canonical URL: http://schema.org/Platform

Thing > Platform

Contains the properties of the platform (satellite) that was used to perform the observation. Use as "id" the URI defined by GCMD to identify the platform. A list of URI can be downloaded from https://gcmdservices.gsfc.nasa.gov/static/kms/platforms/platforms.rdf .

Property	Expected Type	Description
Properties from Platform		
orbitType	Text	OrbitType can be one of geo or leo.
platformSerialIdentifier	Text	Platform serial identifier (e.g. for Seasat : 1).
platformShortName	Text	Platform short name (e.g. "Seasat" or "ENVISAT").

Figure 7: The class Platform as defined in our new schema.org extension

[7] http://geo.spacebel.be/opensearch/myDocumentation/doc/index-en.html#Platform

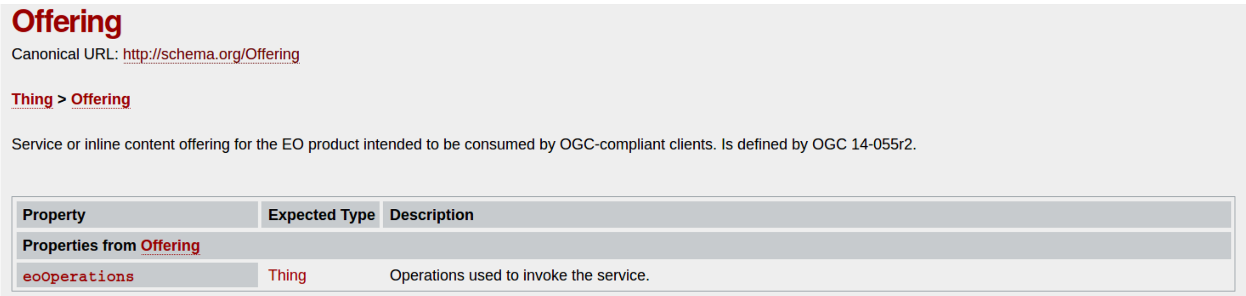
2. *Link* OGC 17-003 class information is not defined as a new class, as the properties it includes can be represented by the url properties that are already defined in *Thing* and *CreativeWork* schema.org classes, as shown in table 1.

Table 1: The mapping between the existing classes Thing and CreativeWork of schema.org and the properties of Link class

OGC 17-003 property	Schema.org property
alternates	Thing : additionalType
data	Thing : mainEntityOfPage
previews	CreativeWork : thumbnailUrl
qualityReport	CreativeWork : contentRating
related property	Thing : sameAs
up	CreativeWork : isBasedOn
via	Thing : url

3. *Offering* OGC 17-003 class information is defined as the schema.org property *eoOffering*, which has as its expected type the newly defined schema.org class *Offering*. We decided to create a new class, as there was not another similar one in the existing version of the core schema.org, and the extended ones. It includes the properties of the OGC 17-003 class *Offering* [8]:

   - *Data properties*: code
   - *Object properties*: operations


Offering

Canonical URL: http://schema.org/Offering

Thing > Offering

Service or inline content offering for the EO product intended to be consumed by OGC-compliant clients. Is defined by OGC 14-055r2.

Property	Expected Type	Description
Properties from Offering		
eoOperations	Thing	Operations used to invoke the service.

Figure 8: The class Offering as defined in our new schema.org extension

For the object property *operation*, a new property was created, the *eoOperations* property. It has as its expected type the existed schema.org class *Thing*, and it includes the datatypes[9] , according to the *OWS context document standard*[10]:

- *Data properties*: code, method, type, requestURL, payload, result, extension

These datatypes can be represented by the properties of the schema.org classes *CreativeWork* and *Thing*, as shown in table 2 .

[8] http://geo.spacebel.be/opensearch/myDocumentation/doc/index-en.html#Offering
[9] http://docs.opengeospatial.org/is/12-084r2/12-084r2.html#60
[10] http://docs.opengeospatial.org/is/12-084r2/12-084r2.html#1

Table 2: The mapping between the existing classes CreativeWork and Thing, and the properties of the class Operation

OGC 17-003 property	Schema.org property
code	Thing : identifier
method	CreativeWork : accessMode
type	CreativeWork : encodingFormat
requestURL	Thing : URL
payload	Thing
result	Thing
extension	CreativeWork

4. *ProductInformation* OGC 17-003 class information is defined as the schema.org property *eoProductInformation*, which has as its expected type the schema.org class *Product*. It includes the properties of the OGC 17-003 class *ProductInformation*[11]:

   - *Data properties*: archivingCenter, archivingDate, availabilityTime, productGroupId, productType, productVersion, size, statusDetail, statusSubType, timeliness
   - *Object properties*: qualityInformation

   All the properties that belong in the class *Product* can be defined as *additionalProperty*[12] properties, provided by the same schema.org class.

3.3 Summary

In this chapter we introduce the specification OGC 17-003 and how we updated it in order to use it in the new schema.org vocabulary extension we are creating for annotating EO product metadata.

[11] http://geo.spacebel.be/opensearch/myDocumentation/doc/index-en.html#ProductInformation
[12] https://schema.org/additionalProperty

4. DATASET ANNOTATION

In this chapter we explain how we can annotate EO datasets using the current version of schema.org. We focus on two EO datasets produced by Sentinel-1 and Sentinel-2, as described in 4.1. In subsection 4.2 we explain how the datasets can be annotated using the current version of schema.org, while in subsection 4.3 we provide the annotation of the data based on the new more detailed extension for representing EO datasets.

4.1 Earth observation datasets

The Copernicus Open Access Hub[1] (previously known as Sentinels Scientific Data Hub) provides complete, free and open access to Sentinel-1, Sentinel-2, Sentinel-3 and Sentinel-5P user products. In this section we explain the importance of data produced by two of the Copernicus sentinels, Sentinel-1 and Sentinel-2. In the following sections of this chapter this data is annotated using our extension of schema.org vocabulary for encoding EO data.

4.1.1 Sentinel-1

Sentinel-1 is the first of the five missions ESA is developing for the Copernicus initiative. Its mission is the European Radar Observatory for the Copernicus joint initiative of the European Commission (EC) and the European Space Agency (ESA). The Sentinel-1 mission includes C-band imaging operating in four exclusive imaging modes with different resolution (down to 5 m) and coverage (up to 400 km). It provides dual polarisation capability, very short revisit times and rapid product delivery. For each observation, precise measurements of spacecraft position and attitude are available[2]. The mission is composed of a constellation of two satellites, Sentinel-1A and Sentinel-1B, sharing the same orbital plane.

Synthetic Aperture Radar (SAR) has the advantage of operating at wavelengths not impeded by cloud cover or a lack of illumination and can acquire data over a site during day or night time under all weather conditions[3]. Sentinel-1, with its C-SAR instrument, can offer reliable, repeated wide area monitoring.

Sentinel-1 is designed to work in a pre-programmed, conflict-free operation mode, imaging all global landmasses, coastal zones and shipping routes at high resolution and covering the global ocean with vignettes. This ensures the reliability of the service required by operational services and a consistent long term data archive built for applications based on long time series.

A Sentinel-1 product is shown in figures 9 and 10. The image produced by Sentinel-1 shows a part of northern Europe, which is the area included in the red polygon of figure 9, while a quicklook of this area is shown in figure 10. The product can be downloaded in the url: `https://scihub.copernicus.eu/dhus/odata/v1/Products('37c2f72a-f1ef-4336-b13c-6056a650918c')/$value`.

[1] https://scihub.copernicus.eu/
[2] https://sentinels.copernicus.eu/web/sentinel/missions/sentinel-1
[3] https://sentinels.copernicus.eu/web/sentinel/missions/sentinel-1/overview

Figure 9: The map of the product of Sentinel-1

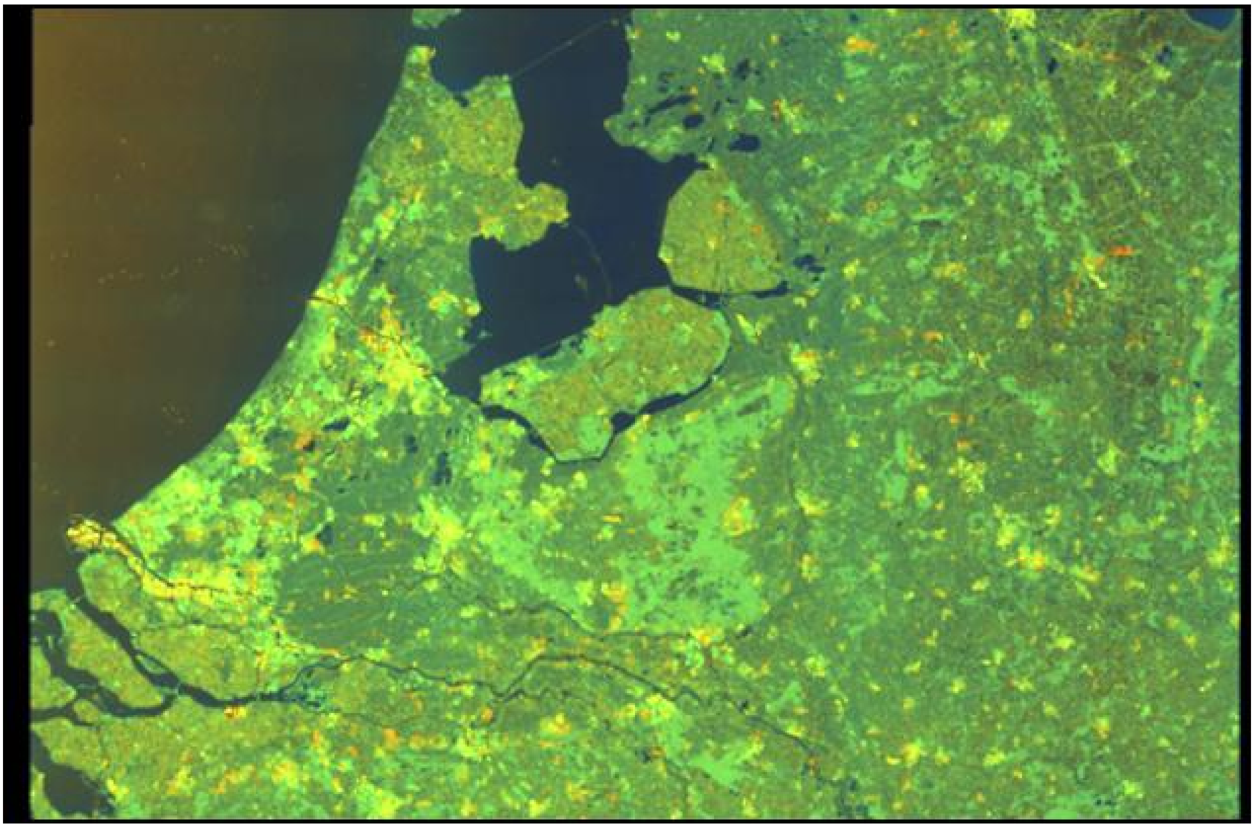


Figure 10: The quicklook of the produced image of Sentinel-1

The metadata provided for this product are shown in tables 3, 4, 5 and 6.

Table 3: General metadata for a Sentinel-1 product

Filename	S1A_IW_GRDH_1SDV_20181107T172504...BDC6.SAFE
Identifier	S1A_IW_GRDH_1SDV_20181107T172504...BDC6
Instrument	SAR-C
Model	IW
Satellite	Sentinel-1
Size	1.62 GB
Date	2018-11-07T17:25:04.147Z

Table 4: Product specific metadata for a Sentinel-1 product

Acquisition Type	NOMINAL
Cycle number	154
Footprint	<gml:Polygon> ... </gml:Polygon>
Format	SAFE
Ingestion Date	2018-11-07T21:29:09.657Z
JTS footprint	POLYGON ((3.458554 ... 52.891983))
Mission datatake id	175948
Orbit number (start)	24485
Orbit number (stop)	24485
Pass direction	ASCENDING
Phase identifier	1
Polarisation	VV VH
Product class	S
Product class description	SAR Standard L1 Product
Product composition	Slice
Product level	L1
Product type	GRD
Relative orbit (start)	88
Relative orbit (stop)	88
Resolution	High
Sensing start	2018-11-07T17:25:04.147Z
Sensing stop	2018-11-07T17:25:29.145Z
Slice number	15
Start relative orbit number	88
Status	ARCHIVED
Stop relative orbit number	88
Timeliness	Fast-24h

4.1.2 Sentinel-2

Sentinel-2 is the second of the five missions that ESA is developing for the Copernicus initiative. It is a European wide-swath, high-resolution, multi-spectral imaging mission. The full mission specification of the twin satellites of Sentinel-2 flying in the same orbit but phased at 180°, is designed to give a high revisit frequency of 5 days at the Equator[4].

[4]https://sentinels.copernicus.eu/web/sentinel/missions/sentinel-2/overview

Table 5: Platform specific metadata for a Sentinel-1 product

Carrier rocket	Soyuz
Launch date	April 3rd, 2014
Mission type	Earth observation
NSSDC identifier	2014-016A
Operator	European Space Agency
Satellite description	https://sentinel.esa.int/web/sentinel/missions/sentinel-1
Satellite name	Sentinel-1
Satellite number	A

Table 6: Instrument specific metadata for a Sentinel-1 product

Instrument abbreviation	SAR-C SAR
Instrument description	https://sentinel.esa.int/web/sentinel/missions/sentinel-1
Instrument mode	IW
Instrument name	Synthetic Aperture Radar (C-band)
Instrument swath	IW

According to ESA, Sentinel-2 carries an optical instrument payload that samples 13 spectral bands: four bands at 10 m, six bands at 20 m and three bands at 60 m spatial resolution. The orbital swath width is 290 km[5]. The twin satellites of Sentinel-2 provide continuity of SPOT and LANDSAT-type image data, contribute to ongoing multispectral observations and benefit Copernicus services and applications such as land management, agriculture and forestry, disaster control, humanitarian relief operations, risk mapping and security concerns.

The Sentinel-2 satellite system was developed by an industrial consortium led by Astrium GmbH (Germany). Astrium SAS located in France is responsible for the MultiSpectral Instrument (MSI). The MSI works passively, by collecting sunlight reflected from the Earth. New data is acquired at the instrument as the satellite moves along its orbital path. The optical design of the MSI telescope allows for a 290 km Field Of View (FOV). A shutter mechanism prevents the instrument from direct illumination by the sun in orbit and to avoid contamination during launch. The same mechanism functions as a calibration device by collecting the sunlight after reflection by a diffuser.

A Sentinel-2 product is shown in figures 11 and 12. The image produced by Sentinel-2 shows a part of northern Europe, which is the area included in the green polygon of figure 11, while a quicklook of this area is shown in figure 12. The product can be downloaded in the url: `https://scihub.copernicus.eu/dhus/odata/v1/Products('c444677e-3484-49a7-b3fc-7e6282a044f9')/$value`.

[5]https://sentinels.copernicus.eu/web/sentinel/missions/sentinel-2/satellite-description

Figure 11: The map of the product of Sentinel-2

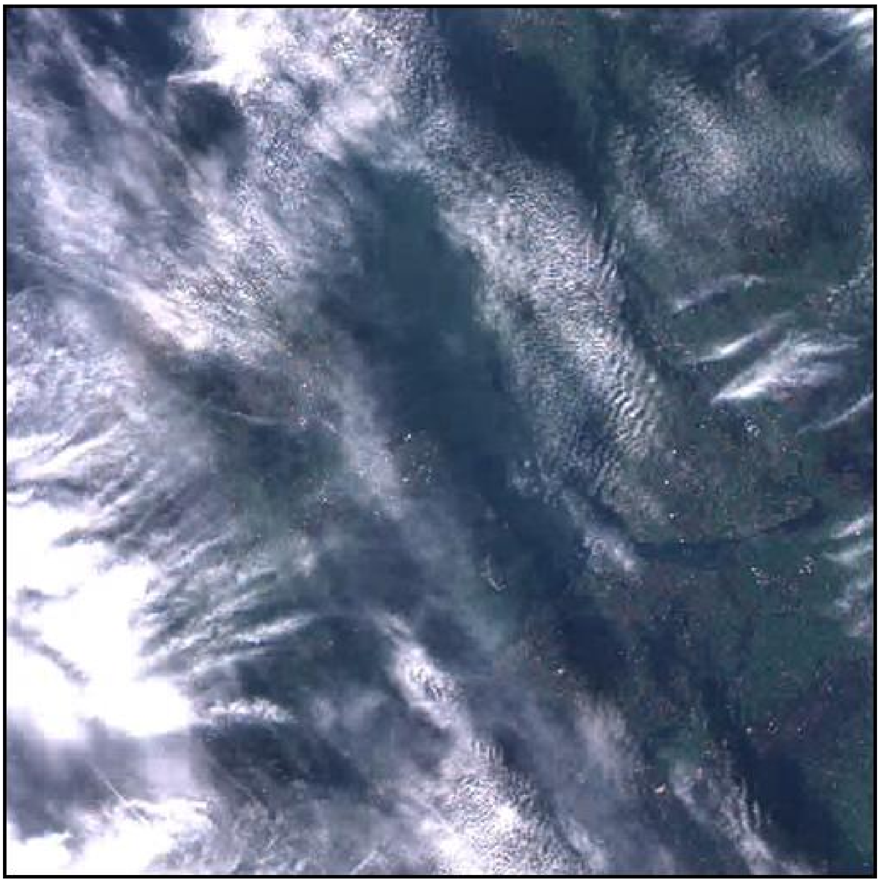


Figure 12: The quicklook of the produced image of Sentinel-2

The metadata provided for this product are shown in tables 7, 8, 9 and 10.

Table 7: General metadata for a Sentinel-2 product

Filename	S2A_MSIL1C_20181107T105231...1341.SAFE
Identifier	S2A_MSIL1C_20181107T105231...1341
Instrument	MSI
Satellite	Sentinel-2
Size	733.68 MB
Date	2018-11-07T10:52:31.025Z

Table 8: Product specific metadata for a Sentinel-2 product

Cloud cover percentage	51.506
Datatake sensing start	2018-11-07T10:52:31.025Z
Degraded ancillary data percentage	0.0
Footprint	<gml:Polygon> ... </gml:Polygon>
Format	SAFE
Format correctness	PASSED
General quality	PASSED
Generation time	018-11-07T11:13:41.000000Z
Geometric quality	PASSED
Ingestion Date	2018-11-07T16:36:06.154Z
JTS footprint	MULTIPOLYGON (((4.4984604283 ...)))
Mission datatake id	GS2A_20181107T105231_017637_N02.07
Orbit number (start)	17637
Pass direction	DESCENDING
Processing baseline	02.07
Processing level	Level-1C
Product type	S2MSI1C
Radiometric quality	PASSED
Relative orbit (start)	51
Sensing start	2018-11-07T10:52:31.025Z
Sensing stop	2018-11-07T10:52:31.025Z
Sensor quality	PASSED
Tile Identifier	31UFU
Tile Identifier horizontal order	UU31F

Table 9: Platform specific for a Sentinel-2 product

NSSDC identifier	2015-028A
Satellite name	Sentinel-2
Satellite number	A

4.2 Annotating using the current schema.org approach

In listings [1,2], we are showing how we can annotate EO datasets using the schema.org class Dataset[6]. These examples are extracted from the products of the Sentinels 1 and

[6]https://schema.org/Dataset

Table 10: Instrument specific for a Sentinel-2 product

Instrument abbreviation	MSI
Instrument mode	INS-NOBS
Instrument name	Multi-Spectral Instrument

2, as described in sections 4.1.1 and 4.1.2.

The current schema.org vocabulary includes the class *Dataset* for data providers to annotate their datasets. Class *Dataset* extends the schema.org class *CreativeWork*, and schema.org class *Thing*. In this way, we can use all the properties the classes *Dataset, CreativeWork,* and *Thing* include for the annotation of our datasets. For Sentinel-1 data the annotated information is shown in table 11, and in listing [1] we provide the *JSON-LD* code we used. In the same way, for Sentinel-2 data the annotated information is shown in table 12, and in listing [2]. The data about the product of the Earth observations which are included in the tables 4 and 8 can be encoded using the property `additionalProperty` provided by the schema.org class *Product*.

Table 11: Current schema.org dataset annotation for Sentinel-1 product

schema.org class	schema.org property	Sentinel value
Dataset	distribution : encodingFormat	SAFE
Dataset	distribution : type	DataDownload
Dataset	distribution : contentUrl	https://scihub ... value.
CreativeWork	copyrightYear	2018
CreativeWork	isAccessibleForFree	true
CreativeWork	temporalCoverage	2018
CreativeWork	author	ESA
CreativeWork	spatialCoverage:geo:polygon	POLYGON((3.4585 ... 52.891983))
CreativeWork	sourceOrganization	ESA
CreativeWork	keywords	”Sentinel-1”
CreativeWork	dateCreated	2018-11-07T17:25:04.147Z
Thing	name	S1A_IW_GRDH ... BDC6.SAFE
Thing	identifier	S1A_IW_GRDH_1SDV ... BDC6
Thing	url	https://scihub...50918c’)

Listing 1: Sentinel-1 data annotation based on the current version of schema.org

```
{"@context": "http://schema.org",
    "@type": "Dataset",
    "name": "S1A_IW_GRDH_1SDV_20181107T172504...SAFE",
    "identifier": "S1A_IW_GRDH_1SDV_20181107T172504...BDC6",
    "dateCreated": "2018-11-07T17:25:04.147Z",
    "author": "ESA",
    "sourceOrganization": "ESA",
    "copyrightYear": "2018",
    "keywords":["Sentinel-1"],
    "spatialCoverage":"POLYGON((52.891983,3.458554...52.891983,3.458554))",
    "temporalCoverage": "2018",
    "isAccessibleForFree": true,
    "distribution": {
        "@type":"DataDownload",
        "encodingFormat":"SAFE",
        "contentUrl":"https://scihub...6056a650918c')/$value"},
    "url": "https://scihub.copernicus...6056a650918c')"}
```

Table 12: Current schema.org dataset annotation for Sentinel-2 product

schema.org class	schema.org property	Sentinel value
Dataset	distribution : encodingFormat	SAFE
Dataset	distribution : type	DataDownload
Dataset	distribution : contentUrl	https://scihub ... value.
CreativeWork	copyrightYear	2018
CreativeWork	isAccessibleForFree	true
CreativeWork	temporalCoverage	2018
CreativeWork	author	ESA
CreativeWork	spatialCoverage:geo:polygon	MULTIPOLYGON((((4.4984604283...)))
CreativeWork	sourceOrganization	ESA
CreativeWork	keywords	”Sentinel-2”
CreativeWork	dateCreated	2018-11-07T10:52:31.025Z
Thing	name	S2A_MSIL1C...11341.SAFE
Thing	identifier	S2A_MSIL1C...11341
Thing	url	https://scihub.copernicus...a044f9’)

4.3 Annotating using our new schema.org extension

The extension of the schema.org vocabulary we created in this diploma thesis includes the class EarthObservation for data providers to annotate their EO datasets. The newly defined class EarthObservation extends the schema.org class Dataset, as described in section 3. In this way, we can use all the specialised classes and properties the class EarthObservation has, as defined in OGC 17-003, and all the properties the classes Dataset, CreativeWork, and Thing include for the annotation of EO datasets.

Listing 2: Sentinel-2 data annotation based on the current version of schema.org

```
{"@context": "http://schema.org",
    "@type": "Dataset",
    "name": "S2A_MSIL1C_20181107T105231...SAFE",
    "identifier": "S2A_MSIL1C_20181107T105231...11341",
    "dateCreated": "2018-11-07T10:52:31.025Z",
    "author": "ESA",
    "sourceOrganization": "ESA",
    "copyrightYear": "2018",
    "keywords":["Sentinel-2"],
    "spatialCoverage":"MULTIPOLYGON((((4.4984604283...)))",
    "temporalCoverage": "2018",
    "isAccessibleForFree": true,
    "distribution": {
        "@type":"DataDownload",
        "encodingFormat":"SAFE",
        "contentUrl":"https://scihub...a044f9')/$value"},
    "url": "https://scihub.copernicus...a044f9')"}
```

In addition to the schema.org classes and properties shown in tables 11 and 12, we can annotate the more specific information provided for the satellites and sensors used during the construction of the Sentinels datasets. For Sentinel-1 data the annotated information is shown in table 13, and in listing [3] we provide the *JSON-LD* code we used. In the same way, for Sentinel-2 data the annotated information is shown in table 14, and in listing [4]. The data about the products of the Earth observations which is included in the tables 4 and 8, and is not already annotated using the the extended vocabulary, can be encoded using the property `additionalProperty` provided by the schema.org class Product.

4.4 Summary

In this chapter we describe how we can annotate two products created by the Sentinels 1 and 2 using the current schema.org vocabulary and our new schema.org vocabulary extension for annotating EO products more precisely.

Table 13: Dataset annotation for Sentinel-1 data based on our schema.org extension

schema.org class	schema.org property	Sentinel value
EO	eoAcquisitionParameters:acquisitionType	Acquisition Type
EO	eoAcquisitionParameters:cycleNumber	Cycle number
EO	eoAcquisitionParameters:ascendingNodeDate	Ingestion Date
EO	eoAcquisitionParameters:acquisitionSubType	Mission datatake id
EO	eoAcquisitionParameters:orbitNumber	Orbit number (start)
EO	eoAcquisitionParameters:orbitNumber	Orbit number (stop)
EO	eoAcquisitionParameters:orbitDirection	Pass direction
EO	eoInstrument:polarisationChannels	Polarisation
EO	eoInstrument:resolution	Resolution
EO	eoAcquisitionParameters:beginningDateTime	Sensing start
EO	eoAcquisitionParameters:endingDateTime	Sensing stop
EO	eoPlatform:platformShortName	Satellite name
EO	eoPlatform:platformSerialIdentifier	Satellite number
EO	eoInstrument:instrumentShortName	Instrument abbreviation
EO	eoInstrument:operationalMode	Instrument mode
EO	eoInstrument:swathIdentifier	Instrument swath
Thing	eoInstrument : identifier	Instrument id
Thing	eoInstrument:name	Instrument name
Thing	eoInstrument : description	Instrument description
Thing	eoPlatform : identifier	Platform id
Thing	eoPlatform : description	Satellite description

Table 14: Dataset annotation for Sentinel-2 data based on our schema.org extension

schema.org class	schema.org property	Sentinel value
EO	eoAcquisitionParameters:beginningDateTime	Datatake sensing start
EO	eoAcquisitionParameters:ascendingNodeDate	Ingestion Date
EO	eoAcquisitionParameters:acquisitionSubType	Mission datatake id
EO	eoAcquisitionParameters:orbitNumber	Orbit number (start)
EO	eoAcquisitionParameters:orbitDirection	Pass direction
EO	eoAcquisitionParameters:beginningDateTime	Sensing start
EO	eoAcquisitionParameters:endingDateTime	Sensing stop
EO	eoAcquisitionParameters:tileId	Tile Identifier
EO	eoPlatform:platformShortName	Satellite name
EO	eoPlatform:platformSerialIdentifier	Satellite number
EO	eoInstrument:instrumentShortName	Instrument abbreviation
EO	eoInstrument:operationalMode	Instrument mode
Thing	eoInstrument : identifier	Instrument id
Thing	eoInstrument:name	Instrument name
Thing	eoPlatform : identifier	Platform id

Listing 3: Sentinel-1 data annotated based on our extended version of schema.org

```
{"@context": "http://schema.org",
"@type": "EarthObservation",

        "As shown in table 4.9"

"eoAquisitionInformation": {
"@type" : "AcquisitionInformation",
"eoInstrument": {
  "@type" :"Instrument",
  "id" : "http://gcmdservices.gsfc.nasa.gov/kms/concept/ed400e7c-229e-48be-
  9a93-84f2fc864448",
  "name" : "Synthetic Aperture Radar (C-band)",
  "instrumentShortName" : "SAR-C SAR",
  "description" : " https://sentinel.esa.int/web/sentinel/missions/
  sentinel-1",
  "polarisationChannels" : "VV VH",
  "operationalMode" : "IW",
  "swathIdentifier" : "IW"},

 "eoPlatform": {
  "@type":"Platform",
  "id": "http://gcmdservices.gsfc.nasa.gov/kms/concept/c7279e54-f7c1-4ee7-
  a957-719d6021a3f",
  "description": "https://sentinel.esa.int/web/sentinel/missions/
  sentinel-1",
  "platformSerialIdentifier":"A",
  "platformShortName":"Sentinel-1"},

 "eoAcquisitionParameters": {
  "@type" : "AcquisitionParameters",
  "acquisitionType" : "NOMINAL",
  "cycleNumber" : 154,
  "ascendingNodeDate" : "2018-11-07T21:29:09.657Z",
  "acquisitionSubType" : "175948",
  "orbitNumber" : "24485",
  "orbitDirection" : "ASCENDING",
  "beginningDateTime" : "2018-11-07T17:25:04.147Z",
  "endingDateTime" : "2018-11-07T17:25:29.145Z"}}
```

Listing 4: Sentinel-2 data annotated based on our extended version of schema.org

```
{"@context": "http://schema.org",
"@type": "EarthObservation",

        "As shown in table 4.10"

"eoAquisitionInformation": {
"@type" : "AcquisitionInformation",
"eoInstrument": {
  "@type" :"Instrument",
  "id" : "http://gcmdservices.gsfc.nasa.gov/kms/concept/081f9b6e-d0a0-4f1d
  -ad8a-638189418480",
  "name" : "Multi-Spectral Instrument",
  "instrumentShortName" : "MSI",
  "operationalMode" : "INS-NOBS"},

 "eoPlatform": {
  "@type":"Platform",
  "id": "http://gcmdservices.gsfc.nasa.gov/kms/concept/2ce20983-98b2-40b9
  -bb0e-a08074fb93b3",
  "platformSerialIdentifier":"A",
  "platformShortName":"Sentinel-2"},

 "eoAcquisitionParameters": {
  "@type" : "AcquisitionParameters",
  "acquisitionType" : "NOMINAL",
  "ascendingNodeDate" : "2018-11-07T16:36:06.154Z",
  "acquisitionSubType" : "GS2A_20181107T105231_017637_N02.07",
  "orbitNumber" : "17637",
  "orbitDirection" : "DESCENDING",
  "beginningDateTime" : "2018-11-07T10:52:31.025Z",
  "endingDateTime" : "2018-11-07T10:52:31.025Z",
  "tileId" : "31UFU"}}
```

5. SHACL

This chapter introduces SHACL, a language for validating RDF graphs against a set of conditions. Moreover, we explain how we use SHACL in order to model and validate RDF EO data graphs.

5.1 What is SHACL?

Shapes Constraint Language (SHACL)[1] is a language based on RDF, for modeling and validating graph - based and object - based data. It has been developed by a *W3C* working group and it is useful for people who work with data that is extracted from different sources, as SHACL allows them to describe ”shapes” and constraints on this data, so that it can be used from the applications more easily and more beneficially. These kinds of descriptions may also be used for more purposes besides validation, including data integration, code generation and interface building.

SHACL can be used to define classes together with constraints on their properties. Program--mers can use some built-in types of constraints, such as *minCount* and *maxCount*, which specify the cardinality of a property. Moreover, more complex constraints can be defined as well. These constraints are expressed as shapes in the form of RDF graphs and are called *shapes graphs*. The RDF data that need to be described or validated against the shape graphs are called *data graphs*.

The programmers use SHACL validation engines in order to validate their data. A *SHACL* validation engine takes as input a data graph and a shapes graph, and produces a validation report. In addition, the data graphs and the shapes graphs can be represented in any RDF serialization formats, such as *Turtle* and *JSON-LD*. In the context of this thesis, *JSON-LD* is used for representing the data graphs, and *Turtle* for representing the shapes graphs that are conducted.

5.2 Shapes and Constraints

The SHACL Core language defines two types of shapes:

1. shapes about the focus node itself, called node shapes - *sh:NodeShape*.
2. shapes about the values of a particular property or path for the focus node, called property shapes - *sh:PropertyShape*.

Definition 5.2.1: A *focus node* is an RDF term which is validated against a shape using the triples that are included in a data graph.

Definition 5.2.2: *Target declarations* of a shape in a shapes graph are triples with the shape as the subject and certain properties as predicates. Target declarations can be used to produce focus nodes for a shape. The *target of a shape* is the union of all RDF

[1] https://www.w3.org/TR/shacl/

terms produced by individual targets that are declared by the shape in the shapes graph. *Definition 5.2.3*: A *shape* is an IRI or blank node *s* that has at least one of the following conditions in the shapes graph:

- *s* is a SHACL instance of *sh:NodeShape* or *sh:PropertyShape*.
- *s* is a subject of a triple that has *sh:targetClass*, *sh:targetNode*, *sh:targetObjectsOf*, or *sh:targetSubjectsOf* as predicate.
- *s* is a subject of a triple that has a parameter as predicate.
- *s* is a value of shape-expecting, non-list-taking-parameter such as *sh:node*, or a member of a SHACL list that is a value of a shape-expecting and list-taking parameters such as *sh:or*.

In the rest of this section, we provide a simple example of a data graph [5, 6], a shapes graph [7], and a validation report [8], which is created when the data graph is validated against the shapes graph.

In listings [5, 6], we show two data graphs including the same information about a person in *Turtle* and *JSON-LD* respectively. The example contains a SHACL instance of the class `schema:Person`. The following conditions are shown in the example:

1. A SHACL instance of `ex:Person` can have at most one value of the properties `schema:familyName, schema:birthDate, schema:deathDate, schema:address`.
2. The properties `schema:birthDate, schema:deathDate` are literals with the datatype `xsd:date`.
3. The property `schema:familyName` is literal with the datatype `xsd:string`.
4. A SHACL instance of `ex:SeverusAddress` can have at most one value of the properties `schema:streetAddress, schema:postalCode`.
5. The property `schema:streetAddress` is literal with the datatype `xsd:string`.
6. The property `schema:postalCode` is literal with the datatype `xsd:integer`.
7. A SHACL instance of `ex:Person` cannot have values for any other property apart from `schema:familyName, schema:birthDate, schema:deathDate, schema:address`.
8. A SHACL instance of `ex:Address` cannot have values for any other property apart from `schema:streetAddress, schema:postalCode`.

The above conditions can be represented as shapes and constraints in the shapes graph shown in listing [7].

We can use the shape declaration above in listings [5, 6] to illustrate some of the key terminology used by SHACL, as shown in listing [7]. The target for the shape `schema:PersonShape` is the set of all SHACL instances of the class `schema:Person`, which is specified using the property `sh:targetClass`. During the validation, these target nodes become focus nodes for the shape. The shape `schema:PersonShape` is a node shape, which means that it applies to the focus nodes. It declares constraints on the focus nodes,

Listing 5: A simple example of a data graph in Turtle

```
@prefix ex: <http://example.org/ns#> .
@prefix schema: <http://schema.org/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

ex:Severus
a schema:Person ;
schema:familyName "Snape" ;
schema:birthDate "1960-01-09"^^xsd:date ;
schema:deathDate "1959-05-02"^^xsd:date ;
schema:address ex:SeverusAddress .

ex:SeverusAddress
schema:streetAddress "1 Alnwick Castle" ;
schema:postalCode 2412 .
```

Listing 6: A simple example of a data graph in JSON-LD

```
{"@context": { "@vocab": "http://schema.org/" },

    "@id": "http://example.org/ns#Severus",
    "@type": "Person",
    "familyName": "Snape",
    "birthDate": "1960-01-09",
    "deathDate": "1959-05-02",
    "address": {
            "@id": "http://example.org/ns#SeverusAddress",
                "streetAddress": "1 Alnwick Castle",
                "postalCode": 2412}}
```

for example using the parameter `sh:closed`. The node shape `schema:PersonShape` declares three other constraints, while the node shape `schema:AddressShape` declares two other constraints. All these constraints include the property `sh:property`, and each of these is backed by a property shape. These property shapes declare additional constraints using parameters such as `sh:datatype`, `sh:minInclusive` and `sh:maxCount`.

Some of the property shapes specify parameters from multiple constraint components in order to restrict multiple aspects of the property values. For example, in the property shape for `ex:postalCode`, parameters from three constraint components are used. The parameters of these constraint components are `sh:datatype`, `sh:maxInclusive` and `sh:minInclusive`. For each focus node the property values of `ex:postalCode` will be validated against all three components.

SHACL validation based on the provided data graph in listings [5, 6] and shapes graph in listing [7] would produce the validation report shown in listing [8].

The validation results are enclosed in the validation report shown in listing [8]. The first

Listing 7: A simple example of a shapes graph in Turtle

```
@prefix dash: <http://datashapes.org/dash#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix schema: <http://schema.org/> .
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

schema:PersonShape
    a sh:NodeShape ;
    sh:targetClass schema:Person ;
    sh:property [
        sh:path schema:givenName ;
        sh:datatype xsd:string ;
        sh:name "given name" ;
    ] ;
    sh:property [
        sh:path schema:birthDate ;
        sh:lessThan schema:deathDate ;
        sh:maxCount 1 ;
    ] ;
    sh:property [
        sh:path schema:address ;
        sh:node schema:AddressShape ;
    ] .

schema:AddressShape
    a sh:NodeShape ;
    sh:closed true ;
    sh:property [
        sh:path schema:streetAddress ;
        sh:datatype xsd:string ;
    ] ;
    sh:property [
        sh:path schema:postalCode ;
        sh:or ( [ sh:datatype xsd:string ] [ sh:datatype xsd:integer ] ) ;
        sh:minInclusive 10000 ;
        sh:maxInclusive 99999 ;
    ] .
```

validation result is produced because `ex:Severus` has a value for `schema:birthDate` less than the value for `schema:deathDate`. The second validation is produced because `ex:SeverusAddress` has a value for `schema:postalCode` that is not greater or equal to value `10000` and less or equal to value `99999`, as defined in properties `sh:minInclusive` and `sh:maxInclusive` respectively.

Listing 8: A simple example of a validation report

```
[
    a sh:ValidationResult ;
    sh:resultSeverity sh:Violation ;
    sh:sourceConstraintComponent sh:LessThanConstraintComponent ;
    sh:sourceShape _:n498 ;
    sh:focusNode <http://example.org/ns#Severus> ;
    sh:resultPath schema:birthDate ;
    sh:value "1960-01-09" ;
    sh:resultMessage "Value is not < value of schema:deathDate" ;
] .
[
    a sh:ValidationResult ;
    sh:resultSeverity sh:Violation ;
    sh:sourceConstraintComponent sh:NodeConstraintComponent ;
    sh:sourceShape _:n501 ;
    sh:focusNode <http://example.org/ns#Severus> ;
    sh:value <http://example.org/ns#SeverusAddress> ;
    sh:resultPath schema:address ;
    sh:resultMessage "Value does not have shape schema:AddressShape" ;
] .
```

5.3 Shapes graphs for validating Earth observation datasets

In this section we describe how the data graphs constructed by Earth observation data can be validated against the constraints described in OGC 17-003. We created the shapes graph of the schema.org vocabulary extension for EO data, which was analysed in section 3 and is available online. To test our shapes and data graphs we used the website *SHACL Playground*[2]. SHACL Playground provides a constraint validator for the Shapes Constraint Language, written in *JavaScript*, and it is a work in progress.

Firstly, in listing [9] we show how data written in *JSON-LD* can be transformed into data graphs, in order to be validated against our shapes graphs about EO data. The example in listing [9] includes information about an instance of the schema.org class *AcquisitionParameters*, as described in sec 3.2.

As it is shown in the example of listing [9], information about the context (`@context`) and the type (`@type`) of the data has changed. These changes were performed so that the schema.org vocabulary is loaded and the types of the schema.org classes are added as well. A more detailed example of a data graph is provided in listing [10], which is an instance of the schema.org class *AcquisitionInformation*, as described in sec 3.2.

[2]http://shacl.org/playground/

Listing 9: A data graph including an instance of AcquisitionParameters class in JSON-LD

```
{
  "@context": { "@vocab": "http://schema.org/" },
  "@type": "AcquisitionParameters",
  "beginningDateTime": "1978-09-27T01:04:30Z",
  "endingDateTime": "1978-09-27T01:04:45Z",
  "aquisitionType": "NOMINAL",
  "aquisitionSubType": "DEFAULT",
  "orbitNumber": 1316,
  "orbitDirection": "DESCENDING",
  "antennaLookDirection": "right",
  "acquisitionAngles": {
    "@type": "AcquisitionAngles",
    "minimumIncidenceAngle": 19.6,
    "maximumIncidenceAngle": 9.6,
    "incidenceAngleVariation": 9.6
  }
}
```

Listing 10: A data graph including an instance of AcquisitionInformation class in JSON-LD

```
{
 "@context" : {"@vocab": "http://schema.org/"},

 "eoPlatform": {
  "@type":"Platform",
  "id": "http://gcmdservices.gsfc.nasa.gov/kms/concept/c7279e54-f7c1-4ee7-
  a957-719d6021a3f",
  "description":"https://sentinel.esa.int/web/sentinel/missions/sentinel-1",
  "platformSerialIdentifier": 1,
  "platformShortName":"Sentinel-1"},

 "eoAcquisitionParameters": {
  "@type" : "AcquisitionParameters",
  "acquisitionType" : "SOMETHING",
  "cycleNumber" : 154,
  "ascendingNodeDate" : "2018-11-07T21:29:09.657Z",
  "acquisitionSubType" : "175948",
  "orbitNumber" : 24485,
  "orbitDirection" : "ASCENDING",
  "beginningDateTime" : "2018-11-07T17:25:04.147Z",
  "endingDateTime" : "2018-11-07T17:25:29.145Z"}

}
```

In the rest of the section, we are describing how we can understand the possible errors that could occur while annotating EO datasets, using our EO schema.org extension . If we validate the data graph of listing [10] against our shapes graph about EO data, we have the validation report that is included in listing [13]. The parts of the *Turtle* code our shapes graph has, which produce the validation report, are included in the listings [11] and [12].

Listing 11: The part of our shapes graph that produces the first validation

```
schema:AcquisitionParametersShape
   rdf:type rdfs:Class ;
   rdf:type sh:NodeShape ;
   rdfs:comment "Contains the properties ... of the data."^^xsd:string ;
   rdfs:label "AcquisitionParameters" ;
   rdfs:subClassOf schema:Product ;
   sh:targetClass schema:AcquisitionParameters;
sh:property [
   sh:path schema:acquisitionType ;
   sh:datatype xsd:string ;
   sh:description "AcquisitionType can ... CALIBRATION,OTHER."^^xsd:string;
   sh:in ("NOMINAL" "CALIBRATION" "OTHER");
   sh:name "acquisitionType" ;
   sh:message "AcquisitionType can be one of: NOMINAL,CALIBRATION,OTHER."];
```

Listing 12: The part of our shapes graph that produces the second validation

```
schema:PlatformShape
   rdf:type rdfs:Class ;
   rdf:type sh:NodeShape ;
   rdfs:comment "Contains ... perform the observation."^^xsd:string;
   rdfs:label "Platform" ;
   rdfs:subClassOf schema:Thing ;
   sh:targetClass schema:Platform;
sh:property [
   sh:path schema:platformSerialIdentifier ;
   sh:datatype xsd:string ;
   sh:description "Platform serial identifier e.g. for Seasat:1"^^xsd:string;
   sh:name "platformSerialIdentifier"];
```

As the validation report in listing [13] describes, we have two violations because of the following reasons:

1. The first violation is about the property *acquisitionType*, as described in *sh:resultPath*. The value of this property is not following the standard described in OGC 17-003, as the user provided a *xsd:string* value that is not one in the domain of this property. As it is shown in *sh:resultMessage*, AcquisitionType can be one of: NOMINAL, CALIBRATION, OTHER, while the user provided the value *SOMETHING*.

2. The second violation is about the property *polarisationChannels*, as described in *sh:resultPath*. The value of this property is not following the standard described in

Listing 13: The validation report produced by the data graph in listing 8 and our shapes graph for EO datasets

```
[
        a sh:ValidationResult ;
        sh:resultSeverity sh:Violation ;
        sh:sourceConstraintComponent sh:InConstraintComponent ;
        sh:sourceShape _:n133 ;
        sh:focusNode _:n467 ;
        sh:value "SOMETHING" ;
        sh:resultPath schema:acquisitionType ;
        sh:resultMessage "AcquisitionType can be one of: NOMINAL,
            CALIBRATION, OTHER." ;
] .
[
        a sh:ValidationResult ;
        sh:resultSeverity sh:Violation ;
        sh:sourceConstraintComponent sh:DatatypeConstraintComponent ;
        sh:sourceShape _:n193 ;
        sh:focusNode _:n472 ;
        sh:value 1 ;
        sh:resultPath schema:platformSerialIdentifier ;
        sh:resultMessage "Value does not have datatype xsd:string" ;
] .
```

OGC 17-003, as the user provided a *xsd:integer* value, while the EO vocabulary should have a ”xsd:integer” value, as shown in *sh:resultMessage*.

Based on the above validation results, the users can check if their EO datasets are annotated correctly, so that search engines can discover them.

5.4 Summary

In this chapter we introduce the language SHACL, which we use for describing and validating EO products annotated with our proposed schema.org extension.

6. CONCLUSIONS AND FUTURE WORK

The objective of the current diploma thesis is to enable the publication of EO datasets on the web and their effective discovery by modern search engines like Google. By developing the extension of the schema.org vocabulary about Earth observation, we aim to make search engines able to discover EO datasets in the same way that they can discover information about actors, movies, etc. The proposed extension has been published at `https://eop-sch.appspot.com/EarthObservation`.

Our approach extended the class Dataset of schema.org with subclasses and properties which cover the EO dataset metadata defined in OGC 17-003. A minimal set of new features was added so that well-known kinds of EO datasets (e.g., optical) and their characteristics (e.g., cloud or snow cover percentage) are covered. In addition, we used SHACL to create a shapes graph, based on OGC 17-003 as well, in order to model and validate EO datasets annotated with our schema.org extension. Moreover, we annotated EO datasets produced by the Copernicus programme as specified in section 4.

As an initial step of our future work, we would like to provide the shapes graphs in *JSON-LD* format as well. Moreover, the resulting extension of the schema.org vocabulary will be submitted to schema.org for adoption.

ACRONYMS

API	Application Programming Interface
CEOS	Committee on Earth Observation Satellites
CLC	Corine Land Cover
CSV	Comma-Separated Values
DAWG	RDF Data Access Working Group
EC	European Commission
EO	Earth Observation
EOP	Earth Observation Product
GML	Geography Markup Language
HTTP	HyperText Transfer Protocol
IRI	Internationalized Resource Identifier
ISA	Interoperability Solutions for European Public Administrations
ISO	International Organization for Standardization
JSON	JavaScript Object Notation
JSON-LD	Javascript Object Notation for Linked Data
LDP	Linked Data Protocol
LOD	Linked Open Data
OGC	Open Geospatial Consortium
O&M	Observations & Measurements
OWL	Web Ontology Language
OWS	OGC Web Services
PDF	Portable Document Format
RDF	Resource Description Framework
RDFS	RDF Schema
SPARQL	SPARQL Protocol and RDF Query Language
UML	Unified Modeling Language
UMM	Unified Metadata Model
UMM-G	Unified Metadata Model for Granules
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
W3C	World Wide Web Consortium
WKT	Well-Known Text
XML	eXtensible Markup Language
XSD	XML Schema Definition Language

NAMESPACE ABBREVIATIONS

alt	http://www.opengis.net/alt/2.1/
atm	http://www.opengis.net/atm/2.1/
atom	http://www.w3.org/2005/Atom/
dct	http://purl.org/dc/terms/
eop	http://www.opengis.net/eop/2.1/
gj	https://purl.org/geojson/vocab#
gsp	http://www.opengis.net/ont/geosparql#
iana	http://www.iana.org/assignments/relation/
ical	http://www.w3.org/2002/12/cal/ical#
lmb	http://www.opengis.net/lmb/2.1/
media	http://search.yahoo.com/mrss/
opt	http://www.opengis.net/opt/2.1/
owc	http://www.opengis.net/owc/1.0/
owl	http://www.w3.org/2002/07/owl#
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs	http://www.w3.org/2000/01/rdf-schema#
sar	http://www.opengis.net/sar/2.1/
skos	http://www.w3.org/2004/02/skos/core#
xs	http://www.w3.org/2001/XMLSchema-datatypes#
xsd	http://www.w3.org/2001/XMLSchema#

REFERENCES

[1] Reiter Baynes, Gazula. UMM-G: Unified Metadata Model for Granules. Technical report, 2015.

[2] Konstantina Bereta, Hervé Caumont, Erwin Goor, Manolis Koubarakis, Despina-Athanasia Pantazi, George Stamoulis, Sam Ubels, Valentijn Venus, and Firman Wahyudi. From big data to big information and big knowledge: A demo from the copernicus app lab project. In *International Conference on Information and Knowledge Management (CIKM)*, 2018.

[3] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. 1998.

[4] Yves Coene. OGC 17-003 - Earth Observation Dataset Metadata Vocabulary. Technical report, 2018.

[5] Open Geospatial Consortium. OGC 10-157r4 Earth Observation Metadata profile of Observations & Measurements. Technical report, 2016.

[6] Open Geospatial Consortium. OGC OWS Context GeoJSON Encoding Standard. Technical report, 2017.

[7] R.V. Guha, Dan Brickley, and Steve Macbeth. Schema.org: Evolution of Structured Data on the Web. *Queue*, 2015.