RDF Validation of Metadata on Person-Level and Aggregated Data

Thomas Bosch[1], Benjamin Zapilko[1], Joachim Wackerow[1], and Kai Eckert[2]

[1] GESIS – Leibniz Institute for the Social Sciences, Germany
{firstname.lastname}@gesis.org,
[2] University of Mannheim, Germany
kai@informatik.uni-mannheim.de

Abstract. For research institutes, data libraries, and data archives, RDF data validation according to predefined constraints is a much sought-after feature, particularly as this is taken for granted in the XML world. The data most often used in research within the community around research data for the social, behavioural, and economic sciences is person-level data, i.e. data collected about individuals (households, businesses). While performing research, the detailed, often access restricted person-level data is aggregated into less confidential publicly available multi-dimensional tables which answer particular research questions and whose purpose is to gain an interest in further more detailed analyses on the underlying person-level data. To ensure high quality and trust, metadata and data must satisfy certain criteria - specified in terms of RDF constraints.

In this paper, we show how metadata and underlying data on different level of aggregation as well as collections of these data sets are represented in RDF and how therefore used vocabularies are interrelated. We explain why RDF validation is important in this context and how (meta)data is validated against RDF constraints to ensure high quality and trust.

Keywords: RDF Validation, RDF Constraints, DDI-RDF Discovery Vocabulary, Disco, RDF Data Cube Vocabulary, Linked Data, Semantic Web

1 Introduction

Thomas: terminology: we use data in singular form / we use data set instead of dataset / We use person-level data and not microdata

For more than a decade, members of the community around research data for the social, behavioural, and economic (SBE) sciences have been developing and using a metadata standard (composed of almost twelve hundred metadata fields) known as the *Data Documentation Initiative (DDI)* [9]. DDI is an XML format designed for the purposes of supporting the dissemination, management, and reuse of the data collected and archived for research purposes. DDI is heavily used by the CESSDA community of European national data archives, the International Household Survey Network community (made up of more than 90 statistical agencies), and ICPSR - the largest SBE data archive in the US. Increasingly, data professionals, national statistical institutes, data archives, data libraries, and government statisticians (e.g. data.gov, data.gov.uk) are very interested in having their data be discovered and used by providing their metadata on the web in form of RDF (e.g. metadata about unemployment rates or income).

Recently, members of the SBE and Linked Data community developed the *DDI-RDF Discovery Vocabulary (Disco)*[3], an effort to leverage the mature DDI metadata model for the purposes of exposing DDI metadata as resources within the Web of Linked Data. For data archives, research institutes, and data libraries, RDF data validation according to predefined constraints is a much sought-after feature, particularly as this is taken for granted in the XML world (DDI-XML documents are validated against diverse XSDs[4]). Several approaches exist to meet this requirement, ranging from using *OWL 2* as a constraint language to *SPIN*[5], a SPARQL-based way to formulate and check constraints. There are also specific constraint languages like *Shape Expressions*[6], *Resource Shapes*[7] or *Description Set Profiles*[8] that more or less explicitly address the aforementioned SBE community. Bosch and Eckert[1] use SPIN as basis to define a validation environment (http://purl.org/net/rdfval-demo) in which the validation of any constraint language[9] can be implemented by representing them in SPARQL. The SPIN engine checks for each resource if it satisfies all constraints (associated with its assigned classes) and generates a result RDF graph containing information about all constraint violations.

2 Motivation

The data most often used in research within the SBE community is *person-level data*, i.e. data collected about individuals (and sometimes also businesses and households) in the form of responses to surveys or taken from administrative registers (such as hospital records, registers of births and deaths). The range of person-level data is very broad (covering many different domains), including census, education, and health data as well as all types of business, social, and labor force surveys. Increasingly, this type of research data is held within data archives or data libraries after it has been collected, so that it may be reused by future researchers. In performing their research, the detailed person-level data is aggregated into less confidential multi-dimensional tables which answer particular research questions. Portals harvest metadata (as well as publicly available data) from multiple data providers in form of RDF. To ensure high quality, the metadata must satisfy certain criteria - specified in terms of RDF constraints.

[3] http://rdf-vocabulary.ddialliance.org/discovery.html
[4] http://www.ddialliance.org/Specification/
[5] http://spinRDF.org/
[6] http://www.w3.org/Submission/shex-primer/
[7] http://www.w3.org/Submission/shapes/
[8] http://dublincore.org/documents/2008/03/31/dc-dsp/
[9] the only limitation is that constraint languages must be represented in RDF

After validating the metadata according to these constraints, portals offer added values to their customers, e.g. by searching over and comparing metadata of multiple providers.

By its nature, person-level data is highly confidential, and access is often only permitted for qualified researchers who must apply for access. The purpose of publicly available aggregated data, on the other hand, is to get a first overview and to gain an interest in further analyses on the underlying person-level data. Researchers typically represent their results as aggregated data in form of two-dimensional tables with only a few columns (so-called *variables* such as *sex* or *age*). The *RDF Data Cube Vocabulary (Data Cube)*[10] is a W3C recommendation for representing *data cubes*, i.e. multi-dimensional aggregate data, in RDF [4]. Aggregate data is derived from person-level data by statistics on groups or aggregates such as counts, means, and frequencies. The SDMX metadata standard[11] – used as the basis for *Data Cube* – and DDI have traditionally made efforts to align their content. Similarly, some of the developers of *Disco* were also involved in the development of *Data Cube*, allowing the RDF versions of these standards to retain that alignment. While *Disco* and *Data Cube* provide terms for the description of data sets, both on a different level of aggregation, the *Data Catalog Vocabulary (DCAT)*[12] enables the representation of these data sets inside of data collections like repositories, catalogs, or archives. The relationship between data collections and their contained data sets is useful, since such collections are a typical entry point when searching for data. Although, in most cases aggregated data is still published in form of PDFs, it is more and more common to publish aggregated data as CSV files, allowing to perform first calculations (either using all variables or only a subset). In 2014, SBE and Linked Data community members developed the *Physical Data Description (PHDD)*[13] vocabulary to represent aggregated and person-level data in a rectangular format. The data could be either represented in records with character-separated values (CSV) or in records with fixed length.

For more detailed analyses, researchers refer to person-level data from which aggregated data is derived from, as person-level data include additional variables needed for further research. One very common example for detailed analyses on person-level data is the content-driven comparison of multiple surveys. Researchers get promising findings (in form of published tables with a few columns) within a metadata portal leading to subsequent research questions like 'How to compare the unemployment rate of different countries (e.g. Germany, UK, and France) in the last 10 years grouped by age?'. The first step is to determine in which countries the unemployment rate is collected and which other variables of each country-specific survey are theoretically comparable and can therefore be used to answer the underlying research question. Variables are constructed out of values (of one or multiple datatypes) and/or code lists. The variable *age*,

[10] http://www.w3.org/TR/vocab-data-cube/
[11] http://sdmx.org/
[12] http://www.w3.org/TR/vocab-dcat/
[13] https://github.com/linked-statistics/physical-data-description

e.g., may be represented by values of the datatype *xsd:nonNegativeInteger*, or by a code list including multiple age clusters (such as '0 to 10' and '11 to 20'). To determine if variables measuring *age* - collected within multiple surveys of different countries (age_{DE}, age_{UK}) - are comparable, both content-driven and technology-driven validation is performed either within our developed validation environment or by matching algorithms. An example for a content-driven validation is to investigate if variables are represented in a compatible way, i.e. are the variables' code lists theoretically comparable. Technically, it can be validated (1) if variable definitions are available, (2) if code lists are properly structured, and (3) if for each code an associated category (a human-readable label) is specified.

Thomas: maybe this paragraph should be moved to one of the subsequent sections and not be part of introduction

Data providers and harvesters do not only offer metadata but also publicly available data on different level of detail. To ensure high data quality and trust, they have to analyze and validate the data (are fundamental data fragments available?, how does valid data look like?). Provenance (where does the data come from?) is an important aspect in evaluating data quality. As data searchers know exactly which data sources they trust and which are reasonable to meet their individual use cases, data validation can only be performed semi-automatically, i.e. an automatic approach serves as basis for intellectual decisions.

This paper aims to address two main **audiences**: (1) metadata practitioners seeking for how to represent metadata on data sets on different aggregation levels and (2) metadata providers and harvesters ensuring high quality metadata by validating metadata on highly complex RDF data sets. In this paper, we show in form of a complete real world running example how to represent metadata on highly complex person-level data (*Disco*), metadata on aggregated data (*Data Cube*), and aggregated data in a rectangular format (*PHDD*) in RDF and how therefore used vocabularies are interrelated (**contribution 1**, sections 3-4). We explain why RDF validation is important in this context and how (meta)data is validated against RDF constraints to ensure high quality (meta)data within this complex of data sets on different levels of aggregation (**contribution 2**, section 5). The remainder of the paper is structured as follows.

3 Running Example

Eurostat[14] is the statistical office of the European Union. Its task is to provide statistics at European level that enable comparisons between countries and regions. Eurostat provides publicly available European aggregated data (downloadable in machine-readable format like CSV files) and its metadata (only textual descriptions). SBE researchers have a strong interest, e.g., in the availability of childcare services across European Union Member States. The variable *formal childcare*[15] (in contrast to childcare at home) captures the measured availability of childcare services in percent over the population. The present data collection

[14] http://ec.europa.eu/eurostat

[15] http://appsso.eurostat.ec.europa.eu/nui/show.do?dataset=ilc_caindformal&lang=en

refers to data on formal childcare and other types of care by the variables *year*, *duration* (0 hours, 1 - 29 HPW; 30 HPW), *age* of the child (0-2 years; 3 to admission age for compulsory school; admission age for compulsory school to 12) and *country*. As Eurostat provides aggregated data (here a two-dimensional table) as CSV files, we can easily represent the aggregated data with *PHDD* in RDF. This conversion enables further aggregations and calculations, e.g., in order to compare *formal childcare* between Northern and Southern Europe or between otherwise grouped countries.

For a broader view of the data framework and more detailed analyses we refer to the metadata on person-level data collected for the series *EU-SILC (European Union Statistics on Income and Living Conditions)*[16]. The *Microdata Information System (MISSY)*[17] is an online service platform that provides systematically structured metadata for official statistics on European person-level data sets. This includes data documentation at the study and variable level as well as documentation materials, tools and further information. The aggregated variable *formal childcare* is calculated on the basis of six person-level variables like *Education at pre-school*[18]. For each person-level variable detailed metadata is given (definitions and descriptions, theoretical concepts, underlying questions, code lists, frequencies and descriptive statistics, countries, year of data collection, classifications) enabling researchers to replicate the results shown in the aggregated data tables from Eurostat. Metadata on person-level data enables researchers to investigate further research questions based on promising findings of other researchers in form of aggregated data. One common research question is ,e.g., the comparison of variables like *formal childcare* between countries, for which the variable is collected within the context of an individual survey, and other European or non European countries (e.g. OSCE).

4 Represent Metadata and Data in RDF

Data Cube is a vocabulary to represent metadata on multi-dimensional aggregate data.

```
EXAMPLE
```

PHDD is a vocabulary to represent aggregated data in a rectangular format. The data could be either represented in records with character-separated values (CSV) or in records with fixed length.

```
EXAMPLE
```

Disco is a vocabulary to represent person-level data in RDF.

[16] http://www.gesis.org/missy/eu/metadata/EU-SILC
[17] http://www.gesis.org/missy/eu/missy-home
[18] http://www.gesis.org/missy/eu/metadata/EU-SILC/2011/Cross-sectional/original#2011-Cross-sectional-RL010

```
EXAMPLE
```

DCAT enables to represent aggregated and person-level data inside of data collections like repositories, catalogs, or archives.

```
EXAMPLE
```

The *Simple Knowledge Organization System (SKOS)* is used multiple times within the context of aggregated and person-level (meta)data, e.g., to represent controlled vocabularies or hierarchies of *SBE theoretical concepts* (e.g. education).

```
EXAMPLE
```

XKOS[19] is a *SKOS* extension to describe formal statistical classifications like the International Standard Classification of Occupations (*ISCO*) and the Statistical classification of economic activities in the European Community *NACE*.

```
EXAMPLE
```

5 RDF Validation of Metadata and Data

Bosch et al. identified in total 74 requirements to formulate RDF constraints; each of them corresponding to a constraint type. We published a technical report[20] in which we explain each requirement (constraint type) in detail and give examples for each (represented by different constraint languages). The knowledge representation formalism *Description logics (DL)*, with its well-studied theoretical properties, provides the foundational basis for each constraint type. Therefore, this technical report contains mappings to DL to logically underpin each requirement and to determine which DL constructs are needed to express each constraint type [2]. Bosch et al. recently published a technical report[21] (serving as appendix of this paper) in which we describe RDF constraints to validate metadata on person-level and aggregated data. We assign each constraint to constraint types corresponding to RDF validation requirements or to data model specific constraint types[22] [3].

Thomas: ToDo: publish technical report

We distinguish two validation types: *Content-Driven Validation* contains the set of constraints ensuring that the data is consistent with the intended semantics of data models, i.e. the integrity of the data according to data models.

[19] https://github.com/linked-statistics/xkos
[20] Available at: http://arxiv.org/abs/1501.03933
[21] Available at: http://arxiv.org/abs/XXXXX
[22] Requirements/Constraint types and constraints are uniquely identified by alphanumeric technical identifiers like *R-1*

Technology-Driven Validation includes the set of constraints which can be generated automatically out of defined data models, such as cardinality restrictions, universal and existential quantifications, domains, and ranges. We determined the default *severity level* (*R-158*) for each constraint indicating how serious the violation of the constraint is. The continuum of severity levels ranges from informational via warning and error to fatal error. Although we provide default severity levels for each constraint, users should be able to specify the severity levels of constraints they need to validate for their individual use cases, i.e., users should be able to define use case specific severity levels for constraints.

—

RDF validation scenarios require the closed-world assumption (CWA) (i.e., a statement is inferred to be false if it cannot be proved to be true).

application of matching algorithms.- the value for a specific data property (e.g. birth date) is needed as prerequisite to perform matching algorithms - 1. step: validation - 2. step: matching - validation as reasonable extension of pattern of Ben (e.g. for pattern: is this digit correct?)

Thomas: @Ben ToDo: macht es sinn matching algorithmen anzuwenden?

5.1 Aggregated (Meta)Data

Thomas: ToDo

- for each dimension there should be a description and code lists.
- for each code list there should be a description.
- there should be a relationship to the underlying person-level data.

5.2 Person-Level (Meta)Data

Surveys. Each survey must have an abstract (*R-46: data property facets*). If the abstract of the survey is missing, a title of the survey has to be stated (*R-71: conditional properties*).

Variables.

- variable does not have associated codes / categories

Theoretical Concepts. Variables must have at least one relationship to a theoretical concept (*R-75: minimum qualified cardinality restrictions*). The variable *Education at pre-school*, e.g., is associated with the theoretical concept *education*. The severity level of this constraint is *warning* and not *error*, as research can be continued without an associated theoretical concept.

A very common research question is to compare variables (*comparison*) of multiple surveys or countries. To compare variables, (1) variables and (2) variable definitions must be present, (3) code lists must be structured properly, (4) for each code an associated category (human-readable label) must be specified, and (5) code lists must either be identical or at least similar. If a researcher wants to get a first overview over comparable variables (use case 1), the first three constraints may be sufficient for this purpose. Thus, the severity level of the first three constraints is stronger than the severity level of the next constraints. If the intention of the researcher is to perform more detailed comparisons (use case 2), however, the violation of the remaining constraints is also more serious.

Constraints of the constraint type *Context-Specific Exclusive OR of Property Groups* (*R-11*) restrict individuals of given classes to have only one of multiple property groups. Within the context of *Disco*, *skos:Concept*s can have either *skos:definition* (when interpreted as theoretical concepts) or *skos:notation* and *skos:prefLabel* properties (when interpreted as codes and categories), but not both.

$$\begin{aligned}
\text{Concept} \sqsubseteq (\neg\ \text{D} \sqcap \text{C}) \sqcup (\text{D} \sqcap \neg\ \text{C})\\
\text{D} \equiv \text{A} \sqcap \text{B}\\
\text{A} \sqsubseteq\ \geqslant 1\ \text{notation.string} \sqcap \leqslant 1\ \text{notation.string}\\
\text{B} \sqsubseteq\ \geqslant 1\ \text{prefLabel.string} \sqcap \leqslant 1\ \text{prefLabel.string}\\
\text{C} \sqsubseteq\ \geqslant 1\ \text{definition.string} \sqcap \leqslant 1\ \text{definition.string}
\end{aligned}$$

It is useful to declare properties to be conditional, i.e. if particular properties exist, then specific other properties must also be present (*R-71: conditional properties*). If a *skos:Concept* represents a code (having a *skos:notation* property) and a category (having a *skos:prefLabel* property), then the property *disco:isValid* has to be stated indicating if the code is valid (*true*) or missing (*false*).

——

Restricting data property ranges $[V_{min}, V_{max}]$ within the context of given classes is a very common requirement (*R-45: literal ranges*) - disco:CategoryStatistics resources, e.g., point to percentage values via the data property *disco:percentage*; these values must be of the datatype *xsd:double* and must be between 0 and 100.

——

It is often useful to declare a given (data) property as the *primary key* (*R-226*) of a class, so that a system can enforce uniqueness and also automatically build URIs from user inputs and imported data. In *Disco*, resources are uniquely identified by the property *adms:identifier*, which is therefore inverse-functional (`funct identifier`$^{-}$), i.e. for each *rdfs:Resource* x, there can be at most one distinct *rdfs:Resource* y such that y is connected by *adms:identifier*$^{-}$ to x. Keys, however, are even more general than inverse-functional properties (*R-58*), as a key can be a data, an object property, or a chain of properties [8]. For this generalization purposes, as there are different sorts of key, and as keys can lead to undecidability, DL is extended with *key boxes* and a special *keyfor* construct (`identifier keyfor Resource`) [6].

——

In many cases, resources must be members of listed controlled vocabularies (*R-32*). *disco:SummaryStatistics*, e.g., can only have *disco:summaryStatisticType* relationships to *skos:Concept*s which must be members of the controlled vocabulary *ddicv:SummaryStatisticType* which is a *skos:ConceptScheme*.

$$\begin{aligned}
\text{SummaryStatistics} \sqsubseteq \forall summaryStatisticType.A\\
A \equiv Concept \sqcap \forall inScheme.B\\
B \equiv ConceptScheme \sqcap \{SummaryStatisticType\}
\end{aligned}$$

——

Depending on property datatypes, two different literal values have a specific ordering with respect to an operator. *P1* and *P2* are the data properties we need to compare and *OP* is the comparison operator (e.g. <, <=, >, >=, =, !=) (*R-43: literal value comparison*). *disco:startDate*s, e.g., must be before (<) *disco:endDate*s. To validate this constraint we bind the variables (P1: *disco:startDate*, P2: *disco:endDate*, OP: <).

—–

Objects/literals can be ordered (*R-121, R-217*) for given properties. *Disco* variables, questions, and codes/categories are typically organized in a particular order. If codes/categories of a *disco:Representation* of a *disco:Variable* should be ordered, the variable representation should also be of the type *skos:OrderedCollection* containing multiple codes/categories (each represented as *skos:Concept*) in a *skos:memberList*.

—–

Mathematical Operations (*R-41, R-42*; e.g. date calculations and statistical computations like average, mean, and sum) are performed to ensure the integrity of data models. The sum of *disco:percentage* (datatype: *xsd:double*) values of all codes (*skos:Concept*s) of a code list (*skos:ConceptScheme* or *skos:OrderedCollection*), serving as representation of a particular *disco:Variable* must exactly be 100.

—–

DCAT Validation

SKOS and XKOS Validation

SKOS is based on RDF, which is a graph-based data model. Therefore, we can concentrate on the vocabulary's graph-based structure for assessing the quality of SKOS vocabularies and apply graph- and network-analysis techniques (*structure*) like (1) a vocabulary should provide entry points (top concepts) to the data to provide efficient access and guidance for human users and (2) concepts, internal to the tree, should not be indicated as top concepts, and (3) a vocabulary should not contain many orphan concepts (An orphan concept is a concept without any associative or hierarchical relations lacking valuable context information. A controlled vocabulary that contains many orphan concepts is less usable for search and retrieval use cases, as, e.g., no hierarchical query expansion can be performed on search terms to find documents with more general content.) [7]. - SKOS validation of hierarchies of theoretical concepts

Thomas: add disco example

XKOS Validation:

Thomas: ToDO: ask Franck if INSEE already expressed well known classification systems with XKOS

Data Integration. use RDF validation for data integration

5.3 Technology-Driven Validation

Constraints of some constraint types are directly and automatically derived from conceptual models of multiple vocabularies. This way, *property domains and ranges, universal and existential quantifications, minimum, maximum, and exact cardinality restrictions* are defined for each property of vocabularies against which one wish to validate. As these constraints directly depend on the intended semantics of data models, associated severity levels are very strong.

Vocabularies should not invent any new terms or use deprecated elements. The *vocabulary* constraint should be the first one to be checked for each reused vocabulary. *Property Domains* (*R-25, R-26*) and *Ranges* (*R-28, R-35*) restrict domains and ranges of properties. Only *skos:ConceptSchemes*, e.g., can have *skos:hasTopConcept* relationships ($\exists$ `hasTopConcept.`$\top \sqsubseteq$ `ConceptScheme`) and *xkos:belongsTo* relationships can only point to instances of the class *skos:Concept* ($\top \sqsubseteq \forall$ `belongsTo.Concept`). A *universal quantification* (*R-91*) contains all those individuals that are connected by a property only to individuals/literals that are instances of particular classes or data ranges. Only *dcat:Catalogs*, e.g., can have *dcat:dataset* relationships to *dcat:Datasets* (`Catalog` $\sqsubseteq \forall$ `dataset.Dataset`). An *existential quantification* (*R-86*) contains all those individuals that are connected by a property to individuals/literals that are instances of given classes or data ranges. Every *qb:SliceKey*, e.g., must be associated with (*qb:sliceKey*) a *qb:DataStructureDefinition* (`SliceKey` $\sqsubseteq \exists$ `sliceKey`$^{-}$`.DataStructureDefinition`).

Minimum/maximum/exact qualified cardinality restrictions (*R-74, R-75, R-76*) contain all those individuals that are connected by a property to at least/at most/exactly n different individuals/literals that are instances of particular classes or data ranges. *Minimum qualified cardinality restrictions* are defined for each property of multiple vocabularies like *Disco*. A *disco:Questionnaire*, e.g., has at least one *disco:question* relationship to *disco:Questions* (`Questionnaire` $\sqsubseteq \geqslant 1$ `question.Question`), a *disco:Variable* has at most one *disco:concept* relationship to a theoretical concept (*skos:Concept*) (`Variable` $\sqsubseteq \leqslant 1$ `concept.Concept`), and every *qb:DataSet* has (*qb:structure*) exactly one associated *qb:DataStructureDefinition* (`DataSet` $\sqsubseteq \geqslant 1$ `structure.DataStructureDefinition` $\sqcap \leqslant 1$ `structure.DataStructureDefinition`). For particular data properties, values of predefined languages must be stated for determined number of times (*R-48, R-49*). Some controlled vocabularies contain literals in natural language, but without information what language has actually been used. Language tags might also not conform to language standards. Some concepts in a thesaurus are labeled in only one language, some in multiple languages. It may be desirable to have each concept labeled in each of the languages that also are used on the other concepts. This is not always possible, but incompleteness of language coverage for some concepts can indicate shortcomings of the vocabulary (severity level: informational) [7].

All properties, not having the same domain and range classes, are defined to be pairwise disjoint (*R-9: disjoint properties*), stating that no individual x can be connected to an individual/literal y by disjoint properties like *phdd:isStructuredBy* and *phdd:column* (*isStructuredBy* $\sqsubseteq \neg$*column*). All *Disco* classes are defined to be pairwise disjoint (*R-7: disjoint classes*; e.g. `Study` $\sqcap$ `Variable` $\sqsubseteq \bot$), i.e. no individual can be at the same time an instance of more than one disjoint class. It is a common requirement to narrow down the value space of properties by an exhaustive enumeration of valid values. *Allowed values* (*R-30, R-37*) for properties can be IRIs (matching one or multiple patterns), any literals, allowed literals (e.g. 'red' 'blue' 'green'), and typed literals of one or multiple type(s) (e.g. *xsd:string*) - *disco:CategoryStatistics*, e.g., can only have *disco:computationBase* relationships to the values *valid* and *invalid* of the datatype *rdf:langString* (`CategoryStatistics` $\equiv \forall$ `computationBase.`$\{$`valid,invalid`$\}$ $\sqcap$ `langString`).

Constraints of some constraint types can be derived from vocabularies' data models for which reasoning can be performed prior to validation which enables to resolve possible constraint violations. *Subsumption* (*R-100*) states that the class *C1* is a sub-class of the class *C2* - *C1* is more specific than *C2*, i.e. each resource of the class *C1* must also be part of the class extension of *C2*. All *disco:Universe*s, e.g., must also be *skos:Concept*s (`Universe` $\sqsubseteq$ `Concept`). *Sub Properties* (*R-54, R-64*) state that the property *P1* is a sub-property of the property *P2* - that is, if individual x is connected by *P1* to individual/literal y, then x is also connected by *P2* to y. If x is connected by *disco:fundedBy* to y, then x is also connected by *dcterms:contributor* to y (`fundedBy` $\sqsubseteq$ `contributor`). *Asymmetric object properties* (*R-62*) restrict that if individual x is connected by the object property *OP* to individual y, then y cannot be connected by *OP* to x. *Asymmetric object properties* are defined for each asymmetric object property for which a semantically equivalent object property pointing from the other direction may also be possible to be specified in the appropriate vocabulary. A *disco:Variable*, e.g., may be based on (*disco:basedOn*) a *disco:RepresentedVariable*. A *disco:RepresentedVariable*, however, cannot be based on a *disco:Variable* ($basedOn \sqcap basedOn^{-} \sqsubseteq \bot$). *Default values* (*R-31, R-38*) for objects/literals of given prooperties are inferred automatically when properties are not stated, in order to pre-populate input forms and to insert a required property that is missing in a web service call. The value *true* for the property *disco:isPublic* (*xsd:boolean*) indicates that the data set (*disco:LogicalDataSet*) can be accessed by anyone. Per default, access to data sets should be restricted (*false*).

—

For future vocabulary versions, out-dated classes and properties can be marked as deprecated. There are constraint types (*context-specific valid classes and properties*) to specify which classes and properties are valid in which context (here a specific vocabulary version). Many properties are not necessarily required but recommended within a particular context (*recommended properties*). The property *skos:notation* is not mandatory for *disco:Variable*s, but recommended to indicate variable names.

—

It has to be ensured that a value is valid for its datatype (*Value is Valid for Datatype*), e.g., that a date is really a date, or that a *xsd:nonNegativeInteger* value is not negative. It is checked if all literals of *xsd:date* properties are really dates (e.g. *disco:startDate*, *disco:endDate*, *dcterms:date*).

—

The validation of instances data (direct or indirect) exploits the sub-class or sub-property link in a given ontology (*Use Sub-Super Relations in Validation*). This validation can indicate when the data is verbose (redundant) or expressed at a too general level, and could thus be improved. If *dcterms:coverage* and one of its sub-properties (*dcterms:spatial*, *dcterms:temporal*) are present, it is checked that *dcterms:coverage* is not redundant with its sub-properties.

—

6 Implementation

- Missy / DDI-RDF export / multiple concrete syntaxes / Triple store / content negotiation / reference implementation of the Disco data model / Missy data model is based in Disco data model

- RDF validator

Validation of Data Cube instances.

– is there an implementation available yet? If not I implement it in our validation environment
– example QB data set / we could use the one from the QB spec

7 Evaluation

– validate constraints on complex example data set
– count Disco, QB, PHDD, XKOS, DCAT constraints by constraint groups

8 Related Work

With RDF validation, one can overcome the drawbacks when validating XML documents. Certain things cannot be validated using XSDs. As a consequence, so-called secondary-level validation tools like Schematron have been introduced to overcome the limitations of XSDs. Schematron generated validation rules and validates XML documents according to them. It cannot be validated if each code of a variable's code list is associated with a category . It cannot be validated that if an element has a specific value, then certain child elements must be present.

Thomas: further explanation needed

Thomas: Achim, which kind of things cannot be validated using XSDs / we also need references

http://www.xmlmind.com/xmleditor/_distrib/doc/xmltool/xsd_structure_limitations.html

https://msdn.microsoft.com/en-us/library/aa468554.aspx

RDF Data Cube Vocabulary. A well-formed RDF Data Cube is an a RDF graph describing one or more instances of *qb:DataSet* for which each of the 22 defined integrity constraints[23] passes. Each integrity constraint is expressed as narrative prose and, where possible, a SPARQL ASK query or query template. If the ASK query is applied to an RDF graph then it will return true if that graph contains one or more Data Cube instances which violate the corresponding constraint [5].

DCAT. are there already constraints defined for DCAT?

9 Conclusion and Future Work

References

1. Thomas Bosch and Kai Eckert. Towards description set profiles for rdf using sparql as intermediate language. *Proceedings of the DCMI International Conference on Dublin Core and Metadata Applications (DC 2014)*, 2014.

[23] http://www.w3.org/TR/vocab-data-cube/#wf

2. Thomas Bosch, Andreas Nolle, Erman Acar, and Kai Eckert. Rdf validation requirements - evaluation and logical underpinning. 2015.
3. Thomas Bosch, Benjamin Zapilko, Joachim Wackerow, and Kai Eckert. Rdf constraint types to validate metadata on highly-complex person-level and aggregated data. 2015.
4. Richard Cyganiak, Simon Field, Arofan Gregory, Wolfgang Halb, and Jeni Tennison. Semantic statistics: Bringing together sdmx and scovo. In Christian Bizer, Tom Heath, Tim Berners-Lee, and Michael Hausenblas, editors, *Proceedings of the WWW 2010 Workshop on Linked Data on the Web*, volume 628 of *CEUR Workshop Proceedings*, 2010.
5. Richard Cyganiak and Dave Reynolds. The rdf data cube vocabulary. W3C recommendation, W3C, January 2014.
6. Carsten Lutz, Carlos Areces, Ian Horrocks, and Ulrike Sattler. Keys, nominals, and concrete domains. *Journal of Artificial Intelligence Research*, 23(1):667–726, June 2005.
7. Christian Mader, Bernhard Haslhofer, and Antoine Isaac. Finding quality issues in skos vocabularies. In *Proceedings of the Second International Conference on Theory and Practice of Digital Libraries*, TPDL'12, pages 222–233, Berlin, Heidelberg, 2012. Springer-Verlag.
8. Michael Schneider. OWL 2 Web Ontology Language RDF-Based Semantics. W3C recommendation, W3C, October 2009.
9. Mary Vardigan, Pascal Heus, and Wendy Thomas. Data documentation initiative: Towards a standard for the social sciences. *International Journal of Digital Curation*, 3(1):107–113, 2008.