SSoLDAC 2023 - ontologies

Mathias Bonduel - Neanex Technologies

neanex

Introduction

Mathias Bonduel - LinkedIn - mathias.bonduel@neanex.com

- **Doctor in Civil Engineering Technology (Jan 2017 - May 2021)**
  - PhD dissertation on Linked Data for built heritage

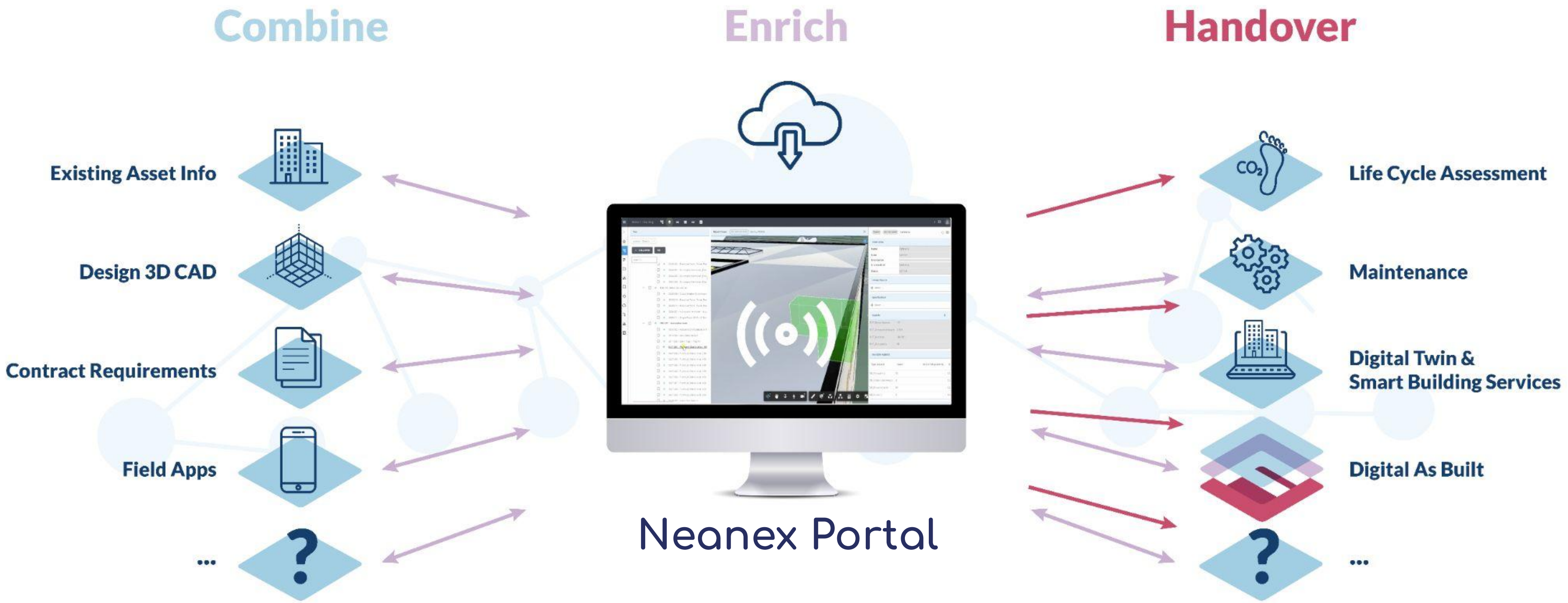
- **Co-chair of the W3C Linked Building Data community group (since Jan 2021)**

- **Working for Neanex Technologies, in Antwerp (since April 2021)**
  - technical product owner and consultant for the Neanex Portal

- **Originally from Bruges, currently living in Ghent**

neanex

Introduction - Neanex Portal



Outline

1. Ontologies - *what & why?*
2. Scope of ontologies
3. Types of ontologies & examples
4. Languages for Linked Data ontologies: RDFS, OWL, SKOS and SHACL
5. Best practices for (Linked Data) ontology *engineering*
6. Best practices for (Linked Data) ontology *publishing*
7. References - further reading

neanex

1. **Ontologies - *what & why?***

neanex

1. Ontologies - *what?*

"An ontology is
a formal, explicit specification of a shared
conceptualisation"
[1, p. 184].

Synonym: conceptual information model
Specializations: Object Type Library, masterdata, a Linked Data ontology, taxonomy, partonomy, dictionary, etc.

[1] R. Studer, V. R. Benjamins, and D. Fensel. "Knowledge Engineering: Principles and methods". In: Data & Knowledge Engineering 25.1-2 (1998), pp. 161–197. doi: 10.1016/S0169-023X(97)00056-6.

neanex

1. Ontologies - *what?*

1. **conceptualization**
   a. it's different from your dataset level
   b. always an approximation
   c. for a specific domain of interest (scope)
2. **formal and explicit**
   a. not only machine-readable, but even machine-interpretable (formal logics - unambiguous)
   b. logics supported by a language: trade-off between expressivity and efficiency
3. **shared**
   a. created for multiple users
   b. shareable (standards)
   c. Application-independent

In summary: *not* an exchange format nor database model > higher abstraction layer

neanex

1. Ontologies - *why?*

European Commission

EU layers of interoperability [2]:

1. Legal → legislation & contracts
2. Organizational → business workflows & exchange requirements
3. Semantic → meaning, content of data
4. Technical → interface specifications, communication protocols & supporting infrastructure

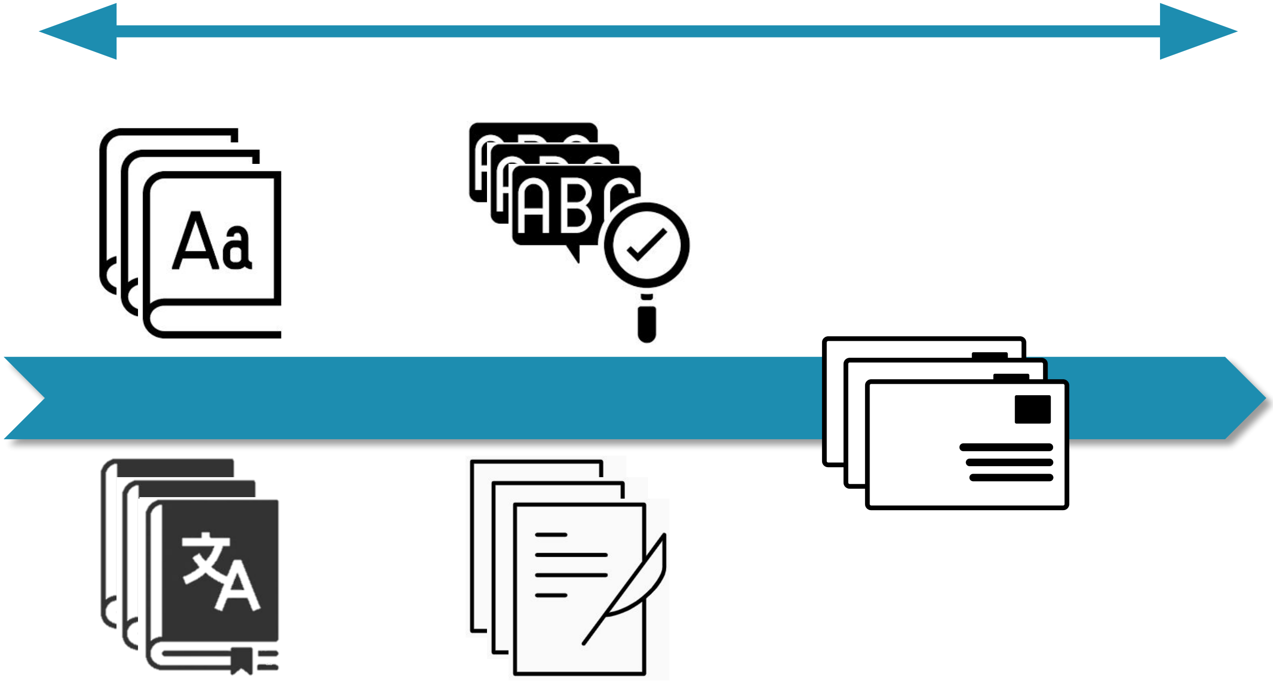
[2] Directorate-General for Informatics (European Commission). New European Interoperability Framework - Promoting seamless services and data flows for European public administrations. Tech. rep. 2017, p. 48. doi: 10.2799/78681.

neanex

1. Ontologies - why?

semantic aspects

technical aspects



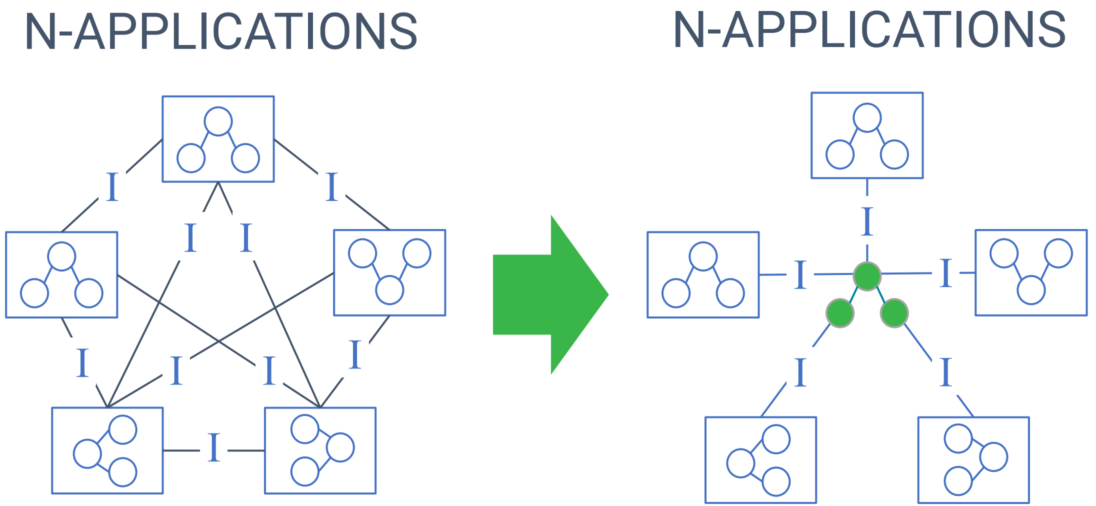
Successful exchange of information between parties

neanex

1. Ontologies - *why?*

Semantic interoperability for:

1. support in: software integrations + kickstart new software developments
2. making knowledge explicit and sharing between individuals, departments and/or organizations



N!/(2*(N-2)!) Interfaces

N-Interfaces

[3] Semmtech - SEm for Infra

neanex

2. Scope of ontologies

neanex

Scope of ontologies > content (“concepts” or “terms”)

1. stable identifier for term
2. text name/label for term
3. text definition for term
4. part of ontology
5. active/depreciated
6. kind of concept (class, relations, properties/attributes, datatypes)
7. hierarchical relation to other concepts (specialization/generalization)
8. optional: alignments to concepts from other ontologies
9. optional: references to external non-ontological sources (specifications, web pages, etc.)
10. for classes
    a. allowed/required relations and properties (cardinalities)
11. for relations
    a. allowed/required source and target classes
12. for properties/attributes
    a. allowed/required source class and target datatype
    b. in case of quanititative properties: allowed/required units
    c. inc case of property with limited options: enumeration list

(items 1–6: dictionary; items 1–7: taxonomy)

neanex

Scope of ontologies > content ("concepts" or "terms")

Coordinate System[c]

back to ToC or Class ToC

IRI: https://w3id.org/gom#CoordinateSystem

A 3D Coordinate System. One or multiple omg:Geometry or omg:GeometryState nodes can link to an instance of this class using gom:hasCoordinateSystem. If no named Coordinate System is linked explicitly to

```
###  https://w3id.org/gom#CoordinateSystem
:CoordinateSystem rdf:type owl:Class ;
                  rdfs:comment "A 3D Coordinate System. One or multiple omg:Geometry or omg:GeometryState
nodes can link to an instance of this class using gom:hasCoordinateSystem. If no named Coordinate System
is linked explicitly to a geometry description, an unnamed Cartesian Coordinate System is assumed. A
custom Coordinate System can be registered in RDF by linking a gom:CoordinateSystemTransformation
instance to this Coordinate System (gom:fromCoordinateSystem) and a second instance of
gom:CoordinateSystem (gom:toCoordinateSystem)"@en ;
                  rdfs:isDefinedBy : ;
                  rdfs:label "Coordinate System"@en .
```

is in range of

from Coordinate System [op], has coordinate system [op], to Coordinate System [op]

neanex

Scope of ontologies > metadata

1. stable identifier for ontology
2. text title for ontology
3. text description for ontology
4. optional: preferred prefix and namespace
5. creator and publisher
6. date of publishing and last edit
7. stable identifier for ontology *version*
8. text description for ontology *version*
9. links to examples

Scope of ontologies > metadata

GOM: Geometry Metadata Ontology

```
<https://w3id.org/gom> rdf:type owl:Ontology ;
                        <http://purl.org/dc/terms/creator> <https://www.researchgate.net/profile/Anna_Wagner13> ,
                                                           <https://www.researchgate.net/profile/Mathias_Bonduel> ,
                                                           <https://www.researchgate.net/profile/Pieter_Pauwels> ;
                        <http://purl.org/dc/terms/description> """The Geometry Metadata Ontology contains terminology to Coordinate Systems (CS), length units and other metadata (file size, software of origin,
etc.). GOM is designed to be at least compatible with OMG (Ontology for Managing Geometry) and FOG (File Ontology for Geometry formats), and their related graph patterns.

In addition, GOM provides terminology for some experimental data structures to manage (marked as vs:term_status = unstable):
* transformed geometry (e.g. a prototype door geometry that is reused for all doors of this type). This is closely related to the transformation of Coordinate Systems"""@en ;
                        <http://purl.org/dc/terms/issued> "2019-10-15"^^xsd:date ;
                        <http://purl.org/dc/terms/modified> "2020-05-18"^^xsd:date ;
                        <http://purl.org/dc/terms/title> "GOM: Geometry Metadata Ontology"@en ;
                        <http://purl.org/vocab/vann/example> "https://raw.githubusercontent.com/mathib/fog-ontology/master/examples/sample_abox_snk_contractor.ttl" ,
                                                             "https://raw.githubusercontent.com/mathib/fog-ontology/master/examples/sample_abox_snk_inspector.ttl" ,
                                                             "https://raw.githubusercontent.com/mathib/gom-ontology/master/examples/gom-demo.json" ;
                        <http://purl.org/vocab/vann/preferredNamespacePrefix> "gom" ;
                        <http://purl.org/vocab/vann/preferredNamespaceUri> "https://w3id.org/gom#" ;
                        rdfs:comment """- Version 0.0.2: adjusted wrong domain, range and label on gom:hasCoordinateSystem; general typos; BREP and NURBS geometry
- Version 0.0.1: initial version"""@en ;
                        owl:versionInfo "0.0.2" .
```

Format JSON LD | Format RDF/XML | Format N Triples | Format TTL

License:

License https://creativecommons.org/licenses/by/4.0/

Visualization:

Visualize with WebVowl

3. Types of ontologies & examples

Ontologies by (sub)domain of interest

- **sensoring and observations**
  - e.g. SOSA/SSN by W3C, SAREF by ETSI, etc.
- **geometry**
  - linking of geometry
    - e.g. Ontology for Managing Geometry (OMG), GeoSPARQL by OGC
  - geometry metadata
    - e.g. Geometry Metadata Ontology (GOM)
  - geometry descriptions (content)
    - e.g. OntoBREP, OntoSTEP, etc.
- **buildings**
  - e.g. BOT, DogOnt, SAREF4BUILDINGS, etc.
- **provenance and metadata in general**
  - e.g. PROV-O by W3C, DCAT by W3C, DublinCore, etc.
- **heritage**
  - e.g. Getty Art and Architecture Thesaurus (AAT)
- …

Ontologies by their languages and underlying data model

expressed in:

- **XSD, JSON schema, etc.**
  - e.g. CityGML in XSD
- **relational data model**
  - e.g. AWV-OTL in sqlite (Flemish road agency)
- **EXPRESS (ISO 10303-11)**
  - e.g. Industry Foundation Classes (IFC) as a schema
- **Digital Twin Definition Language (DTDL, backed by Microsoft)**
  - e.g. Real Estate Core ontology (REC)
- **Linked Data-based (RDF data model)**
  - e.g. BOT and ifcOWL using the RDF(S)+OWL languages, Google's schema.org using RDF(S), REC in Linked Data form using RDF(S)+OWL+SHACL
- ...

neanex

Ontologies by creators / users

- **standardization bodies**
  - W3C
    - e.g. SOSA/SSN as standardized by W3D, BOT as maintained by W3C LBD CG)
  - CEN
    - e.g. Semantic Modeling and Linking (SML) from EN 17632-1:2022 (CEN 442)
  - OGC
    - e.g. GeoSPARQL standardized by OGC
  - ISO
    - e.g. Information Container for Document Delivery (ICDD) from ISO 21597-1:2020
  - buildingSMART international
    - e.g. Industry Foundation Classes (IFC) as a schema
- **national interest organizations**
  - e.g. IMBOR for public spaces by CROW (The Netherlands)
- **individual companies, municipalities, researchers, etc.**
  - e.g. Waternet's OTL, Amsterdam OTL, Google's schema.org, etc.

Ontologies by structure

- **monolithic ontologies**
  - e.g. Industry Foundation Classes (IFC) as a schema, Google's schema.org
  - large and often complex in structure, potentially more difficult to apply
- **top-level ontologies (core ontologies)**
  - e.g. Semantic Modeling and Linking (SML) from EN 17632-1:2022 (CEN 442)
  - define compact set of high-level concepts and possible relations
- **Object Type Libraries (OTL)**
  - e.g. Waternet's OTL
  - potentially large amount of concepts but structured in a simple way, extending from top-level ontologies, no/limited new types of relations beyond top-level ontologies, specific for one organization (e.g. company, municipality)
- **alignments as a stand-alone ontologies**
  - e.g. alignments between BOT and other ontologies (SAREF4BUILDINGS, DogOnt, BRICK, REC, etc.) maintained by W3C LBD
  - linking concepts of two or more ontologies in a separate ontology

neanex

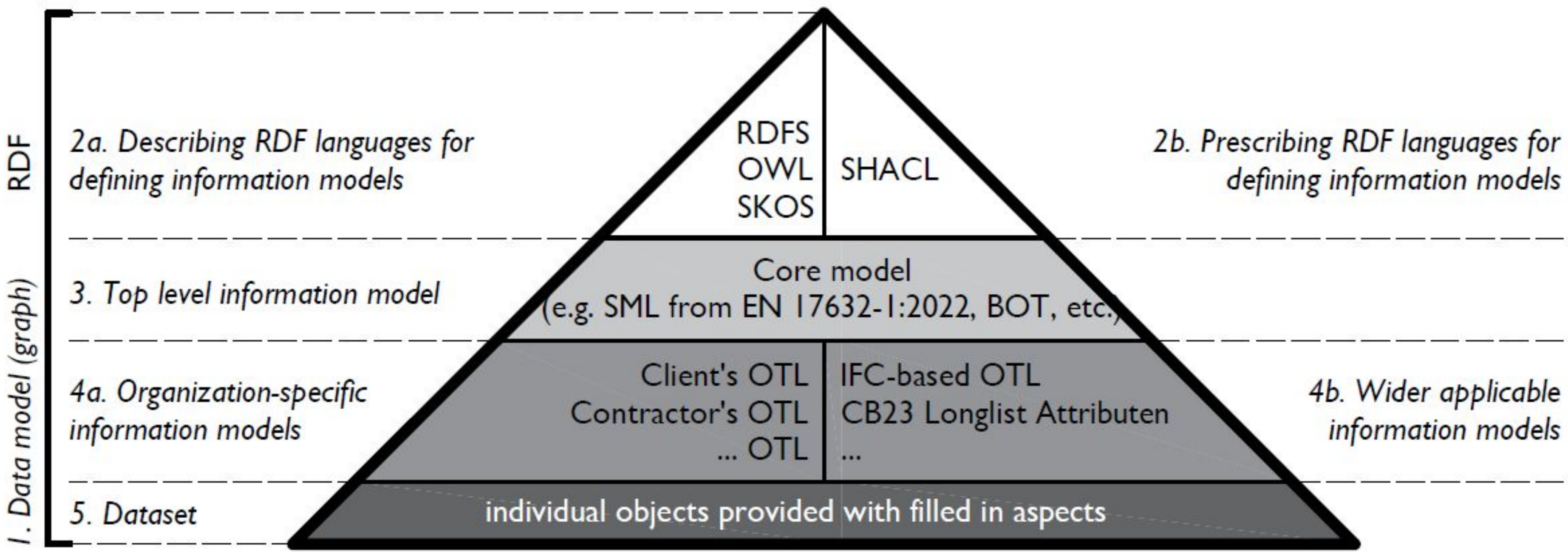
Relevance of *Linked Data* ontologies

Word Wide Web Consortium

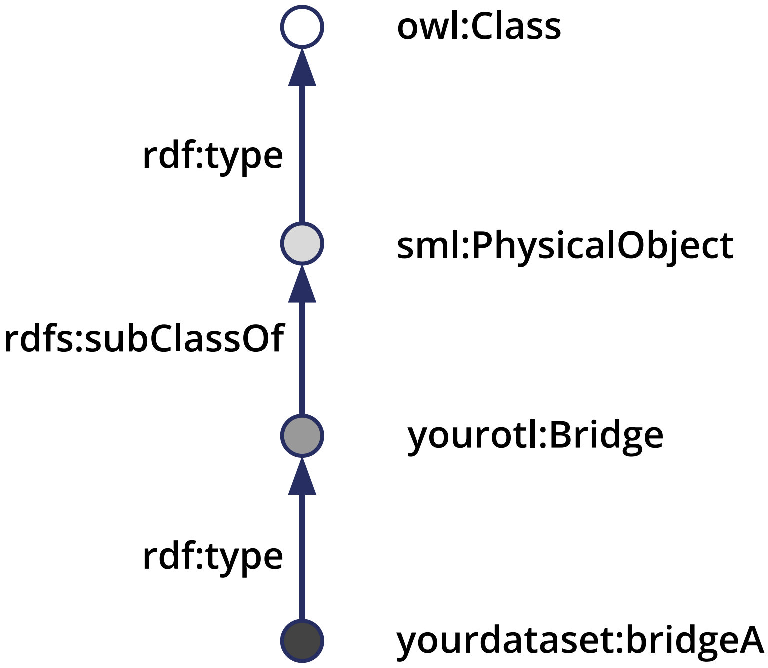
- **standardized languages (RDF(S), OWL, SKOS, SHACL)**
  - unambiguous descriptions which can be shared
  - multitude of available tools for creating, publishing and applying the ontologies, incl. standardized querying, generic reasoning engines, validation tools with standardized outputs, etc.
- **graphs > convenient for linking concepts:**
  - inside the ontology
  - between ontologies (alignments, modularization)
  - always extendible > permissionless innovation
- **graphs > easy to query together with your dataset level (ABox)**
- **web-oriented > convenient for sharing and reusing concepts**
- **note: you don’t really need ontologies to apply Linked Data (RDF) cfr. “schema-less” approaches for NoSQL databases, but there’s an undeniable advantage in using them**

neanex

Linked Data ontologies: bringing it all together



Linked Data ontologies: bringing it all together



neanex

4. Languages for Linked Data ontologies: RDF(S), OWL, SKOS and SHACL

Formal logics

- **Open World Assumption (OWA) <> Closed World Assumption (CWA)**
  - OWA: if a certain statement does not exist in the known dataset, the statement is unknown instead of false
- **No Unique Name Assumption (NUNA) <> Unique Name Assumption (UNA)**
  - NUNA: the assumption that two things with different IDs *might* denote the same thing, unless stated otherwise
- **Terminology/Role box (Tbox/Rbox) <> Assertion box (Abox)**
  - Tbox/Rbox: definition of classes, relations/properties and datatypes > only in ontology
  - Abox: mainly dataset level, but also possible in an ontology (e.g. enumerations for a property)
- **reasoning process**
  - inferring additional statements from asserted statements and ontological axioms with fixed meaning (under OWA and NUNA)
  - different from validating data (requires CWA)

RDF(S) / OWL

[4] p. 30, M. Bonduel, ‘A Framework for a Linked Data-based Heritage BIM’, Ph.D. dissertation, KU Leuven, Ghent, 2021

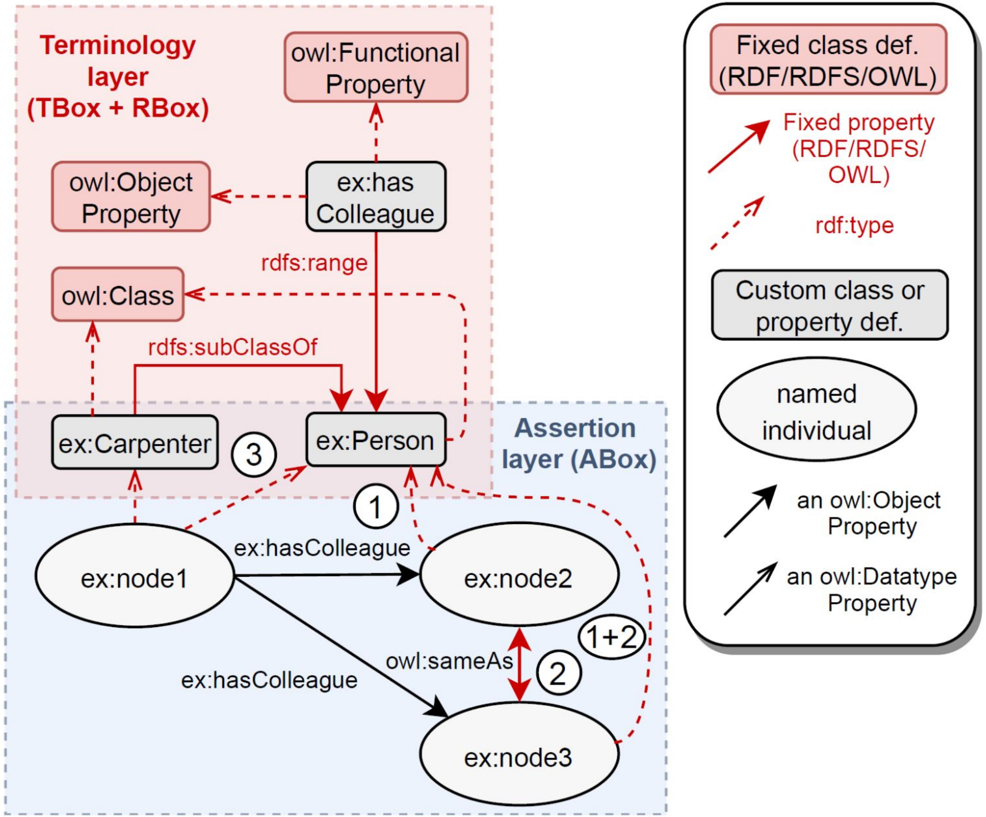


Figure 2.7: RDF graph representing an example OWL 2 ontology including inferred statements resulting from the `rdfs:range` (marked 1), `owl:FunctionalProperty` (marked 2) and `rdfs:subClassOf` (marked 3) axioms.

SKOS

[4] p. 33, M. Bonduel, 'A Framework for a Linked Data-based Heritage BIM', Ph.D. dissertation, KU Leuven, Ghent, 2021

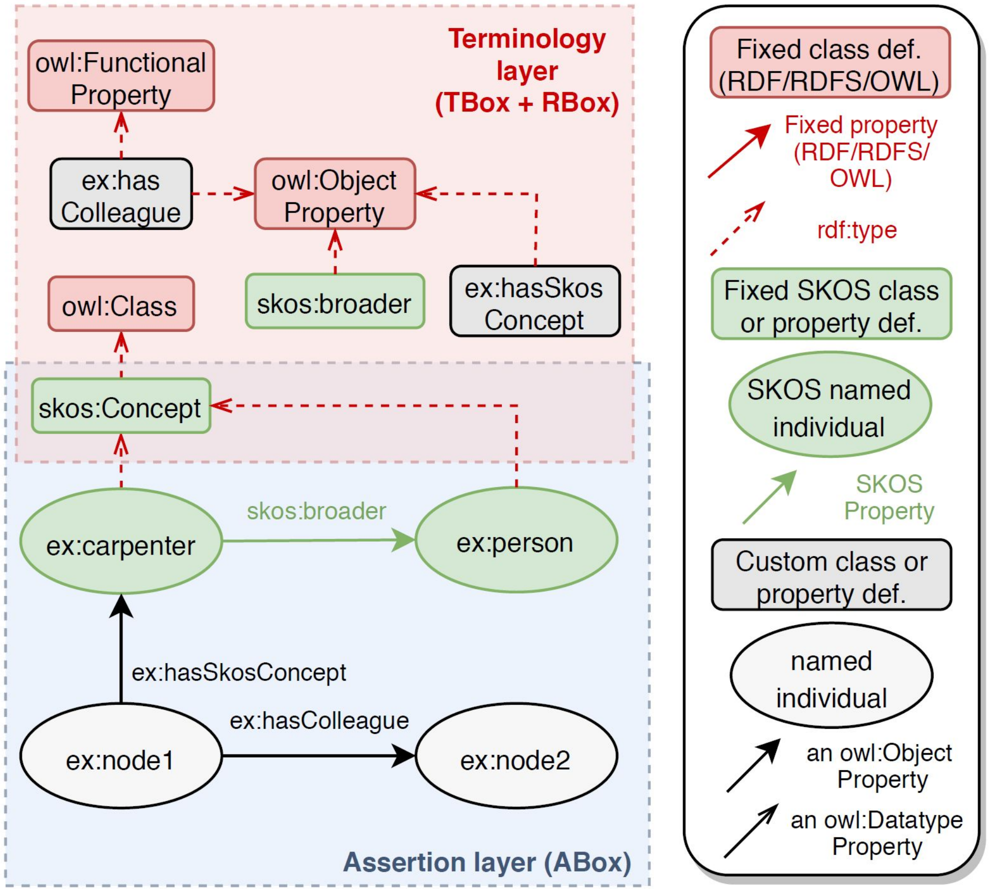
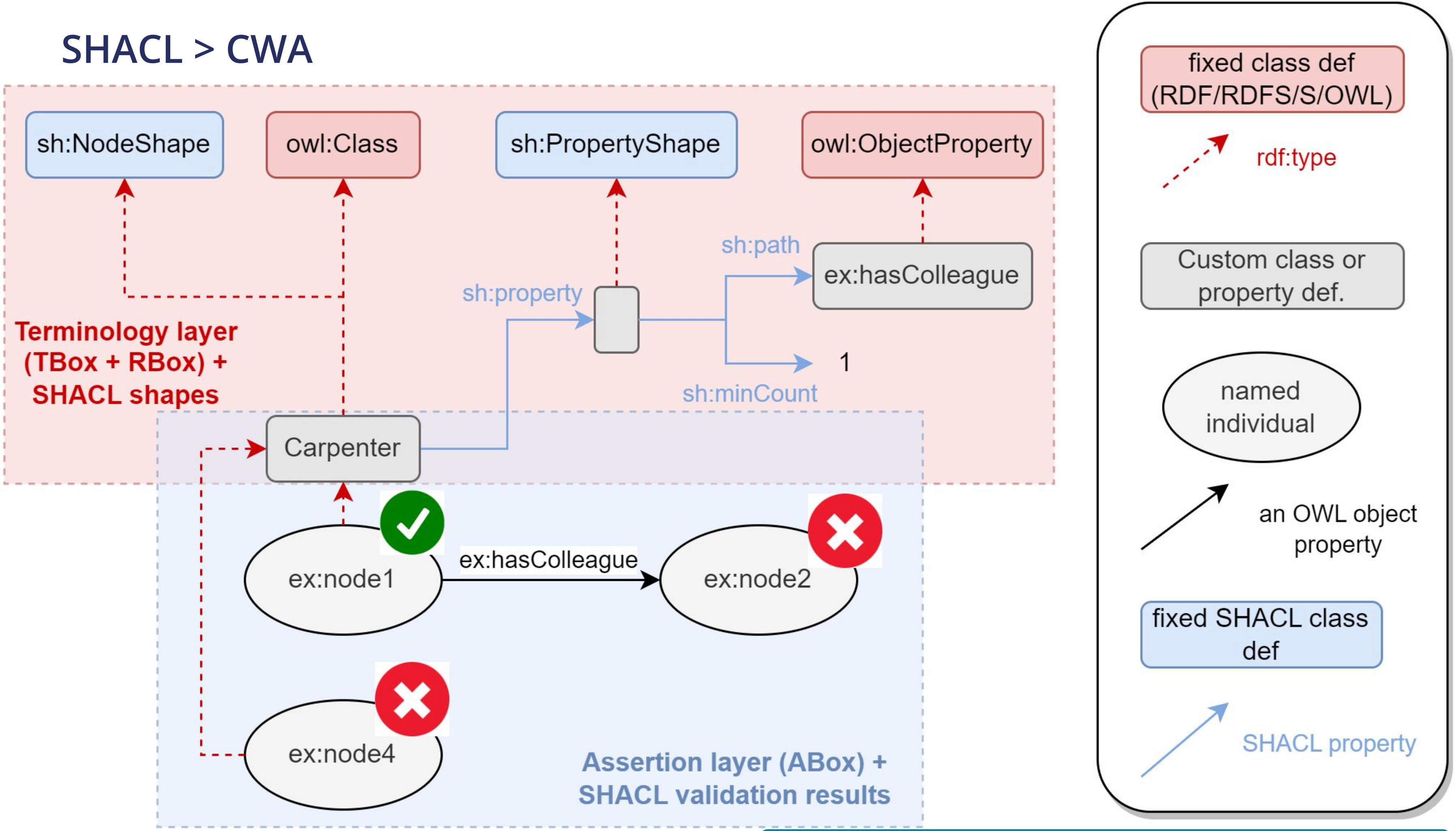


Figure 2.8: RDF graph representing an example SKOS schema, combined with ABox and TBox statements from a custom OWL ontology.

SHACL > CWA


sh:NodeShape
owl:Class
sh:PropertyShape
owl:ObjectProperty
sh:path
ex:hasColleague
sh:property
1
sh:minCount
Terminology layer (TBox + RBox) + SHACL shapes
Carpenter
ex:node1
ex:hasColleague
ex:node2
ex:node4
Assertion layer (ABox) + SHACL validation results
fixed class def (RDF/RDFS/S/OWL)
rdf:type
Custom class or property def.
named individual
an OWL object property
fixed SHACL class def
SHACL property

5. Best practices for (Linked Data) ontology engineering

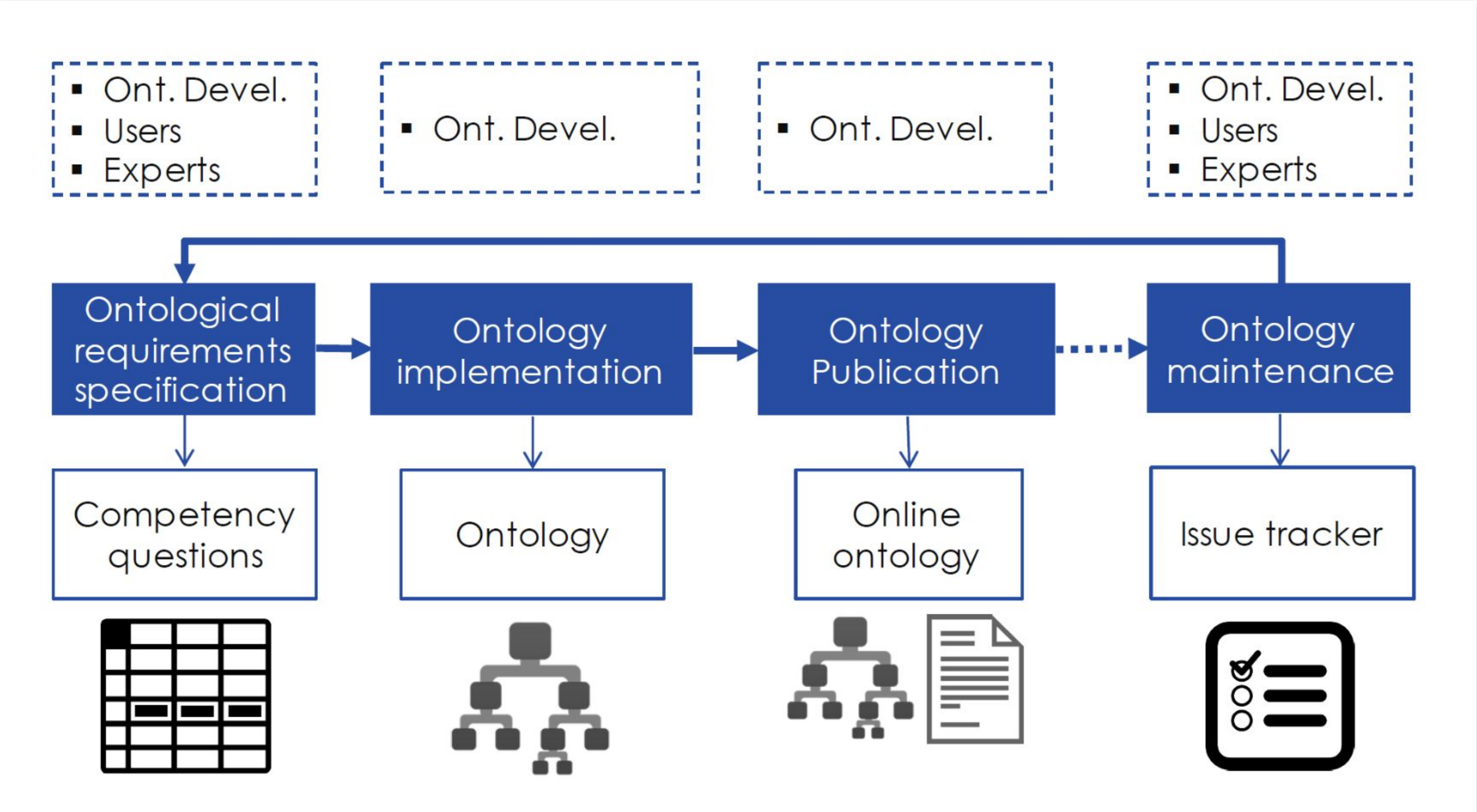
neanex

Structure and content

- **stable identifiers > URIs that don't change (see "Cool URIs don't change")**
  - opaque URIs: avoid human-readable text in URIs
  - ideally dereferenceable URIs
- **don't remove concepts but depreciate whenever possible**
- **modularization can help increase the uptake**
- **if possible: create your ontology with**
  - different use cases in mind
  - multiple stakeholders in mind (certainly in case of top-level ontology)
- **keep your use cases and scope in mind**
  - balance between "correctness" and "completeness" vs "applicability" > reusability + maintainability
  - no exact science > different solutions possible
  - in most cases: simple logics might suffice
- **collecting concepts first > context (relations to others) > definitions**
- **besides application-independent ideally also project independent**

neanex

Ontology development process



[5] slide 54, “Ontology development” presentation by María Poveda Villalón at SSoLDAC 2019 (link)

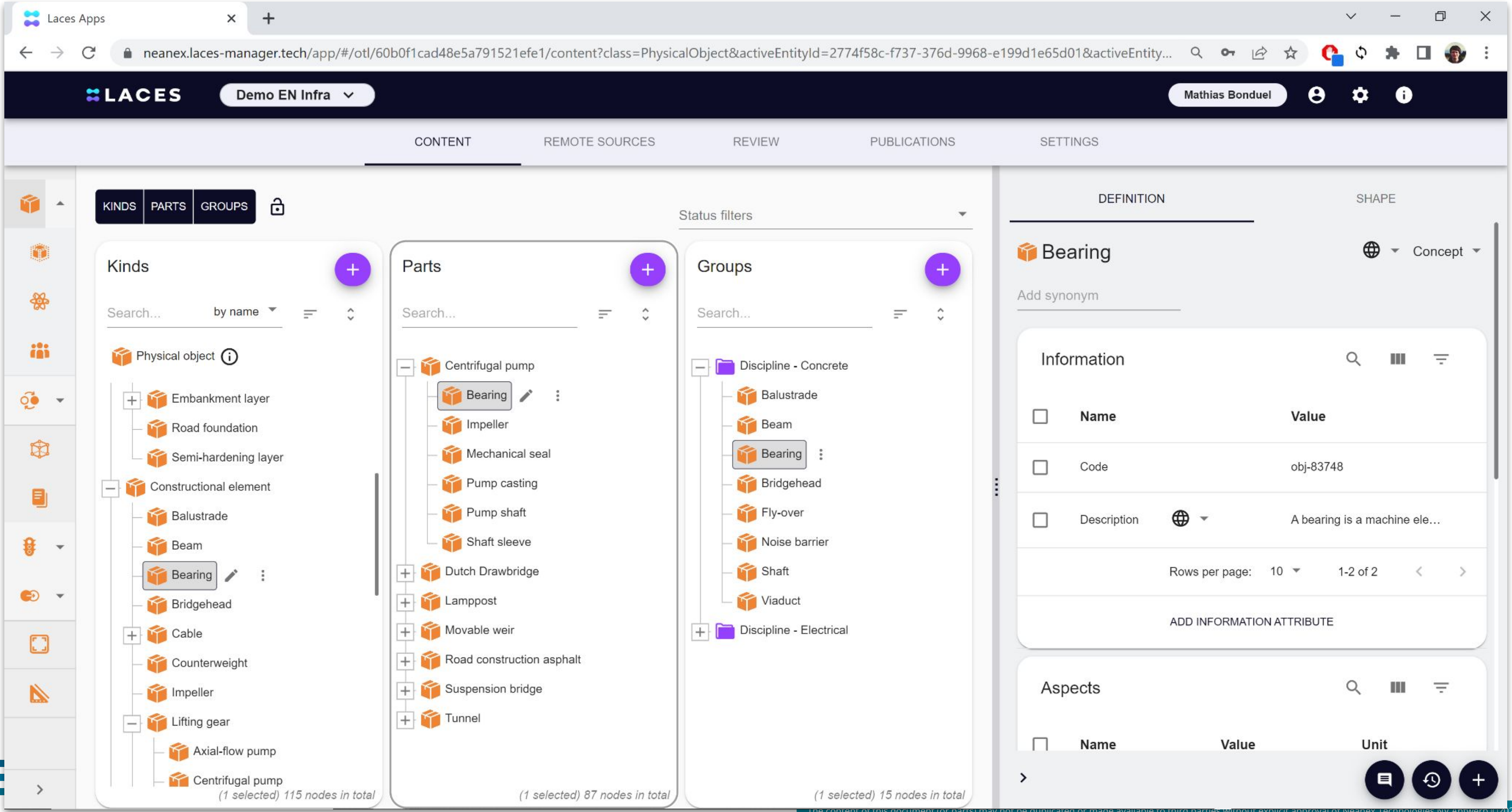
neanex

Tools for editing Linked Data ontologies

- **UI-based**
  - Protégé: open source
  - TopBraid EDG
  - Laces Library Manager (OTL, SPL, PCL and BPL modules) as part of the Laces Suite
    - free trial for academics
    - lowers barrier for domain experts to document and structure their knowledge
- **Code editor**
  - write your RDF directly in the preferred serialization (Turtle, JSON-LD, N-triples, etc.) using your favourite code editor
  - libraries like OWL API
- **UML to OWL:**
  - Chowlk (diagrams.net UML)
  - EA-to-RDF (EnterpriseArchitect UML)

neanex

Laces Library Manager - OTL module



6. Best practices for (Linked Data) ontology publising

- Create a permaId to ensure a **persistent** namespace (also see Cool URIs)
- Register your **prefix** at prefix.cc (ask your coworkers and friends to upvote it)
- Have your RDF files (pref. in multiple serialisations) **available openly and online**
  - HTML human-readable documentation helps users to understand your ideas!
  - Host it on your own server or on GitHub pages / other services
- Provide **example data and queries**
  - Utilise the SPARQL Visualizer to demonstrate your ideas on example data and show your intended workflows!
  - You can host your own SPARQL Visualizer instance or load your JSON file containing the example data

Helpful tools, references and tutorials

- w3id.org
  - Provides permanent identifier
  - HTTP redirects (.htaccess files) to hosted HTML documentations and RDF files
  - Allows simple migration of webspace without breaking links
- prefix.cc
  - Global collection of prefixes and their meanings
- WIDOCO or pyLODE or ...
  - Application to automatically create HTML documentation from RDF files (TTL, JSON-LD, etc.)
  - Java/Python, run from cmd or GUI
- Example documentations / demos
  - BOT (Doc - Demo (hosted individually))
  - OMG (Doc - Demo (loading JSON file))

[6] source: https://github.com/w3c-lbd-cg/lbd/blob/gh-pages/presentations/general/20210323_Group-Discussion.pdf

neanex

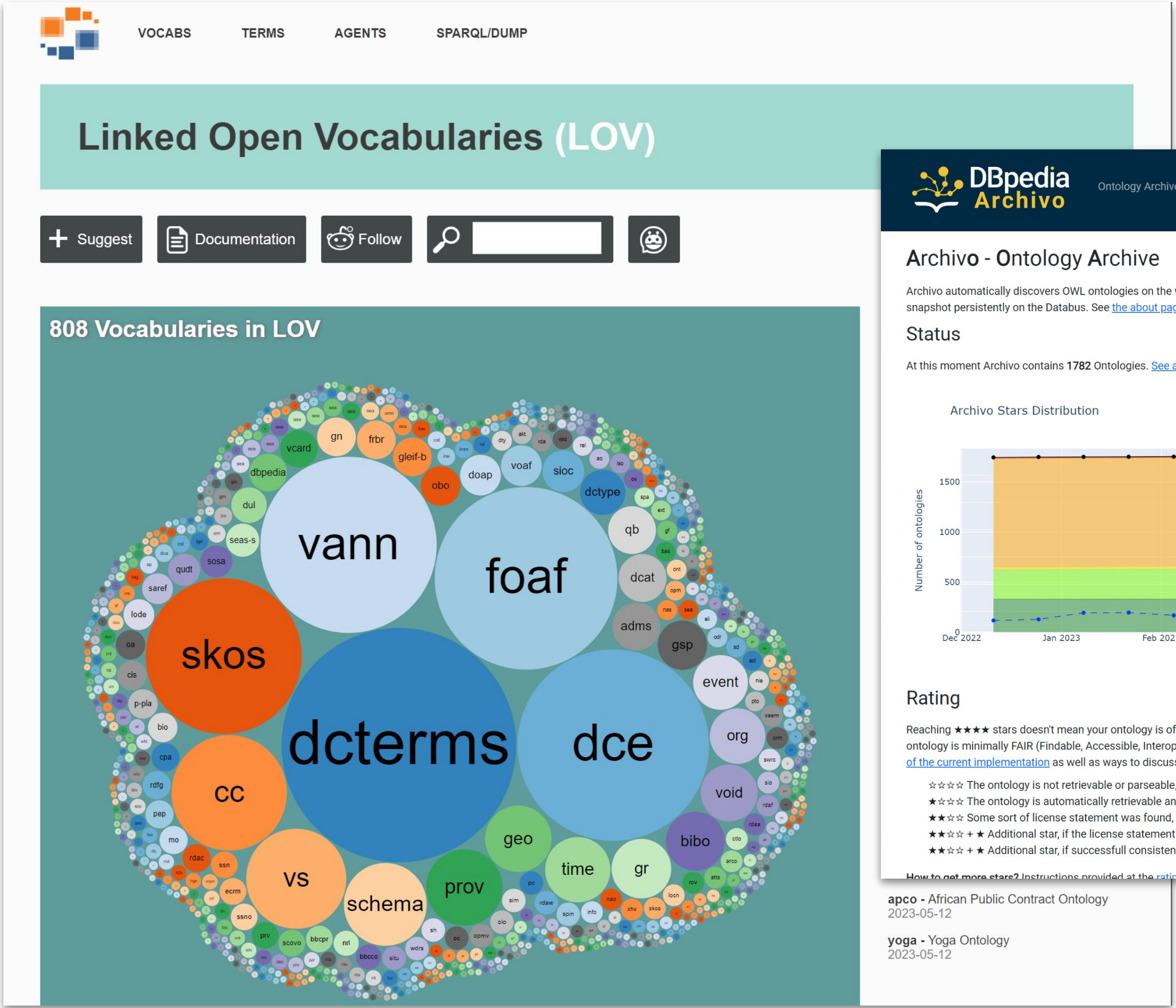
- **Test consistency** of the ontology with a reasoner
- Check your ontology for **pitfalls**
- Evaluate if you provided the **minimal required metadata**
- Attach a **license** to your ontology!
- **Talk about your ontology**
  - Publication in well-known journals
  - Presentations at international conferences
  - W3C calls
- Allow **interaction with users**
  - GitHub issues, forum, contact details, etc.
- **Further reading:**
  - W3C Best Practices for Publishing Linked Data
  - W3C Cool URIs

Helpful tools, references and tutorials

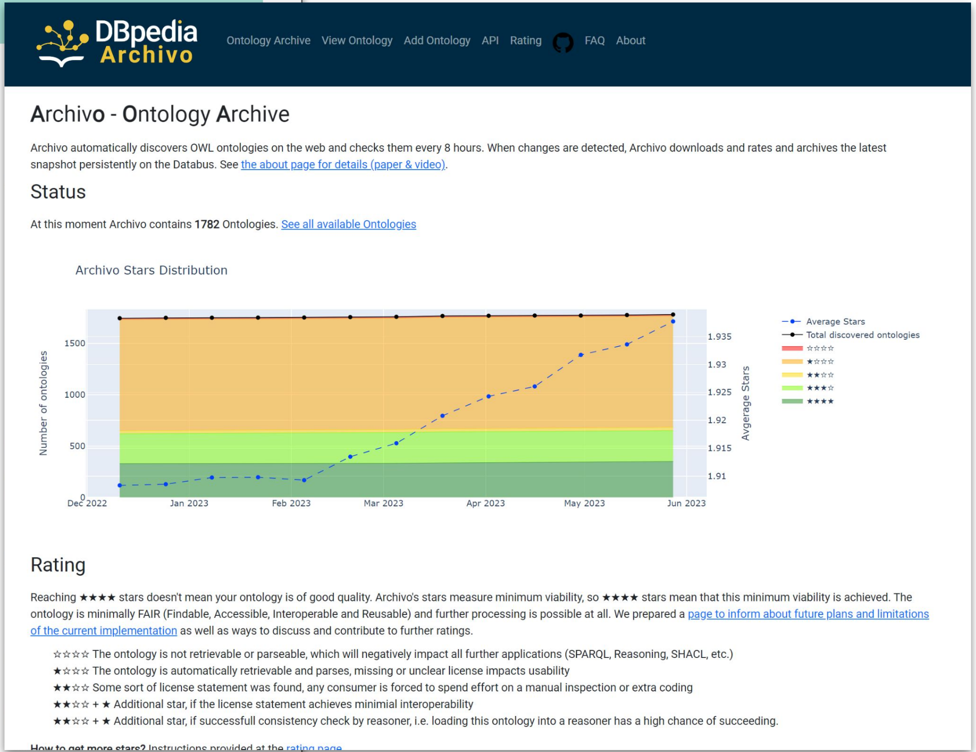
- Evaluation of ontology
  - OntOlogy Pitfall Scanner (OOPS)
  - DBpedia Archivo (if registered/known)
- Discussion on minimal required metadata
  - WIDOCO
  - LOV
- Register your ontology at:
  - Linked Open Vocabulary (LOV)
  - DBpedia Archivo

[6] source: https://github.com/w3c-lbd-cg/lbd/blob/gh-pages/presentations/general/20210323_Group-Discussion.pdf

neanex

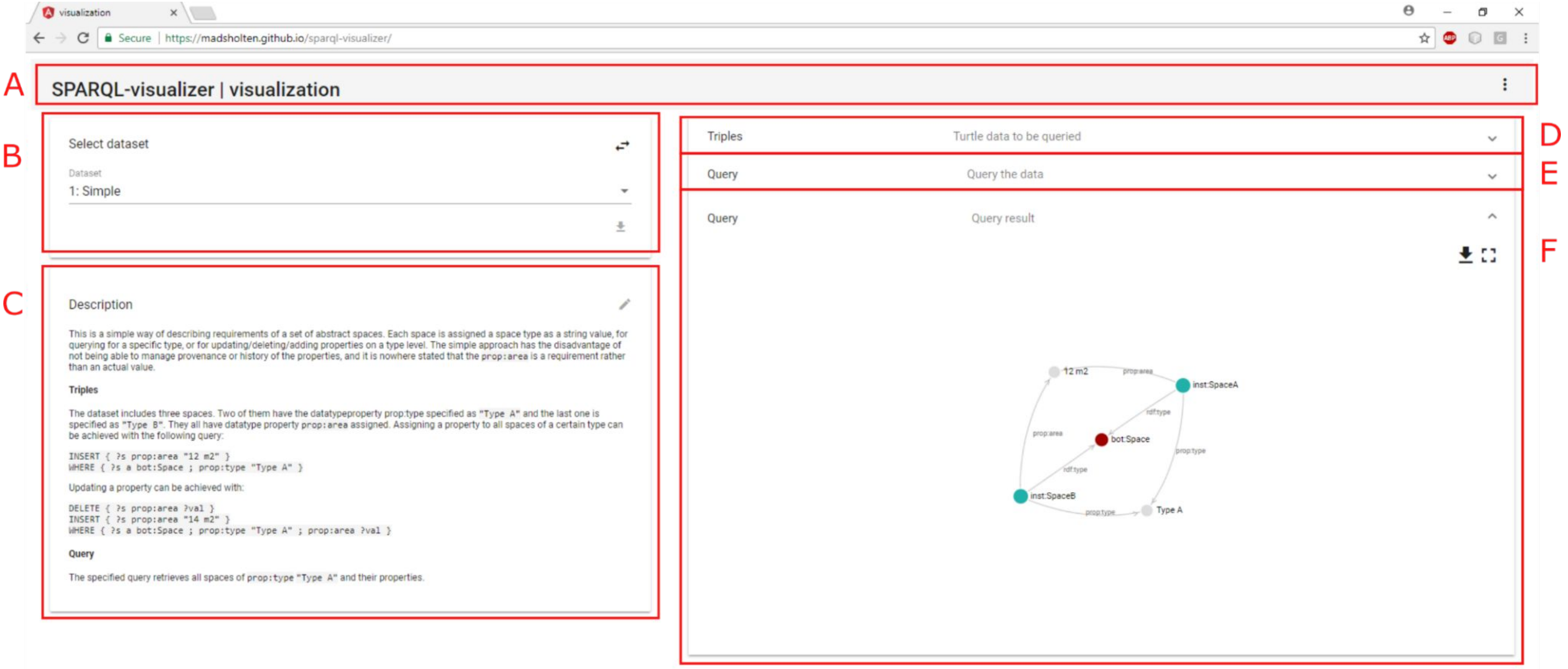


VOCABS
TERMS
AGENTS
SPARQL/DUMP
Linked Open Vocabularies (LOV)
Suggest
Documentation
Follow
808 Vocabularies in LOV
dcterms
foaf
vann
skos
dce
cc
vs
schema
prov
geo
time
gr
bibo
void
org
event
gsp
adms
dcat
qb
dctype
sioc
voaf
doap
obo
gleif-b
frbr
gn
vcard
dbpedia
dul
seas-s
sosa
qudt
saref
lode
oa
cis
p-pla
bio
cpa
rdfg
pep
mo
rdac
ssn
ecrm
ssno
prv
scovo
bbcpr
nrl
bbcco
wdrs
sh
olo
sim
rdaw
spin
info
nao
xhv
skos
locn
apco - African Public Contract Ontology
2023-05-12
yoga - Yoga Ontology
2023-05-12



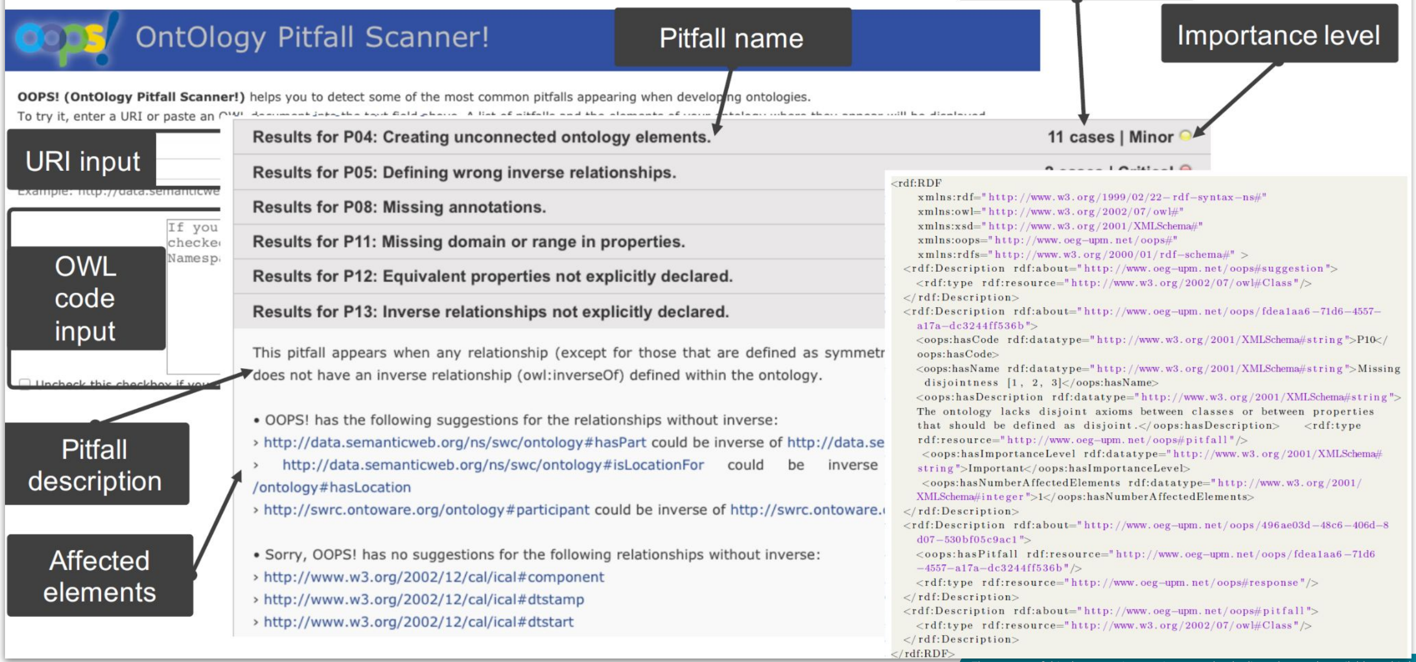
DBpedia Archivo
Ontology Archive
View Ontology
Add Ontology
API
Rating
FAQ
About
Archivo - Ontology Archive
Archivo automatically discovers OWL ontologies on the web and checks them every 8 hours. When changes are detected, Archivo downloads and rates and archives the latest snapshot persistently on the Databus. See the about page for details (paper & video).
Status
At this moment Archivo contains 1782 Ontologies. See all available Ontologies
Archivo Stars Distribution
Number of ontologies
Avgerage Stars
0
500
1000
1500
1.91
1.915
1.92
1.925
1.93
1.935
Dec 2022
Jan 2023
Feb 2023
Mar 2023
Apr 2023
May 2023
Jun 2023
Average Stars
Total discovered ontologies
☆☆☆☆
★☆☆☆
★★☆☆
★★★☆
★★★★
Rating
Reaching ★★★★ stars doesn't mean your ontology is of good quality. Archivo's stars measure minimum viability, so ★★★★ stars mean that this minimum viability is achieved. The ontology is minimally FAIR (Findable, Accessible, Interoperable and Reusable) and further processing is possible at all. We prepared a page to inform about future plans and limitations of the current implementation as well as ways to discuss and contribute to further ratings.
☆☆☆☆ The ontology is not retrievable or parseable, which will negatively impact all further applications (SPARQL, Reasoning, SHACL, etc.)
★☆☆☆ The ontology is automatically retrievable and parses, missing or unclear license impacts usability
★★☆☆ Some sort of license statement was found, any consumer is forced to spend effort on a manual inspection or extra coding
★★☆☆ + ★ Additional star, if the license statement achieves minimial interoperability
★★☆☆ + ★ Additional star, if successfull consistency check by reasoner, i.e. loading this ontology into a reasoner has a high chance of succeeding.

SPARQL visualizer > providing examples for your ontology



Evaluation - OOPS! – OntOlogy Pitfall Scanner!

- Implements the **48** detection methods for **33 pitfalls**
  - Pitfalls selection
  - Selection by dimensions and aspects
- Web user interface http://oops.linkeddata.es/
- Web service http://oops-ws.oeg-upm.net/



[7] slide 47, "Ontology development" presentation by María Poveda Villalón at SSoLDAC 2019 (link)

References - further reading

References - further reading

- W3C specs
  - RDF (primer + official spec)
  - RDFS (official spec)
  - OWL (primer + official specs)
  - SHACL (official spec)
  - SKOS (primer + official spec)
- Validating RDF Data. Ed. by Y. Ding and P. Groth. Vol. 7. Synthesis Lectures on the Semantic Web: Theory and Technology 1. Morgan & Claypool Publishers LLC, 2018. Chap. 1, pp. 1–6. isbn: 9781681731643. doi: 10 . 2200 / S00786ED1V01Y201707WBE016. url: https://book.validatingrdf.com/.
- Handbook on Ontologies. Ed. by S. Staab and R. Studer. 2nd ed. Berlin, Germany: Springer, Berlin, Heidelberg, 2009, pp. 135–152. isbn: 978-3-540-70999-2. doi: 10.1007/978-3-540-92673-3_6.
- A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G. de Melo, C. Gutierrez, J. E. L. Gayo, S. Kirrane, S. Neumaier, A. Polleres, R. Navigli, A.-C. N. Ngomo, S. M. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab, and A. Zimmermann. Knowledge Graphs. 2020. arXiv: 2003.02320 [cs.AI].
- Ontology Design Patterns (ODP) > solutions for recurring ontology modeling challenges

neanex