D4. Principles for a European Road OTL

Document name	INTERLINK-D4-Principles_EuropeanRoadOTL.docx
Version	3.0
Date	13-10-2017
Author (company)	Michel Böhms (TNO), Editor Bart Luiten (TNO) Aonghus O'Keeffe (ROD) Sander Stolk (ST) Lars Wikström (VNS) Matthias Weise (AEC3)

Approval for main versions

	Version	Name	Role	Date
For review PEB MNGT	0.43	Michel Böhms	Editor	21-08-17
Proposed to PEB	1.0	Michel Böhms	Editor	08-09-17
Final	2.0	Michel Böhms	Editor	12-10-17
Authorised	3.0	Bart Luiten	Coordinator	13-10-17
Accepted	4.0	PEB	Client	

Table of Contents

Table of Figures & Tables

Disclaimer

Acknowledgement

This document is a deliverable of the INTERLINK project, which has received funding from the CEDR Transnational Research Programme.

Project Information

Project title	Information management for European roads using linked data		
Acronym - Logo	INTERLINK		
CEDR Topics addressed	CEDR Call 2015: Asset Information using BIM: A. Exploration of procuring asset information B. Exploration of BIM data structures C. Design for common principles for object-type library D. Design and test a basic object-type library and open BIM standards		
Project Coordinator	Bart Luiten - TNO	**E-mail**	bart.luiten@tno.nl
Address	P.O. Box 155 NL-2600 AD Delft The Netherlands	**Tel.**	+31888663122
Partners	Nederlandse Organisatie voor toegepast-natuurwetenschappelijk onderzoek (TNO)	**Country**	NL
	Roughan & O'Donovan Consulting Engineers (ROD)		IE
	Royal HaskoningDHV (RHDHV)		NL
	AEC3		DE
	Trimble Solutions Sandvika AS (TSS)		NO
	interactive instruments (ii)		DE
	Semmtech (ST)		NL
	planen-bauen 4.0 (pb4.0)		DE
Start date **End date**	01/09/2016 31/08/2018	**Duration (in months)**	24
Project Website	http://www.RoadOTL.eu/		

Executive Summary

This INTERLINK deliverable D4 reflects on the earlier findings of the INTERLINK work packages A and B with respect to the identified national roads authority (NRA)'s asset information management needs and issues: their business needs/issues respectively their subsequent data needs/issues. After defining the terms/concepts 'assets' and 'asset management', 'asset information management' is positioned as the key concept for INTERLINK serving as an umbrella concept for the more technology-oriented concepts like Building Information Modelling (BIM), Geo-graphical Information Systems (GIS) and Systems Engineering (SE).

The general conclusion is that NRA's asset data should be 1) better specified: better structured, more complete, more accurate, less ambiguous, up-to-date, consistent, uniformly represented and 2) better communicated and reused. Not only at the NRA but along the asset's whole life-cycle and supply-chain involving multiple perspectives. Better communicated here means 'shared with others by linking' instead of the traditional 'exchanged to others by conversion', keeping data close to their original (and updating) source and avoiding troublesome transformations.

It is observed at the same time that many specifications already exist as partial solutions on International, European, National and company level that solve, or intend to solve part, of the data issues mentioned above. They can be regarded as pieces of the 'data puzzle' that need to be harmonized and aligned before they can fit.

Examples are:

- Open BIM standards like Industry Foundation Classes (IFC) from buildingSmart International (bSI),
- Open GIS standards like:
    - Many fundamental open GIS standards from ISO TC211 [ISO TC211],
    - GML, CityGML and InfraGML1.0 [InfraGML] (the GML encoding of the "LandInfra" conceptual model) from OGC,
    - LandXML2.0 [LandXML], and
    - The European INSPIRE directives [INSPIRE], closely related to both ISO TC211 and OGC.

Both BIM and GIS open data standards tend to focus on geometry aspects which led to the need to develop more meaningful/semantic, object-based specifications, typically referred to as Object Type Libraries (OTLs), in several countries like COINS/CB-NL in the Netherlands, OKSTRA in Germany, the SOSI standard in Norway and CoClass in Sweden. Examples of company-specific OTLs are the RWS-OTL for the Dutch NRA Rijkswaterstaat (RWS) and ANDA for the Swedish NRA Trafikverket (TRV).

All these specifications unfortunately use different underlying data modelling frameworks including different data modelling languages & formats and data modelling styles; and typically overlap in their data structures/meaning. In short, they are difficult to interrelate and reuse in combination.

It was therefore decided in INTERLINK not to start from scratch (i.e. define a European Road Object Type Library fully independently) but first define one coherent and consistent data modelling framework where existing semantic resources can be positioned and reused.

An explanation why this framework is relevant for NRAs and a roadmap how to get there is also included. The framework is defined using the Linked Data/Semantic Web approach and associated modelling and linking guides are included to improve compatibility and the ability to interlink the different LD and non-LD specifications and associated data sets.

In the roadmap, three periods are distinguished:

- Short-term: during the life-time of the INTERLINK project a proof-of-concept of the approach will be developed (WP D2) starting to use some of the selected specifications in three test cases (WP D3). Based on the experience from these cases, a Basic European Road OTL is proposed as a combination of (parts of) used specifications. This will include a suggestion to CEDR how to continue with it organizationally (WP D1).
- Mid-term: after the project, NRAs can start nationally themselves with the Linked Data approach and the suggested Basic European Road OTL, adapting it to their own needs, sharing experience with other NRAs. When commonalities become clear, they are promoted to the European Road OTL in the course of time. The European Road OTL is also a good start point for countries just thinking about developing a national OTL.
- Long-term: NRAs use their influence in relevant standardization bodies (ISO, CEN, W3C, bSI etc.) to promote the approach and specifically the European Road OTL.

1 Introduction

1.1 Document Scope

Based on the outcomes of INTERLINK work packages WPA and WPB reflected in the combined deliverable D2/D3 [INTERLINKpub], this report describes the key principles and guidelines for the development of the European Road Object Type Library (OTL) in work package D.

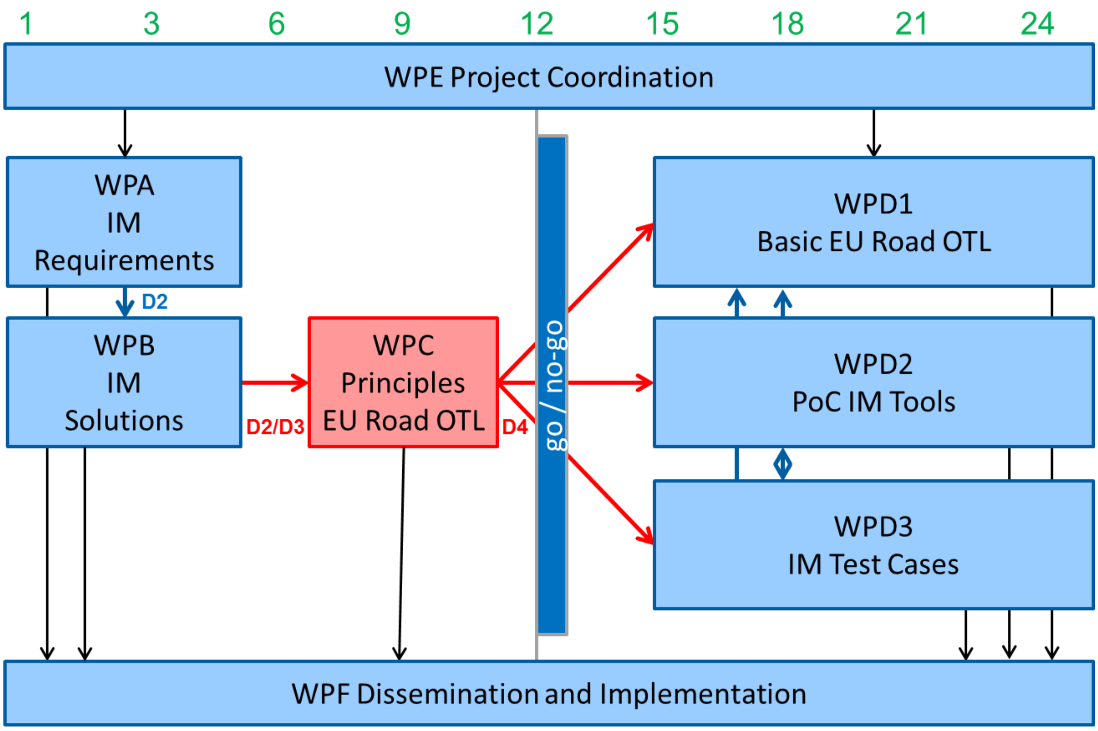


Figure 1: Positioning of Work Package WPC and its deliverable D4

These principles support the conceptual modelling work in work package D1 and the associated software implementation work in work package D2.

The European Road OTL will be a combination of, parts of, existing relevant data structures ('ontologies', 'OTLs', 'schemas', …). Linked together, they form an extendable 'semantic ecosystem' following the W3C Linked Data approach.

Which parts of which data structures become elements in the (initial/basic) European Road OTL is determined after experimentation in the INTERLINK Test Cases (work package D3). Candidates are parts of existing international BIM and GIS standards and common parts of existing national OTLs.

Below, work package C is indicated in the list of work packages with timing and responsibility data.

Table 1: Work Package C Planning Data

WP	Work Package Title	Start Date	End Date	WP Leader
A	IM Requirements	M1	M6	ROD
B	IM Solutions	M1	M6	RHDHV
C	Principles European Road OTL	M7	M12	TNO
E1	Project Management	M1	M12	TNO
F1	Dissemination and Implementation	M1	M12	RHDHV
D1	Basic European Road OTL	M13	M24	TSS
D2	Proof of Concept IM Tools	M13	M24	Semmtech
D3	IM Test Cases	M13	M24	ROD
E2	Project Management	M13	M24	TNO
F2	Dissemination and Implementation	M13	M24	RHDHV

1.2 Target Audience

Although it is relevant to many stakeholders in the area of road infrastructure asset information management, this report is intended in particular for two audience types.

In PART 1 the decision makers at the National Roads Authorities (NRAs) are targeted, giving them a general overview of the business aspects involving asset management and its supporting asset information management. Based on the conclusions from the preceding work packages of this research project, we propose the European Road OTL as solution approach and discuss its essential required features.

In PART 2 the more technical modellers and software implementers are targeted, providing them with specific and detailed guidelines on how to treat the modelling and linking issues involved. In particular, this information is meant as input for Work Packages D1 to D3, where a basic European Road OTL is defined, implemented and tested. This part is however certainly not limited to “internal usage”. It is also meant to provide some common ground for OTL (re)development at national level.

1.3 Reading Guide

PART 1 – General Principles

Chapter 2 reflects (summarizes, generalizes and reformulates) on the outcomes of work packages WPA and WPB, that resulted in the combined deliverable D2/D3 "Information Management for European Road Infrastructure - Investigating the Requirements" [CEDR]. These requirements deal with the existing asset information management practice (WPA/D2) and, more specifically, existing relevant solutions available (WPB/D3).

Chapter 3 introduces some existing initiatives and specifications having direct relevance for INTERLINK.

Chapter 4 introduces in a general way the proposed solution in the form of a European Road OTL Framework, and later a European Road OTL, its nature and special characteristics. A central role is played by the Linked Data/Semantic Web approach by W3C, complemented with modelling and linking guidelines.

Chapter 5 presents an envisaged implementation sequence for the European Road OTL: short-term within the INTERLINK project itself and mid-term/long-term beyond the INTERLINK project. The short-term addresses, based on the initial plans, relevant specifications and data sets for the three INTERLINK test cases (input from WPD3), the more precise steps for the actual development and implementation of the basic European Road OTL (WPD1/WPD2). The mid-term and long-term focus on the reuse of results by NRAs respectively the consolidation of the results in initiatives such as bSI, W3C and ISO/CEN.

PART 2 – Technical Principles

Chapter 6 below defines the principles and, in more technical detail, the guidelines to be used for modelling and linking of the European Road OTL and its associated data sets. This chapter is the main topic of this report and is directly relevant for the conceptual development work in WPD1.

Chapter 7 discusses some software implementation guidelines especially relevant for WPD2.

The final chapters provide lists of abbreviations, glossary of terms and references. Appendices present more detail on some technical matters along with examples of linked data fragments.

PART 1: General Principles

2 Reflection on the Requirements

2.1 Asset Information Management

Any Asset Information Management discussion should start at its context being asset management . I Furthermore, before we start managing something it is good to say a bit more about the thing being managed, in this case an ‘asset’.

In financial accounting, an asset is an economic resource. “Anything tangible or intangible that can be owned or controlled to produce value and that is held by a company to produce positive economic value is an asset. Simply stated, assets represent value of ownership that can be converted into cash” (source: Wikipedia).

INTERLINK deals with the tangible assets of the primary road infrastructure, such as roads, bridges and tunnels. These assets can be seen from four different but related perspectives which some or all are typically dealt with in Object Type Libraries:

- Starting from a **societal**/**business** perspective, in terms of capital investments and people’s/economic value of mobility and interconnection, **realized by:**
- A **functional** perspective, being able to convey traffic of variable volumes quickly and safely between multiple locations, **realized by:**
- A **spatial** perspective, providing 3D corridors via a network of roadways, road segments, lanes and their crossing (bridges, tunnels, …) and junctions, **realized by:**
- A **physical** perspective, realising (the network of) assets and their parts like the physical road objects including asphalt pavement, road restraint systems, traffic control systems and lighting, and the physical realisation of crossings and junctions including bridges, tunnels, roundabouts and exits.

Asset management is the process of managing the whole life-cycle and supply-chain of (a network of) assets and their parts. Therefore, it includes the realisation of new assets and the maintenance, renovation and repurposing (like adding an extra lane to a road increasing capacity) of existing assets. This view is detailed in Figure 2.

In INTERLINK, our main focus is on the process of asset management and especially on the asset information management by National Roads Authorities (NRAs) and their supply chain. An important aspect of this information management is the storage and handling of all the relevant asset data, including specifications, designs, as built records, and asset condition records, all from the aforementioned societal/business, functional, spatial and physical perspectives. Finally, there is data about this data, provenance data, that tells us something about its status, quality, origin and purpose.

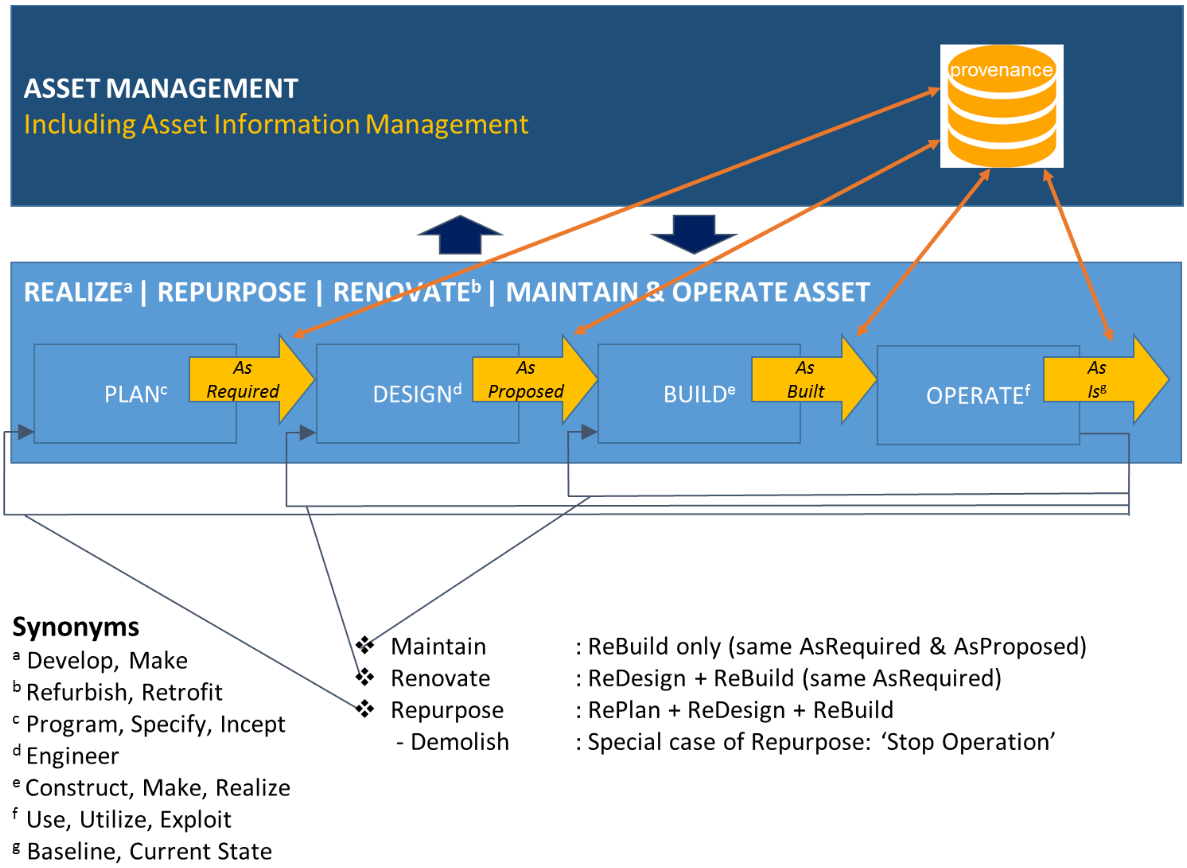


Figure 2: Schematic view on the processes and their relations in the life-cycle of assets

With this process model, the different phases of plan, design and build, are the same for the realization of new assets and the maintenance, renovation and repurposing of existing ones. From a data perspective, we can avail of the following four associated data views:

- AsRequired – what is needed (by clients), allowed by regulations or recommended by experts?
- AsProposed – what is possible, proposed, offered or designed?
- AsBuilt – what is actually constructed or deconstructed?
- AsIs – what is the current state of the asset?

In the plan phase the (governmental) regulations, (client) requirements and (expert) recommendations[1] are specified and input to the design phase, which proposes a solution for the actual build phase where the asset is effectuated (newly realized, repurposed, renovated or 'just' maintained).

Next, a supply-chain dimension is defined relative to a typical decomposition structure for assets, with 'make' (self or subcontract) or 'buy' decisions on each layer (Figure 3).

[1] Both functional requirements and technical boundary conditions

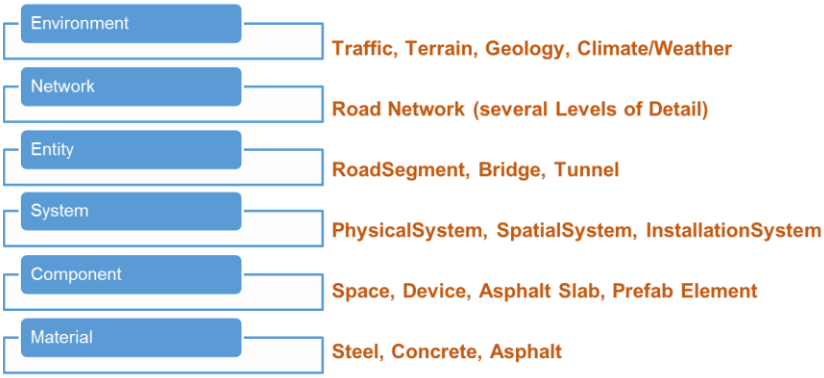


Figure 3: The Supply-Chain/Asset Decomposition dimension

When the life-cycle and the supply-chain dimensions are combined, the following asset management matrix results as context for a Linked Data 'playing field' (Figure 4).

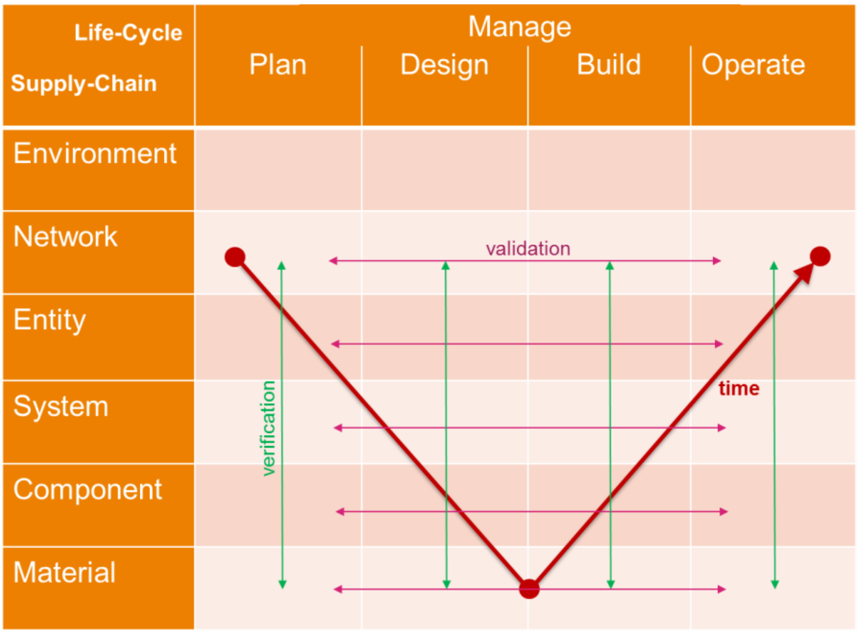


Figure 4: Life-cycle and supply-chain dimensions combined

The result is a typical 'systems engineering[2] image' in the shape of a "V" indicating vertical step-wise 'verification' and horizontal step-wise 'validation'. 'Verification' refers to the check whether the resulting design or product output meets the input requirements, e.g. a specification (source: ISO 9001). 'Validation' refers to the check whether the resulting assets meet the requirements for the intended use.

From this context, current asset information management and the conclusions from the first phase of the INTERLINK project are addressed.

2.2 Current information issues

In the first work packages of the INTERLINK project we analysed the current state of road asset information management in Europe [CEDR]. It was found that, generally speaking:

---

[2] Not that 'system engineering' is a concept touching more topics then addressed here including requirements modelling, design decisions, the earlier mentioned perspectives, variations & versions, level of details etc.

- Information management in the road infrastructure sector is seeing increased investment and standardisation, although much of the progress is in the design and construction stages,
- The use of the ISO 55000 series (for asset management) by public asset owners is becoming more prevalent,
- Various European standards for BIM have been published and the ISO 19650 series (for information management using BIM) is under development, and
- Awareness is growing by NRA senior managers that asset data is an asset in itself.

Asset information management is, in practice, often inadequate for several reasons:

- It is highly software vendor-driven, with systems not compliant to open standards resulting in unconnected data silos and potential vendor lock-in
- Information is often
  - Incomplete (missing data for concepts, attributes or relationships),
  - Inaccurate (not precise enough),
  - Ambiguous (sometimes because of duplicates by exchange),
  - Outdated, or even
  - Inconsistent.
- Even if information is satisfying the aspects above, it is often still not directly (re)usable because it is:
  - Document-based (unstructured, having poor meaning/semantics hence being multi-interpretable),
  - If structured, structured in incompatible ways,
  - Represented in incompatible formats, and/or
  - Stored in incompatible file systems and databases.

Although technological, hardware/software-oriented concepts like Building Information Modelling (BIM), Geographical Information Systems (GIS) and Internet of Things (IoT) are widely addressed, the overarching concept of asset information management is typically not a suitably high priority. In other words, “data” is often not yet recognized as a key asset itself and its management is left to contractors and suppliers. ‘Dealing with the data’ is often confused with the ‘ownership of the data’. Even when data is not dealt with by an NRA, it is still relevant because it is owned by the NRA (which should be able to decide which contractor can deal with the data). Because of the relatively long contract times, principles and technologies agreed at the start of a project are sometimes outdated before the end of a project.

This results in a situation for an NRA where:

- Requirements are non-existent, incomplete or formulated in such a way that they are hard to validate/verify against proposed designs and actually built assets,
- Design data is fragmented and hard to verify and validate since it is bound to/locked into particular software applications,
- The current state of the asset is often not available, incomplete or not having actual data that can be trusted to make predictions and plan maintenance, renovation etc. at the right time and place, and
- Data is often unstructured or even on paper, of bad quality and hard to find and use, especially in combination with other data.

These issues make Asset Information Management inefficient, due to failure costs and time expended on management processes. The also compromise the effectiveness of decision making with respect to the underlying asset delivery and management activities, resulting in higher costs, under-performing assets, or even unsafe situations.

Therefore, there is a need for an asset information management concept ensuring:

- A data-driven, scalable, future-proof approach based on open data standards that are as generic and international as practicable,
- Complete and actual, multi-view information, that is as meaningful (semantically rich) and integrated (linked) as practicable.

Clearly, INTERLINK is not the first initiative to have reached similar conclusions. Many initiatives already provide parts of the solution. In the next chapter we will examine some of these key initiatives. In chapter 6 we will then build upon the results of these initiatives.

3 Relevant Related Initiatives

3.1 Introduction

It was observed that there are many initiatives and associated specifications existing or in development: International/European, country-specific, company-specific or even project-specific, that could be used to deal with relevant infrastructure data like road data [CEDR].

Level	Initiatives/Specifications
International	ISO TC211, TC-59; CEN TC442; bSI IFC, bSDD; OGC InfraGML; NASA/TQ QUDT; W3C SKOS, DC, PROV-O, …
European	INSPIRE Transport Network; V-CON (NL), …
National (Country-specific)	NL CBNL, COINS, BGT/Imgeo; CoClass (SE); OKSTRA (DE); Inframodel (FI); ADD (UK); SOSI (NO); …
Organization-specific	RWS-OTL (NL); ANDA (SE); …
Project-specific	RWS-SAA-OTL (NL); ...

Figure 5: Some of the many existing initiatives/specifications on multiple levels

Some of the key initiatives are discussed below.

3.2 European V-Con Project

This is a finished project of RWS (NL), TNO (NL), Trafic Verket (SE) and CSTB (F). V-Con tendered pre-commercial semantic Asset Information Management software platforms referred to as V-Con Solutions. Software contractors/vendors involved were Arcadis/Semmtech (NL) and TopQuadrant (US/UK). In this project, a modelling and linking guide was defined that was used by the software vendors. Many guidelines in this INTERLINK report initiated from the expertise and experience in V-Con.

3.3 buildingSmart International [bSI] Linked Data Working Group [LDWG]

Also in this group, a modelling and linking guide for linked data is under development lead by TNO. The primary goal of this guide is to support the application of W3C Linked Data (LD) and Semantic Web (SW) concepts and technologies in the AEC/FM industry sector.

The secondary goal of this guide is to improve the compatibility and stimulate the reuse of specifications in the dynamic extension of the static OWL serialization of the Industry Foundation Classes (IFC) specification ('ifcOWL'). 'Extensions' here could be bSI's own bSDD semantics, or additional international, European, national/country-specific, organization-specific and/or even project-specific semantics.

The guidelines in this report are where possible based on the bSI guide always choosing the simplest modelling variants and for the linking just focussing on ontology and data alignment (no complex conversion rule sets).

3.4 W3C Linked Building Data (LBD) [LDB] Community Group (CG)

This is the Building Data on the Web group - a combination of academic and industry partners working together to specifically address the challenge of managing the huge amount of data that is generated across the building life cycle. This group is the generic counterpart of bSI's LDWG in W3C. Its mission, through the use of the linked data / semantic web, is to establish a new Linked Open Data based BIM cloud. The final goal is to enable free and semantically interoperable resources from multiple sources - geometric models, material models, product models, simulation models, weather models, geographic models, energy tariffs, etc. - to be interlinked. They want to optimise the process of integrating existing business solutions and enable the development of new and novel solutions to meet growing demands for energy efficiency well into the 21st century.

This group is considered as the ideal 'liaison point' for discussing the linking between various relevant data sources within the AEC/FM industry sector, such as BIM data, GIS data, sensor data, material data, energy-related data, people data, and so forth. Furthermore, the W3C can provide an appropriate technical background (e.g. Best Practices on the Web, Web standards).

3.5 National initiatives in the infrastructure sector

Many countries have or are developing national specifications for their construction sector or specific parts of it like for geospatial, building or infrastructure. In the Netherlands, there are the COINS, CB-NL and RWS-OTL specifications, in Germany, the OKSTRA specification and in Sweden, the CoClass specification. Most of these initiatives want 'to go Linked Data' and have or are in need of a modelling guide which this report could provide, improve, or harmonize with. Below we give some more info on these national initiatives that will be certainly relevant for INTERLINK.

- **The German OKSTRA/okstraOWL Project**

  OKSTRA was originally defined primarily as a set of EXPRESS subschemas. In 2017, UML was provided as Enterprise Architect (EA) files but also as XMI files. XMI is an OMG standard where a data structure (they call it 'meta-data') is exchanged as OMGs Meta Object facility (MOF) instances. This XMI is input to a converter developed by TU Eindhoven to generate OWL in a similar fashion as ifcOWL (including a specific parser for the many code lists, defined originally as separate SQL statements).

- ***The Dutch COINS/RWS-OTL/CB-NL Projects***

  COINS is a running Dutch project for the exchange of BIM data already applying LD/SW technologies. COINS made some specific choices w.r.t. OWL constructs used and OWA/CWA interpretations that was input to this guide. COINS also covers an own container format for grouping LD and non-LD like IFC SPFF files but also linked PDFs etc. COINS is 'specialized for the Dutch National Road Authority (NRA) Rijkswaterstaat (RWS) via an RWS Reference Framework and an RWS Object Type Library (OTL). It can flexibly handle hierarchies of 'catalogue items' like the Dutch CB-NL library.

- **The Swedish SB CoClass *Project***

  BSAB 96 developed into BSAB 2.0 and will further evolve into an Object Type Library (OTL) referred to as CoClass. BSAB is based on the ISO 12006-2 standard. BSAB is a classification system for buildings, civil engineering works and its constituents and identifies, divides and sorts information (except geometry) for all construction and real estate operations. BSAB is owned by the Swedish Building Centre, a company owned by 32 Swedish construction and FM organizations. CoClass is the new digital classification system (from the 'BSAB 2.0' initiative) for all of the built environment in Sweden. Here exists no formal ontology variant yet in LD/SW technology.

3.6 Generic semantic resources

- **QUDT**
  Many approaches exist worldwide for modelling standard/reference quantities and units. NASA/TQ propose one of the most elaborated ontologies so far: Quantity, Units (of measure), Dimensions and data Types (QUDT) version 2.0. At the time of writing this report (June 2017), the specifications for QUDT2.0 are gradually published. In our recommendations we reuse QUDT2.0 where possible (see for more details paragraph 6.5).
- **Provenance data (meta-data: data about data)**
  Many specifications exists to deal with provenance data. Three main resources are identified:
  - Dublin Core (DC) by the Dublin Core Meta-data Initiative (DCM) [DCMI]
  - Vocabulary of Interlinked Datasets (VOID) in RDFS [VOID]
  - PROV-O [PROVO-O]

PROV-O is of special interest since it is a W3C Recommendation since April 2013.

It starts with three logical main concepts and relationships:

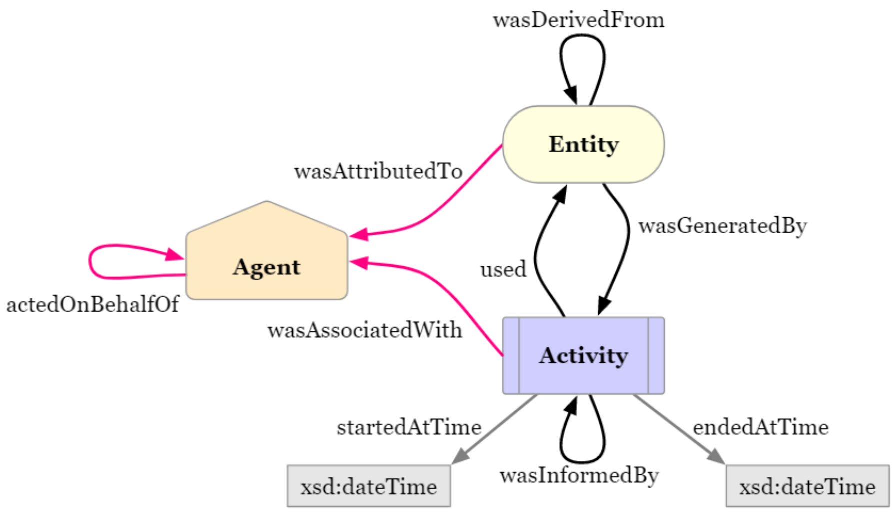


Figure 6: The base model of the PROV-O ontology

Next, it introduces a very useful vocabulary in the form of more specific concepts, attributes and relationships to define various meta-data aspects for individuals. It is unsure at first analysis whether PROV-O can also be used to annotate class-level items. All focus and examples seems on instance level.

4 Proposed Solution Approach

4.1 European Road OTL Framework

In the previous chapter we discussed several initiatives relevant for asset information management. In general there are many potentially relevant specifications, from worlds like BIM, GIS, PLM/SE and even the Internet of Things (IoT) dealing with Big (road) Data from monitoring. All these initiatives use different and typically incompatible ways for modelling civil infrastructures that also make them difficult to map and/or link to each other. They differ in the languages/formats they use, in their modelling style and especially in the meaning they provide expressed in the specifications.

A European Road OTL Framework, enables more comparable and compatible specifications and associated data sets, and uniform mechanisms to link them all together in a kind of ‘semantic ecosystem’ (Figure 7).

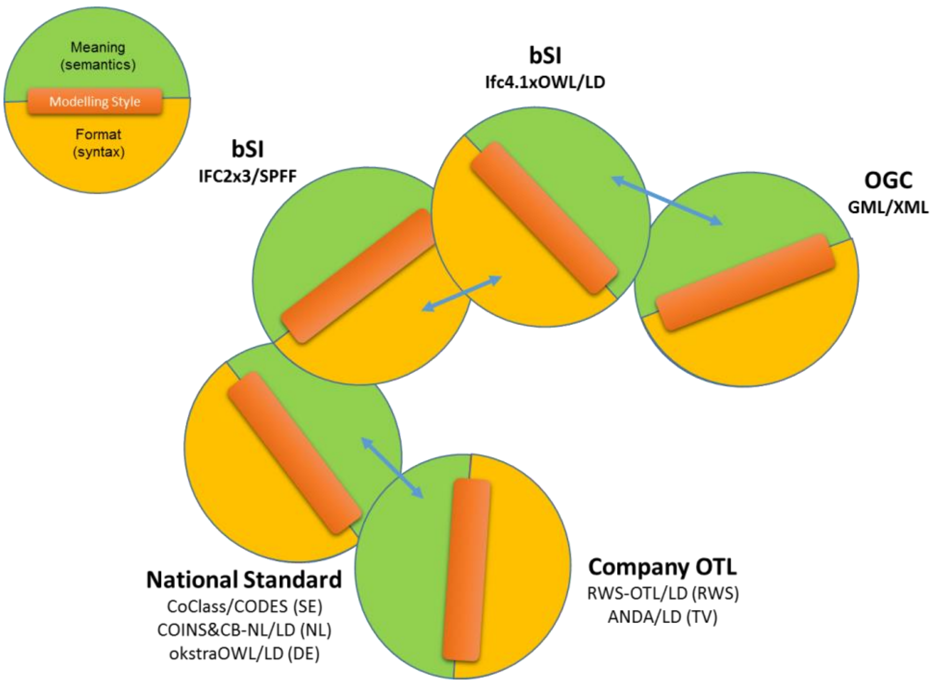


Figure 7: European Road OTL Framework as dynamic ‘Semantic Ecosystem’

(just example, quite incomplete)

With all the relevant specifications linked together we can actually, link all the corresponding data sets together too. In the next figure by Jakob Beetz of TU Eindhoven this situation is nicely visualized. Different sources of bridge data are all linked together: BIM (its spatial/physical structure), GIS (where is it located simply on the map or more realistically in the georeferenced terrain model, including sub-surface layers and geology), tabular cost data, Big (monitoring) Data from sensors, compliance to regulations, etc.

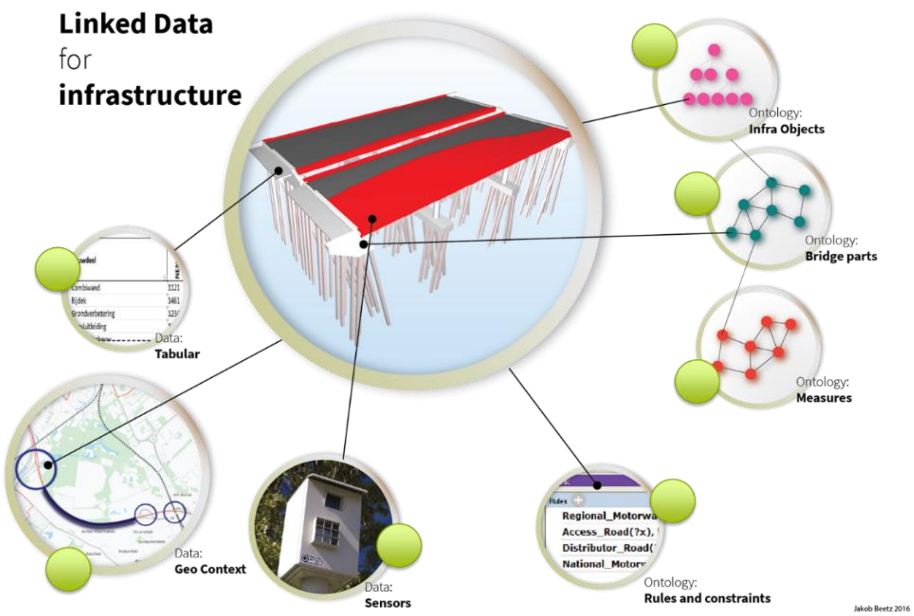


Figure 8: Linking data of different types together (BIM, GIS, PLM/SE, IoT)

To make this possible we also have to support a variety of data mechanisms (translation, referencing, restyling, conversion, linking, etc.) which will be identified and explained in the more technical Part 2 of this report.

4.2 Hybrid Approach

INTERLINK recognizes that each format, style and meaning is potentially relevant for dealing with road data. Hence, INTERLINK proposes a so-called Hybrid Approach, related to hybrid formats (‘multiple languages’), hybrid styles (‘multiple ways of modelling’ in such a language) and hybrid specifications (‘multiple views’). However, to be able to bring all this hybrid-ness together we made a very specific choice for the gluing technology being W3C Linked Data (LD)/Semantic Web (SW).

With this technology we can:

- Model new specifications in LD/SW (according to a preferred modelling guide/style) and model (‘instantiate’) data accordingly,
- Translate existing non-LD/SW specifications towards LD/SW and translate data accordingly (like IFC2x3 to ifc2x3OWL),
- Restyle specifications already in LD/SW towards a preferred modelling guide/style and restyle data accordingly (like RWS-OTL 2.1 to a variant where COINS usage is replaced by CMO usage following the INTERLINK Powerful Modelling Style) ,
- Link different specifications together and link (or convert when really needed, see next paragraph) data accordingly (like expressing that an OKSTRA RoadPart is the equivalent of an RWS-OTL RoadSegment), and
- Refer to non-linked data resources from our linked data (like keeping GML in XSD format and in the data just give references to GML/XML instance data).

4.3 Sharing by Linking & Referencing

Traditionally, many data standards (ISO STEP, bSI IFC, OGC GML) rely on data conversion. Once data is translated to the right format, it is converted from one source meaning to another target meaning so that it can be used, e.g. imported by foreign software that supports that target meaning. Typically this involves a second translation step towards the native software format.

The modern ‘Linked Data’ approach does it differently. As its name already suggests, data is preferably not converted but ‘only’ linked together as a network of data sets where you or your software could browse through (Figure 9). Also the corresponding meanings in the form of ontologies are linked together using the same language in which the ontologies are expressed.

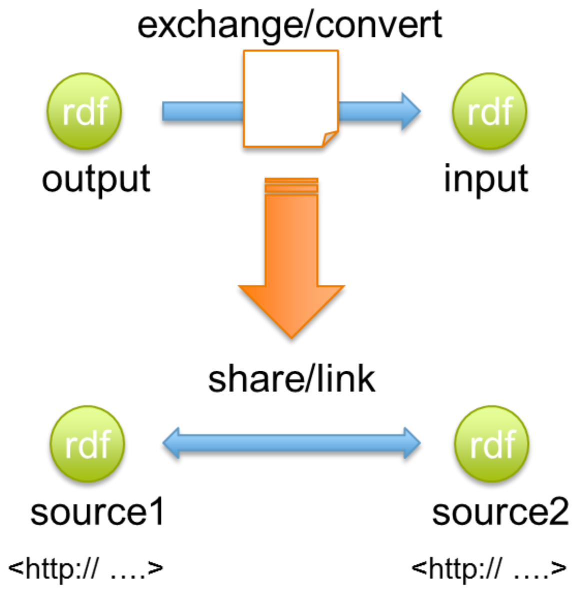


Figure 9: Towards sharing of data

By linking the concepts, attributes and relationships from different ontologies, the linking of data is supported and ‘multi-view/multi-model’ access via the SPARQL query language is greatly improved, making use of the knowledge provided by those links.

Linked Data also provides ways to integrate non-linked data via references via the well-known URIs.

4.4 Focus on Object Data

We distinguish three kinds of data:

1. **Metadata**
   Metadata is all kind of data about (typically object/representation) data: who created it? when was it created? with what precision? is it still current? in what file/document is it stored or referenced to? in what container is it grouped? etc. Meta Data can also cover information about the context where the data is used like life-cycle/supply-chain activities. Another term for this data is ‘provenance’ data and is directly related to the quality of the data and the resulting trust that people have in it.

2. **Object Data**
   Object data is resembling directly the things of interest in the domain at hand. Often these objects are real world physical objects like road assets but they can also be more abstract things like networks, activities, actors, roles, goals or measures.

3. **Representation Data**
   Well-known representations are ‘explicit shape representations’ or ‘geometries’ that somehow spatially represent the Object Data. Examples are Boundary REPresentation (BREPs), Bounding Boxes and Point Clouds. Typically, object data can be represented in multiple ways depending on its intended usage.

This report focuses primarily on object data, covering actual content, the style of modelling this content, the format used and (direct) access aspects. However also, metadata in the form of annotations like labels, descriptions and quantity kinds and units, is also in scope. In the ontologies/OTLs themselves also provenance aspect will be taken into account since they can help to improve the quality of and confidence in the data.

The same is true for representation data aspects typically dealt with in existing international BIM and GIS initiatives/specifications. In ‘BIM for Buildings’ via, for example, existing IFC4 (bounding boxes, BREPs, extruded solids etc.) but also specifically in the future ‘BIM for Infrastructure’ in IFC4x1 and IFC5, by additional representations like 3D Alignments using clothoids etc. currently developed in the bSI Infrastructure Room. This situation is depicted in the next figure.

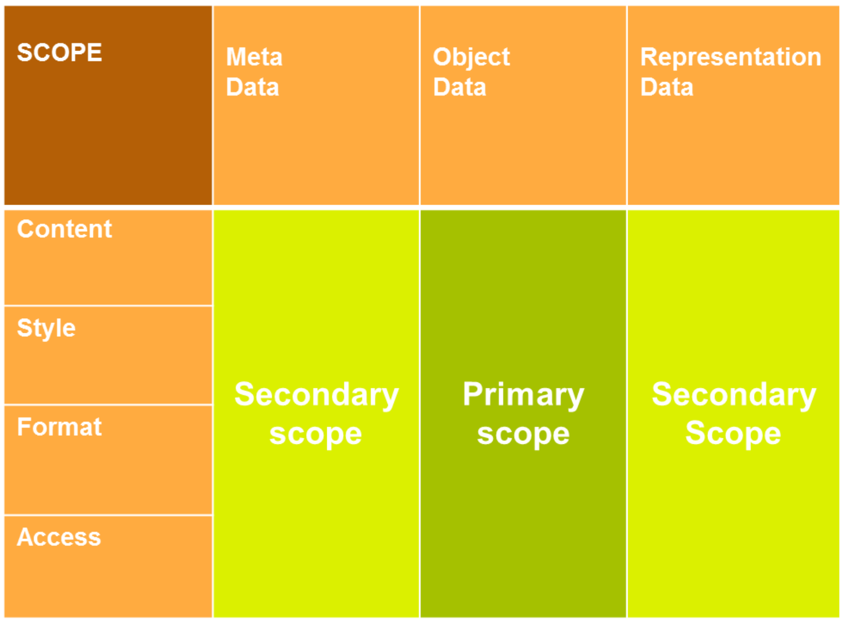


Figure 10: Scope for this report

Next to this object focus, the main interest is in computer-interpretation (i.e. LD instead of PDF) and “open data-formats based on standardized, conceptual models (like GML or IFC) instead of proprietary data-formats (like Autodesk’s DWG)”

4.5 Benefits for Road Asset Information Management

Beyond the general definition and sharing of road data WPA/B also identified some more specific data priorities for NRAs. Below we check them all.

Non-graphical data support

As indicated before, LD/SW is focused on objects, beyond their mere graphical representation/presentations. So, non-graphical, or better, non-shape data, like performance, decomposition structure, material info, current conditions etc. are its key priority.

Information access through GIS

The LD/SW approach makes it easy to assign location data to objects and relate them implicitly to objects at the same location or nearby. It is also possible to explicitly model the relationships with other objects already having a spatial location on the map and/or 3D terrain model. It is therefore quite natural to use a map/terrain model as main data entrance point for all object-related data.

Life-cycle Modelling

Because LD/SW assumes multiple models/views it is the ideal way to model results from different life-cycle phase and link them all together. In maintenance one can always go back to the original requirements and associated design decisions.

Handover support

One of the views is the as-built records that can be qualified and validated against the various requirements, regulations and recommendations.

Governance/Provenance

The same LD/SW approach can be used for data about data i.e. at point in time when data ownership is transferred like the level of development, the contractual status, its revision history etc.

5 Roadmaps for Applying the Principles

5.1 Short-term Roadmap

'Short-term' refers to the use of the defined principles within the INTERLINK project i.e. applying the principles to the specification and software implementation in the three test cases (task WPD3).

On the level of the specifications we summarize the following potential mechanisms/actions:

- Select existing specification & reuse (in LD or non-LD)
- Model ("define") (in LD + preferred modelling style)
- Translate (from non-LD towards LD + preferred modelling style)
- Restyle (within LD)
- Link ("map") (within LD)

Dependent on the actions on the specification level, we have the following potential data set actions relevant for the test cases:

- Select existing data set & reuse (in LD or non-LD)
- Model ("Instantiate") (in LD + preferred modelling style)
- Reference (from LD to non-LD)
- Translate (from non-LD towards LD + preferred modelling style)
    - Or within LD: between equivalent serializations
- Restyle (within LD towards preferred modelling style)
- Convert (within LD)
- Link (within LD)

The various mechanisms/actions on specification level give us a 'semantic ecosystem' consisting of original LD/non-LD-specifications, newly modelled/defined LD-specifications, non-LD specifications translated to LD-variants, restyled versions of existing LD-specifications. All linked together via link sets aligning one specification or data set to another. The idea is that a link set can stay simple in case the involved ontologies adhere to the same modelling style. An example situation is given in the next figure.

It is emphasized that this growing ecosystem is an evolutionary process based on business value, not just academic interest. Initially such eco-system will be specifically supporting a specific test case.

For INTERLINK three Test Cases are planned:

- A Nordic case (Sweden & Norway)
- A Dutch case
- A German case

For each test case the following questions are relevant for this Work Package C (preparing/checking) and Work Package D (making and testing).

- What **processes/activities** are to be supported?
- What International/European, National and/or company/NRA-specific data **specifications** apply?

- What featuring **software** applications are to be connected to the semantic platform for the test cases and to what extent do these applications already support the standards mentioned at bullet point 2?

Furthermore, the following 'preferences' are provided from this work package C:

***Integration** by linking*

Where possible, data linking instead of data transformation ('conversion') should be done when integrating. Experience shows that 'transformation often means trouble' (complex, incomplete, multi-interpretable, …). INTERLINK tries to stick as much as possible to the principle: "store once, use multiple/many times". Conversion processes create duplicates that are to be avoided as much as possible (not to mention the development of the needed complex transformation rule sets). If data really must be read the data by a foreign software application conversion can be performed providing the data is always read-only and edited by the originating software application only.

Hybrid Technology

INTERLINK follows a hybrid approach: 'Linked Data' will coexist for a long time with current Non-'Linked Data'. Therefore, our linking of data will not just be between pure LD data sets but will also include data sets according to other standard approaches, e.g. ISO STEP (like IFC STEP files), W3C XML (like GML files), tabular measurement/monitoring data, and even native software application data like Excel, Revit or PDF files) which we can reference in shallow/global or deep/detailed ways. Although in principle all views are equal, INTERLINK strongly advises to pay attention to the GIS/Geo-spatial view which happens to be so relevant in asset management practice.

Simplicity

It is recommended to limit the amount of 'interconnections/transactions' (i.e. to a maximum of 3) and associated relevant data sets and data structures (schemas, ontologies, …), i.e. one international, one national, one company/NRA-specific as much as possible without losing the essential integration aspect/business benefits you want to show in the end.

Everything in scope should be kept as simple and small as possible (but not too small/simple). Subsets of data structures/data sets should be used where possible, as we are not trying to demonstrating "big data" functionality as part of this project.

Following these guidelines the test cases will result in a kind of framework for OTL adhering to the principles defined in this report with respect to modelling and linking/referencing. An example is depicted in the next figure.

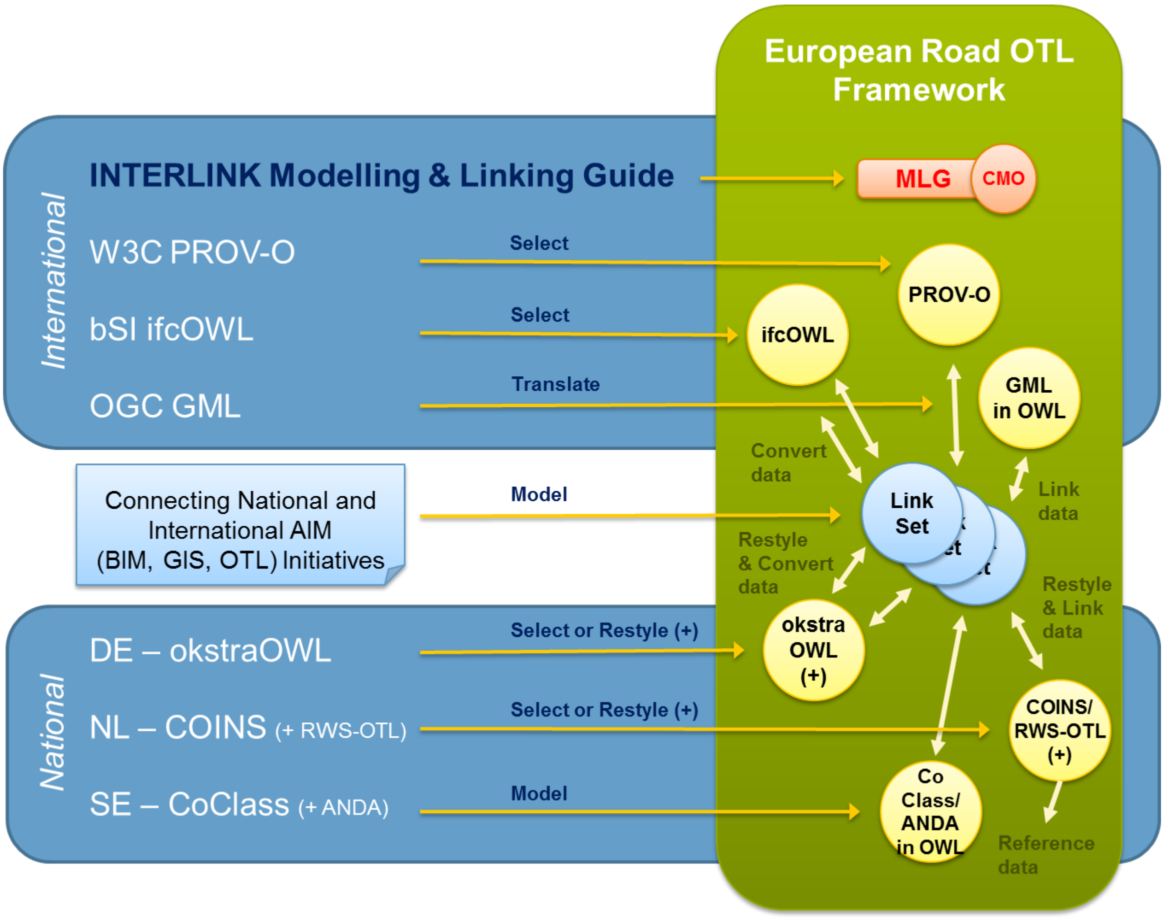


Figure 11: European Road OTL Framework as 'Semantic Ecosystem' example

Next, a first Basic European Road OTL containing/reusing relevant subsets from international standards (BIM, GIS, meta-data, …) and existing country/company OTLs can be defined. This situation is depicted in the next figure.

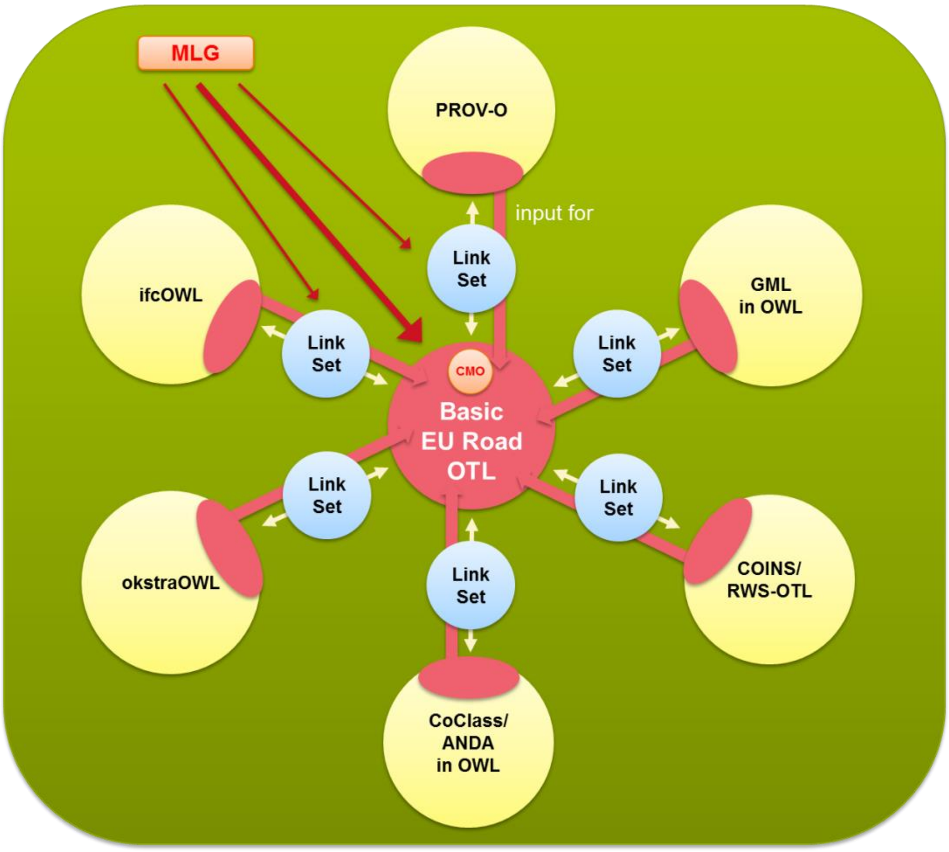


Figure 12: Emergence of a Basic European Road OTL

This Basic European Road OTL will fully conform the INTERLINK modelling and linking guide and its associated CMO upper ontology supporting the recommended modelling styles identified.

The subsets might be restyled for use in the European Road OTL or adapted to have a shared view among existing OTLs. Example content could be:

- Generic semantic resources dealing with basics like time, space, quantities and units
- The road network on different levels of detail
- The clean integration of constructs like bridges and tunnels
- The distinction between functional, spatial, physical objects (the earlier mentioned ‘perspectives’)
- A set of interfacing properties towards IFC and GML 0D, 1D, 2D, 3D geometries
- A practical subset of PROVO-O provenance data classes and properties

5.2 Mid-term Roadmap

After the INTERLINK project, NRAs can start themselves at a national level, with the Linked Data approach and principles, and the suggested Basic European Road OTL, adapting it to their own needs and hereby sharing valuable expertise and experience with other NRAs. When commonalities become clear, they are promoted to the European Road OTL in the course of time. The European Road OTL is also a good start point for countries just thinking about developing a national OTL.

5.3 Long-term Roadmap

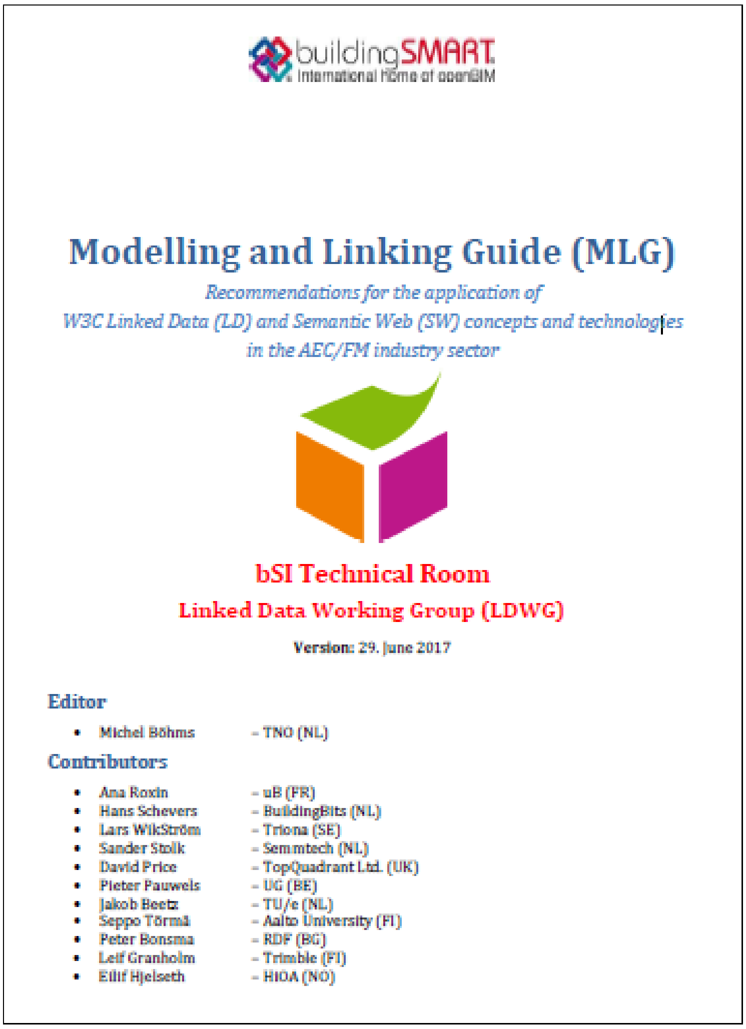
5.3.1 Introduction

Where the short-term concentrates on the use cases and the mid-term on further active experimentation beyond INTERLINK, the long-term focusses on the consolidation of INTERLINK results. The most promising fora for this consolidation where CEDR/NRAs can use their influence to promote both the linked data approach and specifically the European Road OTL are:

- BuildingSmart International (bSI)
- Open Geospatial Consortium (OGC)
- The World Wide Web Consortium (W3C)
- ISO/CEN formal standardization bodies
- CEDR itself

5.3.2 buildingSmart International (bSI)

The INTERLINK modelling and linking approach is currently already promoted in the Linked Data Working Group (LDWG) of the Technical Room, and in the Infrastructure Room of buildingSmart International (bSI). We defined a similar document like D4 but now in the context of bSI where we have to deal with an existing schema (IFC), primarily ISO STEP technology (EXPRESS, SPFF) but also steps towards linked data with ifcOWL variants.


buildingSMART International Home of openBIM

Modelling and Linking Guide (MLG)

Recommendations for the application of
W3C Linked Data (LD) and Semantic Web (SW) concepts and technologies
in the AEC/FM industry sector

bSI Technical Room
Linked Data Working Group (LDWG)

Version: 29. June 2017

Editor

- Michel Böhms - TNO (NL)

Contributors

- Ana Roxin - uB (FR)
- Hans Schevers - BuildingBits (NL)
- Lars Wikström - Triona (SE)
- Sander Stolk - Semmtech (NL)
- David Price - TopQuadrant Ltd. (UK)
- Pieter Pauwels - UG (BE)
- Jakob Beetz - TU/e (NL)
- Seppo Törmä - Aalto University (FI)
- Peter Bonsma - RDF (BG)
- Leif Granholm - Trimble (FI)
- Eilif Hjelseth - HIOA (NO)

The INTERLINK approach has been already promoted as an IFC semantic extension mechanism in the scope of the buildingSmart/BIM Data Dictionary (bSDD) but especially for linking to semantic resources beyond bSI in areas like GIS, SE/PLM and IoT. For the mid-term we will continue to promote our approach (final draft October 2017, to be presented at London Summit end of October 2017.

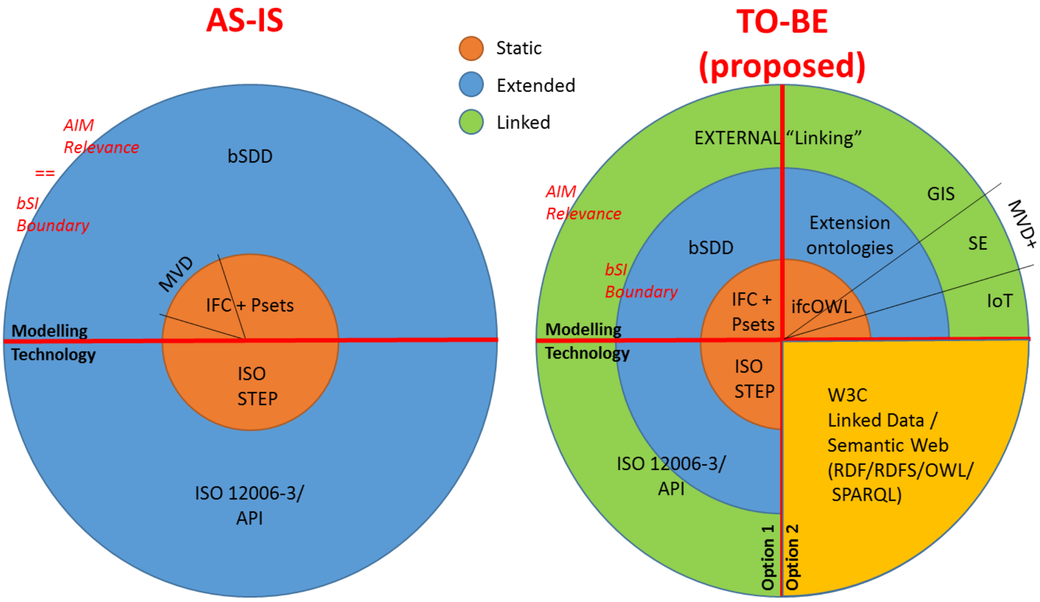


Figure 13: Proposed application of Linked Data approach in bSI

5.3.3 Open Geospatial Consortium (OGC)

The OGC is to (open) GIS, what bSI is for (open) BIM. The OGC produced many of the well-known GIS standard like GML, CityGML, and access standard like WFS (for vector-based data) and WMS (for raster-based data). OGC is currently working on open standards towards the modelling of civil infra: InfraGML. Although more geared towards geometry and not (yet) in linked data technology, it could well find complementary meaning in our European Road OTL.

5.3.4 W3C Linked Building Data (LBD)

In parallel the INTERLINK approach has been discussed in the Linked Building Data (LBD) Community Group (CG) in the World Wide Web Consortium (W3C). It is proposed to target this group in the long-term also especially because of its broad scope (whole life-cycle/supply-chain) and independence of IFC currently being the dominant specification in open BIM standardization.

5.3.5 ISO TC211

This ISO group is a candidate for European Road OTL maintenance since it has been the origin of many fundamental open GIS standards in close cooperation with OGC like:

- ISO 19104 – Terminology, includes «cross domain vocabulary».
- ISO 19107 – Spatial schema. Geometry. From points to clotoids and solids.
- ISO 19111 – Referencing by coordinates. Coordinate reference systems
- ISO 19136 – Geography Markup Language (GML)
- ISO 19148 – Linear referencing. Reference lines

5.3.6 ISO TC59 / CEN TC442

The exact working group is: ISO/TC59/SC13 NP 21597 - Organization of information about construction works -- Information container for data drop (ICDD); Part 1 (Container) and Part 2 (dynamic semantics). The European counterpart for this group is CEN/TC442. This ISO group addresses similar modelling topics as INTERLINK.

This work aims to produce two standard parts, part 1 – Container and part 2 – dynamic semantics. The work focuses on the exchange of data between parties, e.g. contractors and infrastructure owners, in the lifecycle of infrastructure assets and recognizes similar challenges as does the INTERLINK project. Part 1 focuses on a uniform approach for creating data deliveries consisting of data of different representations and formats, such as IFC models, GML documents, drawings, PDF documents, spreadsheets etc., and enables linking, using LD/SW technologies, of different data elements together. This enables use cases such as linking a BOM (Bill Of Material) structure in a spreadsheet to an instance in an IFC model. Part 2 extends the capabilities of part 1 by adding the ability to enrich the data further according to Linked Open Data principles. This functionality is added to allow the extension of a container with user defined data and relations between data from open standards and national or organization-specific standards. This work may have an influence on the work with a European Road OTL and a European Road OTL should be highly relevant for use in an information container.

In the next Part 2 of this report we will explain more technically about data modelling in general and the W3C Linked Data/Semantic Web approach in particular, both conceptually and implementation-wise.

5.3.7 CEDR

Finally CEDR itself could be the place to further maintain the European Road OTL under direct influence of the NRA’s as clients which obviously has its pros and cons.

PART 2: Technical Principles

6 Data Modelling with Linked Data/Semantic Web

6.1 Introduction

In any data (information, knowledge, …) modelling endeavour we have to agree some data modelling methodology. Such methodology has to say something about a data modelling matrix of two dimensions.

First, three Data Levels are distinguished:

1. the data itself, directly reflecting the real-life objects (as required, proposed or realised),
2. the data structure defining the underlying concepts, attributes and relationships for the Data, and
3. the language, defining meta-concepts, meta-attributes and meta-relationships for both the Data and the Data Structures.

Second, three Data Aspects are distinguished (for all three Data Levels):

1. the way the data can be (directly) accessed,
2. the way the data is formatted (for i.e. open import/export aka download/upload), and
3. the actual meaning/content as knowledge and facts.

These two dimensions are fully orthogonal, resulting in the following modelling matrix.

Data Aspects / Data Levels	ACCESS	FORMAT	CONTENT
LANGUAGE	LANGUAGE ACCESS	LANGUAGE FORMAT	LANGUAGE CONTENT
STRUCTURE	STRUCTURE ACCESS	STRUCTURE FORMAT	STRUCTURE CONTENT
DATA	DATA ACCESS	DATA FORMAT	DATA CONTENT

Figure 14: Data Modelling Matrix

<u>Open</u> Data Modelling (like in Open BIM, Open GIS or Open Linked Data)

'Open' means that you agree some standard, non-proprietary choice for some or all of the matrix cells above.

In the next paragraph, we introduce the way W3C Linked Data/Semantic Web populates this matrix.

6.2 W3C Linked Data / Semantic Web

6.2.1 Introduction

'Linked Data' is a W3C concept defined by the World Wide Web Consortium (W3C) positioned on top of the existing World Wide Web (WWW), itself based on the Internet. Data on the WWW is unstructured and configured for human interpretation. Linked Data is structured and therefore 'machine interpretable' by software applications. Another layer of semantics can be defined on top, with ontologies, sometimes referred to as Object Type Libraries (OTLs) containing concepts, properties and restrictions giving powerful meaning to the data utilizing Semantic Web technology. This four-layered 'internet protocol stack is shown in the next figure, demonstrating the evolution of the internet as a communication infrastructure from linked computers, to linked documents, towards Linked Data (LD) and then applying Semantic Web (SW) technology to enhance the LD with knowledge of the things of interest relevant in the road asset domain.

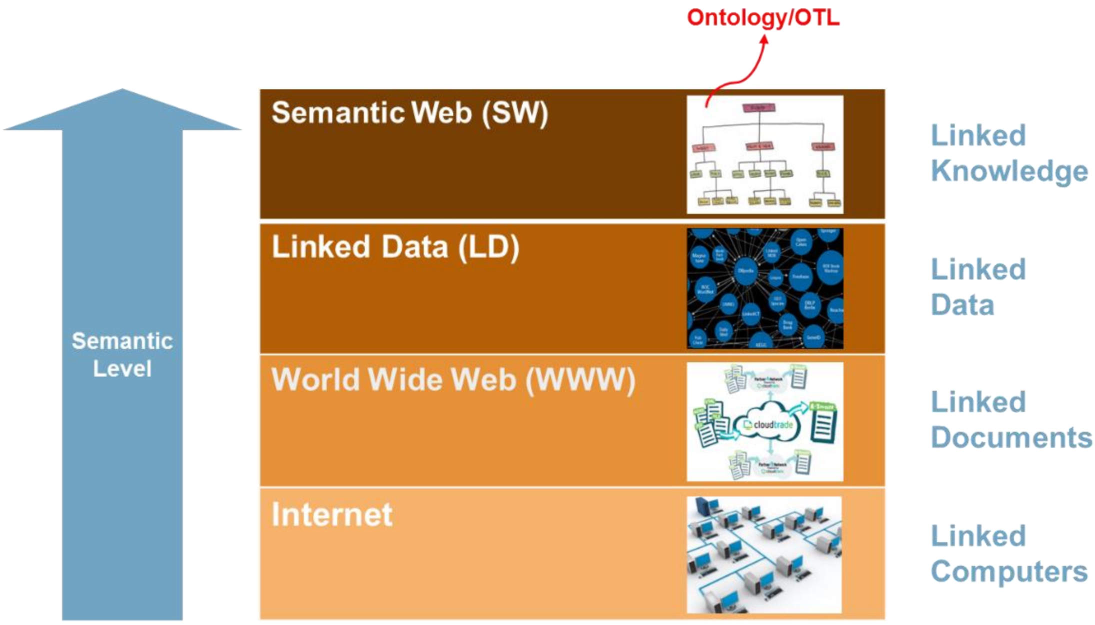


Figure 15: Four-layer protocol stack

Data becomes "Linked Data" (potentially semantically enhanced) when described using the languages provided by the W3C. All data is to be described in the Resource Description Framework (RDF) [RDF], which is a 'data model' with one of the following (equivalent) syntax forms:

- RDF/XML [RDF/XML]
- N-Triples [N-Triples]
- Turtle [Turtle]
- JSON-LD [JSON-LD]

Beyond RDF as data model the W3C provides further semantic vocabularies (in RDF) to define ontologies:

- RDF Schema (RDFS)
- Web Ontology Language (OWL)
- Shape Constraint Language (SHACL) for closed world constraints and rules

Finally, a Query Language (QL) called SPARQL is defined to access the RDF data. Typically, these SPARQL queries can be utilized as (REST-style) Web Services over the WWW [SoH].

On the semantic side, there are much more vocabularies beyond W3C that are already reused extensively like Simple Knowledge Organization System (SKOS) [SKOS], Quantity, Units (of measure), Dimensions and data Types (QUDT) v2.0 [QUDT], etc.

As can be seen in the population of the matrix below, a nice characteristic of the LD/SW approach is that it provides one[3], powerful mechanism for syntax and access (RDF Format resp. SPARQL) for any data level. On the language level for the content we'll find basic RDF data model and vocabulary extended with RDFS, OWL and SHACL vocabularies. With this language stack we define all our ontologies which on their turn define the actual data.

Data Aspects / Data Levels	ACCESS	FORMAT	CONTENT
LANGUAGE	SPARQL	RDF Format • Turtle, • RDF/XML, • N-Triples, or • JSON-LD	OWL SHACL --------- RDFS --------- RDF
STRUCTURE	SPARQL	RDF Format • Turtle, • RDF/XML, • N-Triples, or • JSON-LD	ANY ONTOLOGY
DATA	SPARQL	RDF Format • Turtle, • RDF/XML, • N-Triples, or • JSON-LD	ANY DATA SET

Figure 16: Modelling matrix applied to W3C Linked Data/Semantic Web

6.2.2 Five star data

Another angle for looking at 'Linked Data' is the following figure. Note that this figures does not a priori differentiate between 'data sets' and 'ontologies', both in fact being 'RDF data'.

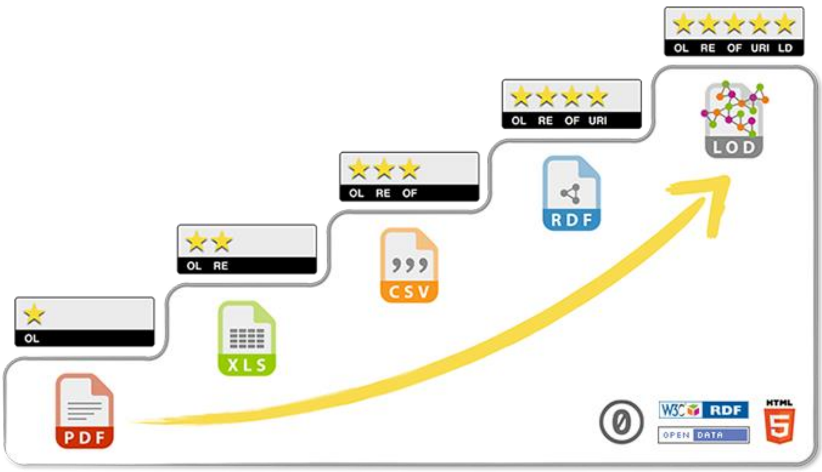


Figure 17: Five-star ambition ([5star])

One star refers to data that is available on the web (potentially only limited by a user access control mechanism). Note this corresponds to the second layer of the previous figure of the protocol stack. The second star is earned when this data is machine-processable in some

[3] Well, to be precise, 'multiple fully equivalent'

format. The third star you get when the actual format is an open standard format. Star number four is assigned when it is not just any open standard but an actual W3C Linked Data standard like RDF (where especially the use of URI for identification is amplified). Note this corresponds to the third layer of the earlier protocol stack. Finally, a fifth star is reached when the RDF data is not provide in an isolated way but closely connected to other data sets giving more context, meaning and usability. Note this differs from the fourth layer of the protocol stack (since for the stars no difference is made between data and ontology and for the layers the 'linkedness' is no factor). So really two views on linked data telling an overlapping but different message.

6.3 Unique Characteristics of LD/SW

6.3.1 Data-driven

Traditionally data modelling starts with a data structure (schema) that is instantiated (by the user) with real-life (or planned real life) data. With LD/SW this is still possible but it can equally go the other way around: you have data (sets of 'individuals'), possibly a lot, Big Data, and you or software start classifying those individuals into classes already defined or also generated from the data in more advanced cases.

6.3.2 Multi-Meta Level

RDF is an abstract language (or 'data model') that can be used to define data instances, data structures and the relationship between them. In other systems one typically needs two separate languages for that. For instance, in ISO 10303 'STEP' we have EXPRESS for data structures and STEP Physical File Format (SPFF) for data instances. This is possible because the top meta-concept of RDF is a 'rdf:Resource' that can be related to any meta-level: it can be an instance (called individual), a class, a meta-class etc. all connected via rdf:type properties. This abstract feature of RDF makes it very generally applicable and a 'one-stop-shop' for modelling. Special schema-level functionality is added by specific RDF vocabularies (RDFS, OWL and SHACL).

6.3.3 Independent Classes & Properties

In traditional modelling approaches including STEP technologies (EXPRESS/SPFF/SDAI) as used for IFC, there is one key/primary modelling (meta-)concept and all others are related. In EXPRESS i.e. we have an Entity as key (meta-)concept having attributes, constraints etc. You would always define an attribute in the context of such an Entity. Not so with the RDF(S)/OWL/SHACL approach where both Class and Property are independent and equally important. Classes and properties only 'meet; when defining constraints or rules.

6.3.4 Open World Assumption (OWA)

Also, very different from other modelling systems is the Open World Assumption (OWA). OWA stems from the principle that 'anybody can say everything about anything'

1. Everything not said in an ontology or data set is "unknown" and not, as with the Closed World Assumption (CWA), "false". In other words: there can always be another source out there that assert it is true or false.
2. It also covers a no Unique name Assumption (no-UNA). There is no (one) Globally Unique IDentifier (GUID) assumption. All identifiers (IDs) for OWL primitives (classes, properties, datatypes, individuals) are never assumed to be globally unique. They can be unique names in themselves but one primitive might have more than one such identifier. OWL provides modelling constructs to explicitly state whether identifiers refer to the same class, property, individual (like owl:equivalentClass,

owl:equivalentProperty and owl:sameAs) or that they are different (like owl:differentFrom).

In the guidelines it is indicated that a Closed World Interpretation (including UNA) is required for the right asset information management with focus on data quality by validation.

6.3.5 Ultimate Normalization

In essence, an RDF model is a "directed labelled graph" made up of triples of the pattern <subject predicate object>. These triples are the 'atoms' of all RDF models. A triple is never edited, only created or deleted. Merging RDF data simply means throwing all triples together and checking for logical consistency. In 'relational terminology' one could say that RDF provides maximal conceptual normalization.

6.3.6 Logic Inference

Because RDF/RDFS/OWL/SHACL reflect some formal logic system (some fragment of First Order Logic (FOL)), data is not only asserted but can also be derived/inferred by generic reasoners that implement these specific FOL fragments.

In summary, with Linked Data/Semantic Web, we can:

- Share (instead of Exchange) data: define once, link & (re)use multiple times
  - Access data from different sources, standards and views (BIM, GIS, PLM/SE, IoT, Big Data, Robotics, Augmented reality, …)
  - For any data type (object data, geometry data, meta data)
  - Non-LD data like documents can also be linked in by reference
  - Maximise the use of the underlying internet/web as common information infrastructure
- Transformations (like syntax translation, restyling or semantic conversion) can be applied where conversion/restyling is best to be avoided where possible
- Interpret data (not just process) in the best way because of sound logic foundations even by generic software components like reasoners and validators
- Configure one's own optimal 'standard' from relevant proven partial solutions (distributed instead of centralised)
- Ease reuse and delivery from/to (up to date) 'external' data resources and legacy systems (both functionality and 'historic' data)
- Address Data & Knowledge Management at the same time and in an integrated way
  - Both knowledge (ontology) and data are treated in the same way including their interconnection

From a business perspective this means:

- Smooth transitions, no loss of earlier IT investments
- National and company-specific or even project-specific modelling needs enabled
- Investments in durable 'future-proof' solutions
- Not limited to a specific market sector and/or country
- Enabling innovations beyond mere 'optimisation' via integration: real innovations in the form of new (big) data-driven software functionalities

To be complete, we also note some potential pitfalls or risks:

- What is the scalability for Big Data sets (i.e. multi-terabyte monitoring data)?
- In hybrid situations, are the data translations good (correct, complete, …) enough?
- Will there be enough software vendors supporting LD/SW-concepts and technology?

- Will data owners embrace the “data-driven” ideas anyway (or keep focussing on software functionalities first)?

6.4 RDF, RDFS & OWL Essentials

6.4.1 RDF Data Model

The most fundamental Linked Data specification is the Resource Description Framework (RDF). In essence, it is a simple but powerful data model or language to define directed, labelled graphs. Such a graph is an unordered set of triples where each triple is of the form as depicted in the figure below.

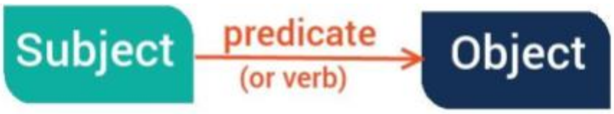


Figure 18: The RDF ‘Atom’: a Triple

The subject and object are typically nouns referring to things in reality. Subjects and objects can be anything, from a concrete things you can point at to a concept in someone’s head. The predicate is typically a verb relating the subject and the object where the semantics is to be understood in the direction from the subject to the object. All three triples components are represented by Uniform Resource Identifiers (URIs) that we already know from the web.

A simple example from the (draft Release Candidate 3) IFC4x1.ttl ontology in the most basic lexical N-Triples syntax form:

```
<http://ifcowl.openbimstandards.org/IFC4x1#IfcTextTransformation>
<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://www.w3.org/2002/07/owl#Class> .
```

To separate different triples in a set of triples, each triple is ended by a dot. This one triple states that there is something like an “IfcTextTransformation” that is an instance of (‘is of type’) “Class”.

This way whole networks/graphs can be described by sets of triples potentially sharing the subject, predicate and/or object.

When we do not exploit this ‘sharing’ the set of triples can be unnecessary large. That is why the RDF format ‘Turtle’ was defined, making use of this sharing and representing the network/graph in a much more efficient way.

For historic reasons RDF is just a bit more than this simple but powerful data model. It has also some very basic predefined vocabulary in the form of predefined predicates most notably:

- rdf:type, and
- rdf:Property.

The complete RDF vocabulary is defined in an OWL ontology[4] at: https://www.w3.org/1999/02/22-rdf-syntax-ns.

The triple concept is here a Statement, the predicate a Property. Such a Statement has a subject, predicate and object rdf:Property. We can forget about this entanglement (of RDF, RDFS and OWL) and just remember for our modelling work that there is an rdf:type property which can be used to classify any subject to some object, typically some real-world object to a class. “rdf:” is just a shorthand or prefix facilitated by the Turtle syntax form, here defined by:

@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

6.4.2 RDF Schema (RDFS), a vocabulary description language for RDF

The RDF data model is a simple way for describing a network (‘directed graph’) interrelated ‘resources’ where the interrelationships are denoted by predicates/properties. Still RDF does not provide any mechanism for declaring such resources and properties. This is the role of RDF Schema (RDFS) [RDFS].
In RDF, the subject and the object within a triple/statement can be anything. In data modelling, however, it is often handy to differentiate between two kinds of ‘anything’:

1. Anything1: Something that has a direct counter-part in reality, an individual thing you can point at if it existed (now) or could point at if it was/could be there in the past or will/can be there in the future, and
2. Anything2: Something that is a concept, potentially grouping multiple anything1’s.

This Table is an individual; A Table is a concept. Car is a concept; MyCar is an Individual. Individuals are typically seen as more dynamic (changing phenomena in the real-world), concepts more stable/invariant in time (inventions of the mind).

To be able to make this differentiation RDFS introduces new vocabulary on top of RDF providing the means for defining terms that will be used in RDF statements and giving them specific meaning. RDFS defines a model for representing simple ontologies; such ontologies can be seen as a graph of RDF statements. RDFS defines not only the properties of a resource (e.g., title, author, subject etc.) but also the types of resources being described (people, paper, web pages, books etc.).

RDFS allows defining the following elements:

- Resources
  - rdfs:Resource – every entity of an RDF model is an instance of this class
  - rdfs:Class
  - rdfs:Literal
  - rdfs:Datatype
- Hierarchies (taxonomies to be more precise)
  - rdfs:subClassOf transitive property defining class hierarchies
  - rdfs:subPropertyOf transitive property defining property hierarchies

[4] This may sound strange since OWL is actually a language defined on top of RDFS, which is itself defined on top of RDF but it is a completely legal form of self-reference.

- Restrictions over properties
  - rdfs:domain for specifying the domain of a property (in terms of classes), in other words to indicate that a particular property applies to a specific class.
  - rdfs:range for specifying the range of a property (in terms of classes or datatypes), in other words to indicate that the values of a particular property are instances of a specific class or are types of specific literals.

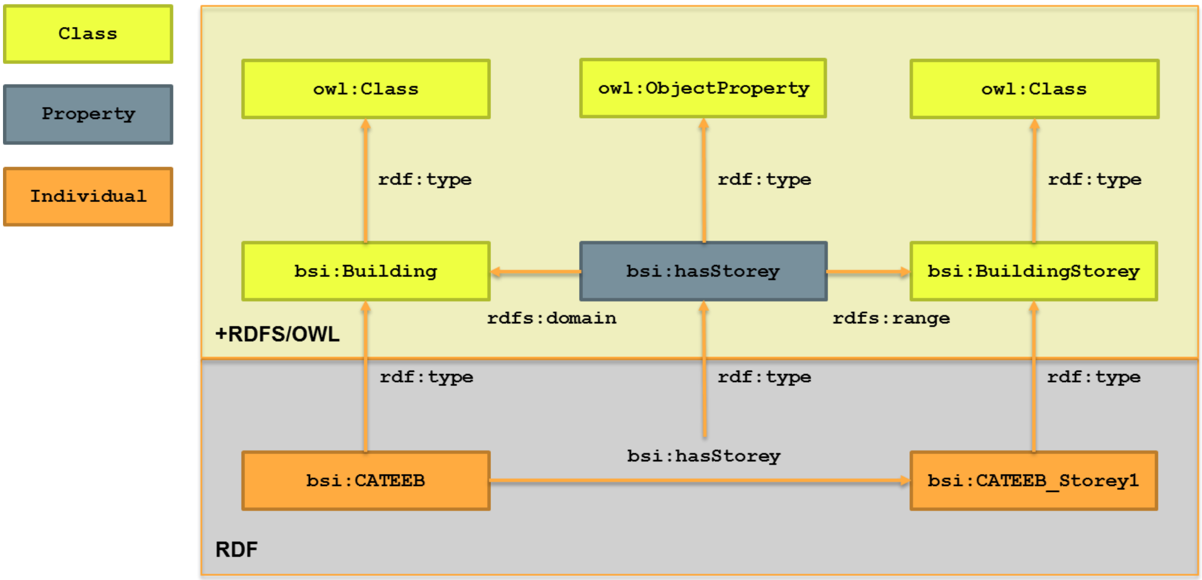


Figure 19: RDFS/OWL vocabulary for RDF

The most important functionality of RDFS/OWL is the ability to make inferences using the RDFS/OWL ‘entailment rules’. This means that one can infer new triples from an OWL ontology/RDF schema and the set of rules. Such triples are often called implicit or inferred triples. Examples are shown in the next figure (red arrows indicate inferred data).

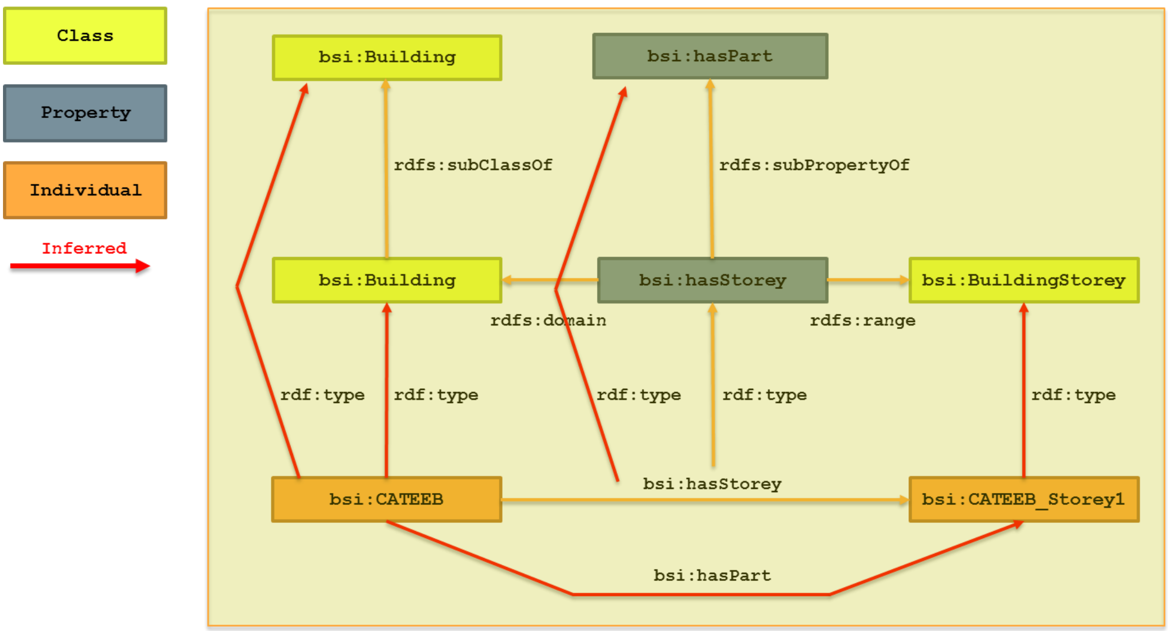


Figure 20: Deriving new data by inference

6.5 Modelling Guide

6.5.1 First Principles

The following first principles are applied as foundational assumptions:

- Maximal reuse of existing LD/SW specifications, so minimal definition of own particular restrictions, extensions and modelling rules:
    - W3C LD/SW Recommendations: RDF (data model and basic vocabulary), SKOS, RDFS, OWL , SHACL (providing extended vocabulary for open and closed world conceptual modelling), SPARQL (for open direct access) & Turtle (as chosen open serialization)
    - Maximal reuse of existing W3C or non-W3C but open 'semantic web' resources (TIME, PROV-O, QUDT etc.) where needed
- Closed World assumption (CWA) to be able to better validate and derive data in an asset information management environment.
    - Hence the importance of SHACL that used to define constraints and rules that should hold for the linked data that can then be validated against such rules (in a CWA fashion).

In this modelling guide we discuss and give recommendations for the use of RDF/SKOS/RDFS/OWL/SHACL (the 'modelling style') but also with respect to the following extended capabilities:

- Quantities and Units
- Individual-level and class-level ('typical') decomposition

The set of extra modelling primitives relevant for the latter is modelled as a generic reusable ontology referred to as CMO. CMO stands for Concept Modelling Ontology and is distributed as a Turtle file: cmo.ttl. This ontology can be imported and reused in/by any other ontology[5].

6.5.2 Guidelines

Ontology Languages

There are several vocabularies available that can serve as language to define ontologies involving varying modelling power:

- SKOS (reusing RDF, RDFS and OWL), mainly for describing class level taxonomies ('knowledge')
- RDFS (reusing RDF), class and data (and their interrelationships) level taxonomies
- OWL (reusing RDF and RDFS), full blown ontologies with many OWA restriction types
- SHACL (reusing RDF and RDFS), providing mechanisms ("shapes") for modelling CWA restrictions/constraints.

In this guide we recommend OWL [OWL] (to be precise, OWL 2) being more expressive than RDFS and enabling RDF-level, 'real-world' data modelling where SKOS is primarily only concerned with knowledge modelling sec. RDFS is however used by OWL and we introduce some optional SKOS vocabulary that might come handy. For CWA constraint and rules we recommend SHACL.

---

[5] It can be directly reused without import as soon as it is dereferenceable at https://w3id.org/cmo. This is also the preferred way: no file names but URIs!

Modelling Styles

Independent of any specific modelling language like Web Ontology Language (OWL) there exist key intuitive meta-concepts from modelling needs in reality/practice when modelling real-world phenomena:

- Concepts
- Instances (having zero or more Concepts as type)
- Value Types
- Attributes (quantitative or qualitative) having Concepts as domain and Value Types as range (having values for instances)
- Relationships (between Concepts/Individuals)
  - (Including ‘classification’, ‘specialization’ and ‘decomposition’)
- Restrictions/Constraints (with respect to values or amounts (‘cardinalities’))

The RDF/RDFS/OWL/SHACL as (layered) language stack from W3C for ontologies and data sets, defines its own key meta-concepts. They are depicted in the following ‘meta-taxonomy’ where the lower level sub-bullets are special cases of the higher level bullets. In **bold** are the RDFS/OWL meta-concepts directly relevant for usage, i.e. for defining end-user ontologies and data sets:

- rdfs:Class
  - owl:**Class**
    - owl:**Restriction**
- rdfs:**Datatype** (reusing basic XSD datatypes)
- rdf:Property
  - owl:**DatatypeProperty**
  - owl:**ObjectProperty**
    - (Predefined meta-individual: rdf:**type)**
    - (Predefined meta-individual: rdfs:**subClassOf**)
    - (Predefined meta-individual: rdfs:**subPropertyOf**)
- owl:NamedIndividual
- owl:Restriction (OWA) and sh:NodeShape & sh:PropertyShape (CWA)

The relationship between real-world meta-concepts and the RDF/RDFS/OWL/SHACL meta-concepts is far from trivial. In practice many approaches can be identified. A good example is the modelling style used for ifcOWL. This style is strongly influenced by backward compatibility issues (stemming from the object-oriented IFC schema modelled in the ISO STEP EXPRESS language). This modelling style has a very specific translation process involving single inheritance, the Liskow substitute principle, the ladder principle for reuse, specific naming conventions etc. ([IfcOWL], [OBS], [LDAC-2015]).

In this report we describe two main modelling styles[6] These modelling styles primarily differ in the way ‘attributes’ and ‘relationships’ are modelled in RDF/RDFS/OWL. In both styles, concepts’ are modelled as owl:Class’es, Instances as owl:NameIndividual’s and Constraints as owl:Restriction’s (OWA) or SHACL shapes (CWA).

- Modelling Style 1: ‘The Simple Way’

[6] In practice, many more variants can be observed. For instance, in ISO 15926-2 having a property value (e.g. a thing having a mass of 4 kilos) is modeled as an instance being a member of a class “things with mass 4.0 kg”)

This variant uses RDF/RDFS/OWL modelling constructs in the most simple and direct way. Value types are always modelled as RDFS datatypes including enumeration types where the enumeration items become just simple ‘allowed strings’[7]. Attributes map to owl:DatatypeProperty’s and relationships are modelled as owl:ObjectProperty’s. Consequently, all constraints, modelled as owl:Restriction’s or SHACL shapes, are also of the simplest form, utilizing RDFS/OWL/SHACL constructs in the most direct way.

Note that with this simple variant, the unit information for instances is easily lost since it is not possible without complex reification or human-interpretation-only annotation modelling, to define a property for a property (like a unit for a height).

The only way now that unit information can be dealt with in a machine-interpretable way is to use it as datatype for the range of a property. This ‘trick’ can only be applied once, so in general annotation is limited. Another know issue is the loss of the actual datatype like xsd:float which now has to be assumed in case of a unit.

- Modelling Style 2: ‘The Powerful Way’

  Now, all attributes and relationships are modelled as owl:Class’es. In particular, also enumeration datatypes are modelled as classes (having the allowed enumeration items as individuals). So the relationship towards such an enumeration class also becomes a class.

  Several extra (own-defined) helper or meta-classes/meta-properties are now needed to tie them all together like: hasAttributeValue, AttributeValue, hasRelationshipObject, RelationshipObject. The main advantage is that it is easier to attach extra meta-data like units, multi-lingual labels, version and status info to these classes and their individuals. This extra flexibility however comes at a high complexity-price, both in terms of space and time. This modelling style reuses QUDT2.0 more extensively than modelling style 1. End-user attributes like height are modelled as subclasses of qudt:QuantityValue from where a qudt:Unit is linked and the actual qudt:numericvalue is assigned.

  Moreover, QUDT-like mechanisms are defined to model qualitative attributes and also relationships (including decomposition).

Formal/standard quantity kind info (like a height being a length) can be inferred in both modelling styles via the available qudt (reference) data. When a height is expressed in unit:M, we can infer that it must have quantityKind:Length.

The recommendation in this report is to use the Simple modelling style whenever possible. The added value of the complex variant is acknowledge but in general the complexity-price is too high.

In Appendix 1 a small/typical example is defined according to the two modelling styles (for simplicity using OWA OWL restrictions instead of CWA SHACL shapes which are not the main issue here). It is evident from that exercise that, if the ontology is kept simple, there is much less need for extra/own language-level constructs and the data stays simple too.

The cmo.ttl ontology (Appendix 3) supports both modelling styles by providing all the meta-modelling constructs needed.

[7] Unless multi-lingualism is really required. In that case they are modelled as classes with enumeration items modelled as individuals together with an object property to refer to them.

Naming (classes, properties, datatypes, individuals)

1. Class/Shape, property and datatype and individual names (the local names within the URIs) are preferably user-friendly names, no codes
   - In case codes ARE used, make sure human-readable labels are used in the form of RDFS or SKOS labels too (some tools can toggle between names and RDFS labels in the GUI)
2. When using names:
   - Class/Shape or datatype names always start with a capital (upper case) letter (“CamelCase”)
   - Property names always start with a non-capital (lower case) letter (“camelCase”)
   - Class/Shape, property and datatype names are in preferably in English and singular

Annotations

In case we want to add extra tags to denote classes, properties and datatypes, improving readability for humans, it is recommended to use the simplest available mechanism being rdfs:label. This label can be multi-lingual and there can be multiple labels in the same language. Labels are certainly recommended when codes are used as names.

Furthermore, if you want to use labels for e.g. generic visualization, you typically need one <u>determined</u> choice per language. In that case we propose the prefLabel and altLabel properties (as sub-properties of rdfs:label) from SKOS [SKOS] in the following way:

- One or more skos:prefLabel’s (exactly one in English (@en), max one per other language)
- Zero or more skos:altLabel’s as synonyms (any language)

For descriptions/definitions it is recommended to use standard rdfs:comment’s.

End-user’s (unique) identifiers or classification codes can always be explicitly modelled in the ontologies/data themselves. Example for data according to the Dutch CROW RAW classification for infra:

```
:ConstructionTask_123
        rdf:type ConstructionTask
        :rawMainCode 31.31.12 ;
        :rawSubCode 224321 .
```

Specialization

Here we use rdfs:subClassOf and rdfs:subPropertOf without any restriction. The only recommendation is to define disjunctness (amoung classes and amoung properties) where possible, enhancing the meaning of the concepts and providing more complete input to reasoners when used.

Decomposition (cmo:hasDirectPart)

If specialization is the most important abstraction mechanism, decomposition is certainly the second one. Inspired by [DECOM] we define for the simple modelling style a non-transitive cmo:hasDirectPart object property. Also its inverse cmo:isPartOf is defined in CMO.

Typical decomposition (resulting in a ‘Meronomy’, see the Glossary for an explanation) is accomplished via Qualified Cardinality Restrictions (QCRs) on this cmo:hasDirectPart object property.

In case attributes and relationships (and so also a hasPart relationship) are objectified as in the powerful modelling style, we can use cmo:hasDirectPartObject / cmo:DirectPartObject / cmo:objectReference (and its inverse cmo:referredBy) to enable the same functionality.

Units & quantities & Enumeration Data Types

The handling of units and quantity information is strongly dependent on the modelling style (see paragraph 6.5.1). In both styles we reuse QUDT2.0 in the most efficient/minimal way. Furthermore, if the enumeration is more like a 'type- or kind-attribute' it is recommended to introduce subclasses instead.

Reusability warning for property restrictions

We just give a warning when using property restrictions like rdfs:domain and rdfs:range.

Once specified, they cannot be 'undone' by say a using ontology. So when a domain is specified (using OWL restrictions or SHACLE shapes) such as :Bridge for :height, another ontology importing it, cannot use the height for a Building only: if there is a height there always has to be something in the domain being a Bridge AND a Building, which is typically not the intended semantics.

Property relevance by minCardinality/minQualifiedCardinality being 0

In case of optional properties (datatype properties or object properties like cmo:hasDirectPart), explicitly modelling minCard=0 (which is implicitly the default in OWL) for a property, marks it as a special property. This approach is introduced because in principle, anything can be a property of, or related to (like being a part of), anything else when not constrained explicitly.

Often, it is convenient to make this (optional) relevance explicit so that the end-user can be given a template involving typical/relevant parts for an individual having a certain class as type.

Closed World Assumption (CWA)

Restrictions/Constraints with a Closed World Assumption (CWA) enable validation of data against these restrictions/constraints. There are several ways to obtain CWA:

- Just use OWL semantics but now interpreted the CWA-way (i.e. via reasoner configuration), or
- Use of CWA-based constraint/rule languages like SHACL [SHACL].

Typically in CWA, also unique names are assumed (UNA). This way it is not necessary to explicitly state for each class, property and individual that it is different from all other classes, properties, individuals (which is in general not feasible).

Below is a typical SHACL shape constraint, saying that all persons have at least one first name.

```
family:Person
  rdf:type owl:Class ;
  rdf:type sh:NodeShape ;
  rdfs:subClassOf rdfs:Resource ;
  sh:property [
      sh:path family:firstName ;
      sh:datatype xsd:string ;
      sh:description "A person's first name" ;
      sh:minCount 1 ;
      sh:name "first name" ;
    ] ;
```

Here de typing to an OWL class is combined with the node shape definition but this can also be separated by importing the OWL ontology and then adding:

sh:targetClass familiy:Person ;

in the shape graph.

The question that arises is: how can we best combine in one ontology both OWA and CWA involving both OWL and SHACL. It should be possible to use the ontology in OWA-mode for inferencing and in CWA-mode for validation without specifying things double.

This is not a trivial question given also the facts that:

- SHACL constraints can be automatically generated from OWL restrictions (for one time transformation or for keeping them both in in parallel over time)
- SHACL has more constraint/rule power then OWL especially because of the extensible SHACL advanced features involving SPARQL or even JavaScript.
- Besides validation SHACL also provides advanced inferencing (via SHACL rules)

The note [SHACL-OWL] addresses this issue asking the question “when to use OWL, SHACL or both?”. In that note it is recommended to split between RDFS, OWL and SHACL (the OWL file importing the RDFS file and the SHACL file importing the RDFS file) for a maximum ‘separation of concerns’ as depicted in the next figure.

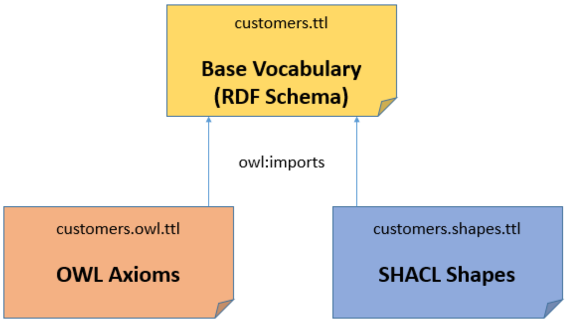


Figure 21: Future combination of RDFS, OWL and SHACL?

In this INTERLINK modelling guide, however, we recommend a bit simpler approach splitting in RDFS/OWL and SHACL. The OWL/RDFS file covers the RDFS/OWL constructs plus the restrictions (if any) for OWA inferencing, and the SHACL file(s) covers the SHACL constraints and rules for CWA validation and inferencing according to one or more close world view (next figure).

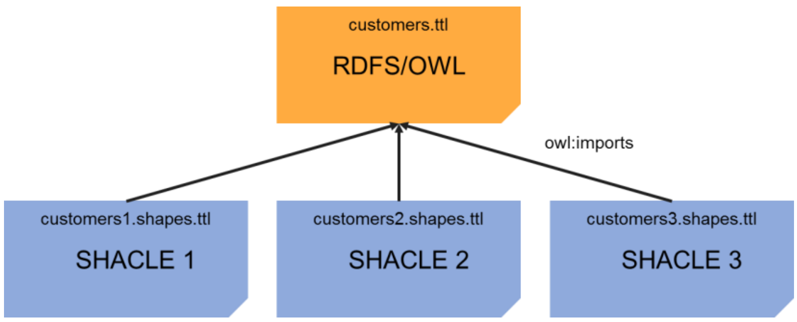


Figure 22: Recommended combination of RDFS, OWL and SHACL in INTERLINK

With closed world SHACL shapes it is much easier to do closed world data validation resulting in meaningful validation results as depicted in the next figure.

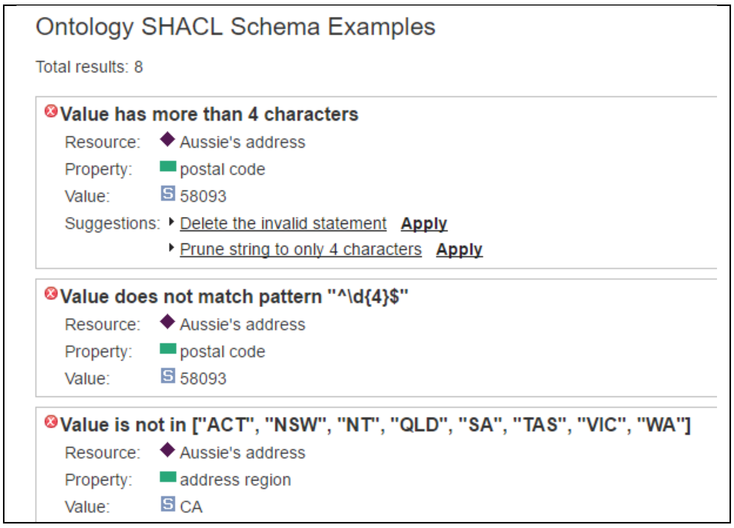


Figure 23: Validation results (in TopBraid Composer v.5.4beta)

More information on SHACL can be found at [SHACL], [SHACL-OWL] and [SHACL Play].

6.5.3 Modelling Heuristics

Role-based classes

Classes are declared ('introduced') in ontologies but really only get meaning by explicitly modelling the restrictions (or, with another term 'constraints') that apply to them. These restrictions/constraints are two-fold:

- Restrictions related to attributes (datatype properties), and
- Restrictions related to relationships (object properties).

The second category gives rise to so-called "role-based classes" that are defined with specific roles in mind towards other classes. Individuals of a role class are not (only) typed based on their own intrinsic datatype properties but on their relationship with other individuals, which are more likely to change over time resulting in changes in the applicable typing.

For example: the class Person is non-role-based Class. The subclass Father is role-based since it depends on a restriction on say a 'hasChildren' object property. A particular person's type will change as soon as he gets children. The class Wheel is not role-based but a subclass FrontWheel is, since it depends on a location relationship with other car part individuals.

Theoretically, one can argue that all classes in the end are somehow role-based. The kind of restrictions are often more involving 'statistics' than 'modelling'. The concept of a Person is for instance also determined by the fact that someone of type Person can have children or in 72% actually has children etc. Such knowledge is however not encoded in simple cardinality constraints like "0 or 1", "1 or more" etc.

So, role-based classes are totally fine and often needed i.e. in cases where you want to add more knowledge on Fathers or FrontWheels only. However in their usage/instantiation they might have some unexpected side-effects so it is recommended to first decide on the non-role based classes and their sub-classes based on intrinsic datatype property restrictions. At least one should be aware of the difference and the consequences for interactions with other classes. Mixing constraints for attributes and relationships for subclassing from one layer to another can for instance result in non-coherent sister classes.

Strong modelling ("necessary" versus "sufficient")

If we define an ontology with classes having no restrictions, all semantics is only "in the name of the class". Labels, descriptions, definitions of codes can add knowledge but only for human interpretation.

Only by adding more necessary conditions in the form of restrictions/constraints we add semantics to the classes making them 'smarter'. Technically in OWL, this means that we define a restriction class and specify that our class is a subclass of it (via rdfs:subClassOf).

If we can replace this subclass relation by an equivalent class (owl:equivalentClass) it means that if the restriction holds (typically a set of restrictions combined with AND/OR-logic) we know the class assertion holds. Said otherwise: we can automatically classify any individual to a defined class that is satisfying the restrictions. We say the restrictions are "sufficient" (for class membership).

Clearly, an ontology having only classes that are defined with 'necessary AND sufficient' properties is smarter than an ontology having only empty class declarations. As a consequence, the smarter an ontology the more information can be inferred from it (like the mentioned automatic classification or in linked data terms: inference of rdf:type statements) and/or individuals as data can be validated against it. Therefore the guideline here is to try to capture as much as knowledge reflected in necessary or even better, sufficient properties (this can be both value or cardinality type restrictions involving both datatype and object properties) for all relevant classes identified. The other guideline is a warning not to over-specify a class this way, making it less flexible for future instantiation (is a FlinstoneCar really a Car?).

6.6 Linking Guide

6.6.1 Introduction

'Linking' is an approach for loosely-coupled, decentralized knowledge/data integration. It does not require large-scale upfront specification and transformation efforts; instead, existing data or knowledge (in the form of OTLs/ontologies) can be linked from the outside, establishing pathways across them.

Whereas linking can be used also across models stored in centralized repositories, its unique power becomes evident in decentralized data environments - typical in the construction industry - where different OTLs/ontologies and datasets are produced and maintained by independent parties, often in parallel, and residing in different hosts over the net.

Linking can support the evolution of a 'semantic ecosystem' for the built environment by incrementally establishing the access paths between data/knowledge sources as required by new or evolving software application functionalities.

In practise 'links' are typically not modelled explicitly but derived 'on the fly' based on properties. That can be 'same IDs' (in a closed world situation or in situations where those IDs are agreed between parties) for individuals or concepts, a location/area in space and/or a point/period in time for individuals. Another variant of implicit linking is in people heads when they program functionality like making assumptions about individuals or concepts when defining a SPARQL query over several ontologies/data sets.

From here we will focus on explicit linking where specific vocabulary is used to represent the links. There are many ways to link two semantic resources involving since all existing RDF/RDFS/OWL/SHACL predicates can be utilized not only to link internally but also externally between resources of different sources from independent parties.

It is important to note that our Closed World Assumption (CWA) really holds "per independent party" so when linking externally we have again a kind of Open World Assumption ("anybody can have its own closed world") involving/linking multiple Closed Worlds! So where we do not need e.g. owl:sameAs within one CWA ontology (since all things with the same uri/name are the same and with a different uri/name are not the same, by default) we do need owl:sameAs to indicate two things from different parties are actually the same.

The actual links between different party data can be defined at one end or better, separately in the middle not disturbing one of the two ends (many such interactions involving more than two resources will become quickly chaotic).

6.6.2 Levels of Linking

Linking can take place at different levels/granularities:

1. **Class-level linking** in Linking Ontologies.

   How the classes and properties in different ontologies correspond to each other? Class-level linking makes it possible to interpret data belonging to different types of be interpreted - that is, understood, compared and manipulated - in a same way. An example would be to state that a 'CityGML Building' is equivalent to an 'IfcBuilding'.

2. **Model-level linking**

   How different models (data sources, containers or streams) are related to each other? Especially in AEC projects there is often information in advance about the models to be created and the relations that these models will have with each other, even before any content for the models have been created. For instance, in a construction project it is known in advance that an architectural model, structural model and MEP model will be produced, and that the structural model and MEP models will be based on the architectural model (and thus can use it as a reference model), and the MEP model and structural model share same physical space (and thus must be checked for collisions). Model-level links are metadata about models and are useful to guide the workflows of information production.

3. **Instance-level linking** in Linking Data Sets

   How are individual objects related to each other? This means concrete relations such as which structural walls correspond to which architectural walls, which building object a status event relates to, or what is the space that a sensor is providing information from. Instance-level linking provides information pathways between objects. For instance, if we want to know the status of a architectural wall, we can access the structural walls linked to it, and then access the status events related to structural walls. There are numerous other similar examples. One important area is the coordination of changes between models.

In this guide we focus on 1. and 3. which are depicted in the next figure. The basic mechanism is a Link Set 'in the middle' that can be on class-level a Linking Ontology (LO) or on data-level a Linking Data Set (LDS).

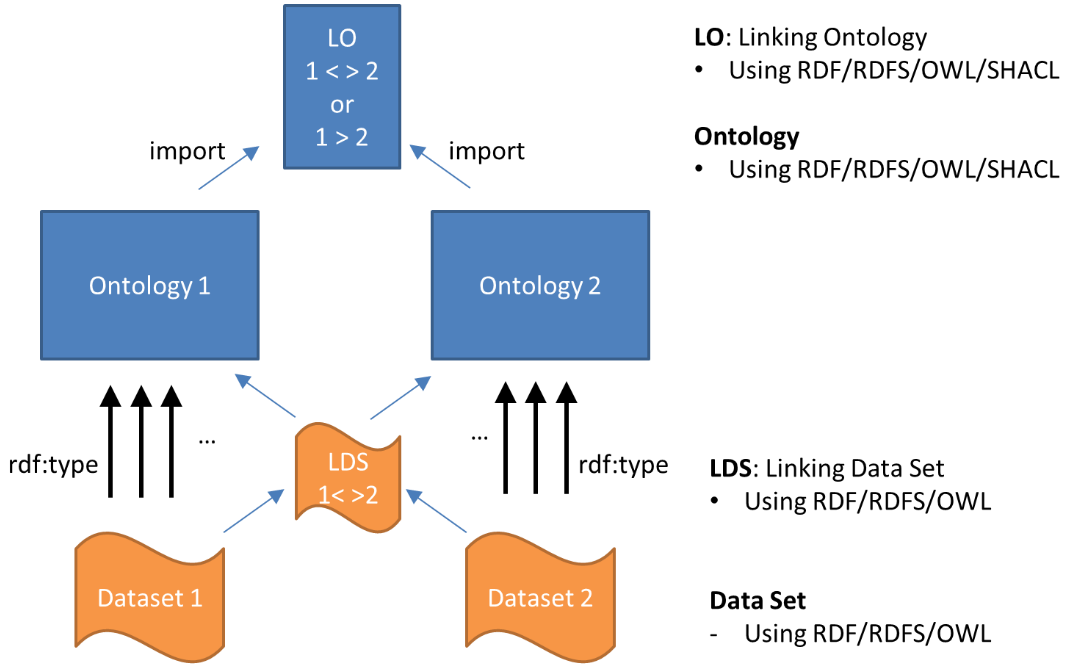


Figure 24: General situation for 'linking' with Link Sets

In this figure we see two ontologies that are linked class-level together by a Linking Ontology (LO) and two datasets that are linked together by a Linking Data Set (LDS). 'What' is asserted or inferred/calculated and 'how' is very much dependent on the kind of situation. It is recommended to distinguish at ontology-level between two main situations with two types of Linking Ontologies:

1. Conversion Rule Sets (CRSs) for Data Conversion/Restyling
2. Alignment Ontologies (AOs) for Data Linking

Data Conversion/restyling is sometimes needed to be able to make it possible that data from one software application can be interpreted by another software application (beyond "translation" that make the data 'just' processable by another application). Data conversion (together with data translation) has been a key theme for the last 30 years in systems integration including all the trouble that comes with it (incompleteness, wrong mappings, multiple data sets representing the same real-world objects etc.). In the previous figure it would mean we have Dataset 1 and convert it into Dataset 2 with the help of a Conversion Rule Set (CRS).

To be as complete as possible, we typically need a lot of mapping power (complex OWL/SHACL restrictions complemented with complex, sometimes procedural SHACL constraints and rules involving SPARQL code or even programmed functions underneath according to some programming language binding (JavaScript in advanced SHACL) Quantitative/numerical conversions involving calculations are a typical example.

Data Linking, on the other hand, represents the LD/SW paradigm shift, where data is not converted all the time, but just kept and edited at their original source according to their own ontology and linked to other data sources. Clearly we expect more from (semantic) software applications in case it needs to interpret 'foreign-ontology' data. The Linking Ontologies can however stay simpler, we call them Alignment Ontologies (AOs). We do not need procedural approaches (like querying/coding in advanced SHACL) but can stay declarative in RDF/RDFS/OWL/SHACL to support the data linking using basic reasoning.

The main recommendation is to clearly determine the purpose of the linking in the actual INTERLINK Test Cases and to avoid conversion/restyling as much as possible. In Appendix 4: Linking Background Information it is shown how the actual ontology and data alignment(/conversion) can be accomplished using a RDF/RDFS/OWL (subset) and standard associated reasoning.

6.6.3 Translation to/from 'non-LD-formatted' resources

A key success factor for the uptake of LD/SW is possibility to also deal with non-LD/SW data. This could be a one-time activity to bring non-RDF to RDF (like generating ifcOWL from an ISO STEP EXPRES-based IFC schema) but also a continuous activity to translate data from a non-RDF compliant software application to RDF (like translating an OGC GML XML file to RDF or vice versa).

Alternatively one can always decide not to translate at all but just reference non-RDF data using for instance an 'rdfs:seeAlso' property that is just pointing to some URI containing an HTTP address where the resource can be found. A more file-based approach is actually the LDP Container [LDP] in which RDF and non-RDF resources can be grouped together.

In general the translation process can be visualized as in the next figure. Because our context is IFC and hence primary the ISO STEP technology we choose this as the counter-part of LD/SW technology to translate to and from. A similar figure could be drawn for GIS-based specifications and data involving XSD Schemas and XML data.

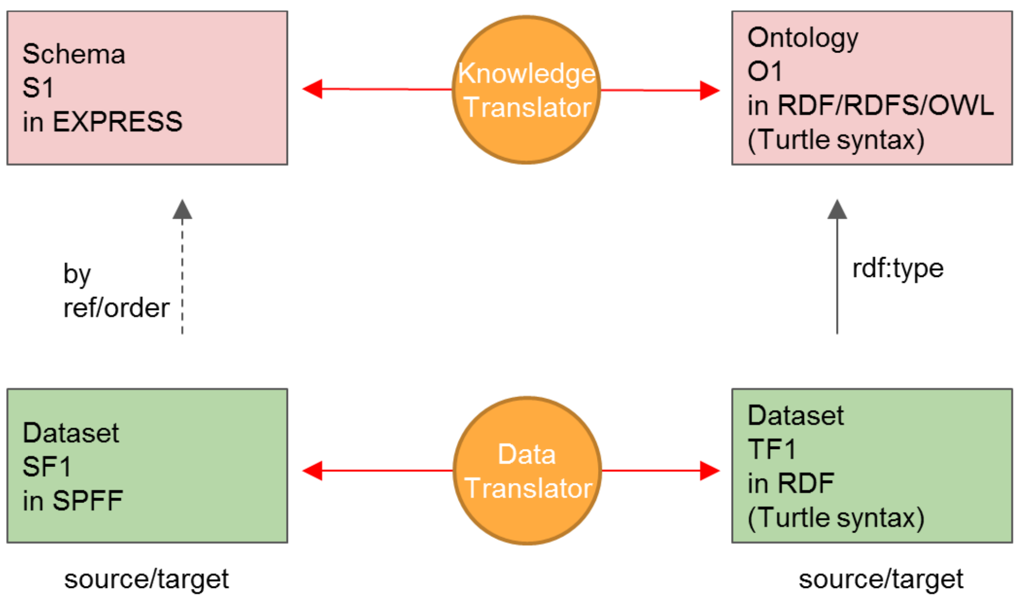


Figure 25: Data Translation

First, on the ontology-level the knowledge is translated from one format to another (here EXPRESS to OWL) and next, the SPFF data is translated to an RDF dataset choosing one the equivalent RDF formats like Turtle. Note that in this case we have two languages on the left and one on the right (also used for interrelating ontology and semantic data using 'rdf:type').

6.7 Explicitly not in scope of this report

Data grouping mechanisms like containers

Specific data access aspects with respect to containers that group sets of data according to some view/dimension for data exchange are dealt with in ISO TC59/SC13[8]/WG8 under the term Information Container for Data Drop (ICDD) and on a European level in CEN TC442/WG2.

In these standardization groups several initiatives bring their ideas and results that are intended to be harmonized (DRUM (FI), Mephisto Multi-Model Containers (MMC) (DE), bSI BCF2.0 (INT), COINS (NL). The goal here is to pave the way for data platforms that can collect, merge and write such containers. Therefore the topic will not be dealt with in this report.

It is preferred however that the resulting specifications are compliant to the existing W3C Linked Data Platform (LDP) [LDP] specification most notably its section on containers.

INTERLINK is also in contact with the participants in these groups (especially from COINS) and harmonization has taken place for the modelling and linking recommendations that will also be addressed in that group.

Regulations, requirements and recommendations

Specifying governmental regulations, client requirements and expert recommendation are topic already addressed by other working groups like the bSI Regulatory Room. Although the interdependencies with a way of modelling are obvious, these kind of specifications will not be addressed in this report other than those requirements that are 'definitional' like OWL Restrictions or SPIN constraints/SHACL [SHACL] shapes constraining classes and properties in ontologies.

Data Set level Versioning

Although guidelines for ontology versioning are given data set versioning is not considered. Although the same guidelines could be applied, the data sets are typically much larger requiring more advanced forms of versioning (like on a more granular level of individuals or even RDF triples).

Importing Strategies

No guidelines are defined on importing strategies (import or just reference). All have pros and cons w.r.t. factors like:

- What would the theoretically the best option?
- What are the effects on availability?
- What are the performance implications?
- Use of local offline files is also up to the software tools used.
- Because with OWL always whole ontologies are imported (and not just some items from them) it might sometimes be handy to define you own subsets; again this is all up to the modeller.

Publishing Aspects

We just follow best practices already well documented in [BPLD] & [COOLURIS] for aspects like:

- Server .htaccess files
- HTML document links + multiple LD serializations
- URI Strategies

---

[8] The same group defining the ISO/DIS 19650 series

- File extensions: Like: no file extensions in URIs, following best practice

It is highlighted that ‘not being in scope’ does not mean that is not relevant for the test cases. So where needed we will borrow results from initiatives that do touch these issues (like data containers). For the more technical topics ‘out of development scope’ the software platform implementer (Semmtech) will make appropriate choices, e.g. on data set level versioning, importing strategies and publishing aspects.

7 Software Implementation Guidelines

7.1 Introduction

'Implementation' is a term that can be interpreted in many ways. It can refer to 'implementation in software' but also to further, more detailed, technical choices still independent of a particular software implementation. The latter meaning is the one relevant for this chapter and also the area we provide guidelines for.

For LD/SW such 'implementation' guidelines are especially relevant since there are typically several technical ways to do the same conceptual thing.

A good example is the actual format chosen to represent ontologies and their associated data sets. There are several alternatives around that are mostly equivalent (that can be translated into each other without loss). We also discuss alternatives to access the data (file-based involving the formats or more direct access utilizing a standard Query Language.

7.2 Formats & Access

7.2.1 Formats

Several, to a large extent equivalent, formats (syntax forms/serializations) exist for serialization of RDF/RDF/OWL like:

- RDF/XML (primary syntax, XML-based, RDF-level)
- N-Triples (non-XML-based, RDF-level)
- Turtle (non-XML-based, RDF-level, more efficient variant of N-Triples)
- Recently added: JSON-LD (equivalent to the other formats in case certain restrictions on the use of JSON-LD are applied) (non-XML-based, RDF-level)

XML-based syntax forms are often liked by people using XML software tools. JSON-LD is popular with JavaScript-oriented semantic web application developers.

Our first preference is for syntax forms that are 'RDF-level'. They closely resemble the fact that OWL (and also RDFS) can be regarded as 'just a special vocabulary' in RDF. In other words, they keep the layered structure which promotes a modular approach. They are typically also much more efficient in terms of space and time (think performance when reading in) that i.e. XML-based serializations. The preferred RDF-level syntax form for use in bSI is Turtle [Turtle], being more efficient in space than (its subset) N-Triples. In Turtle we can use so-called prefixes that can make the code much more human-friendly:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
PREFIX owl: <http://www.w3.org/2002/07/owl#> .
PREFIX ifc: http://ifcowl.openbimstandards.org/IFC4x1#> .

ifc:IfcTextTransformation rdf:type owl:Class .
```

Now all subjects, predicates and object can have a 'prefix' that is just a shorthand for some URI-fragment that has to be prefixed for the actual identifier to obtain its full URI. Note that rdf:type is a predefined RDF predicate to indicate that a subject is of a certain object type.

Furthermore, Turtle enables us to introduce three more abbreviation mechanisms when modelling the an RDF graph:

- Parallel: same subject via ; separator, so one subject having multiple predicates,
- Parallel: same subject-predicate via , separator, so one subject-predicate combination having multiple objects,
- Sequential: the end of one triple is start of the next triple via […] grouping

To illustrate Turtle, assume we have the following triple set:

```
a1 b1 c1.
a1 b2 c2.
a1 b2 c3.
a1 b3 c4.
c4 b4 c5.
```

We can then apply all abbreviation types as shown in the next figure.

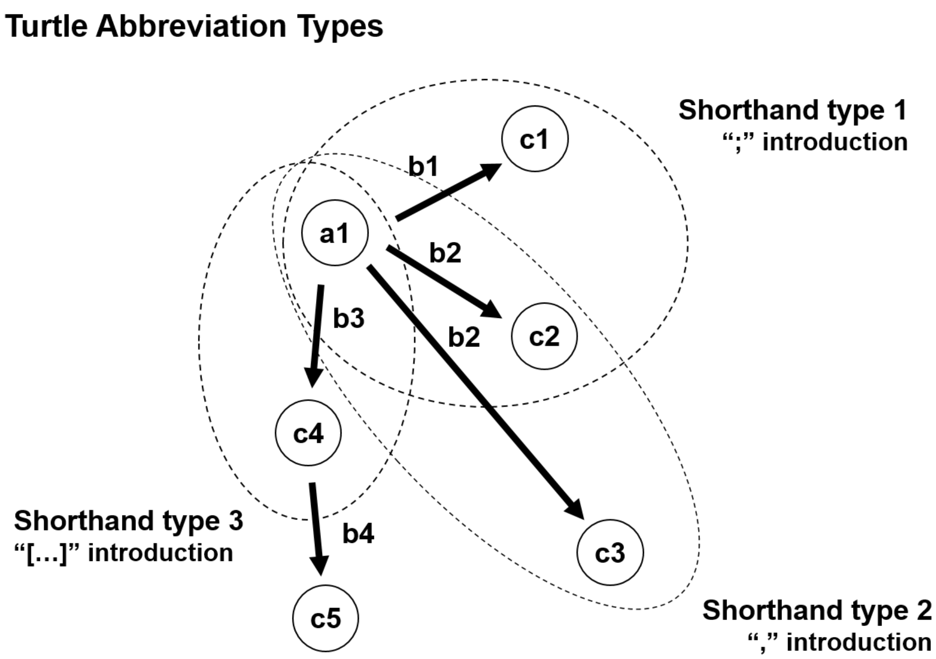


Figure 26: Turtle format explained

Resulting in the very efficient:

```
a1 b1 c1; b2 c2,c3; b3 [c4 b4 c5].
```

Especially the first abbreviation (;) is applied very often when multiple predicates for the same subject are relevant:

```
:RoadSegment_123
   :hasWidth "14.8"^^xsd:float ;
   :hasLength "1800.0"^^xsd:float ;
   :hasDepth ""0.8"^^xsd:float ;
   :hasMaterial :OpenAsphaltConcrete-12 ;
   :hasExplicitShape :IFC-BoundingBox-321 .
```

7.2.2 Direct Access

In addition to import/export (aka upload/download) of Turtle (or RDF/XML, JSON-LD, etc.) files, we assume direct access to RDF Data Sets (ontologies and/or individuals) through the standard SPARQL query language [SPARQL, SPARQL-T].

This access might be in future extended/specialized via development of a higher-level QL on top of SPARQL.

Besides this language-independent direct access several 'programming language'-dependent application programming interfaces (APIs) exist. A good example is the Java API for SHACL [SHACL API] itself build on top of JENA ([JENA]) (an often used Java API for RDF).

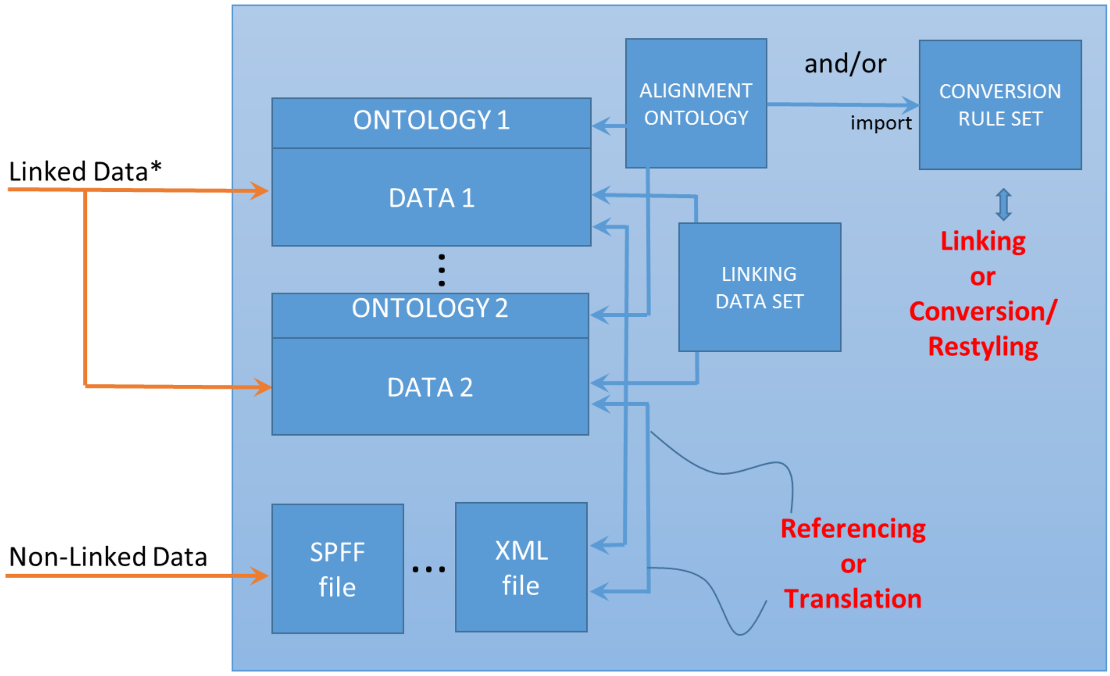
7.3 Versioning of ontologies

- No version info in URIs for classes and properties etc. in ontologies
- Only versions in URIs for ontologies as a whole
- The 'latest version' will be adapted (all version info out of the URIs). This version will be (by default) referenced by other ontologies. If needed ontologies can refer to specific earlier versions
- All to be handled by semantic 'negotiating' servers

A simple example is provided in appendix 2.

7.4 Software Platform Architecture for Implementation

In the following figure we indicate the basic functionalities for a Linked Data Management Platform.



* Meaning: formatted according to an official LD format like Turtle, RDF/XML, JSON-LD

Figure 27: Typical Platform Architecture for Linked Data & Non-Linked Data management

- Linked Data in any linked data format (at least the efficient and user-friendly Turtle) can be imported according to any ontology,
- Non-‘Linked Data‘ like ISO STEP Files or W3C XML files can be imported,
- Non-‘Linked Data‘ can be translated to Linked Data or just referenced from the Linked Data
- Linked Data can be
    - Linked by Linking Data Sets optionally under control of Alignment Ontologies, or
    - Restyled or converted under control of Alignment Ontologies and/or Conversion Rule Sets.

This architecture is directly relevant for the proof-of-concept software to be developed/provided in task WPD2, taking into account the right mix of vocabularies recommended in this guide (RDF/SKOS/RDFS/OWL/SHACL) for ontologies, data sets and link sets.

8 Abbreviations

Abbreviation	Explanation [optional context], '|' means alternative
BASt	Bundesanstalt für Straßenwesen [DE]
BIM	Building Information Model(ling) | Building Information Management
bSDD	buildingSmart Data Dictionary [bSI/Product Room]
bSI	buildingSmart International
CB-NL	Concepten-Bibliotheek NL (Dutch Concept Library for the Dutch construction industry) [NL]
CBIM	COINS BIM
CEDR	Conference of European Directors of Roads (the Platform for cooperation between National Road Authorities in Europe)
CSTB	Centre Scientific et Technique du Bâtiment [FR]
CMO	Concept Modelling Ontology [TNO/CSTB/RDF.bg]
COINS	Constructive Objects and INtegration of Processes and Systems [NL]
CWA	Closed World Assumption
DX	Deliverable X [INTERLINK]
DC	Dublin Core [DCMI]
DCMI	Dublin Core Meta-data Initiative
DWG	DraWinG binary format [Autodesk]
EU	European Union
GIS	Geographical Information Systems
GUI	Graphical User Interface
GUID	Globally Unique IDentifier
HTTP	HyperText Transfer Protocol [IETF]
HVAC	Heating, Ventilation and Air Conditioning
ICDD	Information Container for Data Drop (ICDD) [ISOTC59/SC13/WG8]
IETF	Internet Engineering Task Force
IFC	Industry Foundation Classes [bSI]
IM	Information Management
INSPIRE	INfrastructure for SPatial Information in EuRopE Directive [EU]
INTERLINK	INformation managemenT for European Roads using LINKed data [CEDR]

Abbreviation	Explanation [optional context], '|' means alternative
IoT	Internet of Things (web-based sensors & actuators)
ISO	International Standardization Organization
IT	Information Technology
LBD CG	Linked Building Data Community Group [W3C]
LC	Life-Cycle
LD	Linked Data [W3C]
LDP	Linked Data Protocol (including also a container specification for combined RDF & non-RDF -based resources) [W3C]
LDWG	Linked Data Working Group [bSI/Technical Room]
LoD	Level of Detail | Level of Development
MLG	Modelling and Linking Guide [VCON | LDWG | INTERLINK]
MNGT	Management
MOF	Meta Object Facility [OMG]
MVD	Model View Definition [bSI]
N3	Notation3 [W3C]
NRA	National Road Authority
OGC	Open Geospatial Consortium
OKSTRA	Objekt Katalog für das Straßen- und Verkehrswesen [DE]
OMG	Object Management Group
OTL	Object Type Library
OWA	Open World Assumption [SW]
OWL	Web Ontology language [W3C]
PDF	Portable Document Format [Adobe]
PEB	Programme Executive Board [CEDR]
PoC	Proof of Concept [INTERLINK]
PROV-O	Provenance Ontology [W3C]
QCR	Qualified Cardinality Constraint [OWL]
QL	Query Language
RDF	Resource Description Framework [W3C]
RDFS	RDF Schema [W3C]
RVT	Revit file/format [Autodesk]
RWS	Rijkswaterstaat (the Dutch NRA)
SC	Supply-Chain

Abbreviation	Explanation [optional context], '|' means alternative
SE	Systems Engineering (life-cycle/supply-chain data management including requirements definition)
SPARQL	Simple Protocol and RDF Query Language [W3C]
SPFF	STEP Physical File Format [STEP]
STEP	Standard for the Exchange of Product model data [ISO]
SW	Semantic Web [W3C]
TN	Transport Network [INSPIRE]
TV	Trafikverket (the Swedish NRA)
Turtle	Terse RDF Triple Language [W3C]
UNA	Unique Name Assumption [SW]
V-Con	Virtual Construction for roads [EU]
W3C	World Wide Web Consortium
WPL	Work Package Leader [INTERLINK]
WWW	World Wide Web [W3C]
XML	eXtensible Markup Language [W3C]
XSD	eXtensible Schema Definition language [W3C]
Y1, Y2	Year 1, Year 2 [INTERLINK]

9 Glossary

Key concept	Definition
Asset Management	The management processes covering the whole life-cycle and supply-chain of (networks of) assets like buildings, roads, bridges, tunnels, dikes, airports, ports, etc., and their parts, including the realisation of new assets and the maintenance, renovation and repurposing of existing assets. Typically excluding the management of its operation (like Traffic Management).
Asset Information Management	All information management activities relevant for Asset Management including data acquisition/monitoring, storage, analysis, calculation/simulation, exchange and sharing of asset data (i.e. the whole 'information life-cycle').
Taxonomy	A taxonomy is a kind of hierarchy based on specialization. In LD/SW the rdfs:subClassOf en rdfs:subPropertyOf properties are used.
Meronomy	A meronomy is a kind of hierarchy based on (typical) decomposition (here via restrictions on cmo:hasDirectPart).
Ontology	An ontology is a shared abstract view of a part of reality to be represented for some specific purpose. An ontology is essentially a set of concepts, value types, attributes, relationships and constraints. An often used synonym[9] for 'ontology' is Object Type Library (OTL).Typically, a Taxonomy or sometimes a Meronomy, or both, constitute the 'backbone' of an Ontology.
Data Integration approach	In this report two main approaches for data integration are distinguished 1. Data Exchange 2. Data Sharing, keeping the data according their original semantics and/or format and: a. In case of Linked Data: just linking to it b. In case of a non-Linked Data: just referencing to it Option 2a is preferred, option 2b in case no (good) data translation is available (second best). Option 1 typically involves unwanted complex data transformations and associated conversion/restyling rule sets.
Data Transformation	In this report three types of data transformations are relevant: 1. Data Translation, same meaning, different format 2. Data Conversion, same format, different meaning

[9] Some would say 'special kind of' since it focusses on semantic domain objects rather than say representation of those objects like with geometry

Key concept	Definition
	3. Data Restyling (in between 1&2), same format, same meaning but different modelling style (where 'modelling style' is in between format & meaning) In this report we recommend: 1. Assume Linked Data formats for all data sets or assuming available complete bi-directional translations to/from it where needed, 2. By referencing the non-linked data if option 1. is not feasible 3. By linking the resulting Linked Data instead of data conversion/restyling. 4. If really needed restyle/convert data

10 References

[5star] http://5stardata.info/en/, Tim-Berners Lee.

[bSI] buildingSmart International, http://buildingsmart.org/.

[bSDD] bSI Data Dictionary, http://buildingsmart.org/standards/standards-library-tools-services/data-dictionary/.

[BPLD] Best Practices for Publishing Linked Data, W3C Working Group Note 09 January 2014, http://www.w3.org/TR/ld-bp/.

[CEDR] CEDR INTERLINK, WPA/WPB deliverable D2/D3, "Information Management for European Road Infrastructure using Linked Data; Investigating the requirements", Final Issue, 31st March 2017.

[COINS] Constructieve Objecten en de INtegratie van Processen en Systemen, http://www.coinsweb.nl; http://www.coinsweb.nl/wiki2.

[COOLURIS] Cool URIs for the Semantic Web, W3C Interest Group Note 03 December 2008, https://www.w3.org/TR/cooluris/.

[DCMI] Dublin Core Metadata Initiative (DCMI), http://dublincore.org/.

[DECOM] http://www.w3.org/2001/sw/BestPractices/OEP/SimplePartWhole/.

[DWBP] Data on the Web, Best Practices, W3C Candidate Recommendation, 30 August 2016, http://www.w3.org/TR/dwbp/.

[IFC4x1] http://www.buildingsmart-tech.org/mvd/IFC4x1/Alignment/1.0.1/html/.

[ifcOWL] bSI BuildingSMART Final Standard: The ifcOWL ontology, LDWG – Pieter Pauwels, Walter Terkaj and Jakob Beetz, Final, 29. March 2017.

[InfraGML] OGC InfraGML Encoding Standard encoding concepts supporting land and civil engineering infrastructure facilities specified in the OGC Land and Infrastructure Conceptual Model Standard (LandInfra), OGC 15-111r1, http://www.opengeospatial.org/standards/infragml.

[INSPIRE] https://inspire.ec.europa.eu/SDICS/isotc-211.

[INTERLINKpub] http://www.roadotl.eu/publications.html.

[HALPIN] When owl:sameAs isn't the Same: An Analysis of Identity Links on the Semantic Web, Harry Halpin, Ivan Herman (W3C-ERCIM), Patrick Hayes (Institute for Human and Machine Cognition), https://www.w3.org/2009/12/rdf-ws/papers/ws21, December 2009.

[INSPIRE] INSPIRE DS TN, http://inspire.ec.europa.eu/id/document/tg/tn.

[ISO TC-211] Geographic information/Geomatics, https://committee.iso.org/home/tc211.

[JENA] https://jena.apache.org/.

[LandXML] http://LandXML.org.

[LDAC-2015] http://ldac-2015.bwk.tue.nl/LDAC_2015_workshopreport.pdf.

[LDB] https://www.w3.org/community/lbd/.

[LDP] Linked Data Platform 1.0, W3C Recommendation, 26 February 2015, https://www.w3.org/TR/ldp/.

[LDWG] Linked Data Working Group, part of the bSI Technical Room, http://buildingsmart.org/standards/standards-organization/groups/linked-data-working-group/.

[OBS] http://openbimstandards.org/standards/ifcowl/how-is-ifcowl-generated/.

[ONT101] Ontology Development 101: A Guide to Creating Your First Ontology, http://protege.stanford.edu/publications/ontology_development/ontology101.pdf. Ontology 101: An introduction to knowledge representation & ontology development, http://smartdata2015.dataversity.net/uploads/handouts/TUE_0830_Kendall_Elisa_McGuinness_Deborah_COLOR_7736.pdf.

[OWL] Authoritative reference: OWL 2 Web Ontology Language Primer (Second Edition), W3C Recommendation 11 December 2012, http://www.w3.org/TR/owl2-overview/.

[PROV-O] PROV-O: The PROV Ontology, W3C Recommendation, 30 April 2013, https://www.w3.org/TR/prov-o/.

[PUB] Best Practice Recipes for Publishing RDF Vocabularies W3C Working Group Note 28 August 2008, http://www.w3.org/TR/swbp-vocab-pub/.

[RDF] RDF 1.1 Concepts and Abstract Syntax, W3C Recommendation 25 February 2014, http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/Overview.html.

[RDFS] RDF Schema 1.1, W3C Recommendation 25 February 2014, http://www.w3.org/TR/rdf-schema/.

[RDF-XML] https://www.w3.org/TR/rdf-syntax-grammar/, RDF 1.1 XML Syntax, W3C Recommendation 25 February 2014.

[Reinhart] Property Reification RDF Vocabulary, a strawman draft for an extension to RDFS that allows to express mapping between RDF properties and RDF classes that reify the property, Simon Reinhardt, 30 October 2009, https://www.w3.org/wiki/PropertyReificationVocabulary.

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

[SHACL API] https://github.com/TopQuadrant/shacl.

[SHACL-OWL] SHACL and OWL Compared, http://spinrdf.org/shacl-and-owl.html.

[SHACL Play] SHACL Playground, http://shacl.org/playground/.

[SKOS] https://www.w3.org/2004/02/skos/.

[SoH] SPARQL over HTTP, https://www.w3.org/TR/2013/REC-sparql11-protocol-20130321/.

[SPARQL] SPARQL 1.1 Overview, W3C Recommendation 21 March 2013, http://www.w3.org/TR/sparql11-query/.

[SPARQL-T] SPARQL, Tutorial, https://jena.apache.org/tutorials/sparql.html.

[TURTLE] RDF 1.1 Turtle, Terse RDF Triple Language, W3C Recommendation 25 February 2014, http://www.w3.org/TR/turtle/.

[VOID] Vocabulary of Interlinked Datasets, http://vocab.deri.ie/void/, http://semanticweb.org/wiki/VoID.html#Related_Specifications

Appendix 1: Examples Modelling Styles

Introduction

The example situation modelled: there are three concepts: physical objects, specialized into bridges and vehicles. Bridges are described with exactly one height attribute and vehicles have an optional velocity attribute. Vehicles also have an optional loadLevel attribute which can be Light, Normal or Heavy. Bridges furthermore optionally serve certain vehicles (at a certain point in time). Finally, a bridge always has as part exactly one Deck and a deck always has as part one or more Slabs.

We have one instance of a bridge, with a height of 50 meters serving an instance of Vehicle, with loadLevel being Heavy and having a velocity of 128 km/hour. The instance of bridge has as part a deck instance where this deck instance has three slab instances.

Example 'Modelling Style the Simple Way'

```
simple:PhysicalObject
  rdf:type owl:Class ;
.
simple:Bridge
  rdf:type owl:Class ;
  rdfs:subClassOf simple:PhysicalObject ;
  rdfs:subClassOf [
      rdf:type owl:Restriction ;
      owl:cardinality "1"^^xsd:nonNegativeInteger ;
      owl:onProperty simple:height ;
    ] ;
  rdfs:subClassOf [
      rdf:type owl:Restriction ;
      owl:onClass simple:Deck ;
      owl:onProperty cmo:hasDirectPart ;
      owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
    ] ;
.
simple:Deck
  rdf:type owl:Class ;
  rdfs:subClassOf simple:PhysicalObject ;
  rdfs:subClassOf [
      rdf:type owl:Restriction ;
      owl:minQualifiedCardinality "1"^^xsd:nonNegativeInteger ;
      owl:onClass simple:Slab ;
      owl:onProperty cmo:hasDirectPart ;
    ] ;
.
simple:Slab
  rdf:type owl:Class ;
  rdfs:subClassOf simple:PhysicalObject ;
.
simple:Vehicle
  rdf:type owl:Class ;
  rdfs:subClassOf simple:PhysicalObject ;
  rdfs:subClassOf [
      rdf:type owl:Restriction ;
      owl:minCardinality "0"^^xsd:nonNegativeInteger ;
      owl:onProperty simple:loadLevel ;
    ] ;
  rdfs:subClassOf [
      rdf:type owl:Restriction ;
      owl:minCardinality "0"^^xsd:nonNegativeInteger ;
      owl:onProperty simple:velocity ;
    ] ;
.
simple:height
  rdf:type owl:DatatypeProperty ;
  rdfs:range unit:M ;
.
simple:velocity
  rdf:type owl:DatatypeProperty ;
  rdfs:range unit:KM-PER-HR ;
.
simple:loadLevel
```

```
  rdf:type owl:DatatypeProperty ;
  rdfs:range [
      rdf:type rdfs:Datatype ;
      owl:oneOf (
          "Light"
          "Normal"
          "Heavy"
        ) ;
    ] ;
.
simple:currentlyServingVehicle
  rdf:type owl:ObjectProperty ;
  rdfs:range simple:Vehicle ;
.

# the actual data

simple:Bridge_1
  rdf:type simple:Bridge ;
  cmo:hasDirectPart simple:Deck_1 ;
  simple:currentlyServingVehicle simple:Vehicle_1 ;
  simple:height "50.0"^^unit:M ;
.
simple:Deck_1
  rdf:type simple:Deck ;
  cmo:hasDirectPart simple:Slab_1 ;
  cmo:hasDirectPart simple:Slab_2 ;
  cmo:hasDirectPart simple:Slab_3 ;
.
simple:Slab_1
  rdf:type simple:Slab ;
.
simple:Slab_2
  rdf:type simple:Slab ;
.
simple:Slab_3
  rdf:type simple:Slab ;
.
simple:Vehicle_1
  rdf:type simple:Vehicle ;
  simple:loadLevel "Heavy" ;
  simple:velocity "128.0"^^unit:KM-PER-HOUR ;
.
```

Example 'Modelling Style the Powerful Way'

```
powerful:PhysicalObject
  rdf:type owl:Class ;
.
powerful:Bridge
  rdf:type owl:Class ;
  rdfs:subClassOf powerful:PhysicalObject ;
  rdfs:subClassOf [
      rdf:type owl:Restriction ;
      owl:onClass powerful:Deck ;
      owl:onProperty cmo:hasDirectPartObject.objectReference ;
      owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
    ] ;
  rdfs:subClassOf [
      rdf:type owl:Restriction ;
      owl:onClass powerful:Height ;
      owl:onProperty cmo:hasAttributeValue ;
      owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
    ] ;
.
powerful:Deck
  rdf:type owl:Class ;
  rdfs:subClassOf powerful:PhysicalObject ;
  rdfs:subClassOf [
      rdf:type owl:Restriction ;
      owl:minQualifiedCardinality "1"^^xsd:nonNegativeInteger ;
      owl:onClass powerful:Slab ;
      owl:onProperty cmo:hasDirectPartObject.objectReference ;
    ] ;
.
powerful:Slab
  rdf:type owl:Class ;
  rdfs:subClassOf powerful:PhysicalObject ;
.
powerful:Vehicle
  rdf:type owl:Class ;
  rdfs:subClassOf powerful:PhysicalObject ;
  rdfs:subClassOf [
      rdf:type owl:Restriction ;
      owl:minQualifiedCardinality "0"^^xsd:nonNegativeInteger ;
      owl:onClass powerful:CurrentlyServingVehicle ;
      owl:onProperty cmo:hasRelationshipObject ;
    ] ;
  rdfs:subClassOf [
      rdf:type owl:Restriction ;
      owl:minQualifiedCardinality "0"^^xsd:nonNegativeInteger ;
      owl:onClass powerful:HasLoadLevel ;
      owl:onProperty cmo:hasAttributeValue ;
    ] ;
  rdfs:subClassOf [
      rdf:type owl:Restriction ;
      owl:minQualifiedCardinality "0"^^xsd:nonNegativeInteger ;
      owl:onClass powerful:Velocity ;
      owl:onProperty cmo:hasAttributeValue ;
    ] ;
.
```

```
powerful:Height
  rdf:type owl:Class ;
  rdfs:subClassOf qudt:QuantityValue ;
.
powerful:CurrentlyServingVehicle
  rdf:type owl:Class ;
  rdfs:subClassOf cmo:RelationshipObject ;
.
powerful:Velocity
  rdf:type owl:Class ;
  rdfs:subClassOf qudt:QuantityValue ;
.
powerful:HasLoadLevel
  rdf:type owl:Class ;
  rdfs:subClassOf cmo:RelationshipObject ;
.
powerful:LoadLevel
  rdf:type owl:Class ;
.
powerful:Light
  rdf:type powerful:LoadLevel ;
.
powerful:Heavy
  rdf:type powerful:LoadLevel ;
.
powerful:Normal
  rdf:type powerful:LoadLevel ;
.

# the actual data

powerful:Bridge_1
  rdf:type powerful:Bridge ;
  cmo:hasAttributeValue powerful:Height_1 ;
  cmo:hasDirectPartObject powerful:DirectPartObject_1 ;
  cmo:hasRelationshipObject powerful:CurrentlyServingVehicle_1 ;
.
powerful:Deck_1
  rdf:type powerful:Deck ;
  cmo:hasDirectPartObject powerful:DirectPartObject_2 ;
  cmo:hasDirectPartObject powerful:DirectPartObject_3 ;
  cmo:hasDirectPartObject powerful:DirectPartObject_4 ;
.
powerful:Slab_1
  rdf:type powerful:Slab ;
.
powerful:Slab_2
  rdf:type powerful:Slab ;
.
powerful:Slab_3
  rdf:type powerful:Slab ;
.
powerful:Vehicle_1
  rdf:type powerful:Vehicle ;
  cmo:hasAttributeValue powerful:Velocity_1 ;
  cmo:hasRelationshipObject powerful:HasLoadLevel_1 ;
.
powerful:Height_1
```

```
  rdf:type powerful:Height ;
  qudt:numericValue "50.0"^^xsd:float ;
  qudt:unit unit:M ;
.
powerful:HasLoadLevel_1
  rdf:type powerful:HasLoadLevel ;
  cmo:objectReference powerful:Heavy ;
.
powerful:CurrentlyServingVehicle_1
  rdf:type powerful:CurrentlyServingVehicle ;
  cmo:objectReference powerful:Vehicle_1 ;
.
powerful:Velocity_1
  rdf:type powerful:Velocity ;
  qudt:numericValue "128.0"^^xsd:float ;
  qudt:unit unit:KM-PER-HR ;
.
powerful:HasLoadLevel_1
  rdf:type powerful:HasLoadLevel ;
  cmo:objectReference powerful:Heavy ;
.
powerful:DirectPartObject_1
  rdf:type cmo:DirectPartObject ;
  cmo:objectReference powerful:Deck_1 ;
.
powerful:DirectPartObject_2
  rdf:type cmo:DirectPartObject ;
  cmo:objectReference powerful:Slab_1 ;
.
powerful:DirectPartObject_3
  rdf:type cmo:DirectPartObject ;
  cmo:objectReference powerful:Slab_2 ;
.
powerful:DirectPartObject_4
  rdf:type cmo:DirectPartObject ;
  cmo:objectReference powerful:Slab_3 ;
.
```

Appendix 2: Example Ontology Versioning

OntologyX.ttl – the used ontology in version 0.1

```
@prefix x:          <http://www.test.nl/ontologies/OntologyX/0.1/OntologyX#> .
@prefix owl:        <http://www.w3.org/2002/07/owl#> .
@prefix rdfs:       <http://www.w3.org/2000/01/rdf-schema#> .
@prefix rdf:        <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xsd:        <http://www.w3.org/2001/XMLSchema#> .
x:OntologyX rdf:type owl:Ontology .
x:Road rdf:type owl:Class .
```

OntologyX.ttl – the used ontology in version 0.2 (class added)

```
@prefix x:          <http://www.test.nl/ontologies/OntologyX/0.2/OntologyX#> .
@prefix owl:        <http://www.w3.org/2002/07/owl#> .
@prefix rdfs:       <http://www.w3.org/2000/01/rdf-schema#> .
@prefix rdf:        <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xsd:        <http://www.w3.org/2001/XMLSchema#> .
x:OntologyX rdf:type owl:Ontology .
x:Road rdf:type owl:Class .
x:MainRoad rdfs:subClassOf x:Road .
```

OntologyX.ttl – the used ontology in latest version (content copied from v. 0.2)

```
@prefix x:          <http://www.test.nl/ontologies/OntologyX/OntologyX#> .
@prefix owl:        <http://www.w3.org/2002/07/owl#> .
@prefix rdfs:       <http://www.w3.org/2000/01/rdf-schema#> .
@prefix rdf:        <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xsd:        <http://www.w3.org/2001/XMLSchema#> .
x:OntologyX rdf:type owl:Ontology .
x:Road rdf:type owl:Class .
x:MainRoad rdfs:subClassOf x:Road .
```

OntologyY.ttl – latest version using ontology X (default in latest version)

```
@prefix y:          <http://www.test.nl/ontologies/OntologyY/OntologyY#> .
@prefix owl:        <http://www.w3.org/2002/07/owl#> .
@prefix rdfs:       <http://www.w3.org/2000/01/rdf-schema#> .
@prefix rdf:        <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xsd:        <http://www.w3.org/2001/XMLSchema#> .
@prefix x:          <http://www.test.nl/ontologies/OntologyX/OntologyX#>
y:OntologyY rdf:type owl:Ontology .
y:Network rdf:type owl:Class .
y:hasElements rdf:type  owl:Objectproperty ;
      rdfs:range x:MainRoad .
```

OntologyZ.ttl – some version using ontology X (in older version 0.1)

```
@prefix z:          <http://www.test.nl/ontologies/OntologyY/0.8/OntologyZ#> .
@prefix owl:        <http://www.w3.org/2002/07/owl#> .
@prefix rdfs:       <http://www.w3.org/2000/01/rdf-schema#> .
@prefix rdf:        <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xsd:        <http://www.w3.org/2001/XMLSchema#> .
@prefix x:          <http://www.test.nl/ontologies/OntologyX/0.1/OntologyX#>
z:OntologyZ rdf:type owl:Ontology .
z:Network rdf:type owl:Class .
z:hasElements rdf:type  owl:Objectproperty ;
    rdfs:range x:RoadSegment .
```

Appendix 3: Concept Modelling Ontology (CMO)

```
@prefix cmo: <https://w3id.org/cmo#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix quantitykind: <http://qudt.org/2.0/vocab/quantitykind/base/> .
@prefix qudt: <http://qudt.org/2.0/schema/qudt/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix unit: <http://qudt.org/2.0/vocab/unit/space-and-time/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<https://w3id.org/cmo>
  rdf:type owl:Ontology ;
  owl:imports <http://qudt.org/2.0/schema/qudt> ;
  owl:imports <http://qudt.org/2.0/vocab/quantitykind/base> ;
  owl:imports <http://qudt.org/2.0/vocab/unit/space-and-time> ;
  owl:versionInfo "1.0" .

# supporting the simple modelling style

qudt:Unit
  rdfs:subClassOf rdfs:Datatype .

cmo:hasDirectPart
  rdf:type owl:ObjectProperty ;
  rdfs:label "hasDirectPart"@en .

# supporting the powerful modelling style

qudt:QuantityValue
  rdfs:subClassOf cmo:AttributeValue .

cmo:hasAttributeValue
  rdf:type owl:ObjectProperty ;
  rdfs:label "hasAttributeValue"@en ;
  rdfs:range cmo:AttributeValue .

cmo:AttributeValue
  rdf:type owl:Class ;
  rdfs:label "AttributeValue"@en .

cmo:QualityValue
  rdf:type owl:Class ;
  rdfs:label "QualityValue"@en ;
  rdfs:subClassOf cmo:AttributeValue ;
  rdfs:subClassOf [
      rdf:type owl:Restriction ;
      owl:cardinality "1"^^xsd:nonNegativeInteger ;
      owl:onProperty cmo:nonNumericalValue ;
    ] .

cmo:nonNumericalValue
  rdf:type owl:DatatypeProperty ;
  rdfs:label "nonNumericalValue"@en .
```

```
cmo:stringValue
  rdf:type owl:DatatypeProperty ;
  rdfs:label "stringValue"@en ;
  rdfs:range xsd:string ;
  rdfs:subPropertyOf cmo:nonNumericalValue .

cmo:booleanValue
  rdf:type owl:DatatypeProperty ;
  rdfs:label "booleanValue"@en ;
  rdfs:range xsd:boolean ;
  rdfs:subPropertyOf cmo:nonNumericalValue .

cmo:hasRelationshipObject
  rdf:type owl:ObjectProperty ;
  rdfs:label "hasRelationshipObject"@en ;
  rdfs:range cmo:RelationshipObject .

cmo:RelationshipObject
  rdf:type owl:Class ;
  rdfs:label "RelationshipObject"@en ;
  rdfs:subClassOf [
      rdf:type owl:Restriction ;
      owl:cardinality "1"^^xsd:nonNegativeInteger ;
      owl:onProperty cmo:objectReference ;
    ] .

cmo:objectReference
  rdf:type owl:ObjectProperty ;
  rdfs:label "objectReference"@en .

cmo:referredBy
  rdf:type owl:ObjectProperty ;
  rdfs:label "referredBy"@en ;
  rdfs:range cmo:RelationshipObject ;
  owl:inverseOf cmo:objectReference .

cmo:hasDirectPartObject
  rdf:type owl:ObjectProperty ;
  rdfs:label "hasDirectPartObject"@en ;
  rdfs:range cmo:DirectPartObject ;
  rdfs:subPropertyOf cmo:hasRelationshipObject .

cmo:DirectPartObject
  rdf:type owl:Class ;
  rdfs:label "DirectPartObject"@en ;
  rdfs:subClassOf cmo:RelationshipObject .

cmo:isDirectPartOf
  rdf:type owl:ObjectProperty ;
  rdfs:label "isDirectPartOf"@en ;
  owl:inverseOf cmo:hasDirectPart .
```

Note:

As soon as the NASA/TQ QUDT 2.0 resources become ‘dereferenceable’ (planned for year 2017) the imports clauses for them can and will be deleted.

Appendix 4: Linking Background Information

Separate MS PowerPoint presentation:

- Appendix4-INTERLINK-D4-LinkingBackgroundInformation.pptx