European Commission

DBL

DIGITAL BUILDING LOGBOOK

DBL Semantic Data Model

Providing standard form and meaning to digital building logbook data.

Authors: Michel Böhms (TNO), Ed.

Martin van der Ende, Michael Flickenschild (Ecorys)

Tom Borst, Niko Raes, Amy Cai, Yvon Gankema,

Robbin Schinkel (Arcadis)

23. May 2023 (Final)

DG GROW

Contents

List of acronyms and definitions

Term	Definition
API	Application Programming Interface. Interfacing software that allows programmers to access existing software functionalities or data sources without needing to know their internal details (like data formats). Especially relevant are web-based API's (or Web API's) like RESTful API's over HTTP having no memory. REST stands for representational state transfer.
CEN	European Committee for Standardization.
CityGML	A standard to store and exchange 3D city models with semantics in the GIS domain. Source: Open Geospatial Consortium (OGC).
DID	Decentralised Identifiers (DIDs) v1.0, Core architecture, data model, and representations, W3C Recommendation 19 July 2022.
Digital Twin	Digital Twins are meant to bridge the gap between real-world physical objects and their virtual/digital representations. They typically comprise of data and software functionalities having a direct link to reality via sensors and sometimes even actuators. There are many kinds of digital twins from simple "stand-alone" via "predictive" to fully "autonomous" variants. For any kind of digital twin, the right data, including semantic alpha-numeric data, geometric data and linked documents, is essential.
DBL	Digital building logbook, "a common repository for all relevant building data. It facilitates transparency, trust, informed decision making and information sharing within the construction sector, among building owners and occupants, financial institutions and public authorities." Source: Study on the development of a European Union framework for digital building logbooks. A DBL can be seen as a simple "standalone" Digital Twin for buildings focussing on the data aspects.
DCAT	Data CATalog vocabulary recommendation by W3C. A catalogue is a collection of meta-data about data sets and data services.
DMS	Data Management System. In the building sector also often referred to as a Common Data Environment (CDE). There is a large variety of DMS's. Some are document-based, some are more semantic. Some treat data sets only as a whole, some can go into these data sets, etc. A well-known open-source DMS is CKAN powering data hubs/portals worldwide especially in the governmental sector for public data.
DPP	Digital product passports (DPP) aim to gather data on a product and its supply chain and share it across entire value chains so all actors, including consumers, have a better understanding of the materials and products they use and their embodied environmental impact.
GIS	Geographic Information System.

GML	Geography Markup Language – "GML is an XML grammar for expressing geographical features." Source: https://www.ogc.org/standards/gml.
IFC	Industry Foundation classes – "IFC is a standardized, digital description for the built asset industry." Source: buildingSMART International (bSI).
JSON	JavaScript Object Notation – "JSON is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate." Source: https://www.json.org/json-en.html.
JSON-LD	A web-based specialisation of JSON (IDs become URI's). Also used for serialising linked data next to Turtle. Source: https://www.w3.org/TR/json-ld11/.
Master data or reference data	The project-independent (internal") data facts typically in the context of a specific organisation.
Ontology	A semantic data model in the W3C linked data/semantic web context. It defines the concepts, attributes, datatypes, relations and constraints/rules relevant for the actual (master/reference or project) data. Based on source: https://www.w3.org/standards/semanticweb/ontology.
OGC	Open Geospatial Consortium.
OWL	Web Ontology Language – "The W3C Web Ontology Language (OWL) is a Semantic Web language designed to represent rich and complex knowledge about things, groups of things, and relations between things." Source: https://www.w3.org/OWL/.
QL	A computer language for querying (and updating) databases. Can also be seen as a "flexible API". Examples are Structured Query Language (SQL) for relational databases and SPARQL for semantic databases.
RDF	Resource Description Framework by the World Wide Web Consortium (W3C). De most basic language underpinning the W3C Linked Data/Semantic Web approach. It is based on "triples" forming the atoms of the data.
Semantic data models	Semantic data models or information models are data specifications that define meaning for an information set. In linked data technology such a model is called an ontology.
SKOS	Simple Knowledge Organisation System.
SML	Semantic Modelling and Linking standard, Part 1 (generic) and Part 2 (domain-specific) by CEN TC442 WG4 (support data dictionaries) ([11], [12]).
SPARQL	SPARQL Protocol and RDF Query Language.
STEP-technology	The technology behind the ISO 10303 - Standard for the Exchange of Product model data (STEP) standard like the EXPRESS language for defining data models and STEP Physical File Format (SPFF) as data format.
TC442	CEN Technical Committee on Building Information Modelling (BIM).

Turtle	Terse RDF Triple Language is an RDF format or as they call it, an RDF serialisation. A Turtle document allows writing down an RDF graph in a compact textual form. Equivalent formats include RDF-XML and JSON-LD. Turtle is more user-friendly.
UOI	An object may have many identifiers with unique values for various purposes/views. One of them is a Unique Object Identifiers which can be used to link all others. Often a code assigned by a national registry is used as UOI for buildings although the UOI of parts of buildings could come from other sources like international product codes like UPC (bar codes) and EPC (RFID).
URI	Uniform Resource Identifier, which is a unique sequence of characters that identifies a logical or physical resource used by web technologies.
UUID	Universally Unique IDentifier. A generated globally unique ID (GUID) according to certain rules. Their uniqueness does not depend on a central registration authority or coordination between the parties generating them, unlike most other numbering schemes.
W3C	World Wide Web Consortium: the international community that develops open standards for the web.

1. Introduction

In the earlier EU study on a European framework for building logbooks [1] it was stated:

"According to the experts, the main purpose of an EU harmonisation or standardisation process for a digital building logbook should be to establish a semantic data model of the core digital building logbook elements".

This deliverable defines such a Digital Building Logbook (DBL) Semantic Data Model and its underlying principles and technologies. This semantic data model is proposed as a European 'core model' for EU Member States and related data providers. It provides agreed common meaning and terminology for the various relevant logbook data sets.

The deliverable is structured as follows:

- A short introduction to the use case type of **data exchange and sharing** between building data web portals.
- A description of a **standard global data architecture** identifying all the data aspects that have to be taken care of.
- The **basic design principles** for our proposed DBL Semantic Data Model.
- The **technological choices** made concerning all data aspects facilitating or even enabling these principles.
- The **key state-of-the-art developments** that our DBL approach should be aligned with.
- Finally, we propose a **first version of a core DBL Semantic Data Model** that can be used as a European DBL and reused and adapted/extended by Member States or even building owners and construction professionals following the same principles and utilising the same technologies.

We reused as much as possible existing semantics from the existing European INSPIRE initiative, especially that part dealing with buildings, building units, cadastral parcels and addresses ([8], [9] and [10]).

The DBL concept is to be carefully positioned towards strongly related concepts like building passports, building information models, geo-spatial information models, building material passports and digital product passports. The latter especially for construction-related products. The DBL view on this positioning is depicted in Figure 1.

In here, the DBL is seen as a global entrance to the typically semantic and distributed data and links to data sets as a whole. Other, typically more detailed, data sets for BIM, GIS/GEO, BMP and DPPs would directly or indirectly provide data to the DBL and/or can be referenced by the DBL.

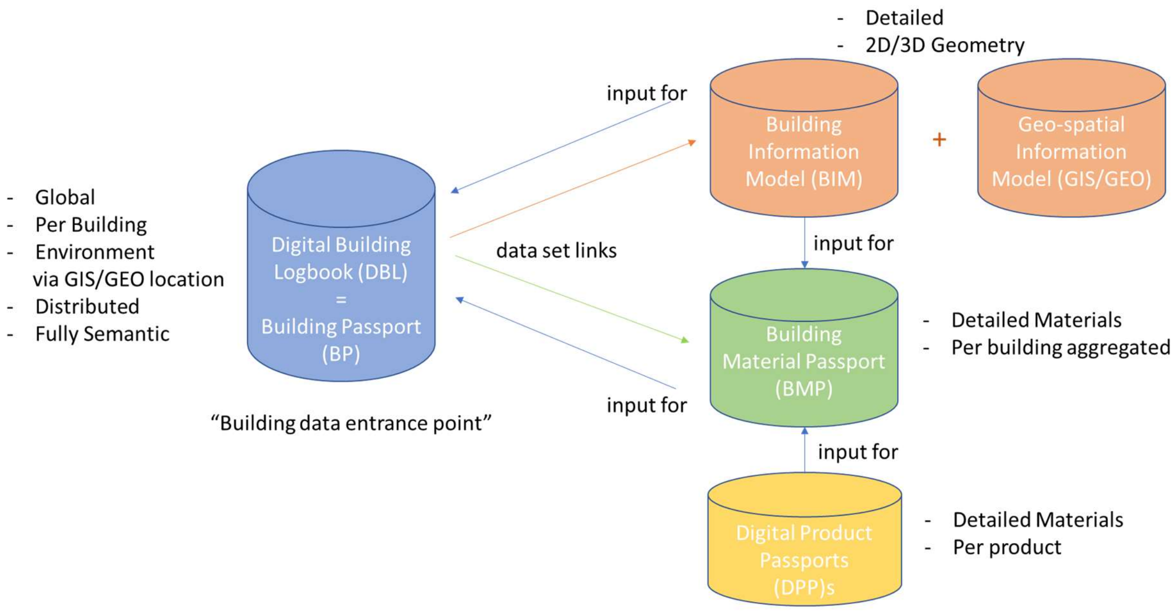

Figure 1: The context of DBL

2. Use Case Types

The generic use case types underlying the development of the European framework for digital building logbooks are the data exchange and/or data sharing and the subsequent integration of building-related data between different stakeholder types related to different geographic levels:

- The EC at the EU level;
- The national agencies at the Member State level; and
- The actual data providers like building owners and construction professionals.

All three stakeholder types have bilateral communication links that can greatly benefit from agreement on all data aspects: technically, syntactically and moreover semantically.

Data exchange refers to the process of sending whole data sets from sender to receiver. In general, this approach is fine as long as the data provenance is clear. Especially in case of a change of data ownership, this is often the way to go. In other situations, however, where there is no change of ownership, this process would result in multiple copies of the same data often leading to data getting out of sync because of uncontrolled changes.

In this scenario, data sharing is preferred: describing data once and utilising it in many ways. Data sets are now not communicated via import/export (in a web situation: upload/download) but directly accessed/published, typically at some web location via some direct access method. In practice today we observe many mixed forms where data is exchanged and not touched at the target side (i.e. read-only) but updated in appropriate time intervals with obvious drawbacks on actuality.

Without going into too much detail here (as this will be done in other deliverables) a typical scenario is shown in Figure 2 where a national portal would provide or publish its DBL data towards a European portal. Several 'modes of interaction' are relevant here typically needing data transformers/enrichers to provide existing data sets in the right format (translated), according to the right semantic data model (converted) and annotated in an agreed way.

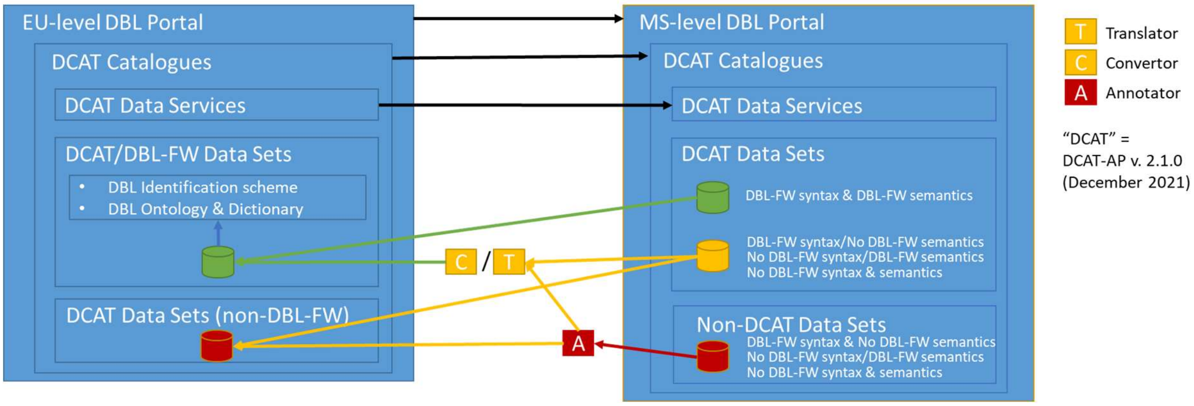

Figure 2: Modes of interaction between building data portals

Data Catalog Vocabulary (DCAT) [6] is an important existing W3C linked data specification regarding providing meta-data about the data sets without going into the content of those data sets. It greatly enhances findability and access. Our proposed DBL framework will go further than that, providing interoperability and reusability of the content in those data sets.

In future, we will always have a hybrid situation of framework-compliant data and non-compliant data, not even DCAT-compliant data, involving data transformations to cope with this hybrid situation.

Figure 2 shows communication between EC and Member States but is equally relevant between Member State agencies and the actual data providers. It is actually relevant for all stages in the whole data supply chain.

Any scenario for this use case type and any mode of interaction requires agreement on various data aspects. These aspects will be introduced in the next chapter based on standard data architecture.

3. Data Architecture

ISO 8000 Part 110 ('Master data — Exchange of characteristic data: Syntax, semantic encoding, and conformance to data specification') defines a data architecture for, as they call it, the 'exchange of data without loss of content and meaning'. This data architecture is extended to make it complete for the purpose of DBL development (see the extensions coloured in red) in Figure 3.

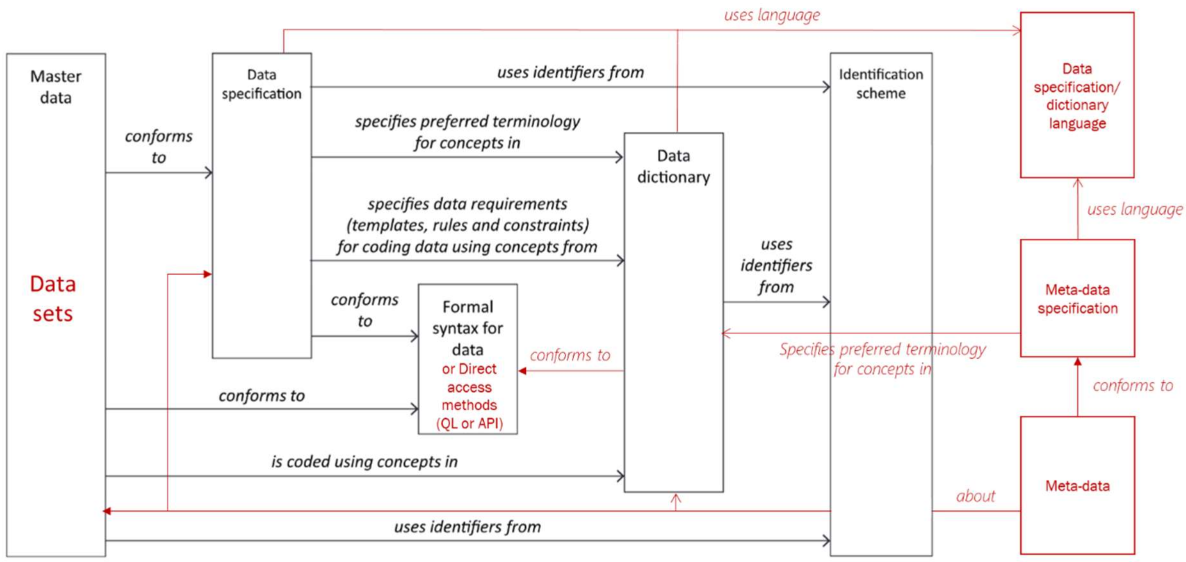

Figure 3: ISO 8000 Data Architecture extended in red

Master data is a kind of reference/project-independent data bound to a specific stakeholder. We apply the architecture here to 'data in general', so even, data sets relevant for multiple stakeholders in the building sector, covering both internal master data and internal/external project data. Data sets can refer to data resources in any form like databases and data files.

For any set of (master) data we have to specify:

- A *data specification* that defines all the possibilities and/or impossibilities for the content in the data sets. This way it defines the meaning or 'semantics' of the data. Such specification typically takes the form of a graphical diagram or a lexical schema or ontology. This specification typically defines concepts/classes, attributes, datatypes, relations and constraints/rules that should hold for the data. Say otherwise: a data specification is used to correctly interpret the data. The backbone of any ontology is often formed by two hierarchies of concepts ('classes'): a specialization hierarchy (also known as a 'taxonomy') and a typical decomposition hierarchy (also known as a 'meronomy'). The meronomy should not be confused with an actual decomposition of the instances of concepts/classes, often called a System Breakdown Structure (SBS) or "object tree".

- A *formal syntax* or as we refer to it, a 'format' or in 'linked data'-speak, a 'serialisation'. The format can be abstracted by defining a functional direct access method like a (low-level, schema/ontology-agnostic) Application Programming Interface (API) or more flexibly, a Query Language (QL). Often predefined queries in some QL can be grouped into a more static 'snapshot' API that can be regarded as a schema/ontology-aware high-level API as alternative for the schema/ontology itself. The same content of a data set in a different format is often referred to as a 'distribution'.

- A *data dictionary* that defines all the terms used as names or labels for the concepts, attributes and relations used in the data specification. Often a data specification and a data dictionary are integrated into one specification. Dictionaries can be multi-lingual and also indicate synonyms and homonyms for terms. For DBL we keep the data specification (DBL Ontology) and the data dictionary (DBL Dictionary) separate but interrelated in a standard way.

- An *identification scheme* is needed for all elements in the master data, data specifications and data dictionaries. The actual IDs following this scheme can be distributed over the various stakeholders/geographic levels. IDs can take the form of meaningful names (for humans) or meaningless codes like the generated Universally Unique IDs (UUIDs). UUIDs are unique by themselves but the other way round, the buildings, their parts or aggregates can still have multiple UUIDs. One of them that is unique regarding the asset/part/aggregate is often referred to as a Unique Object Identifier (UOI).

- A *data specification/dictionary language* is needed to provide the language constructs to define a data (or meta-data) specification and/or a data (or meta-data) dictionary. Typically, such a language contains elements like 'term' and 'definition' (for the dictionary) and 'concept', 'attribute', 'datatype', 'relation' and 'constraint' (for the specification).

- *Meta-data*, or 'data about data', includes data about data sets (or their elements) or data about their data specifications/dictionaries (or their elements). This meta-data, like the 'normal' data, also needs a data specification potentially referring to a data dictionary, in this case, a *Meta-data specification*. For DBL, the meta-data for data sets as a whole will be based on the European profile of the W3C DCAT standard as already utilized at 'data.europa.eu' involving the open source Data Management System (DMS) called CKAN. CKAN assets themselves are held in trust by the Open Knowledge Foundation (https://okfn.org/), a non-profit organization, with best practice policies on governing openly and for use of the trademark.

For DBL the situation is as follows:

Geographic level/ Stakeholder	Formats & Direct access methods	Identification scheme	Language	DBL Semantic Data Model (ontology & multi-lingual Data Dictionary (incl. DCAT 'meta-data' specs)	DBLs (content incl. IDs and meta-data)
EU/EC	like JSON-LD & SPARQL	like Unique Object Identifiers(UOIs) involving UUIDs	like RDF, SKOS, RDFS, OWL, SHACL	Key result of this DBL project	asserted/inferred by EC
Member States/ National agencies				reused by agencies	asserted/inferred by agencies
Data Providers (building owners & construction professionals)				reused by data providers	asserted/inferred by data providers

Figure 4: Relating data aspects to stakeholders (types) & geographic levels

Our DBL framework will be defined on the EU/EC level, but the semantic data model part can be reused, that is selected, extended, and adapted, by the Member States on a national level or by building owners/construction professionals on an asset level.

As a consequence, the DBL data follows the same pattern: EU-level data (asserted or inferred) following the DBL data model; national-level data following the same reused semantic data model; and finally building owner/professional level data following this further extended data model. Note that the EU-defined DBL semantic data model is typically populated by the other two levels via the provision of their data.

4. Guiding principles

Simplicity

The first principle for making any data endeavour a success is simplicity. Many initiatives in the past have failed because they were too complex. People did not understand it or thought they understood but interpreted it in different ways. Better be able to do the simple things first then go for complex solutions that hamper the solution of the simple issues.

FAIRness

The key guiding principle is FAIRness (https://www.go-fair.org/) for digital assets, where FAIR stands for Findable, Accessible, Interoperable and Reusable (in that order). GO FAIR is a bottom-up, stakeholder-driven and self-governed initiative that aims to implement the FAIR data principles, making data Findable, Accessible, Interoperable and Reusable (FAIR).

First data should be **findable**, using some data identification scheme. When found, it should be **accessible** via the right user Identification, Authentication and Authorisation (IAA). When accessible the data should be readable/processible via agreed data formats or data access mechanisms and thereby also **interoperable**. Finally, the data should be **reusable** by being well-defined by data specifications/dictionaries denoting the data's intended meaning.

For the DBL we assume:

- Data is findable (in the cloud)
- Data is accessibility (in the cloud).

    - Many initiatives are currently discussed to make data more secure and reliable like European Data Spaces (EDS), iSHARE, etc. For open/free data this might be less relevant.

- Data is at least partially computer-processible and/or computer-interpretable (we want to go beyond 'data sets as a whole only') so that they can in principle be translated resp. converted to some agreed syntax resp. semantics.

Portals that make data findable and accessible are often referred to as Data Access Points. They do not care about the format and structure of the data but treat data sets on the meta-level involving meta-data. The European data portal (data.europa.eu) is a typical example. Our DBL will also be on that level but then go beyond by formatting and structuring some of the key data in the content of the data sets explicitly. In other words, making those selected parts also interoperable and reusable.

Remark on Data Quality

Reusability in FAIR focusses on data being well-defined (by specifications/information models like schema or ontologies). Surely there are many more indicators for reusability like the many data quality aspects that typically depend on the purpose in some contexts like:

- Relevance (usefulness, timeliness)
- Completeness
- Consistency
- Precision
- Simplicity
- Traceability
- Scalability
- Adaptability (changeability, extendibility)

It is however not the scope of this project to address those. In this study, we focus on the basic requirements for making the data FAIR.

Levels of Information Need (LOIN)

LOIN (as defined by the European norm EN 17412-1) distinguishes three levels of information needs:

- Alphanumerical data sets, aka semantic data (think parameters/variables with values).
- Geometric data sets, describing explicit location/shape representations
- Documents, with unstructured information content

All three levels are relevant and can be treated at the 'meta-level' described by meta-data (data about data sets) or we can go into their content assuming some predefined data structure. In general, the trend is indicated in the next figure where more and more documents and also geometries are replaced with machine-processible and interpretable data.

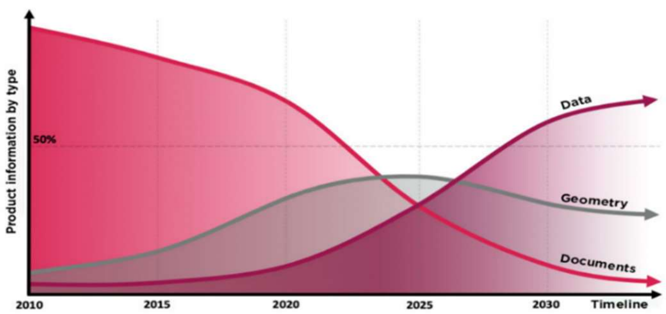

Figure 5: Documents and geometry replaced by (semantic) data

For DBL we will address all three levels of information need with a mix of meta-level and content-level. For example, the 'footprint' of a building in M2 will be content-level, where a link to a BIM model will be meta-level (just give a link to the BIM model as a whole, not going inside it).

Keeping Data at its Source

We distinguish two main ways to communicate data:

- By data exchange (sometimes referred to as data transfer), and
- By data sharing.

The second option is preferred if no ownership of the data is changed. This way avoids multiple copies potentially getting out of sync and keeps the data at the location where it is consistently updated by the humans or machines/software who have the right to do so. Only when ownership is changed, like in a supply-chain where data goes from contractor to client, data transfer is the more logical form of communication, after which it can be shared again.

Data sharing typically asks for a distributed/federated solution whether data is not copied to a central place but is made available it.

Although not in the scope of LOIN (that focusses on data) the same is true about software functionalities like web data services that perform actions on the data. Here it is even more logical to keep the services at the source. So existing portals publishing catalogues of relevant data sets and data services should best not upload those centrally but make them remotely accessible by data/service sharing.

5. Data Technology

There are many existing data technologies around, all with their own history and typical use cases. A quick overview without further deep analysis or comparison:

- ISO/IEC 9075-1:2016 Relational technology by ISO/IEC
    - Large subset implemented by Open Source Software PostgreSQL, popular in the GEO-world
- ISO 10303 STEP technology by ISO
    - STEP Physical File Format (SPFF)
    - EXPRESS language
    - Used as primary technology for bSI IFC, the international Open BIM initiative
- Model-driven technology by OMG
    - (graphical) Unified Modelling Language (UML)
    - Non-graphical (lexical) representations involving the so-called Meta Object Facility (MOF)
- Internet technology by IETF
    - HyperText Transfer Protocol (HTTP)
    - RESTful APIs (over HTTP)
- Web technology by W3C
    - URI/URL for identification & location

- HyperText Markup Language (HTML), actually a textual format (for documents)
- CSS for rendering HTML to an end-user

- W3C XML technology by W3C
    - eXtensible Markup Language (XML), actually a 'format' in our terminology
    - XML Schema Definition (XSD) language (for structured data)
- WebDev technology by various
    - JSON format, GraphQL direct access mechanism (graph query language)
    - JSON Schema language, GraphQL Schema language
    - OpenAPI's (Docker, Swagger, …)
- W3C Linked Data/Semantic Web (LD/SW) technology by W3C
    - Basic framework: Resource Description Framework (RDF)
    - RDF-serialisations as format (RDF-XML, Turtle, JSON-LD)x
    - Languages (RDFS, OWL, SHACL (Core and Advanced Features), SKOS)
    - SPARQL Query Language

The key technology chosen for DBL is the W3C Linked Data/Semantic Web technology. This technology has the needed power to define our DBL terminology and semantics and is well-defined in math, logic and graph theory and also well-maintained at the World Wide Web Consortium (W3C). It is a logical extension of the normal document/human-centred world wide web as we know it but now for structured, machine-readable and machine-interpretable data. Moreover, it fully supports our FAIR data principle.

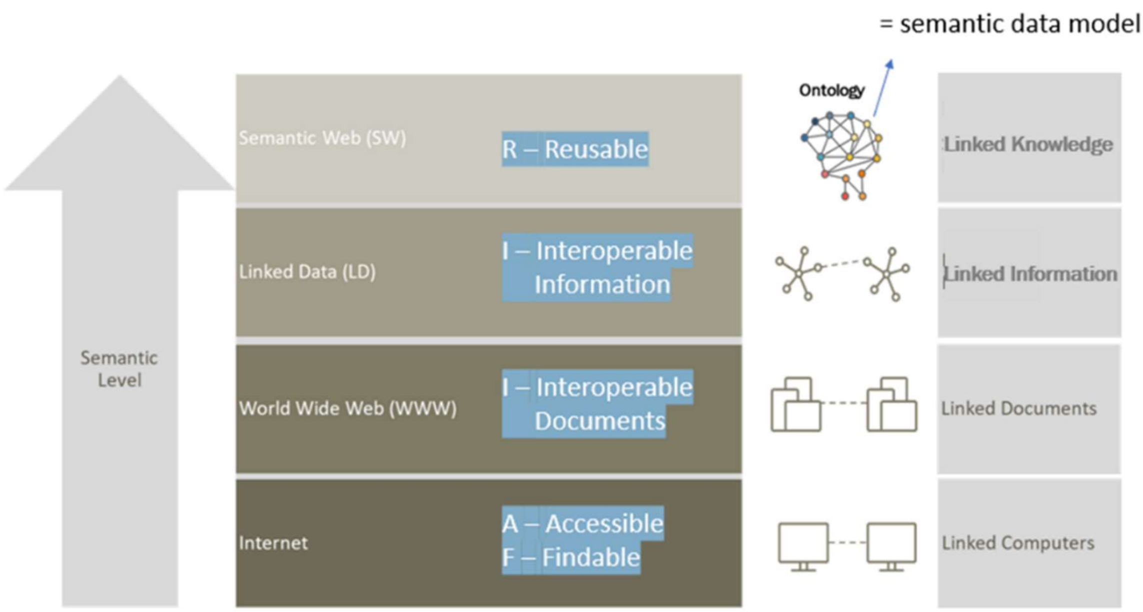

Figure 6: Key enabler for FAIRness: W3C Linked Data/Semantic Web technology

First of all, this technology is fully Internet-based. So findable becomes 'findable in the cloud'. For open data, it is directly accessible. For more restricted data, there are many technical developments underway like GAIA-X, IDS and iSHARE, as generic trust framework capabilities for federative data sharing and data spaces.

Furthermore, the WWW itself, as an application on top of the Internet, already provides standard formatted documents for human consumption via standards like HTML, CSS and JavaScript. Where the internet links computers (more precise computer networks), the WWW links documents together. The LD/SW is adding two layers on top of that: first, the linked data provides standards formats to make this data syntactically interoperable, next the semantic web adds data specifications in the form of ontologies and dictionaries providing semantic interoperability or as FAIR refers to it: reusability.

Linked Data

The key basic linked data language (or better 'framework') is Resource Description Framework (RDF) ([13]). RDF captures the idea that all data is built from atomic triples of the form:

"Subject – Predicate – Object" (in that order)

Sets of these interrelated triples form the semantic graphs of data; both master data and their data specifications in the form of ontologies, and data dictionaries.

LD/SW provides various equivalent options for both RDF serialisations (the term used for formats in the context of LD/SW) and languages to define ontologies (the term used for information models in the context of LD/SW). LD/SW RDF serialisations:

- RDF-XML/TriX (a specialisation of XML)
- Turtle/TriG
- JSON-LD (a specialisation of JSON involving URIs for the identification or said simply: making JSON web-aware)

TriX and TriG are just multiple-graph variants of RDF-XML respectively Turtle.

For DBL we formally select Turtle and JSON-LD. Primarily we use the more user-friendly Turtle and generate autaomatically JSON-LS without any loss. JSON-LD is a specialisation of JSON in general and therefore 100% compatible with this format that is ubiquitously used in modern web-based software developments supported by all major programming languages like C, C++, C#, Java, JavaScript, Perl and Python. JSON-LD is essentially JSON with some special keywords (@graph, @context, @id and @type).

JSON-LD is more and more becoming the choice of modern data initiatives. It is also chosen by W3C Distributed IDs (DID) which in its turn is chosen by the SOLID project (https://solidproject.org/) working on the next generation of the fully distributed read-write semantic web. Finally, people can use GraphQL as a simpler alternative to SPARQL if needed.

For the examples in this specification, however, the more compact and readable Turtle serialisation is used which can be directly translated into the more verbose JSON-LD.

Example

Turtle:

```
ex:Building_1
  rdf:type dbl:Building ;
  dbl:heightAboveGround [
      qudt:numericValue 5.3 ;
      qudt:unit unit:M ;
    ] .
```

JSON-LD:

```
{
  "@graph" : [ {
    "@id" : "_:b0",
```

```
      "numericValue" : "5.3",
      "qudt:unit" : {
        "@id" : "unit:M"
      }
    }, {
      "@id" : "ex:Building_1",
      "@type" : "dbl:Building",
      "heightAboveGroundt" : "_:b0"
    } ],
    "@context" : {
      "heightAboveGround" : {
        "@id" : "https://data.europa.eu/dbl/def#HeightAboveGround",
        "@type" : "@id"
      },
      "numericValue" : {
        "@id" : "http://qudt.org/schema/qudt/numericValue",
        "@type" : "http://www.w3.org/2001/XMLSchema#decimal"
      },
      "unit" : "http://qudt.org/vocab/unit/",
      "qudt" : "http://qudt.org/schema/qudt/",
      "dbl" : "https://data.europa.eu/dbl/def#",
      "ex" : "https://data.europa.eu/dbl/example1/id#",
      "rdf" : "http://www.w3.org/1999/02/22-rdf-syntax-ns#",
    }
  }
```


Semantic Web

For giving meaning to the (linked) data we need data specifications and dictionaries expressed in some data modelling language. The semantic web provides us with several language options:

- RDF Schema (RDFS) for defining simple ontologies.
- OWL on top of RDFS adds open-world constraints suited for logical- data inference.
- SHACL on top of RDFS adds closed-world, more powerful constraints, suited for data verification.
- SHACL-AF (Advanced Features) is not yet final and covers, among others, data derivation rules via SPARQL Construct queries.
- SKOS, actually an instantiation of OWL, for modelling multi-lingual dictionaries of terms with their human-readable labels, definitions, examples and interrelationships.

For DBL we select the following options:

- RDFS for the DBL Ontology; where needed to be extended with SHACL for closed-world constraint modelling, and
- SKOS for the DBL Dictionary.

It is always possible for Member States to apply more complex languages to cover their more complex country-specific information modelling needs. For EU/EC-wide common DBL Ontology/Dictionary, RDFS and SKOS are good enough since they have the capabilities to define common (RDFS) classes and (RDF) properties and datatypes (representing attributes or relations).

LD/SW direct access mechanism

The direct access mechanism from LD/SW is the query language SPARQL ([14]). SPARQL is RDF-based so it supports all languages, ontologies and data sets that are defined in it. By using SPARQL the actual serialisation becomes agnostic: SPARQL does not have to know the actual serialisation. SPARQL can also directly access a relevant fragment of an ontology/data set instead of needing to download/upload a total model.

For people who think that SPARQL is too complex for them (inherently since it supports a graph view on data), there are software tools around to swap it for GraphQL (not graph-based but tree-based). Several approaches exist here: a GraphQL frontend to SPARQL assuming JSON-LD serialisation or GraphQL Schema generation from LD/SW language-based ontologies which can then be queried (next to the data sets) by GraphQL no matter its type of serialisation.

Flexible SPARQL or GraphQL access can be provided over the web using tools like Postman or Yasgui that translate the data location, the actual query and optional basic credentials into HTTP REST points get/post calls. Simplification can be obtained by just predetermining a set of static (often useful) queries (as API) communicating only the input and outputs over HTTP.

All these technological choices can be positioned in the earlier Data Architecture (Figure 7).

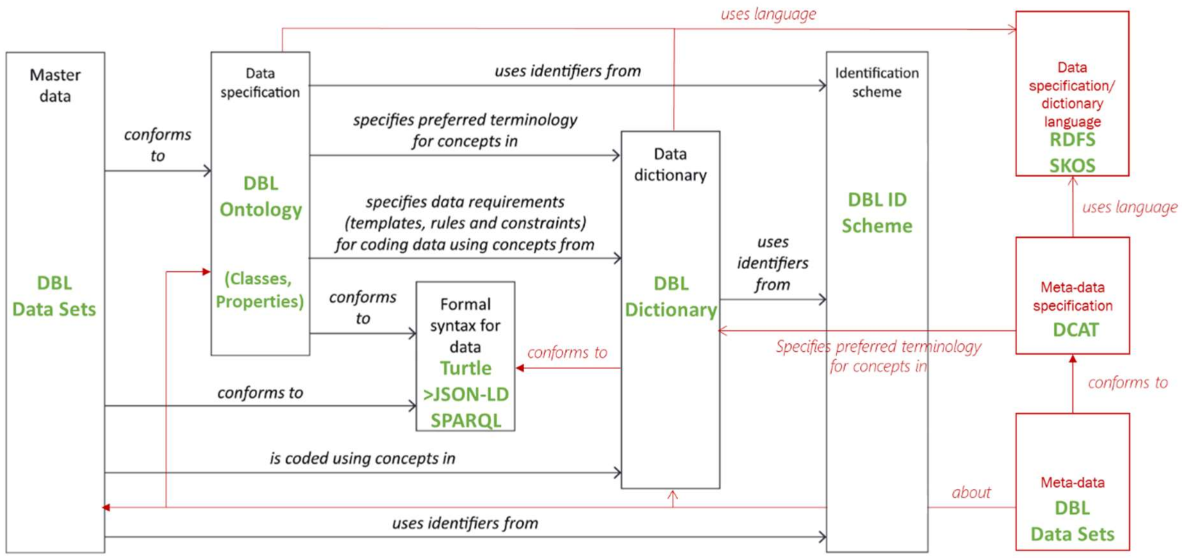

Figure 7: The data architecture filled in the linked data way

6. State-of-the-Art

New European Interoperability Framework (EIF)

The (new) EIF ([2]) is meant to be a generic framework applicable to all public administrations in the EU. It lays out the basic conditions for achieving interoperability, acting as the common denominator for relevant initiatives at all levels including European, national, regional and local, embracing public administrations, citizens and businesses. This document is addressed to all those involved in defining, designing, developing and delivering European public services. The EIF defines recommendations for six interoperability layers:

1. Interoperability governance
2. Integrated public service governance
3. Legal interoperability
4. Organisational interoperability

5. Semantic interoperability
6. Technical interoperability

Our DBL Framework assumes technical interoperability (web-based availability & secure access) and focusses on semantic interoperability. On this Semantic interoperability, it is said:

"it ensures that the precise format and meaning of exchanged data and information is preserved and understood throughout exchanges between parties, in other words, 'what is sent is what is understood'. In the EIF, semantic interoperability covers both semantic and syntactic aspects:

- The semantic aspect refers to the meaning of data elements and the relationship between them. It includes developing vocabularies and schemata to describe data exchanges, and ensures that data elements are understood in the same way by all communicating parties;
- The syntactic aspect refers to describing the exact format of the information to be exchanged in terms of grammar and format".

Classifying "syntax" under semantic interoperability is a bit special definition but these two aspects are exactly the issues tackled by our European DBL Framework.

EC Regulation 305/2011: Construction Product Regulation (CPR)

Construction assets come in a 'buy' and 'make' form, or, alternatively formulated, as 'construction products' and 'construction works' (Figure 8).

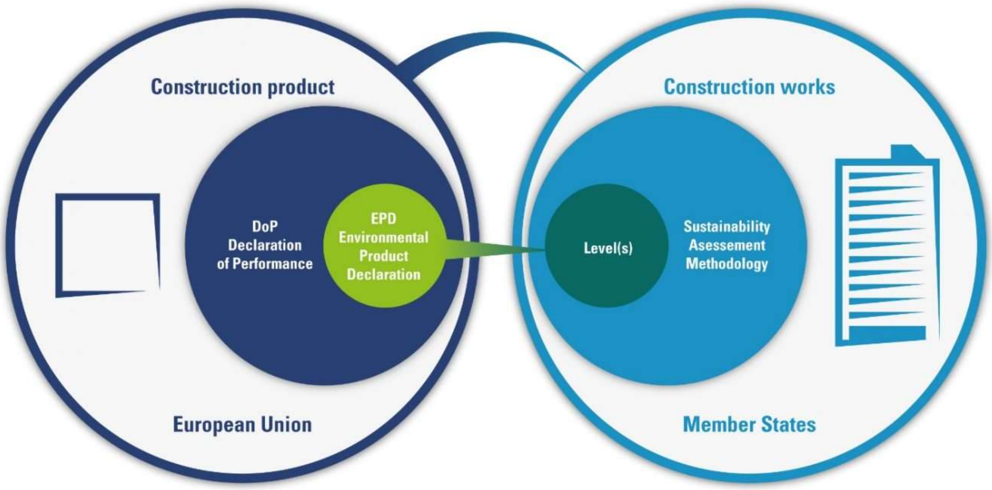

Figure 8: CPR context (source: https://www.construction-products.eu/publications/cpr/)

The CPR was initiated by compatible declarations of performance (DoP) via an Environmental Product Declaration (EPD). CPR can be seen as the "paper" variant of harmonizing DoPs to be formalized by CEN TC442's data templates. This would also create a better link with BIM.

This approach is also referred to as "Smart CE marking". The SML approach followed for DBL ([11], [12]) can be seen as one of the keys to implementing data template standards (bindings for W3C linked data technology).

This construction product area is strongly related to the generic concept of Digital Product Passports (DPPs).

EC Energy Performance of Buildings Directive (EPBD) guidelines

Revised in 2018, the directive will help reach the building and renovation goals set out in the European Green Deal. The EPDB specifies requirements for:

- Building energy performance & documentation,
- Technical system installation, configuration and use (especially their thermostatic control),
- Obligatory examinations, and
- Obligatory building automation and electrical charging equipment.

In 2018 the original EPBD was amended/modernized. In 2022 there is another proposal for an update supporting the "energy renovation wave strategy". In 2018 the role of European energy standards was empowered especially:

- ISO 52000-1 – EPB framework
- ISO 52003-1 – EPB indicators
- ISO 52010-1 – climatic data
- ISO 52016-1 – internal temperature and energy needs
- ISO 52018-1 – thermal balance & fabric

Especially the last 4 contain properties relevant to the energy performance of the building and are thus relevant inputs for the DBL on building performance and the building services realizing that performance.

CEN TC442 ("BIM")

The DBL Ontology will follow all syntactic, semantic and pragmatic recommendations by the European data standard developed in WG4 ("Support Data Dictionaries")/TG3: EN 17632-1:2022 - Semantic Modelling and Linking (SML) in the built environment ([11], [12]). This means that it selects its data languages, data serialisations, direct access methods and modelling patterns for example for IDs from this standard.

Where relevant, other CEN TC442 "data template" standards will be applied such as:

- EN ISO 19650 series on information management using BIM
- EN 17412-1:2020 level of information need (LOIN)
- EN ISO 23386:2020 methodology for properties (involving meta-properties referred to as 'attributes' here)

EU INSPIRE for buildings (https://inspire.ec.europa.eu/)

Inspire stands for "Infrastructure for spatial information in Europe". This infrastructure enables the sharing of environmental spatial information among public sector organisations, facilitates public access to spatial information across Europe and assists in policy-making across boundaries.

Registries in Member States are typically required to communicate their data according to the European INSPIRE syntax and semantics.

The key concepts for our DBL are based on those parts of Inspire dealing with cadastral buildings, building units and cadastral parcels. These concepts are however treated in a much simpler way:

- leaving out many detailed explicit geometry aspects (coming from OGC GML/CityGML);
- leaving out the generic (again CityGML-based) BuildingPart subdivision

and grouping remaining key entities:

INSPIRE BuildingAndBuildingUnitInfo	> dbl:BuildingOrBuildingUnit
INSPIRE AbstractConstruction	> dbl:Building
INSPIRE AbstractBuilding	> dbl:Building
INSPIRE Building	> dbl:Building
INSPIRE BuildingInfo	> dbl:Building
INSPIRE AbstractBuildingUnit	> dbl:BuildingUnit
INSPIRE CadastralParcel	> dbl:CadastralParcel
INSPIRE AddressRepresentation	> dbl:Address

Many attributes and relations for these key concepts are directly reused from this INSPIRE data specification again typically applied in a simplified/flattened way and now represented in the modern Linked Data technology.

Sometimes we deviated from the exact INSPIRE terminology for better consistency. Example: INSPIRE always uses 'XValue' pattern for enumeration types except for the property 'CurrentUse'. We decided for the term 'CurrentUseValue'.

Especially the address is treated as 'complex datatype' not having a life-cycle (and identifier) of its own. In linked data, the address can be treated as a 'blank node'. The OGC GML reused by Inspire is reused by DBL via the latest GeoSPARQL implementation.

buildingSmart International (bSI)

A lot of building data relevant to a DBL is created in the design phase. More and more, a Building Information Model (BIM) is defined using BIM authoring software such as Autodesk Revit, Graphisoft ArchiCAD or one of the many other software vendors. Typically these software packages can export their data according to the international Open BIM standard Industry Foundation Classes (IFC) as defined by buildingSmart International (bSI). IFC is a data specification for buildings modelling sites, zones, spaces and building elements with all their details and representing 3D geometry.

The IFC schema is primarily defined in the standard ISO Standard for the Exchange of Product model data (STEP) language EXPRESS, the related data in STEP Physical File Format (SPFF). Provided alternatives include ifcXML in eXtensible Markup Definition (XSD) language and ifcOWL in Web Ontology Language (OWL). The ifcOWL is generated from the EXPRESS schema for backward compatibility, resulting in a very technical, non-efficient ontology. On the positive side, this ifcOWL and corresponding data can be queried using SPARQL making data sharing beyond mere data exchange of SPFF files feasible. Currently, a JSON variant is being developed[1]. This schema and data can now be queried using the JSON-based query language GraphQL.

The IFC model can be extended via concepts and property sets from different sources ("domains" as they call them) that are collected in the buildingSmart Data Dictionary (bSDD). In theory, these extensions have to be modelled by a meta-model defined by ISO 12006-3 (sometimes referred to as IFD-International Framework for Dictionaries, complementing in name IFC). In practice, however, there are ongoing experiments applying much simplified JSON Schema based specifications.

Some people may assume that with BIM in IFC, one has the DBL too. However, for historic reasons, most IFC data is geared towards the actual build phase of a building's life-cycle. At the same time developments are underway to improve this situation. A good example is the IFC for Facility

[1] https://github.com/buildingSMART/ifcJSON

Management (FM) project at bSI covering also maintenance aspects. For now, IFC is seen as just one (albeit important) data source for the DBL. A BIM/IFC model can be linked to by a DBL or some of its content can form the basis for some more explicit data (think lengths, areas, volumes etc.).

Dotbim (https://dotbim.net/)

This is a minimalistic, simple, JSON-based file format for BIM by MIT as a response to the perceived complexity of other Open BIM initiatives like bSI. It supports only one representation (triangularization) and attaches key-value pairs of data to them. Many consider this as “too” simple.

Building Topology Ontology (BOT)

This is a draft report/specification from the W3C Linked Building Data (LBD) community group. It sits in between IFC and dotbim for complexity. It is maintained at: https://w3c-lbd-cg.github.io/bot/. This minimal OWL ontology might still be too detailed for a building logbook but its top-level is of great inspiration.

data.europa.eu

This web site is THE data portal for (currently ~1.5 million) European data sets. Currently, this data is only available in the ‘meta-way’. This data portal is powered by the open-source CKAN data management system (ckan.org) that already fully supports ‘Linked Data’-based meta-data. All data sets are classified and annotated using a standard set of meta-classes/attributes called DCAT Version 2 or more specifically, a DCAT Version 2 application profile currently in version 2.1.0. DCAT is a W3C Recommendation defined in 2020.

DCAT enables a publisher to describe datasets and data services in a catalogue using a standard model and vocabulary that facilitates the consumption and aggregation of metadata from multiple catalogues.

The key definition of DCAT is a Data Set defined as “a collection of data, published or curated by a single source, and available for access or download in one or more formats”.

An actual data set consists of a description according to DCAT and one or more resources that can be local or remote. A web portal that is providing such data sets is called a catalogue and is servicing data sets or services related to those data sets.

The meta-part of our DBL should be 100% in line with this portal. The more explicit part of DBL involving actual data in the data sets could be seen as a semantic extension made available by a gateway service.

NL Basisregistratie Adressen en Gebouwen (BAG)

The cadastral register for (all) addresses and buildings in the Netherlands ([7]) is a good example of building data published. The portal is freely available at bagviewer.kadaster.nl.

Figure 9: The NL BAG site

When searching for a specific address you get info like in figure 5.

- 2D location/representation
- Identifier
- Construction year
- Area in m2
- Purpose (here: residential)
- Status
- And all kinds of meta-data (like the source of the data)

This BAG can be considered as NL-specific DBL. The format for the total view is PDF but internally open standards are applied such as the Dutch “Basisregistratie Grootschalige Topografie“ (BGT) and the more detailed “Basisregistratie Topografie” (BRT) maps. XML-based extracts are also possible for a city or the whole of NL.

The BAG is also experimentally available in Linked Data: https://labs.kadaster.nl/cases/bag-ld. Now all data is available as linked data and can be queried by SPARQL.

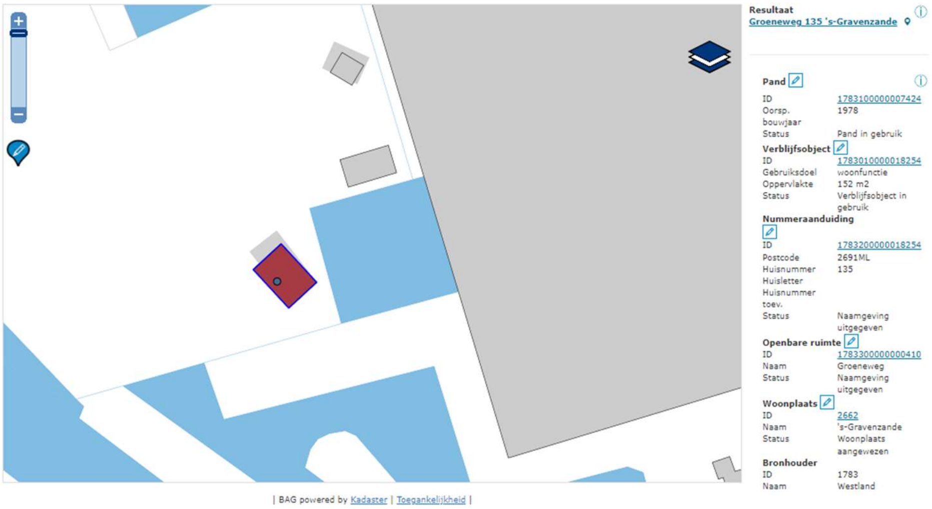

Figure 10: A BAG object example

Also experimentally, a 3D variant for the BAG is available at (https://3dbag.nl/en/viewer) where BAG data is combined with height measurements (the Dutch "Actueel Hoogtebestand Nederland" (AHN3)).

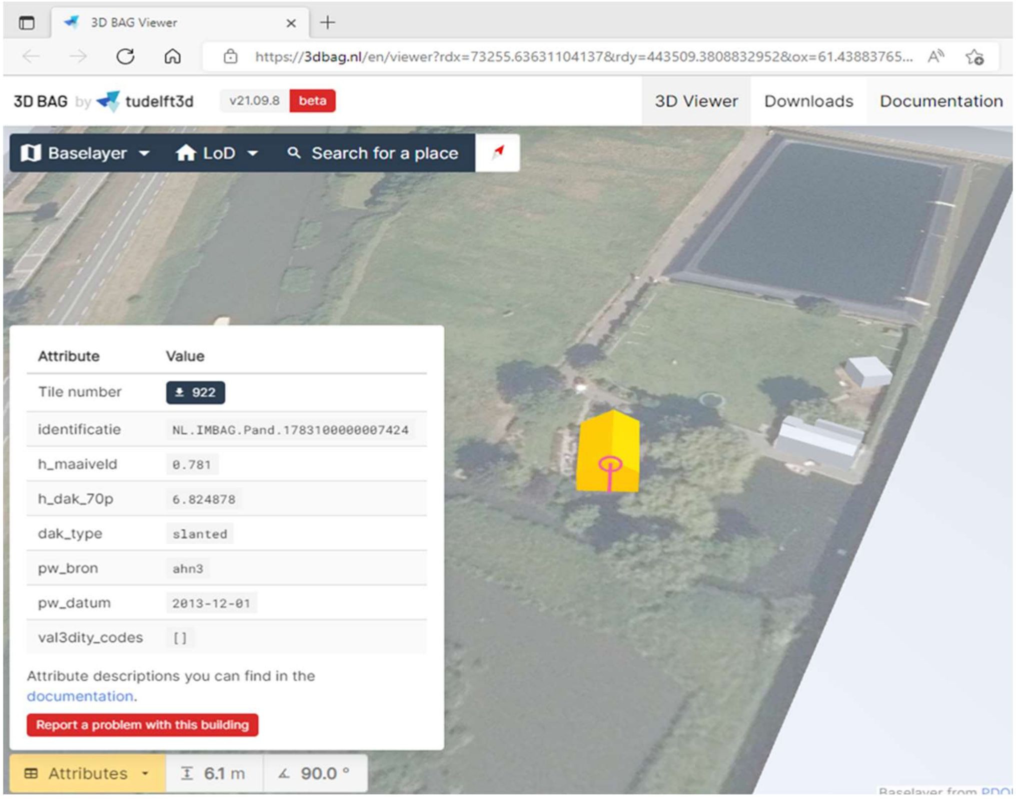

Figure 11: Same object, now in 3D

All this data is typical data for DBL providing links to the service, links to the data sets with meta-data or even links to the actual buildings with explicit attribute values.

Data from the BAG is often combined with other public data sets by commercial web services. A good example is vastgoedloep.nl which is further analyzed as example in report D2.3.

NL Platform ‘Circulair Bouwen’ (CB)’23

This platform connects all relevant initiatives and parties in NL on circular construction via the exchange of knowledge, Identifying and scheduling obstacles and setting up sector-wide agreements. A key example of such an agreement is a guideline for the use of passports in construction:

“CB23 ‘Leidraad Paspoorten voor de bouw”.

Version 3 of this guideline written in June 2022 says in its introduction:

A passport is structured like a pyramid, where the information at a higher level is made up of information from objects of underlying scale levels (e.g. a building that consists of a collection of elements). The levels of scale are 'nestable' so that it transfers the information from its underlying level can take place. For this, it is one primary requirement that the device of the data for material passports comply with NEN 2660-2.

This NEN-2660-2 is the earlier Dutch counterpart of the CEN SML standard as used in this DBL Study.

They also developed a ‘quick start guide’ (https://paspoorten.platformcb23.nl) to derive passport information needs depending on type of asset, level of detail, life cycle phase and passport application area (like ‘circularity’). The output is an Comma Separated Value (CSV) file or Microsoft Excel file with the properties relevant.

7. DBL Identification scheme

For identification, the used SML standard distinguishes between the identification of individuals and the identification of concepts and properties (attributes and relations). The latter case is quite limited and often subject to human interaction. Therefore, we use user-friendly names like “Building” and “energyPerformance” following also the SML guidelines for names. The only drawback is the case of (same-spelled) homonyms which now have to be made explicit (like “Stone” and “RockMusic” instead of just “Rock”).

On both demand (EU) and supply (standardization bodies) sides, there is an enormous interest in electronic IDs (eIDs) as a key ingredient for better cross-border communication. At the European level, there is the eIDAS Regulation and the concept of Self-Souverain Identities (SSIs).

At W3C we see technical solutions for distributed eIDs like WebIDs ([3]) and Distributed IDs (DIDs) ([4]). The latter is now a W3C Recommendation and is also considered at eIDAS ([5]). Both approaches are web-based in the sense that they use URIs patterns. DIDs seem to have the future, not needing a central registry and providing verifiers for both issuers and holders of credentials (Figure 12).

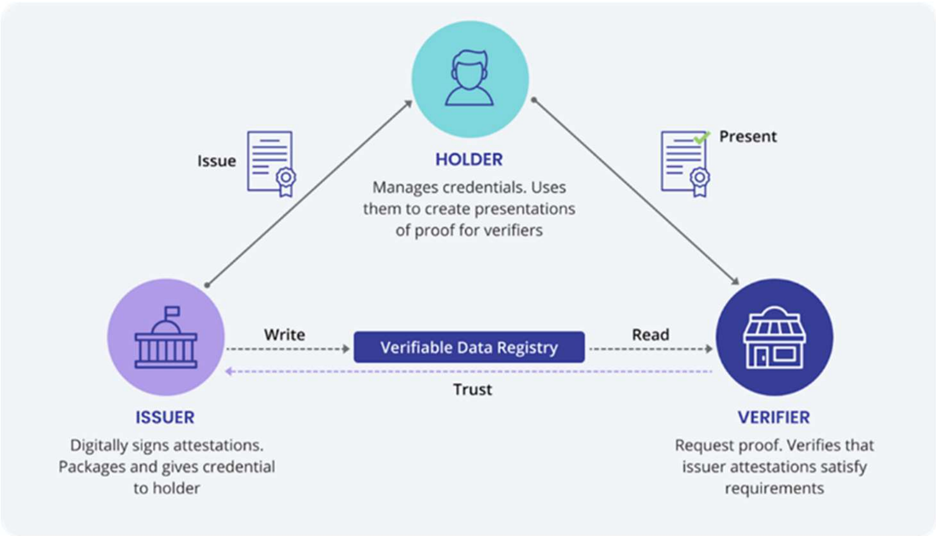

Figure 12: key participants for DIDs

For DBL we stick for now to the CEN SML URI pattern and generated UUIDs. This way we are ready to easily adapt to any stable approach emerging in the coming years.

8. DBL Ontology

Relationship with dictionary data

Since the DBL Ontology uses user-friendly element (concept, attribute, relation) names we do not need SKOS (preferred/alternative) labels and definitions in the ontology (ontologies using <u>codes</u> as names sometimes need these labels and definitions also in the ontology for correct human-friendly rendering to the end-user).

Instead, we refer from the elements via the 'semantically weak' relation called 'rdfs:seeAlso' to the relevant terms for these elements in a separate DBL Dictionary. This dictionary then provides preferred labels and definitions potentially in multiple languages. A simple example is shown in Figure 13.

DBL Ontology

```
dbl:Building a rdfs:Class ;
  rdfs:seeAlso dbl-term:Building .
dbl:footprint a rdf:Property ;
  rdfs:domain dbl:Building ;
  rdfs:range sml:QuantityValue .
```

DBL Data Dictionary

```
dbl-term:Building a skos:Concept ;
  skos:definition "Enclosed constructions above and/or underground
                   which are intended or used for the shelter of humans, animals, things
                   or the production of economic goods and that refer to any structure
                   permanently constructed or erected on its site."@en ;
  skos:definition "Ingesloten constructies boven en/of ondergronds die bedoeld zijn of
                   worden gebruikt voor de beschutting van mensen, dieren, dingen of de
                   productie van economische goederen en die verwijzen naar elke
                   constructie die permanent op haar site wordt gebouwd of opgericht."@nl ;
  skos:prefLabel "Building"@en ;
  skos:prefLabel "Gebouw"@nl ;
  skos:example "Multi-apartment building"@en ;
  skos:example "Multi-appartement gebouw"@nl .
```

DBL Data

```
ex:d6897181-3188-420a-af1d-3fa3bec33f43 a dbl:Building ;
  sml:isDescribedBy [ a sml:InformationObject; rdfs:seeAlso
<https://3dbag.nl/en/viewer?rdx=73255.63631104137&rdy=443509.3808832952&ox=28.464074678500765&oy=39.095544912425325&oz=28.46407467848621> ] .
  dbl:footPrint [qudt:numericValue 320.0 ; qudt:unit unit:M2 ] .
```

Figure 13: Distinguishing and linking the DBL Ontology and DBL Dictionary

DBL Relationship with IFC & BOT

The concepts we introduce for DBL can in general be seen as subclasses (or sometimes even equivalent classes[2]) of IFC/BOT classes (like IfcBuilding/bot:Building). Our DBL Ontology will however be a standalone ontology not importing those. Where needed, links can be introduced in future to flexibly connect them if/when needed (via the standard rdfs:subClassOf or owl:equivalentClass and owl:equivalentProperty relations). This keeps the DBL Ontology as clean and simple as possible.

DBL Property modelling

"Properties" take a centre stage in our DBL approach. In the end all concrete data is in properties defined for buildings, their parts and the parcels they are built on. Remember that a property is a generic term for both attribute and relation. CEN TC442 SML introduces several variants of property modelling. Suppose we have an instance "Building_1" of the concept dbl:Building having a valued attribute dbl:heightAboveGround of 5.36 meters. We can state this fact in the following ways ranging from simple to complex:

For each variant we have the Building concept defined as a class:

```
dbl:Building
  a rdfs:Class .
```

1. SML-1, simple, implicit unit

```
dbl:heightAboveGround
  a rdf:Property ;
  rdfs:domain dbl:Building ;
  rdfs:range xsd:float .

:Building_1
  a dbl:Building ;
  dbl:heightAboveGround "5.36"^^xsd:float .
```

2. SML-1, simple, human interpreted unit in property definition

```
dbl:heightAboveGround_in_m
  a rdf:Property ;
  rdfs:domain dbl:Building ;
  rdfs:range xsd:float .

:Building_1
```

[2] Note that 'equivalence' is the same as 'two-way subclassing'

```
  a dbl:Building ;
  dbl:heightAboveGround_in_m “5.36”^^xsd:float .
```

3. SML-1, simple, human-interpreted unit in value

```
dbl:heightAboveGround
  a rdf:Property ;
  rdfs:domain dbl:Building ;
  rdfs:range xsd:string .

:Building_1
  a dbl:Building ;
  dbl:heightAboveGround “5.36_in_m”^^xsd:string .
```

4. SML-1, simple, unit as datatype

```
unit:M
  a qudt:Unit;
  rdfs:subClassOf rdfs:Datatype .

dbl:heightAboveGround
  a rdf:Property ;
  rdfs:domain dbl:Building ;
  rdfs:range unit:M .

:Building_1
  a dbl:Building ;
  dbl:heightAbovGround “5.36”^^unit:M .
```

5. SML-1, simple, unit as meta-data

```
dbl:heightAboveGround
  a rdf:Property ;
  qudt:unit unit:M ;
  rdfs:domain dbl:Building ;
  rdfs:range xsd:float .
```

Note: when defined in OWL i.s.o. RDFS this would involve "OWL pruning" not necessarily resulting in the "OWL Full" variant.

6. SML-1, complex, explicit unit on instance-level

```
dbl:heightAboveGround
  a rdf:Property ;
  rdfs:domain dbl:Building ;
  rdfs:range sml:QuantityValue .

:Building_1
  a dbl:Building ;
  dbl:heightAboveGround :QuantityValue_1 .

:QuantityValue_1
  a sml:QuantityValue ;
  qudt:numericvalue "5.36"^^xsd:float ;
  qudt:unit unit:M .
```

7. SML-1, complex, explicit unit on type-level

```
dbl:heightAboveGround
  a rdf:Property ;
  rdfs:domain dbl:Building ;
  rdfs:range sml:QuantityValue ;
  qudt:unit unit:M .

:Building_1
  a dbl:Building ;
  dbl:heightAboveGround :QuantityValue_1 .

:QuantityValue_1
  a sml:QuantityValue ;
  qudt:numericvalue "5.36"^^xsd:float .
```

8. SML-2, more complex ('complicated'), involving the W3C SOSA ontology

(not only objectified value but also objectified property definition and property assignment)

```
:Building_1
  a sosa:FeatureOfInterest ;
  :isFeatureOfInterestOf : Observation_1 .

:Observation_1
  a sosa:Observation ;
  sosa:observedProperty :HeightAboveGround ;
  sosa:hasResult :Result_1 .

:HeightAboveGround
  a sosa:ObservableProperty ;
  qudt:hasQuantityKind quantitykind:M .

:Result_1
  a sosa:Result ;
  qudt:numericValue “5.36”^^xsd:float ;
  qudt:unit unit:M .
```

(alternatively, the unit:M can here also be attached at type-level for ‘:HeightAboveGround’)

Note 1: Instead of defining the chosen unit (in various variants) one can also specify the right (QUDT) ‘quantity kind’ having already a set of allowed units. In that case however, the actual chosen unit for the value at hand has still to be specified too.

Note 2: Values like “5.36”^^xsd:float can also be simplified in Turtle as simple 5.36 for floats and “5.36_in_m” for strings.

For the DBL it is decided to go for variant 6: ‘complex properties, units at instance-level’ for encoding all relevant DBL properties (attributes and relations) for the three key concepts: Building, BuildingUnit and CadastralParcel. Only the attributes related to identification where there is no need for state/meta-modelling and all meta-properties (like official area references) are modelled the simple way as in variant 5. Finally, all attributes for helper concepts like for ‘Address’ are also modelled the simple way. The imported items from other ontologies come as they are.

This approach is considered to be the right ‘compromise’ between functionality and simplicity for our DBL use case types.

Modelling and Linking documents and/or non-LD data sets

SML defines a sml:Object that distinguished between two subtypes:

- sml:PhysicalObject
- sml:InformationObject

A generic sml:isDescribedBy relation links the two.

The W3C DCAT ontology defines several dcat:Resource's among which a dcat:Dataset.

For DBL we classify information objects as both sml:InformationObject and dcat:Dataset so that they get all relevant properties.

Via sml:PhysicalObject such instance gets sml:isDescribedBy that can be specialized where needed (like into dbl:isDescribedByBIM). Via dcat:Dataset one or more dcat:Distribution's can be related which can be further related to the actual document via dcat:accessURL or dcat:downloadURL. A dcat:Distribution can also be given a dcat:format.

Graphically, this situation is depicted in Figure 14 below.

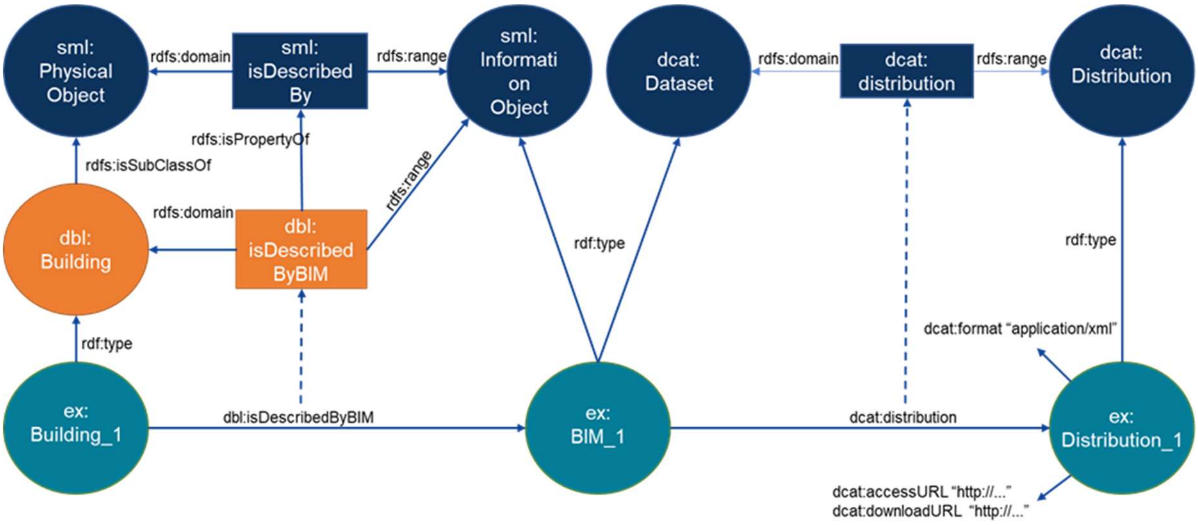

Figure 14: Relating semantics and documents

Core DBL Ontology

In the following sub-sections we describe the DBL core ontology.

Key concepts

A building logbook includes instance data classified according to the following three key concepts heavily based on the European INSPIRE specification:

- Building
- BuildingUnit, being a part of a building related to one facility/activity complex (INSPIRE-term) or functional entity (SML-term)
- CadastralParcel

In terms of the top-level ontology of SML, all three concepts are seen as a subclass of both sml:SpatialRegion and sml:TechnicalEntity.

- Various generic “helper concepts” like Address, Agent (Person or Organisation), Location, Geometry etc.

All have sets of (pre)defined attributes and relations where relations can also target a whole data set (sml:InformationObject’s and/or dcat:Dataset’s to be precise). Because we use Linked Data, such a link can be any local or remote data resource depending on its URI-based ID.

Bi-temporal logic

For all three key entities, building, building unit and cadastral parcel, we apply ‘bi-temporal logic’ grasping their lifespan states in time but also relevant assertion times on the meta-level.

Many current approaches apply these ideas on the object/instance level enabling traceability and reproducibility and avoiding data loss. For flexibility, we decided to apply it on the (lower) property level since properties can come from many data sources all having their independent life span states and assertion times by different data sources/parties. This way, we obtain a more continuous situation for the object which in this case has to be derived. In other words: the object state becomes emergent. Note that properties cover both attributes and relations. So a change in building owner is also considered a state change for a building.

This approach combines well with the decision to model the properties in the SML complex variant way allowing for not only units but also other kinds of meta-data like these various timing aspects.

This way we introduce a kind of “property-states” that when combined at a certain point in time can be on-the-fly result in an overall object state (sml:State in SML) typically having “begins” and “ends” relations from some event (sml:Event).

NOTE 1: SML states are only dealing with object states, not property states. SML states (and events connecting them) can still be used where:

- *Object states are not asserted but derived when needed (from the various more detailed/atomic property value states)*
- *Object states can differ in multiple property values*

NOTE 2: Other metadata properties than actual value or unit result in parallel property-states: a new property state is only changed based on its value change; not its life cycle ‘status’ indication or ‘reference’ indication.

Aspects involved

All properties (attributes or relations) are grouped according to a limited set of seven aspects These aspects are chosen based on data sets collected in WP1.

1. Identification
    a. Building ID ; building unit ID ; cadastral parcel ID ; online link ID like InspireID
2. General
    a. Relationships between key DBL entities (Building, BuildingUnit, CadastralParcel)
    b. Indication of types or subtypes (via a simple attributes not full taxonomy)
    c. Address, placement indicators like geo coordinates
    d. Use function (residential, non-residential and sub-categories) at building level and building unit level ; user profile like students, seniors, asylum seekers etc.
    e. Dates of construction, permits, renovation etc.

   f. Documents : Urban licenses, renovation proposals, renovation passports etc.
   g. Documents: BIM model, technical drawings etc.
3. Legal and Finance
   a. Property tax valuation ; lifecycle cost ; annual maintenance cost ; rental value and maximum rental value ; sales transaction value etc.
   b. Dates of valuations
   c. Documents: sales deed, tenancy agreement, insurance policy, clean soil statement, rule violations
4. Dimensions
   a. Lengths, gross and net areas & volumes
   b. Linked geometric representations (0D, 1D, 2D, 3D)
5. Performance
   a. Functionality offered incl. connection to utility services, indoor health & comfort levels
   b. Energy performance label, circularity label, energy and water use label, CO_2 and N_2 emissions label, smart readiness indicator etc.
   c. Actual energy and water consumption and production levels
   d. Document: Energy Performance Certificate (EPC)
6. Structure & Material
   a. Number of / breakdown in zones, floors, spaces/rooms, elements, components, products, materials
   b. U-values for various element types
   c. Year of latest materials inspection, asbestos check etc.
   d. Certain materials as asbestos for authorised DBL users only
7. Building Services
   a. Energy (gas/electricity/solar/thermal/city heating, …), production and consumption installations
   b. Ventilation system
   c. Water and sewerage installations
   d. Number of elevators, balconies, swimming pools etc.
   e. Building automation parameters
   f. Security for authorised DBL users only
   g. Telecommunications connectivity
   h. Year of latest installation or repair
   i. Year of latest fire security check, elevator and balcony safety check, swimming pool salmonella check etc.
   j. Documents: elevator and fire safety inspection certificates, maintenance contract, utilities contracts

These aspects are all coded as standard SML groups represented by rdfs:Container having members (via rdfs:member).

NOTE 1: Links to unstructured documents or non-LD structured data sets are modelled via sml:InformationObject and/or dcat:Dataset as a whole, linking from there to the actual files via rdfs:seeAlso respectively dcat:distribution/(dcat:accessURL and/or dcat:downloadURL).

Life cycle phase indication

Information on buildings, building units and parcels is typically associated with, or even created in, a specific life cycle phase:

- Programming (As-required);
- Design (As-designed);
- Build (As-built); and
- Use (As-used).

Note:

- Maintenance is regarded as a new as-built & new as-used;
- Renovation is regarded as new as-designed & new as-built & new as-used;
- Repurposing is regarded as a new as-required & new as-designed & new as-built & new as-used; and
- Demolishment/Recycling is regarded as an ultimate form of repurposing (“no more purpose”).

As with the unit information, this status is optionally attached to the property value level via a dbl:status property that points to an enumeration type with the following allowed values:

- dbl:As-required
- dbl:As-designed
- dbl:As-built
- dbl:As-used

Note that changing just the status of a property and not the property value itself is not regarded as a new property-state since a required value stays relevant in parallel if an as-designed or as-built value is added. Formulated the other way round: there can be multiple property states with the same status (but in different time periods) reflecting maintenance, renovation or repurposing events.

Resulting Core DBL Ontology

LEGEND

• = Concept (or sub-Concept)

○ = Grouping

- = Property (in *italics* the proposed non-INSPIRE additions)[3]
    - an attribute, having an XSD datatype or enumeration type as range
    - a relation, having an RDFS class as range

NOTE 1 Not all eight aspects/groupings are relevant for all concepts.

NOTE 2 Indented attributes under attributes are meta-attributes ("attributes of attributes").

- DBL-Root (subtype of sml:SpatialRegion, sml:TechnicalEntity)
    - Identification
        - dbl:inspireId xsd:STRING
    - Legal & Finance
        - *dbl:owner foaf:Agent*
        - *dbl:tenant foaf:Agent*
        - dbl:officialValue xsd:float
            - dbl:officialValueReference dbl:OfficialValueReferenceValue ([TransactionPriceSimple, TransactionPriceMedium, TransactionPriceFull, RentalIncome])
    - Structure and material
        - *dbl:numberOfSwimmingPools: xsd:integer*

- dbl:BuildingOrBuildingUnit (subtype of dbl:DBL-Root)
    - General
        - dbl:cadastralParcel dbl:CadastralParcel
        - dbl:address dbl:Address
        - dbl:currentUse dbl:CurrentUseValue ([Residential, IndividualResidential, CollectiveResidential, TwoDwellings, MoreThanTwoDwelling, ResidenceForCommunities, Agriculture, Industrial, CommerceAndServices, Office, Trade, PublicServices, Ancillary])
    - Dimensions
        - *dbl:grossVolume xsd:float*
        - *dbl:netVolume xsd:float*
        - dbl:officialArea xsd:float
            - dbl:officialAreaReference dbl:OfficialAreaReferenceValue ([ConstructedArea, ExternalArea, InternalArea, InternalPrimaryArea, InternalOtherArea, InternalResidualArea, InternalServiceArea])
    - Performance

[3] In the implementation code, the range of these properties will become a subclass of sml:Objectification: sml:QuantityValue (for floats and integers), sml:QualityValue (for strings and booleans) or sml:RelationReference. The original range can then be encoded as a constraint (like an OWL restriction or a SHACL shape) on the qudt:value/qudt:numericValue property of the objectification (in the context of the relevant class, here: Building, BuildingUnit or CadastralParcel).
Except for 1) identifiers like inspireId, geographicalName and nationalCadastralReference having a simple xsd:string as range, 2) meta-properties. In these two cases we stick to simple, direct, 'no-state' properties.

    - dbl:connectionToElectricity xsd:boolean
    - dbl:connectionToGas xsd:boolean
    - dbl:connectionToSewage xsd:boolean
    - dbl:connectionToWater xsd:boolean
    - dbl:energyPerformance dbl:EnergyPerformanceValue ([A, B, C, D, E, F, G][4])
        - dbl:dateOfAssessment xsd:dateTime
        - dbl:assessmentMethod xsd:string
    - *dbl:circularityPerformance dbl:CircularityPerformanceValue ([CP1, CP2, CP3][5])*
        - dbl:dateOfAssessment xsd:dateTime
        - dbl:assessmentMethod xsd:string
    - *dbl:smartReadinessIndicator dbl:SmartReadinessIndicatorValue ([SRI1, SRI2, SRI3][6])*
        - dbl:dateOfAssessment xsd:dateTime
        - dbl:assessmentMethod xsd:string
    - *dbl:yearlyUseOfWater xsd:float*
    - *dbl:yearlyReuseOfWater xsd:float*
    - *dbl:yearlyUseOfGas xsd:float*
    - *dbl:yearlyUseOfElectricity xsd:float*
  - Structure & Material
    - dbl:numberOfRooms xsd:integer
  - Building Services
    - dbl:heatingSource dbl:HeatingSourceValue ([Biogas, Electricity, LiquidFuels, Naturalgas, SolidFuels, Straw, WarmWaterOrSteam])
    - dbl:heatingSystem dbl:HeatingSystemValue ([CentralHeating, DistrictHeating, ElectricRadiators, HeatPump, PortableGasHeating, SolarHeating, Stove, Missing])
    - dbl:ventilationSystem dbl:VentilationSystemValue ([Natural, Mechanical, Hybrid, Missing])

- dbl:Building (subtype of dbl:BuildingOrBuildingUnit)
  - Identification
    - dbl:geographicalName xsd:string
  - General
    - dbl:buildingUnit dbl:BuildingUnit[7]

---

[4] These codes are just placeholders to be filled in by member states maybe involving extra options like A+, A++ and A+++..
[5] Idem.
[6] Idem. Example: a score from 0 to 3 for each of seven dimensions (0 = nothing, 1 = clock control (heating turns on at set times), 2 = occupancy detection control (heating turns on when someone enters), 3 = air quality sensor control, 4 = sensor control with local flow regulators as define at https://ec.europa.eu/newsroom/ener/newsletter-archives/39573.
[7] Not modelled as a subclass of sml:hasPart since we need an objectified variant here having state

- dbl:buildingNature dbl:BuildingNatureValue ([Arch, Bunker, Canopy, Castle, CaveBuilding, Chapel, Church, Dam, Greenhouse, Lighthouse, Mosque, Shed, Silo, Stadium, StorageTank, Synagogue, Temple, Tower, Windmill, WindTurbine])
- *wgs84_pos:location wgs84_pos:SpatialThing*
- *locn:location dct:Location*
- dbl:dateOfConstruction xsd:dateTime
- dbl:dateOfRenovation xsd:dateTime
- dbl:dateOfDemolition xsd:dateTime
- *dbl:isDescribedByNativeBIM dbl:NativeBIM, dcat:Dataset*[8]
- *dbl:isDescribedByOpenBIM dbl:OpenBIM, dcat:Dataset*[9]
- *dbl:isDescribedByNativeGIS dbl:NativeGIS, dcat:Dataset*[10]
- *dbl:isDescribedByOpenGIS dbl:OpenGIS, dcat:Dataset*[11]

- Dimensions
  - dbl:elevation xsd:float
    - dbl:srsName xsd:string
    - dbl:elevationReference dbl:ElevationReferenceValue ([AboveGroundEnvelope, BottomOfConstruction, EntrancePoint, GeneralEave, GeneralGround, GeneralRoof, GeneralRoofEdge, HighestEave, HighestGroundPoint, HighestPoint, HighestRoofEdge, LowestEave, LowestFloorAboveGround, LowestGroundPoint, LowestRoofEdge, TopOfConstruction])
  - *dbl:footprint xsd:float*
  - dbl:heightAboveGround xsd:float
    - dbl:heightReference dbl:ElevationReferenceValue (same as elevation reference value)
    - dbl:lowReference dbl:ElevationReferenceValue (same as elevation reference value)
    - dbl:heightStatus dbl:HeightStatusValue ([Estimated, Measured])
  - dbl:heightBelowGround xsd:float
  - dbl:roofType dbl:RoofTypeValue ([ArchRoof, ConicalRoof, DomedRoof, DualPentRoof, FlatRoof, GabledRoof, HalfHippedRoof, HippedRoof, HyperbolicParabaloidalRoof, MansardRoof, MonopitchRoof, PavilionRoof, PyramidalBroachRoof, SawToothRoof])
  - dbl:geometry geo:Geometry (INSPIRE-variant)
  - *geo:hasGeometry geo:Geometry (GeoSPARQL-variant)*
  - dbl:horizontalGeometry geo:Geometry
  - dbl:horizontalGeometryReference dbl:HorizontalGeometryReferenceValue ([AboveGroundEnvelope, Combined, EntrancePoint, Envelope, Footprint, LowerFloorAboveGround, PointInsideBuilding, PointInsideCadastralParcel, RoofEdge])
- Performance

[8] Not modelled as a subclass of sml:isDescribedBy since we need an objectified variant here having state
[9] Idem.
[10] Idem.
[11] Idem.

    - dbl:conditionOfConstruction dbl:ConditionOfConstructionValue ([Declined, Functional, Demolished, Projected, Ruin, UnderConstruction])
  - Structure & Material
    - dbl:numberOfBuildingUnits xsd:integer
    - dbl:numberOfDwellings xsd:integer
    - dbl:numberOfFloorsAboveGround xsd:integer
    - dbl:numberOfFloorsBelowGround xsd:integer
    - *dbl:numberOfElevators: xsd:integer*
    - *dbl:numberOfBalconies: xsd:integer*
    - dbl:materialOfFacade dbl:MaterialOfFacadeValue ([Adobe, Asbestos, CeramicTiles, Composite, Concrete, Glass, Limestone, Masonry, Metal, NaturalStone, Vegetated or Wood])
    - dbl:materialOfRoof dbl:MaterialOfRoofValue ([Asbestos, CeramicTile, ClayTile, Composition, ConcreteTile, CurrugatedSheet, Glass, HotMoppedAsphalt, Metal, ReinforcedConcrete, Slate, Thatch, VegtatedGreenRoof or WoodShinglesOrShakes])
    - dbl:materialOfStructure dbl:MaterialOfStructureValue ([ReinforcedConcrete, ReinforcedMasonry, RubleStoneMasonry, Steel, StoneMasonryBlock, Wood, AdobeBlockWalls, ConcreteBlockMasonry, Earth, FiredBrickMasonry, InformalConstructions, MassiveStoneMasonry, MobileHomes, MudWalls or PrecastConcrete])
    - *dbl:uValueFacades xsd:float*
    - *dbl:uValueRoofs xsd:float*
    - *dbl:uValueWindows xsd:float*
    - *dbl:uValueFloors xsd:float*
  - Building Services
    - *dbl:solarSurfacePotential xsd:float*
    - *dbl:solarSurfaceActual xsd:float*
    - *dbl:renewableEnergyProduction xsd:float*
    - *dbl:numberOfEVChargingPoints xsd:integer*
    - *dbl:kindOfCommunicationConnection KindOfCommunicationConnectionValue ([TelephoneLine, Cable, Optical, WiFi, 4G, 5G])*

- dbl:BuildingUnit (subtype of dbl:BuildingOrBuildingUnit)

- dbl:CadastralParcel (subtype of dbl:DBL-Root)
  - Identification
    - dbl:nationalCadastralReference xsd:string
  - General
    - *dbl:isDescribedByNativeGIS sml:InformationObject dcat:Dataset* (rdfs:subPropertyOf sml:isDescribedBy)

    - *dbl:isDescribedByOpenGIS sml:InformationObject dcat:Dataset* (rdfs:subPropertyOf sml:isDescribedBy)
  - Legal & Finance
    - dbl:administrativeUnit xsd:string
    - *dbl:hasCleanSoilStatement xsd:boolean*
  - Dimensions
    - dbl:areaValue xsd:float
    - *dbl:circumference xsd:float*
    - dbl:referencePoint geo:Point
    - dbl:geometry geo:Geometry (INSPIRE-variant)
    - *geo:hasGeometry geo:Geometry (GeoSPARQL-variant)*

Helper entities (not having state-properties)

- dbl:Address
    - dbl:adminUnit1stOrder (country) xsd:string
    - dbl:adminUnit2ndOrder (state, province, ...) xsd:string
    - dbl:adminUnit3rdOrder (municipality) xsd:string
    - dbl:postName (city) xsd:string
    - dbl:thoroughfare (streetName) xsd:string
    - dbl:locatorDesignator (streetNumber) xsd:string
    - dbl:locatorName xsd:string
    - dbl:postCode xsd:string

- sml:QuantityValue, sml:QualityValue, sml:RelationReference or instances of sml:EnumerationType
    - dbl:assertionTimeStart xsd:dateTime
    - dbl:assertionTimeEnd xsd:dateTime
    - dbl:stateTimeStart xsd:dateTime
    - dbl:stateTimeEnd xsd:dateTime
    - dbl:status statusValue ([dbl:As-required, dbl:As-designed, dbl:As-built, dbl:As-used])

- foaf:Agent
    - foaf:Person
    - foaf:Organization

Location

- locn:location dct:Location (http://purl.org/dc/terms/Location) named place or spatial region or

- wgs84_pos:location wgs_pos:SpatialThing
- wgs84_pos:SpatialThing
    - wgs84_pos:lat xsd:float (unit: decimal degree)
    - wgs84_long xsd:float (unit: decimal degree)
    - wgs84_alt xsd:float (unit: decimal degree)

Geometry

DBL reuses like INSPIRE OGC's GML Simple Features for geometry:

GM_Object and its subtypes like GM_Point, GM_Curve, GM_MultiCurve, GM_LineString, GM_Polygon, GM_Surface and GM_MultiSurface.

The 'Linked Data'-implementation here is the GeoSPARQL (version 1.1) ([18]) specification providing GML_Literals as datatype being XML-fragments of GML used by geo:Geometry.

EXAMPLE

ex:Building_1 dbl:geometry/geo:hasGeometry ex:Geometry_1 .

ex:Geometry_1 geo:asGML

```
""" <gml:Point
srsName=\"http://www.opengis.net/def/crs/OGC/1.3/CRS84\"
xmlns:gml=\"http://www.opengis.net/gml/3.2\"> <gml:pos>-83.38
33.95</gml:pos> </gml:Point>
"""^^http://www.opengis.net/ont/geosparql#gmlLiteral .
```

Specific information objects/data sets

- sml:InformationObject, dcat:Dataset
- dbl:NativeBIM
- dbl:OpenBIM
- dbl:NativeGIS
- dbl:OpenGIS

NOTE 1: The SML top-level concepts: smL:GeometricEntity and sml:TemporalEntity are not used for DBL. For the first we reuse GeoSPARQL (geo:hasGeometry/geo:Geometry), for the second, we simply reuse xsd:dateTime0 (and not the more complex W3C Time Ontology ([15]) e.g.).

NOTE 2: INSPIRE hierarchies in enumeration type values are lost: all individuals are equal. The generic and specific ones can all be used independently. The reason is that we think such semantics should have been modelled via a taxonomy instead involving vlasses i.s.o. individuals.

In deliverable D2.2 of task 2.2, all these concept and properties (attributes and relations) are implemented in RDFS code (primary in Turtle syntax, derived JSON-LD code in appendix.

9. DBL Dictionary

Concept terms (all reused from INSPIRE)

Building: Enclosed constructions above and/or underground which are intended or used for the shelter of humans, animals, things or the production of economic goods and that refer to any structure permanently constructed or erected on its site.

BuildingUnit: BuildingUnits are subdivisions of (parts of) Buildings with their own lockable access from the outside or from a common area (i.e. not from another BuildingUnit), which are atomic, functionally independent, and may be separately sold, rented out, inherited, etc.

NOTE 1: A building unit is homogeneous in management aspects. Its key mandatory attribute is the external reference to some official register where the BuildingUnit is identified and described. It is generally the cadastral register but may be another information system, e.g. a register of public properties.

CadastralParcel: Areas defined by cadastral registers or equivalent. As much as possible, cadastral parcels should be forming a partition of the national territory. The cadastral parcel should be considered as a single area of Earth surface (land and/or water), under homogeneous real property rights and unique ownership, real property rights and ownership being defined by national law.

NOTE 2: As much as possible, in the INSPIRE context, cadastral parcels should be forming a partition of the national territory. The cadastral parcel should be considered as a single area of Earth surface (land and/or water), under homogeneous real property rights and unique ownership, real property rights and ownership being defined by national law (adapted from UN ECE 2004 and WG-CPI, 2006). Unique ownership is means that the ownership is held by one or several joint owners for the whole parcel.

Address: An identification of the fixed location of the property (like a building) through a structured composition of geographic names and identifiers.

Property/relation/group terms (reused from INSPIRE where available)

For dbl:DBL-Root

- **dbl:inspireId**: External object identifier of the spatial object. For DBL we will apply the CEN SML (URI-based) identification scheme.
- **dbl:officialValue**: The value of the building, building unit or cadastral parcel as registered in an official information system.
- **dbl:officialValueReference**: The reference market price that the official value aims to assess. One of the following: TransactionPriceSimple, TransactionPriceMedium, TransactionPriceFull or RentalIncome. Meta-property for dbl:officialValue.

For dbl:BuildingOrBuildingUnit

- **dbl:grossVolume:** The total volume of all interior spaces in a building or building unit over the gross floor area. This total volume is enclosed by the outer boundary surfaces of the foundation, the exterior walls and the roof (including the dormers and skylights) (DIN 277-1 2005).
- **dbl:netVolume:** The net volume of a building or building unit is the gross floor space minus the construction volumes (the latter being the spaces occupied by vertical construction elements such as walls).
- **dbl:officialArea:** The area of the building or building unit as registered in an official information system.
  - **dbl:officialAreaReference:** The type of the official area. One of the following: ConstructedArea, ExternalArea, InternalArea, InternalPrimaryArea, InternalOtherArea, InternalResidualArea or InternalServiceArea. Meta-property for dbl:officialArea.
  - **dbl:currentUse**: Activity hosted within the building or building unit. This attribute addresses mainly the buildings hosting human activities. One of the following: Residential, IndividualResidential, CollectiveResidential, TwoDwellings, MoreThanTwoDwelling, ResidenceForCommunities, Agriculture, Industrial, CommerceAndServices, Office, Trade, PublicServices or Ancillary.
  - **dbl:connectionToElectricity**: An indication if the building or building part or building unit is connected or not to the public electricity network.
  - **dbl:connectionToGas**: An indication if the building or building part or building unit is connected or not to the public gas network.
  - **dbl:connectionToSewage**: An indication if the building or building unit is connected or not to the public sewage network.
  - **dbl:connectionToWater**: An indication if the building or building unit is connected or not to the public water network.
  - **dbl:energyPerformance**: The energy performance of the building or building unit. One of the following: A+++, A++, A+, A, B, C, D, E, F or G.
  - **dbl:dateOfAssessment:** The date when the energy, circularity, smart readiness (or other) performance of the building or building unit was assessed. Meta-property for dbl:energyPerformance, dbl:circularityPerformance and dbl:smartReadinessIndicator.
  - **dbl:assessmentMethod:** The reference to the method or document describing the assessment method of performance. Meta-property for dbl:energyPerformance, dbl:circularityPerformance and and dbl:smartReadinessIndicator.
  - **dbl**:**circularityPerformance**: A total performance label for circularity related to material scarcity and environmental impacts, based on the total life cycle of a building or building unit. Taking into account all its products/materials that are part of it. One of the following (placeholder): CP1, CP2, CP3.
  - **dbl:smartReadinessIndicator**: The Smart Readiness Indicator (SRI) of a building or building unit is an indicator that informs on the rating of smart readiness of a building or building unit in line with Directive 2010/31/EU. [17]. One of the following (placeholder): SRI1, SRI2, SRI3.

- **dbl:yearlyUseOfWater**: The total use of fresh water for a building or building unit typically in M3 used for drinking, cooking, cleaning, toilet flushing, gardening etc.
- **dbl:yearlyReuseOfWater**: The part of the yearlyUseOfwater that is directly or indirectly reused by a building or building unit in a year, again typically in M3.
- **dbl:yearlyUseOfGas**: The total use of natural gas for a building or building unit typically in M3 used for heating, washing, cooking etc.
- **dbl:yearlyUseOfElectricity**: The total use of natural gas for a building or building unit typically in kWh used for heating, washing, cooking, appliances etc.
- **dbl:numberOfRooms**: The number of different rooms in a building or building unit available for end-user processes.
- **dbl:heatingSource**: The source of energy used for the heating like electricity or natural gas. One of the following: Biogas, Electricity, LiquidFuels, Naturalgas, SolidFuels, Straw or WarmWaterOrSteam.
- **dbl:heatingSystem**: The system of heating like a stove, central heating or a heat pump. One of the following: CentralHeating, DistrictHeating, ElectricRadiators, HeatPump, PortableGasHeating, SolarHeating, Stove or Missing.
- **dbl:ventilationSystem**: The system of ventilation. One of the following: Natural, Mechanical, Hybrid or Missing.

For dbl:Building

- **dbl:geographicalName**: Name of the construction.
- **dbl:buildingNature**: Characteristic of the building that makes it generally of interest for mappings applications. The characteristic may be related to the physical aspect and/or to the function of the building. One of the following: Arch, Bunker, Canopy, Castle, CaveBuilding, Chapel, Church, Dam, Greenhouse, Lighthouse, Mosque, Shed, Silo, Stadium, StorageTank, Synagogue, Temple, Tower, Windmill or WindTurbine. These items have to be extended for "normal" buildings used for residential, office purposes etc.! like SingleFamilyHouse, MultiFamilyHouse, TerracedHouse, ApartmentBlock etc.
- **dbl:dateOfConstruction**: Date of construction.
- **dbl:dateOfRenovation**: Date of last major renovation.
- **dbl:dateOfDemolition**: Date of demolition.
- **dbl:elevation**: Vertically-constrained dimensional property consisting of an absolute measure referenced to a well-defined surface which is commonly taken as the origin (geoïd, water level, etc.).
- **dbl:elevationReference**: Element where the elevation was measured. One of the following: AboveGroundEnvelope, BottomOfConstruction, EntrancePoint, GeneralEave, GeneralGround, GeneralRoof, GeneralRoofEdge, HighestEave, HighestGroundPoint, HighestPoint, HighestRoofEdge, LowestEave, LowestFloorAboveGround, LowestGroundPoint, LowestRoofEdge or TopOfConstruction. Meta-property for dbl:elevation.
- **dbl:srsName:** The name of the spatial reference system. Meta-property for dbl:elevation.
- **dbl:footprint**: The ground plate of a building. The geometry of this plate is defined by a horizontal geometry reference.

- **dbl:heightAboveGround**: Height above ground. Vertical distance (measured or estimated) between a low reference and a high reference.
- **dbl:heightReference**: Element used as the high reference. One of: same as dbl:elevationReference. Meta-property for dbl:heigthAboveGround.
- **dbl:lowReference**: Element as the low reference. One of: same as dbl:elevationReference. Meta-property for dbl:heigthAboveGround.
- **dbl:heightStatus**: Element used as the high reference. One of: Estimated or Measured. Meta-property for dbl:heigthAboveGround.
- **dbl:heightBelowGround:** Height below the ground of the building.
- **dbl:numberOfElevators:** The number of elevators in a building.
- **dbl:numberOfSwimmingPools:** The number of swimming pools for a building.
- **dbl:numberOfBalconies:** The number of balconies for a building.
- **dbl:roofType**: The shape of the roof. One of the following: ArchRoof, ConicalRoof, DomedRoof, DualPentRoof, FlatRoof, GabledRoof, HalfHippedRoof, HippedRoof, HyperbolicParabaloidalRoof, MansardRoof, MonopitchRoof, PavilionRoof, PyramidalBroachRoof or SawToothRoof.
- **dbl:conditionOfConstruction**: Status of the construction. One of the following: Declined, Functional, Demolished, Projected, Ruin or UnderConstruction.
- **dbl:numberOfBuildingUnits**: Number of building units in the building. A BuildingUnit is a subdivision of a Building with its own lockable access from the outside or from a common area (i.e. not from another BuildingUnit), which is atomic, functionally independent, and may be separately sold, rented out, inherited, etc.
- **dbl:numberOfDwellings**: Number of dwellings as residential units which may consist of one or several rooms designed for the occupation of households.
- **dbl:numberOfFloorsAboveGround**: Number of floors above ground.
- **dbl:numberOfFloorsBelowGround**: Number of floors below ground.
- **dbl:solarSurfacePotential**: The potential / maximum possible surface area for solar electricity production.
- **dbl:solarSurfaceActual**: The used surface area for solar electricity production.
- **dbl:renewableEnergyProduction**: The total generated energy by the building.
- **dbl:numberOfEVChargingPoints**: The number of charging points for electric vehicles.
- **dbl:kindOfCommunicationConnection**: The kind of communication connection(s) to the environment. One of the following: TelephoneLine, Cable, Optical, WiFi, 4G or 5G.
- **dbl:materialOfFacade**: Material(s) of the building facade. One of the following: Adobe, Asbestos, CeramicTiles, Composite, Concrete, Glass, Limestone, Masonry, Metal, NaturalStone, Vegetated or Wood.
- **dbl:materialOfRoof**: Material(s) of the building roof. One of the following: Asbestos, CeramicTile, ClayTile, Composition, ConcreteTile, CurrugatedSheet, Glass, HotMoppedAsphalt, Metal, ReinforcedConcrete, Slate, Thatch, VegtatedGreenRoof or WoodShinglesOrShakes.
- **dbl:materialOfStructure**: Material(s) of the building structure. One of the following: ReinforcedConcrete, ReinforcedMasonry, RubleStoneMasonry, Steel, StoneMasonryBlock, Wood, AdobeBlockWalls, ConcreteBlockMasonry, Earth, FiredBrickMasonry, InformalConstructions, MassiveStoneMasonry, MobileHomes, MudWalls or PrecastConcrete.

- **dbl:uValueFacades**: The amount of energy (heat) lost through a square metre (m2) for facades for every degree (K) difference in temperature between the inside and the outside.
- **dbl:uValueRoofs**: The amount of energy (heat) lost through a square metre (m2) for roofs for every degree (K) difference in temperature between the inside and the outside.
- **dbl:uValueWindows**: The amount of energy (heat) lost through a square metre (m2) for windows for every degree (K) difference in temperature between the inside and the outside.
- **dbl:uValueFloors**: The amount of energy (heat) lost through a square metre (m2) for floors for every degree (K) difference in temperature between the inside and the outside.

For dbl:CadastralParcel

- **dbl:nationalCadastralReference**: Thematic identifier at a national level, generally the full national code of the cadastral parcel. Must ensure the link to the national cadastral register or equivalent.
- **dbl:administrativeUnit**: The administrative unit of the lowest administrative level containing this cadastral parcel.
- **dbl:hasCleanSoilStatement**: The possession of proof that a parcel has clean soil.
- **dbl:areaValue**: The area value of the parcel typically in M2.
- **dbl:circumference**: The total circumference of a parcel.

For dbl:Address

- **dbl:adminUnit1stOrder**: Position derived from the related administrative unit of 1st order. For DBL we interpret this to be a country.
- **dbl:adminUnit2ndOrder**: Position derived from the related administrative unit of the 12th order. For DBL we interpret this to be a state, province, etc. as a region within a country.
- **dbl:adminUnit3rdOrder**: Position derived from the related administrative unit of 3rd order. For DBL we interpret this to be a municipality.
- **dbl:postName:** One or more names created and maintained for postal purposes to identify a subdivision of addresses and postal delivery points. For DBL we interpret this to be a city.
- **dbl:thoroughfare**: The name or names of a passage or way through from one location to another like a road or a waterway. For DBL we interpret this to be a name of a street.
- **dbl:locatorDesignator**: A number or a sequence of characters which allows a user or an application to interpret, parse and format the locator within the relevant scope. A locator may include more locator designators. For DBL we interpret this to be a street number.
- **dbl:locatorName**: Proper noun(s) applied to the real-world entity identified by the locator. Like the “Belvedere Building”.
- **dbl:postCode**: A code created and maintained for postal purposes to identify a subdivision of addresses and postal delivery points.

The definitions for the allowed ‘one of’ values are added in the SKOS code in deliverable D2.2.

Relation terms (part of 'general' property group)

Definitions for relations are reused from the INSPIRE directive.

- **dbl:cadastralParcel** from dbl:BuildingOrBuildingUnit to dbl:CadastralParcel
  - Definition: The cadastral parcel(s) to which the building or building part or building unit is officially related.
- **dbl:buildingUnit** from dbl:Building to dbl:BuildingUnit
  - Definition: The building unit(s) belonging to the building or building part.
- **dbl:address** from dbl:BuildingOrBuildingUnit to dbl:Address

New relations (DBL defined):

- **dbl:isDescribedByNativeBIM** from dbL:Building to dbl:BIM, dcat:Dataset
  - Definition: A linked Building Information Model in a proprietary format/semantics. Like a REVIT model.
- **dbl:isDescribedByOpenBIM** from dbl:Building to dbl:OpenBIM, dcat:Dataset
  - Definition: A linked Building Information Model in an open format/semantics. Like a an IFC or gbXML model.
- **dbl:isDescribedByNativeGIS** from dbl:Building/dbl:CadastralParcel to dbl:NativeGIS, dcat:Dataset
  - Definition: A linked GIS/GEO model in a proprietary format/semantics. Like an ESRI Shape file.
- **dbl:isDescribedByOpenGIS** from dbl:Building/dbl:CadastralParcel to dbl:OpenGIS, dcat:Dataset
  - Definition: A linked GIS/GEO Model in an open format/semantics. Like a GML/CITYGML or CityJSON model.
- **dbl:owner** from dbl:DBL-Root to foaf:Agent
  - Definition: A person who owns something; in this case a building, a building unit or a cadastral parcel.
- **dbl:tenant** from dbl:DBL-Root to foaf:Agent
  - Definition: A person who occupies land or property rented from an owner; in this case a building, a building unit or a cadastral parcel.
- **dbl:address** from dbl:BuildingOrBuildingUnit to dbl:Address
  - Definition: The address of a building or building unit.
- **dbl:referencePoint** from dbl:CadastralParcel to geo:Geometry
  - Definition: A point within the cadastral parcel.
- **dbl:geometry** from db:Building/dbl:CadastralParcel to geo:Geometry
  - Definition: The explicit location/shape representation of a building or cadastral parcel according to the GeoSPARQL implementation of GML Simple Features.

- **dbl:horizontalGeometry** from dbl:Building to geo:Geometry
  - Definition: The explicit horizontal shape representation of a building according to the GeoSPARQL implementation of GML Simple Features.
- **dbl:horizontalGeometryReference** meta-property for dbl:horizontalGeometry
  - Definition: Values indicating the element considered to capture a horizontal geometry of a building. One of the following: AboveGroundEnvelope, Combined, EntrancePoint, Envelope, Footprint, LowerFloorAboveGround, PointInsideBuilding, PointInsideCadastralParcel or RoofEdge.

Group terms

All these properties are grouped in seven aspects with the following definitions:

Identification: The aspect of identification refers to all properties that somehow uniquely denote an abstract or concrete thing, existing in reality or only planned. Identification is typically the first step in Identification, Authentication & authorisation.

General: This generic aspect refers to all properties that are hard to classify towards any other aspect defined here like relations to other objects, locations and specific life-cycle events in time.

Legal and Finance: The aspect of legal & finance refers to all properties that are used in formal transactions like proof of ownership and taxation values.

Dimensions: The aspect of dimensions refers to all properties that give information about its location, orientation and inner (interior) or outer (boundary) shape/geometry. Typically involving some coordinate reference system (CRS).

Performance: The aspect of performance refers to all properties that tell us about how it performs in its environment.

Structure & Material: The aspect of structure & material refers to all properties related to the breakdown and materialization of an object.

Building Services: The aspect of building services refers to all properties related to the technical installations of a building.

In deliverable D2.2 of task 2.2, all these definitions are implemented in SKOS code (primary in Turtle syntax, derived JSON-LD code in an appendix). Except for terms related to properties having range classes being individuals of sml:EnumerationType. These are defined equivalently to the definitions for their range classes.

References

[1] Study on the Development of a European Union Framework for Digital Building Logbooks, FINAL REPORT, December 2020, EUROPEAN COMMISSION, Executive Agency for Small and Medium-sized Enterprises (EASME), Unit A.1.2 — Competitiveness & Internationalisation, European Commission B-1049 Brussels.

[2] New European Interoperability Framework, Promoting seamless services and data flows for European public administrations, European Union, 2017, https://op.europa.eu/en/publication-detail/-/publication/bca40dde-deee-11e7-9749-01aa75ed71a1/language-en.

[3] WebID 1.0, Web Identity and Discovery, W3C Editor's Draft 28 May 2014, https://www.w3.org/2005/Incubator/webid/spec/.

[4] Decentralized Identifiers (DIDs) v1.0, Core architecture, data model, and representations, W3C Recommendation 19 July 2022, https://www.w3.org/TR/did-core/.

[5] eIDAS Regulation, Regulation (EU) No 910/2014 of the European Parliament and of the Council of 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC, https://digital-strategy.ec.europa.eu/en/policies/eidas-regulation.

[6] Data Catalog Vocabulary (DCAT) - Version 2, W3C Recommendation 04 February 2020, https://www.w3.org/TR/vocab-dcat-2/.

[7] NL Kadaster: Basisregistratie Adressen en Gebouwen (BAG), in SKOS, https://catalogus.kadaster.nl/bag/nl/.

[8] Data Specifications > Themes > Buildings | INSPIRE (europa.eu), in UML/XSD, https://inspire.ec.europa.eu/Themes/126/2892.

[9] Data Specifications > Themes > Cadastral parcels, in UML/XSD, https://inspire.ec.europa.eu/Themes/122/2892.

[10] Data Specifications > Themes > Addresses | INSPIRE (europa.eu), in UML/XSD, https://inspire.ec.europa.eu/Themes/79/2892.

[11] EN 17632-1:2022 Building Information Modelling (BIM) - Semantic Modelling and Linking (SML) - Part 1: Generic modelling patterns, CEN TC442, 07-12-2022.

[12] prEN 17632-2:2023 Building Information Modelling - Semantic Modelling and Linking (SML) - Part 2: Domain-specific modelling patterns, CEN TC442 (planned out for enquiry in March 2023).

[13] RDF 1.1 Concepts and Abstract Syntax, W3C Recommendation 25 February 2014, https://www.w3.org/TR/rdf11-concepts/.

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

[15] Time Ontology in OWL, W3C Candidate Recommendation Draft, 15 November 2022, https://www.w3.org/TR/owl-time/.

[16] Guide, Measuring circularity, Working agreements for circular construction, Version 2.0 – 02 July 2020 Platform CB'23, https://platformcb23.nl/images/downloads/Platform_CB23_Guide_Measuring_circularity_2.0.pdf.

[17] Commission delegated regulation (EU) 2020/2155 of 14 October 2020 supplementing Directive (EU), 2010/31/EU of the European Parliament and of the Council by establishing an optional common European Union scheme for rating the smart readiness of buildings, https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32020R2155&rid=8.

[18] OGC GeoSPARQL - A Geographic Query Language for RDF Data, Open Geospatial Consortium, External identifier of this OGC® document: Internal reference number of this OGC® document: 22-047, Version: 1.1, Category: OGC® Implementation Specification, https://opengeospatial.github.io/ogc-geosparql/geosparql11/spec.html.

DBL

DIGITAL BUILDING
LOGBOOK

LinkedIn | @EU_Growth

European
Commission