

SEMIC Style Guide for Drafting Semantic Specifications

Table of Contents

1. Introduction	2
1.1. Current status	2
1.2. How to contribute	2
1.3. Future works	2
2. Need	3
2.1. Description	3
2.2. Branding	3
2.3. Ownership and governance.	3
2.4. Pragmatism and ease of use	3
2.5. Broad applicability	3
3. General Guidelines	7
3.1. Naming and versioning	7
3.2. Reuse of external vocabularies	10
4. UML Modelling	17
4.1. General guidelines	17
4.2. eGovernment Core Vocabularies	23
4.3. Application Profiles	40
5. RDF guidelines	47
5.1. Vocabulary annotations in RDFS	47
5.2. Vocabulary semantics in RDFS/OWL	84
5.3. Application Profiles	94
6. SHACL guidelines	96
6.1. SHACL Approach	96
6.2. SHACL and Application Profiles	125
7. XSD guidelines	136
7.1. XSD Design	136
7.2. Application Profiles	166
7.3. XSD Annotations	169
8. Publishing	171
8.1. Artefacts	171
8.2. LOD Resources	174
9. Samples	181
9.1. Core RDFS	181
9.2. SHACL	183
9.3. Semantics RDFS	187
9.4. SHACL Application Profiles	188
9.5. XSD Vocabularies	189
9.6. XSD Constraints and AP	190



The views expressed in this report are purely those of the Author(s) and may not, in any circumstances, be interpreted as stating an official position of the European Commission.

The European Commission does not guarantee the accuracy of the information included in this study, nor does it accept any responsibility for any use thereof.

Reference herein to any specific products, specifications, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favouring by the European Commission. All care has been taken by the author to ensure that s/he has obtained, where necessary, permission to use any parts of manuscripts including illustrations, maps, and graphs, on which intellectual property rights already exist from the titular holder(s) of such rights or from her/his or their legal representative.

Chapter 1. Introduction

This document defines the style guide to be applied to the formal representations of SEMIC's specifications, notably to the [eGovernment Core Vocabularies](#). It provides rules on naming conventions, syntax, and organisation for the SEMIC specifications. It is complemented with technical artefacts and rules which allow for automatic validation.

The content of these guides is part of the [ISA2 Action 2016.07 Promoting semantic interoperability amongst the EU Member States](#) with the objective of promoting the use of standards by, for example, offering guidelines and expert advice on semantic interoperability for public administrations.

1.1. Current status

The status of this release of the SEMIC Style Guide is ***draft version for revision***. It is intended for the European Commission's officers, collaborating consultants, and stakeholders to read, comment, and improve.

1.2. How to contribute

Any comments, suggestions, corrections, and other feedback is expected to be addressed in this [Issues](#) workspace.

Please use the template and labels provided therein and follow the provided instructions for optimal management and communication with the SEMIC team.

1.3. Future works

- **Revision of the eGovernment Core Vocabularies:** This SEMIC Style Guide will be used as the basis for revision and evolution of the eGovernment Core Vocabularies.
- **Tooling:** In alignment with the SEMIC Style Guide, a technical solution is being developed to automate the production of the semantic artefacts based on the conceptual representations (as UML) of the Vocabularies.
- **Dissemination:** The deliverables related to the SEMIC Style Guide, including the tooling, will be made publicly available under the [SEMIC licensing policy](#). Any European Public Administration, or stakeholder, is encouraged to use and extend them.

Communication events, technical documentation, and training materials will be provided by the European Commission to raise awareness on the existence of the SEMIC Style Guide and related solutions, as well as to promote their use.

Chapter 2. Need

2.1. Description

This deliverable establishes guidelines for drafting eGovernment Core Vocabularies and Application Profiles. It has been considered that defining and aligning common guidelines is beneficial for – at least – the following reasons:

2.2. Branding

SEMIC aims to present these standards as one large group under a single brand. Such branding will only work when all standards use the same modelling approach, follow the same naming conventions, notations, guidelines and rules, and are published with identical documentation output.

2.3. Ownership and governance.

In the future, SEMIC will make sure it has ownership over the vocabularies and can therefore govern them in a timely manner aligned with stakeholders' needs. This is intended to lead to a policy with less dependence on external vocabularies and external organisations. For this reason, own namespace(s) will be claimed, and the core concepts and relationships needed will be defined using these own namespace(s).

2.4. Pragmatism and ease of use

Previous versions of these standards were only oriented to the Semantic Web and Linked Open Data. However, these technologies are not yet universally understood nor used and are considered to be too difficult. Hence, having unified guidelines and good practice advice is essential for keeping the eGovernment Core Vocabularies pragmatic and straightforward. Besides, the majority of users use established data formats such as XML and JSON. As far as these contexts are concerned, the present guidelines also include XML Schema and detailed guidelines on how to use and implement these in alignment with the semantic web-oriented assets (including JSON Schema guidelines for the future as soon as they will come to a stable version).

2.5. Broad applicability

These standards should have general applicability. Accordingly, the eGovernment Core Vocabularies aim is to define simplified, reusable, and extensible data models that capture only the fundamental characteristics in a context-neutral manner.

This will lead to:

- low ontological commitment, meaning that semantics which would lead to inferences/deductions that could limit reuse while being too specific, will not be added;
- limited constraints; constraints are better defined at the application level layer.

The following is an example of a ‘too heavy’ ontological commitment:

Suppose there is a property with a name ‘identifier’ which has been assigned a class ‘Public Service’ as its domain. When this property ‘identifier’ is also used on an instance of a class ‘Membership’ – which is allowed since rdf properties are global and can be used everywhere – then this same ‘Membership’ instance would also become an instance of ‘Public Service’. This is not a desired outcome and even a wrong one if the respective classes have been defined as disjoint. And so, the semantics of this ‘identifier’ property result that it cannot be as widely used as possible.

This document’s objective can be summarised as the following:

- Provide a brief background on Semantic Interoperability along with the technological means to materialise this interoperability by using the SEMIC specifications;
- Develop and suggest a set of human-oriented guidelines and recommendations for the conceptual representation, design and implementation of the SEMIC specifications (i.e. the reusable ontologies and the eGovernment Core Vocabularies;
- Develop and suggest a collection of reference artefacts ^[1] for the formal representation, implementation and validation of SEMIC’s specifications;
- Develop and propose an initial set of naming and design rules for approaching the automated generation of the ontologies and eGovernment Core Vocabularies in different syntax languages (RDF, XML, JSON), and from one single point of maintenance (the conceptual representation). This document is mainly addressed to ICT developers involved in the definition, design, coordination, and delivery of the SEMIC standards and the application profiles themselves. A large number of these recommended guidelines have been collected in order to make naming, styling, and lay-out as consistent as possible.

Additional to the developers, the second targeted audience is the users of these standards and application profiles. They, being supported by the Style Guides, should be able to read, understand, and implement the proposed data models.

Finally to mention, policymakers as well as public administration officers at various governmental levels that need to build on the standards by further extending or refining the existing SEMIC ones can significantly benefit from this document. Based on the input gained in Step One and Two, as well as the considerations outlined in Step Three, the following modelling choice was established.

The agreed starting points of eGovernment Core Vocabularies and APs is the formal representation using a conceptual model in the UML Class Diagram. Such UML can be translated automatically into other individual artefacts ^[2]. These unique artefacts must, in effect, cover the needs of open-world (inference model of LOD) as well as the closed world (constraint checking in application) to satisfy the needs of all stakeholders.

The modelling choice is visualised in more detail below (Figure 2) and consists of the following parts.

- Conceptual model represented in UML which serves as the "master" owner of the data model and may import existing External Assets.
- Core Core (Human-oriented annotations), formalising the classes and properties with their human-oriented meaning.

- Open World Inference, modelling aiming to generate additional facts/knowledge in the LOD.
- Closed World Constraint Checking, modelling designed to validate data in the Object-Oriented applications.

Both Open World and Closed World models can be further refined and extended in the context of the application profiles.

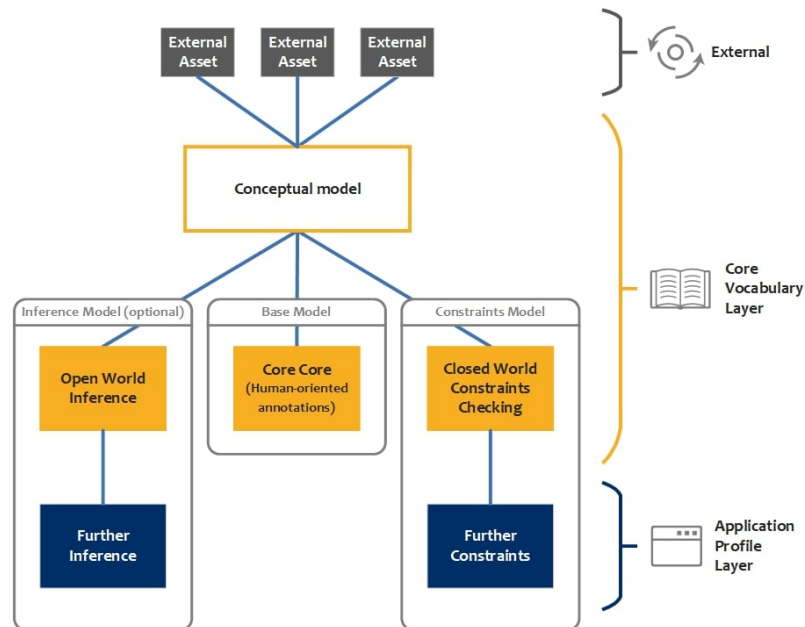


Figure 1. Modelling Approach

On a more concrete level, the UML model for controlled vocabularies is translated into the following expression.

- An RDFS file with human-oriented semantics containing the classes and properties of the vocabulary with their labels, definitions and usage notes. This core model is a semantic asset.
- (OPTIONAL) an RDFS/OWL file, importing and containing constructs explicitly aimed at inferring new facts. `rdfs:domain` and `rdfs:range` belong here and are created intentionally (not automatically derived from the closed model).
- A SHACL file, importing and containing shape for every class from the vocabulary with the included attributes and relationships and their occurrence and value constraints. SHACL over ShEx is chosen since it is a W3C standard and has more mature tool support.

For application profiles, since these are more focusing on constraints, the proposal is to use only a SHACL file, regularly importing the SHACL file from the eGovernment Core Vocabulary for extension and refinement, which itself imports the base vocabulary file with human-oriented information. Of course (optional) RDFS files are useful in the case when additional classes and properties are needed or if one wants to add machine semantics, e.g. inference. The goal in this context is that the base RDFS file and the SHACL file, at minimum, can be generated from the UML model.

An example using actual filenames:

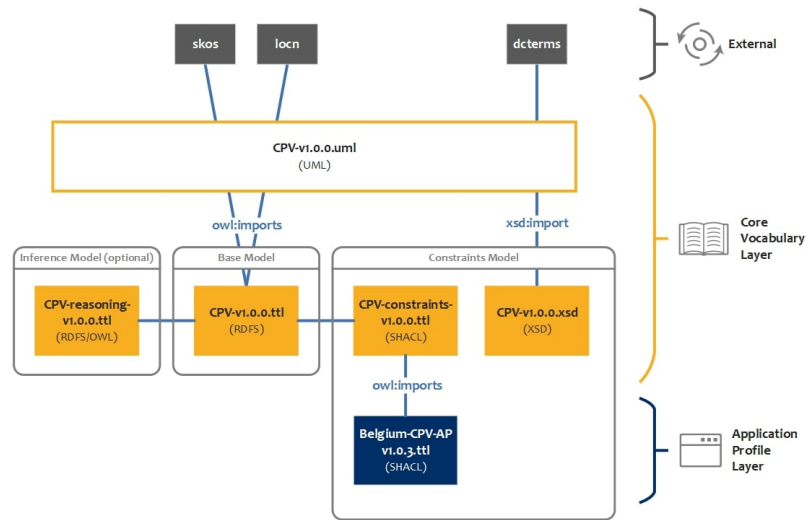


Figure 2. Different modelling artefacts with their dependencies

[1] Within the context of this document, semantic assets are distinguished from reference artefacts. The former includes all types of assets that materialise the Semantic Interoperability (like ontologies, taxonomies, vocabularies, application profiles) while the latter are related to the modelling choice and the respective model presented in subsection Step four: Modelling choice made

[2] As proven by work of the OSLO initiative (see Error: Reference source not found)

Chapter 3. General Guidelines

3.1. Naming and versioning

The first section addresses the generic style guide which applies to all parts of the eGovernment Core Vocabularies.

It covers the following main topics:

- the naming convention for the eGovernment Core Vocabularies/AP
- namespaces for eGovernment Core Vocabularies/AP, constraints and reasoning
- metadata properties and relationship for eGovernment Core Vocabularies and AP
- reuse of foreign/external vocabularies.

Guideline 1: Follow the standard vocabulary naming pattern**

Summary

For the eGovernment Core Vocabularies managed by SEMIC.

- The **topic** of the vocabulary SHOULD be indicated with the first capital letters of the words in the title (Capitalise Each Word).
- For multi-topic vocabularies, and/or SHOULD be dismissed.
- SEMIC CAN be used as **Prefix** in case of overlaps with other vocabularies. **Prefix** is optional.

Description

In case of one topic:

{Prefix}? Core {Topic} Vocabulary

For multi-topic vocabulary – i.e. vocabulary using various topics – the rule should be:

{Prefix}? (Core {Topic} (and|or)) * Core {Topic} Vocabulary

Example

Topic: Person → Core Person Vocabulary, abbreviated as CPV

Topic: Criterion and Evidence → Core Criterion and Core Evidence Vocabulary, abbreviated as CCCEV

Guideline 2: Follow the standard AP naming pattern

Summary

For the Application Profiles managed by SEMIC:

- The **Source** of the vocabulary SHOULD be indicated with the first capital letters of the words in the title (Capitalise Each Word).
- **Publisher** SHOULD define the geographical coverage, usually represented as the organisation/country publishing the AP.

Description

{Publisher}-{Source}-AP

Example

Source: DCAT, Publisher: SEMIC → SEMIC-DCAT-AP

Source: AM, Publisher: Belgium → Belgium-AM-AP

Guideline 3: Follow the standard version naming pattern

Summary

For the eGovernment Core Vocabularies and Application Profiles managed by SEMIC:

- **Topic** SHOULD indicate the eGovernment Core Vocabulary or Application Profile as an abbreviation.
- The version number SHOULD be indicated as vMAJOR.MINOR.PATCH following the semantic versioning ^[3] principle.

Description

{Topic} v{Major}. {Minor}.{Patch}

Where:

- **{Major}** = version when introducing changes that are not backwards compatible;
- **{Minor}** = version when added functionality is backwards-compatible;
- **{Patch}** = version when the change contains only backwards-compatible bug fixes.

Example

Topic: Core Person Vocabulary, Major version: 1, Minor version: 2, Patch: 3 → Core Person Vocabulary v1.2.3, abbreviated as CPV v1.2.3

Guideline 4: Use the standard namespace naming pattern

Summary

For the eGovernment Core Vocabularies and Application Profiles managed by SEMIC:

- The name and version of the standard in the namespace SHOULD be in lower case. See Artefact naming and versioning.
- In constraint namespaces ^[4], ‘constraints’ SHOULD be in between the name and the version number.
- In reasoning namespaces ^[5], ‘semantics’ SHOULD be in between the name and the version number.

Description

Namespace: `{Domain}/{Context}/ns/{Topic}_{Version}`

Constraint and Reasoning Namespace: `{Domain}/{Context}/ns/{Topic}({Type})?{Version}`

Where:

- `{Domain}`= data.europe.eu
- `{Context}`= semanticassets
- `{Topic}`= name of the eGovernment Core Vocabulary | application profile
- `{Type}`= indication of the type of the vocabulary= ‘constraints’ | ‘semantics’
- `{version}`= indication of the version=`v{Major}.{Minor}.{Patch}`

Example

Namespace:

- eGovernment Core Vocabularies namespace: https://data.europe.eu/semanticassets/ns/cpv_v1.0.0
- Application profiles namespace: https://data.europe.eu/semanticassets/ns/dcat-ap_v2.0.1
- Constraints Namespace: https://data.europe.eu/semanticassets/ns/cpv-constraints_v1.0.0.
- Reasoning Namespaces ^[6]: https://data.europe.eu/semanticassets/ns/cpv-semantics_v1.0.0.

Guideline 5: Use the standard file naming pattern

Summary

For the eGovernment Core Vocabularies and Application Profiles managed by SEMIC:

- `{Topic}` SHOULD indicate the name of the eGovernment Core Vocabulary or Application Profile where spaces are replaced with “_”
- The `{fileextension}` SHOULD indicate the file extension: TTL, JSON-LD and XSD.

Description

For the core file:

`{Topic}_{Version}.{fileextension}`

For the semantics and constraints files in rdfs and SHACL:

`{Topic}_{constraints|semantics}_{Version}.{fileextension}`

Where:

`{Version}=v{Major}.{Minor}.{Patch}`

Example

eGovernment Core Vocabularies file name:

- Core_Person_Vocabulary-v1.0.0.ttl
- Core_Person_Vocabulary-v2.1.6.jsonld
- Core_Criterion_and_Core_Evidence_Vocabulary-v2.1.2.xsd
- CPV-v1.0.0.ttl
- CPV-v2.1.6.jsonld
- CCCEV-v2.1.2.xsd

For the semantics and constraints files in rdfs and SHACL:

- Core_Person_Vocabulary-constraints-v1.0.0.ttl
- Core_Person_Vocabulary-semantics-v2.1.6.jsonld
- Core_Criterion_and_Core_Evidence_Vocabulary-v2.1.2.xsd
- CPV-constraints-v1.0.0.ttl

3.2. Reuse of external vocabularies

Given the number of existing vocabularies available in the market, reusing them is considered good practice rather than creating their own or new definitions. For this to be done correctly, it is crucial to approach such reuse with consistency.

The guidelines below ensure such consistency is followed during the reuse process.

Guideline 6: Validate the semantics of reused constructs

Summary

The semantics of the constructs for reuse SHOULD be validated.

Description

Only use constructs with semantics (human and machine-readable) that support the use case or domain. A similar reflex needs to be considered by reusing properties and classes from other vocabularies.

As a general rule, it is safe to reuse annotation properties since no reasoning (semantics) is done with those.

If in doubt, define the class or property in an owned namespace.

Example

The property 'dcterms:title' ^[7] carries only limited semantics. It has a range of 'rdfs:Literal' which represents the set of all literals. While using 'dcterms:title' one adds a statement with 'dce:title' ^[8] due to the sub-property relationship.

On the contrary, 'dcterms:type' ^[9] has a range of 'rdfs:Class', which might be not what one wants if the intention is the use of a 'skos:Concept' ^[10] as value for this attribute. +0.xsd.

Guideline 7: Prioritise maintained vocabularies

Summary

Quality of maintenance and governance SHOULD be reviewed before reuse. Preference SHOULD be given vocabularies that are well maintained and governed.

Description

N/A

Example

The property 'dcterms:title' ^[7] carries only limited semantics. It has a range of 'rdfs:Literal' which represents the set of all literals. While using 'dcterms:title' one adds a statement with 'dce:title' ^[8] due to the sub-property relationship.

On the contrary, 'dcterms:type' ^[9] has a range of 'rdfs:Class', which might not be what one wants if the intention is the use of a 'skos:Concept' ^[10] as value for this attribute. +0.xsd.

Guideline 8: Prioritise vocabularies with machine-readable versions

Summary

Priority SHOULD be given to vocabularies with machine-readable versions.

Description

Preference should go to external vocabularies of which the machine-readable version can be retrieved (dereferenced).

The dereference-ability and continuous maintenance are cornerstones of the Linked Open Data vision and should be critically evaluated when deciding to reuse a vocabulary.

Example

A vocabulary which has problems with several of the guidelines is vann ^[11]. The URI cannot be dereferenced, and the vocabulary appears not to be maintained anymore.

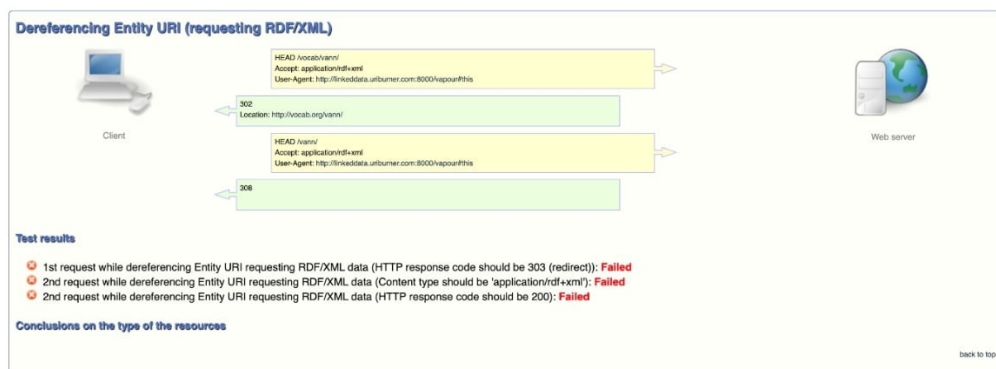


Figure 3. URI dereferencing

Guideline 9: Import reused foreign vocabularies

Summary

Foreign vocabularies SHOULD be reused using import.

Description

When reusing parts of vocabulary, it is crucial to always import the foreign vocabulary in own vocabulary, not copy and paste (in the machine-readable files).

Experience indicates that copying and pasting leads to copy, paste, and edit, often resulting in what is called semantic hijacking. Importing external vocabularies in a dedicated RDF editing environment makes the re-use of the foreign properties and classes secure and error-free.

However, a potential caveat is that RDFS/OWL does not allow selective imports. This means that an import can be too heavy, importing many classes and properties (being the case when, e.g. importing the schema.org vocabulary) while only using a few. In these cases, importing the vocabulary is not advised.

When ‘do not copy and paste’ is mentioned here, it is meant in the machine-readable files. For obvious reasons of convenience, it is advised to copy the content into the human-oriented documentation.

Example

N/A

Guideline 10: Create subclasses in own namespace

Summary

Subclass SHOULD be created in own namespace.

Description

If a class is reflecting a particular subset of items from a class coming from an external vocabulary, then make an explicit subclass in one’s namespace instead of reusing the URI of the external vocabulary.

Example

N/A

Guideline 11: Respect meaning of foreign vocabulary

Summary

Meaning of the foreign vocabulary SHALL NOT be altered.

Description

In case copy and paste is still done, do not change the information. Hijacking another vocabulary is considered bad practice. If there are more specific definitions, semantics, and usage notes, place them in classes and properties in their namespace. One could then still map these, even later on, to foreign constructs.

Example

According to the AAA principle, anything can be stated – e.g. that the label of `dcterms:title` is ‘placename’, but it is important to imagine what users see at a merged version of one’s vocabulary and the source vocabulary:

Property Form

URI: <http://purl.org/dc/terms/title>

Annotations

`rdfs:comment`

- A name given to the resource. (@en-US)
- This property contains a name given to the Catalogue. This property can be repeated for parallel language versions of the name. (@en)
- This property contains a name given to the Data Service. This property can be repeated for parallel language versions of the name. (@en)

`rdfs:isDefinedBy`

- <http://dublincore.org/documents/dcmi-terms/#terms-title>

`dcterms:`

`rdfs:label`

- Catalogue Record - title (@en)
- Data Catalogue - title (@en)
- Data Service - title (@en)
- Title (@en-US)

Figure 4. Merged Vocabularies

What needs to be achieved is the ability to supply the user with context-dependent tips. The issue is that in RDFS, properties are and stay global, meaning this is not the way to accomplish the ultimate legitimate goal. How to achieve this correctly is addressed later in this document.

The same issue but with heavier semantics is presented in the following figure:

Property Form

URI: <http://www.w3.org/ns/org#purpose>

Annotations

Property Axioms

`rdfs:domain`

- cv:PublicOrganisation
- org:Organization

Figure 5. Semantic Hijacking

The above is an example of misusing domain to attach this property to a class in its namespace.

In this context, the triples:

lead to :CompanyX being a cv:PublicOrganisation.

[15] <http://xmlns.com/foaf/0.1/>

[16] <http://www.w3.org/2004/02/skos/core#>

Guideline 14: Diagonal lines

Summary

Straight lines are more comfortable to follow, therefore diagonal or curved lines SHOULD be avoided.

Description

The reader of the diagrams needs to have a clear understanding of the diagram content. If the diagram is well organised, and if associations between classes are well separated and run in parallel – i.e. lines never cross or the least possible – the diagram will gain clarity and will become less error-prone.

Example

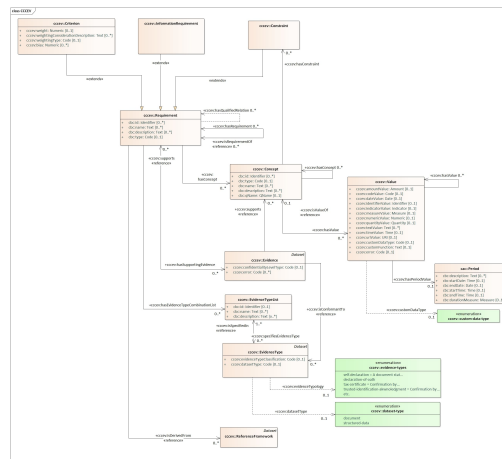


Figure 8. UML diagram example

Guideline 15: Leave space between lines

Summary

Lines SHOULD have sufficient spacing in between them.

Description

The reader of the diagrams needs to have a clear understanding of the diagram content. If the diagram is well organised, and if associations between classes are well separated and run in parallel – i.e. lines never cross or the least possible – the diagram will gain clarity and will become less error-prone.

Example

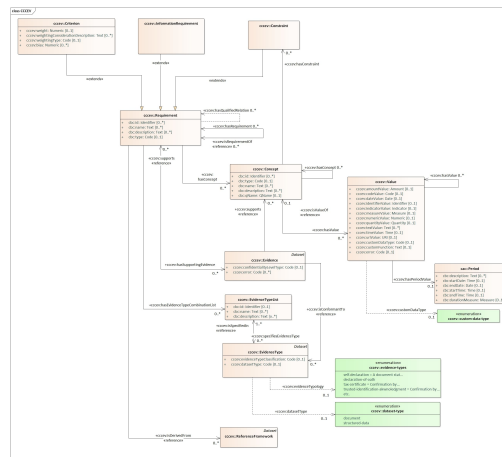
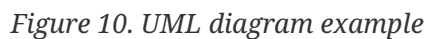


Figure 9. UML diagram example

Summary

Description

Example



Guideline 17: Symbol size

Summary

Symbols **SHOULD** be of the same size (unless one wants to draw attention to a specific element), as nothing should draw more attention unless aimed.

Description

The reader of the diagrams needs to have a clear understanding of the diagram content. If the diagram is well organised, and if associations between classes are well separated and run in parallel – i.e. lines never cross or the least possible – the diagram will gain clarity and will become less error-prone.

Example

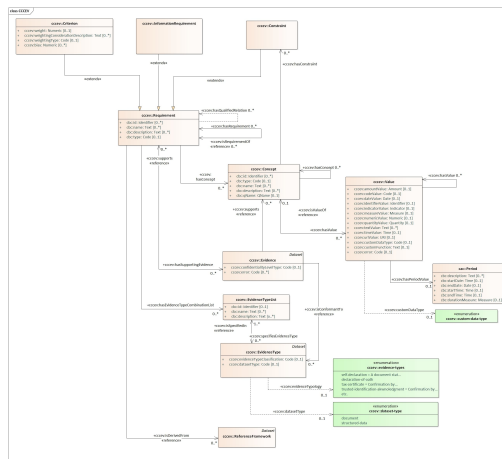


Figure 11. UML diagram example

Guideline 18: Reading direction

Summary

In Europe, people generally read left to right and top to bottom, hence diagrams **SHOULD** be organised left to right and top to bottom.

Description

The reader of the diagrams needs to have a clear understanding of the diagram content. If the diagram is not well organised and the associations between classes are not clear, the diagram becomes more error-prone.

Example

N/A

Guideline 19: Short and concise

Summary

Too many details may diminish readability and understandability, thus diagrams **SHOULD** contain only the necessary level of detail.

Description

If the diagram offers the reader too much detailed information, it can overburden them and hinder their understanding of the diagram elements.

Example

N/A

Guideline 20: Notations use

Summary

More specialised symbols have the danger of not being broadly understood, therefore only well-known notations **SHOULD** be used.

Description

If symbols are used that are too specific for the reader to understand the diagram's content could be improperly conveyed the potentially less knowledgeable readers.

Example

N/A

Guideline 21: Control diagrams size

Summary

Larger diagrams can be difficult to read, thus they **CAN** be split into multiple smaller ones.

Description

The reader of the diagram needs to have a clear understanding of the diagram's content. If the diagram lines are too long to be comfortably followed the diagram will become more error-prone.

Example

N/A

Guideline 22: One-page diagrams are preferred

Summary

As they are more comfortable to read, one-page diagrams SHOULD be used when possible.

Description

The reader of the diagrams needs to have a clear understanding of the diagram's content. If the diagram is divided into many small versions, the reading effort is higher, becoming more error-prone.

Example

N/A

Guideline 23: Element metadata

Summary

Each UML element SHALL be annotated with the following metadata:

- there SHALL be one `rdfs:label` containing the human-oriented label in English;
- there CAN be more `rdfs:label` but only one per language;
- there SHALL be one `rdfs:comment` in English (1..1) containing the concept's intention, being the definition of the element (the class, the attribute, the association);
- `skos:altLabel` (0..n) CAN be used to indicate synonyms;
- `skos:scopeNote` in English (0..1) CAN be used to clarify how, where and when to use the UML element.

Description

To ensure a minimum content richness, certain metadata should be added to the UML elements.

Example

N/A

4.2. eGovernment Core Vocabularies

eGovernment Core Vocabularies can be considered generic domain models that capture the essential concepts (classes) in a domain together with the appropriate attributes and associations. This means that they stay at a rather abstract level and independent of a concrete application. The main aim is reducing them globally and helping in the interchange of data, primarily of data related to base registries managed by governments, such as base registries of businesses, addresses, parcels, and so on.

The following image depicts the above:

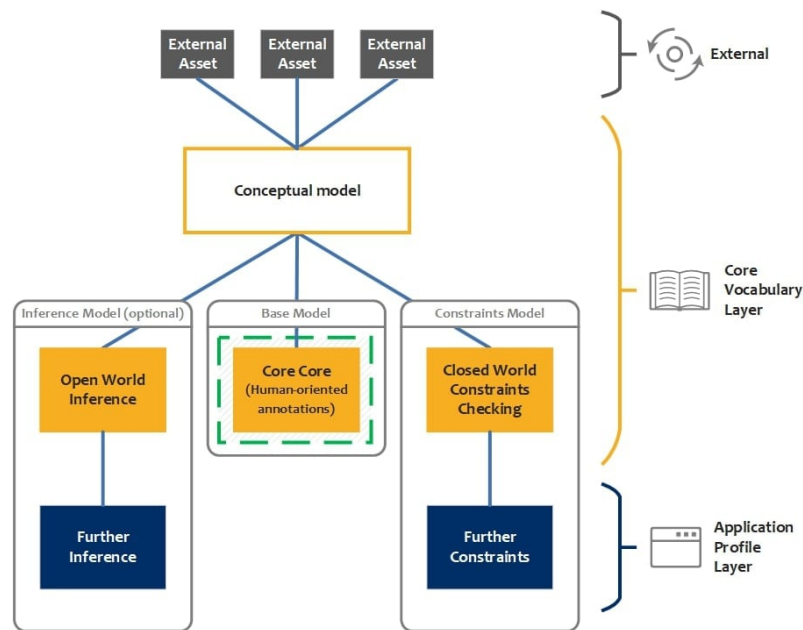


Figure 12. Core vocabularies relation with external domain layers

The following UML artefacts are covered in this section:

- [Packages](#) which group all modelling elements of a eGovernment Core Vocabulary and which translate into an owl:Ontology in the semantic world
- [Classes](#)
- [Attributes](#)
- [Associations](#)
- [Generalisations](#)
- [Aggregations](#)
- [DataTypes](#)

Packages

A package is a mechanism that allows grouping of modelling elements such as classes or data types, but can also contain other packages itself. The following guidelines describe the way a package must be created.

Guideline 24: One package per vocabulary

Summary

A package SHALL exist for each vocabulary.

Description

Every vocabulary should be contained in its own container so that it can be easily imported in other environments.

Example



Figure 13. Package name

Guideline 25: Use PascalCase to name the package

Summary

A package SHALL have a name in UpperCamelCase (or PascalCase). This can be the plural of the name of the main concept. For example: Agents, Organisations, Locations, AcademicEducations.

Description

UpperCamelCase is an established practice. The plural name reflects the containment of multiple elements.

Example

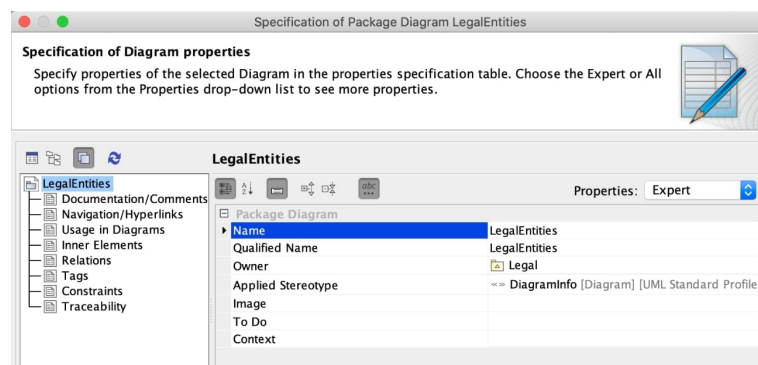


Figure 14. Diagram properties

Guideline 26: Use package URI annotation

Summary

A package SHALL be annotated with the namespace identifier as discussed in Section [Artefact naming and versioning](#) at the URI property of the package.

Description

Every element needs to have a unique identifier.

Example

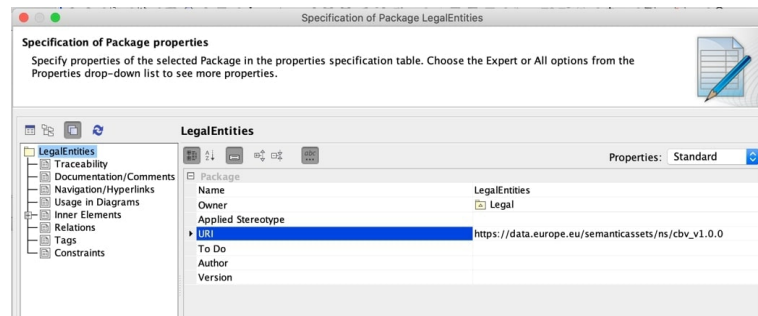


Figure 15. Example of a package URI

It is assumed that this URI is the URI of the eGovernment Core Vocabulary, e.g. ontology and that this URI concatenated with "#" constitutes the base URI.

Example:

Ontology URI: https://data.europe.eu/semanticassets/ns/cbv_v1.0.0

Base URI: https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#

Guideline 27: Import external packages

Summary

Packages of external namespaces SHOULD be imported, if available.

Description

Importing external packages ensures the long-term availability thereof.

Example

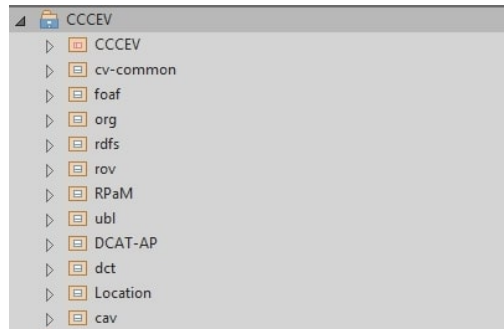


Figure 16. Example of an imported package

Classes

A class represents a concept from one's domain, and defines a set of 'similar' objects such as persons or cars. The symbol to represent a class is a rectangle with multiple compartments. The guidelines below are related to the creation of a class in a UML diagram.

Guideline 28: Use one class per domain concept

Summary

Every domain concept SHALL be represented as a class.

Description

This is an established practice.

Example

The concept 'Agent' will be represented by a Class.

Guideline 29: Use singular nouns for naming classes

Summary

As common practice dictates, each class SHALL have a name in the form of a singular noun in UpperCamelCase (Pascal Case).

Description

A real name in the domain or a name that stakeholders already know and use, should be used.

Example



LegalEntity

Figure 17. Class name example

Attributes

An attribute represents a structural feature (i.e. properties) of a class, for which each object in the class can have data values. Most often, there is only one data value (height, weight, id, etc.), but multiple values are also possible (e.g. phone numbers). Attributes are to be included in the second compartment of the class diagram.

The guidelines below describe how attributes need to be formulated within a class.

Guideline 30: Use one attribute per atomic value

Summary

Every structural feature that has an **atomic value** (preferably using a built-in XML schema simple type) or an **enumeration type** SHALL be represented as an attribute.

Description

Every structural feature that has a **class** as data type shall **not** be represented as an attribute. Instead, this class and an association to the corresponding class will be included in the model. See the example below. One could consider adding an attribute "registeredAddress" of type Address to a class "LegalEntity". Instead, Address should be modelled as a class, and an association "registeredAddress" between LegalEntity and Address should be added.

Example

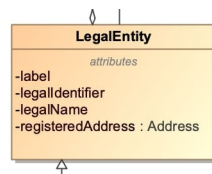


Figure 18. Incorrect structural features representation



Figure 19. Correct structural features representation

Incorrect representation of structural features with a class as datatype (top), should be modelled as an association to the respective class instead of an attribute (bottom).

Guideline 31: Use lowerCamelCase and singular for attribute naming

Summary

Each attribute SHALL have a name in the form of a singular noun in lowerCamelCase (or Dromedary Case).

Description

Use the real name in the domain or a name that stakeholders already know and use. This name appears in the second compartment of the rectangle.

Example

Attribute name:

firstName, lastName, primaryPhone

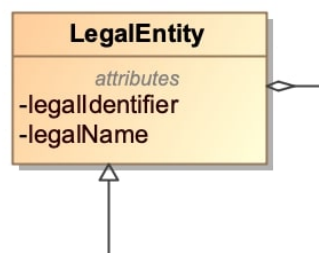


Figure 20. Attribute name example

Guideline 32: Specify XSD type

Summary

The XSD type (or the enumeration type of the attribute) SHALL be indicated after its name followed by ‘.’.

Description

See [DataTypes](#) for more details on the data types allowed.

Example

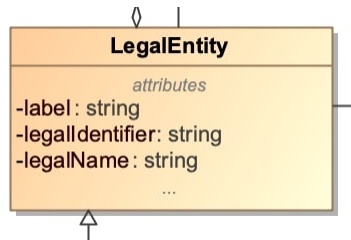


Figure 21. Attributes with simple data type (e.g. `xsd:string`)

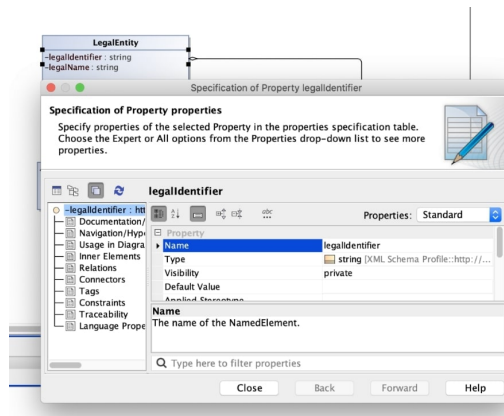


Figure 22. XSD Type indication example

Guideline 33: Specify attribute multiplicity

Summary

All attributes at the eGovernment Core Vocabulary level SHALL be indicated with the multiplicity of 0..* (with one exception, label, as noted in the next Guideline).

Description

See [DataTypes](#) for more details on the data types allowed.

Example

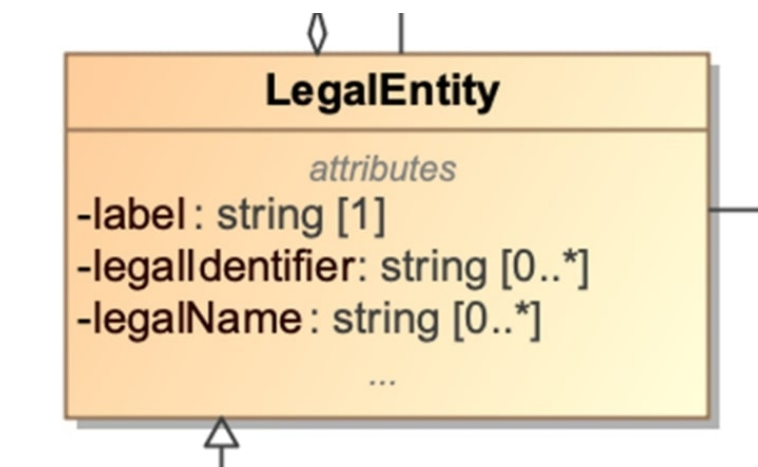


Figure 23. Multiplicity example

Guideline 34: Specify attribute label

Summary

Every class SHALL have a label attribute attached (with the multiplicity of 1..*) with the following constraints:

- There SHALL be one label in English;
- There CAN be more `rdfs:labels` but only one per language;
- This attribute needs to be annotated as belonging to the <http://www.w3.org/2000/01/rdf-schema#> namespace (`rdfs:label`).

Description

This is a fundamental usability requirement. Every instance of any class should have one `rdfs:label`. This `rdfs:label` will be used then when presenting the data as linked open data. The user gets a human-readable label instead of an unreadable URI.

Example



Figure 24. Required label on every class

Associations

Associations model possible relationships between instances of the classes. In Semantic Web languages, such as RDF, an association is a binary relation: it is used to link two individuals. Due to this nature we focus on binary associations in UML. This binary association also translates easily into an xml serialisation.

```
<Subject id="S1"><relatedTo><Object id="O1"/></relatedTo></Subject>
```

In UML, by default, navigability in both directions is assumed. For translation to RDF, navigability should be explicitly specified, except in the case of inverse relationships.

The following guidelines concern the association relationships between the two classes.

Guideline 35: Use solid lines for class associations

Summary

For the association relationships between the two classes:

- associations SHALL be drawn using a solid line between the classes involved;
- navigability SHALL be specified and illustrated with an open arrowhead pointing from source to target.

Description

The choice was made to make associations more explicit by drawing them as a line. Since RDF triples describe a binary relationship in 1 direction, this is already reflected in the UML class diagram.

Example

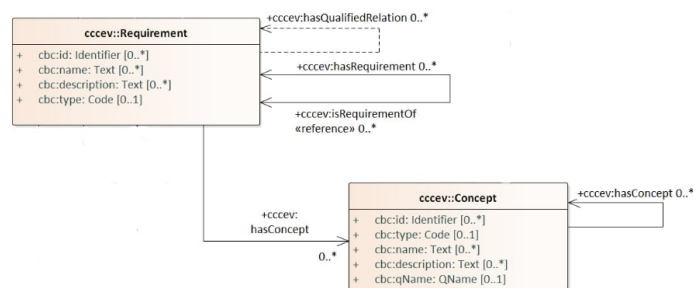


Figure 25. Association between `cccev:Requirement` and `cccev:Concept`

Guideline 36: Use meaningful phrases for class associations

Summary

The navigable association end SHALL be named as indicated below.

- Using a verb phrase that is meaningful on its own and also creates a meaningful contextualised sentence following the (source)ClassName-verbPhrase-(target)ClassName format. See first image in the example below.

Note: A verb like 'has' on its own is not a meaningful verb phrase and therefore should not be used.

- Using a noun that is meaningful on its own and in the context of the association qualifies the target Class name. See second image in the example below.

Description

This label needs to be readable and meaningful for the stakeholders to understand the association between the classes.

Example

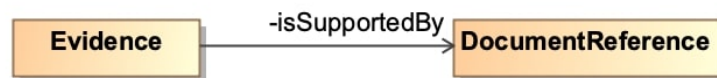


Figure 26. Association example named according to (source)ClassName-verbPhrase-(target)ClassName

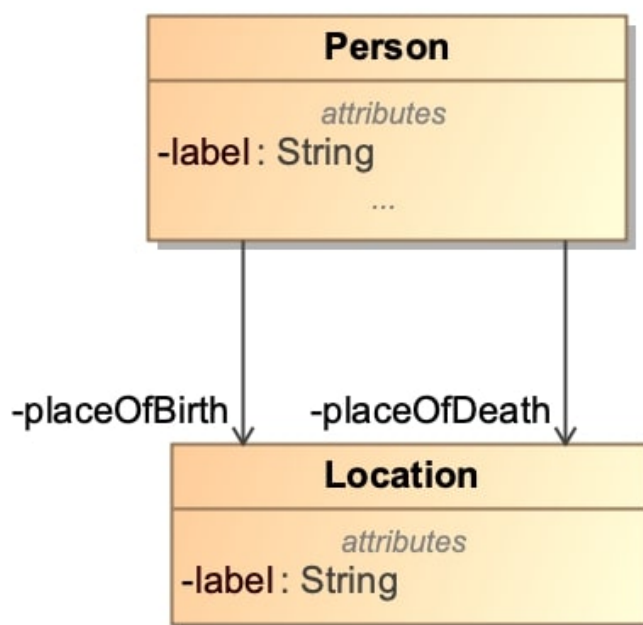


Figure 27. Association example named according to the qualification of target ClassName

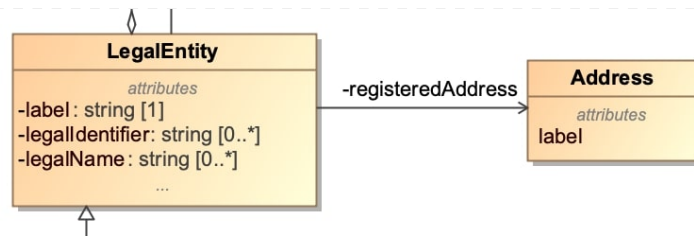


Figure 28. Association Example

Guideline 37: Depict inverse class associations

Summary

When an inverse association (relation in the other direction) represents the exact inverse of an already modelled relationship, this **SHOULD NOT** be drawn using a separate directed line.

Instead, this **SHOULD** be indicated as an association with navigability in both directions. Both ends shall have an appropriate name.

Description

The readability of the association could be hindered by using two different lines to depict the relation.

Example

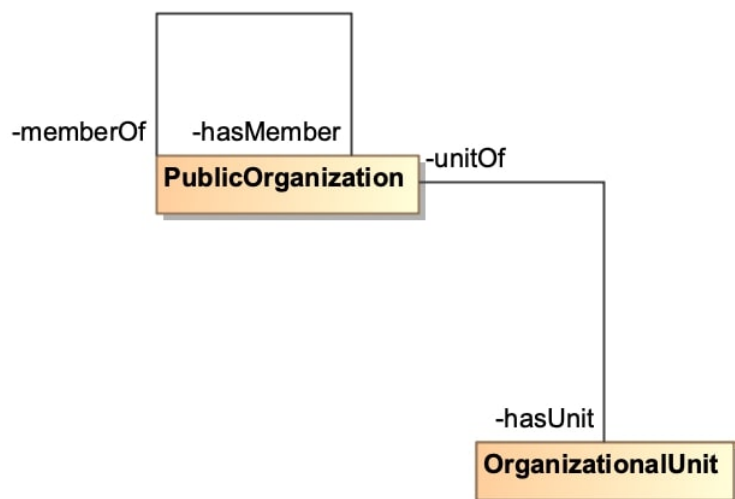


Figure 29. Inverse associations resulting from bi-directional navigability

Guideline 38: Specify association multiplicity

Summary

All associations at the eGovernment Core Vocabulary level SHALL be indicated with the multiplicity of 0..*.

Description

The readability of the association could be hindered by using two different lines to depict the relation.

Example

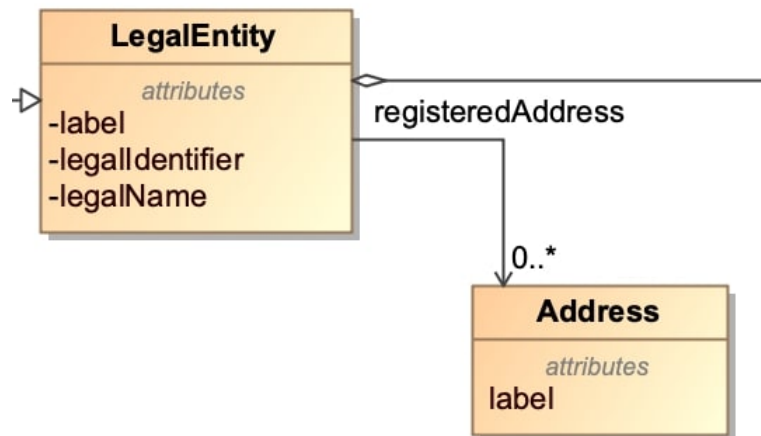


Figure 30. Multiplicities on associations

Generalisations

Generalisation allows deriving more specific classes from existing classes as a result of this inheriting the attributes and the associations that are specified for the superclass. These are passed on to the subclasses.

The following guidelines below focus on the generalisation relationships between the two classes.

Guideline 39: Use generalisation notation for inheritance

Summary

As it is an established practise, the generalisation notation SHALL be used to indicate inheritance and subclassing.

Description

The inheritance relation between two classes must be clear.

Example

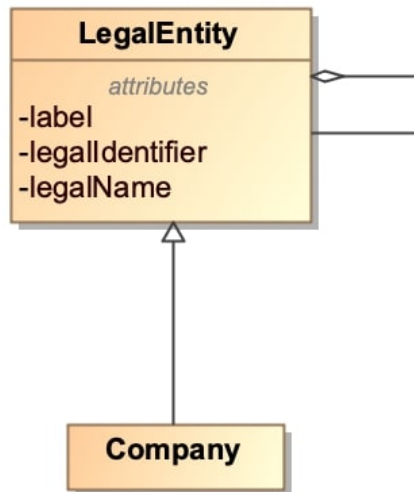


Figure 31. Generalisation example

Aggregations

An aggregation is a particular form of association used to express that instances of one class **are parts of** an instance of another class.

A distinction is made between two types:

1. **Shared aggregation:** expresses a weak belonging of the parts to the whole, meaning that the parts can still exist if the whole is deleted.
2. **Composition:** expresses a strong belonging of the part to the whole; its existence depends on the presence of its container.

A shared aggregation is indicated by a hollow diamond, while a composition is indicated by a solid diamond at the side of the containing class.

The UML semantics are however unclear, in particular for shared aggregation. The composition is more clearly defined, but mainly refers to the storage, namely "the composite object has responsibility for the existence and storage of the composed objects" ^[19]. The degree of belonging (weak or strong) leads to different serialisations in the XML artefacts as explained in [Associations](#).

The composition association can be used when such containing serialisation effects are desired.

Guideline 40: Use binary association

Summary

The notion of shared aggregation SHALL NOT be used. Instead, a regular binary association shall be used. ^[20]

Description

N/A

Example

N/A

Guideline 41: Avoid Composite aggregation

Summary

The notion of composite aggregation SHALL ONLY be used when a part has no own identifier and needs to be stored inside the whole.

Description

When no reference to an identifier is possible, the parts need to be contained in the whole. This also maps easily on XML containment.

Example

An OrderLine only exists as part of an Order and is identified using the Order.

DataTypes

UML makes a distinction between data types, primitive data types, and enumerations.

- Data Types are visualised in almost the same way as a class. One difference is that the name of the data type is annotated with the keyword `<< datatype >>`. Data Types can have an internal structure in the form of attributes.
- Primitive data types `<< primitive >>` do not have an internal structure.
- Enumerations `<< enumeration >>` are data types whose values are defined in a list. These values are literals.

Guideline 42: Use XSD simple data types

Summary

The UML class diagram SHOULD use the built-in XSD simple data types ^[19] (similarly to their usage in XSD and RDFS).

Description

It is always good to consider if `<< enumeration >>` lists CAN be expressed as a `skos:ConceptScheme` with the list nodes as `skos:Concepts`.

Example

The “Type” in the image below:

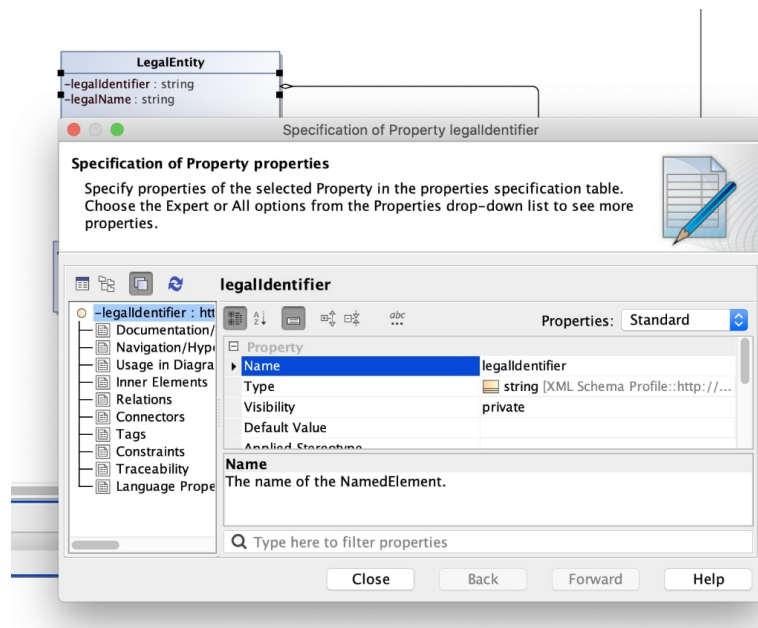


Figure 32. XSD Type indication example

4.3. Application Profiles

Application profiles define a data model to be used in a more specific context or application. For this reason additional constraints and rules are expected to be added to the more generic models as supplied by the eGovernment Core Vocabularies. The focus of this section is to provide the guidelines for the artefacts used in the Application Profiles of the eGovernment Core Vocabularies.

- [Packages](#) that group all modelling elements of a eGovernment Core Vocabulary, and that translate into an owl:Ontology in the semantic world.
- [Classes](#)
- [Attributes](#)
- [Associations](#)
- [Generalisations](#)
- [Aggregations](#)
- [DataTypes](#)

Packages

A package is a mechanism that allows for the grouping of modelling elements such as classes or data types, but which can also contain other packages itself. The guidelines that follow describe the way a package has to be created.

Guideline 43: One package per vocabulary

Summary

A package SHALL exist for each vocabulary.

Description

Every vocabulary should be contained in its own container so that it can be easily imported in other environments.

Example

.Agents package example image::one_package_per_vocabulary.jpg[Agents package example, 60%, 60%, align="center"]

Guideline 44: Naming the package

Summary

A package SHALL have a name in UpperCamelCase (or PascalCase). This can be the plural of the name of the main concept. For example: Agents, Organisations, Locations, AcademicEducations.

Description

UpperCamelCase is an established practice. The plural name reflects the containment of multiple elements.

Example

Package Name: Agents, Organisations, Locations.

Guideline 45: Package URI annotation

Summary

A package SHALL be annotated with the namespace identifier as discussed in Section [Naming and versioning](#) at the URI property of the package.

Description

Every element needs to have a unique identifier.

Example

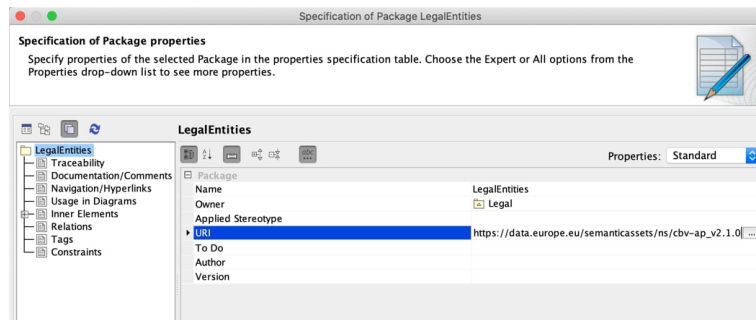


Figure 33. Namespace example

Ontology URI: https://data.europe.eu/semanticassets/ns/cbv-ap_v2.1.0

Base URI: https://data.europe.eu/semanticassets/ns/cbv-ap_v2.1.0#

Classes

The guidelines formulated for the eGovernment Core Vocabularies [Classes](#) can be followed here as well. With an additional optional guideline.

Guideline 46: Specify use

Summary

A class CAN contain an annotation that denominates whether it is the following:

- **Mandatory:** a receiver of data SHALL be able to process information about instances of the class; a sender of data SHALL provide information about instances of the class.
- **Recommended:** a sender of data SHOULD give information about instances of the class; a sender of data SHALL provide information about instances of the class if such information is available; a receiver of data SHALL be able to process information about instances of the class if such information is available.
- **Optional:** a receiver SHALL be able to process information about instances of the class if such information is available; a sender MAY provide the information but is not obliged to do so.

Description

This is an established practice.

Example

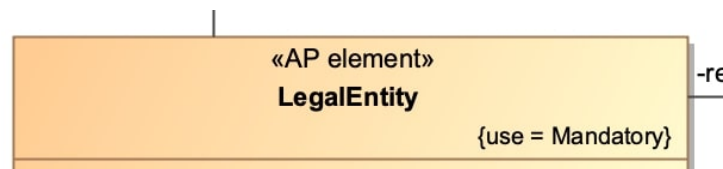


Figure 34. Use annotation example

Note: this annotation has been implemented as a tag with name 'use' with the tree allowed values enumerated on a stereotype with name "AP element".

Attributes

The guidelines formulated for the eGovernment Core Vocabularies [Attributes](#) can be followed here as well. With an additional optional guideline.

Guideline 47: Specify attribute multiplicity and use

Summary

All attributes SHALL contain a multiplicity indication.

In addition, any association CAN contain an annotation that denominates whether it is:

- **Mandatory:** a receiver SHALL be able to process the information for that property; a sender SHALL provide the information for that property. This is aligned with multiplicities (1..1 or 1..*).
- **Recommended:** a receiver SHALL be able to process the information for that property; a sender SHOULD provide the information for that property if it is available. This is aligned with multiplicities (0..1, 0..*).
- **Optional:** a receiver SHALL be able to process the information for that property; a sender MAY provide the information for that property but is not obliged to do so. This is aligned with multiplicities (0..1, 0..*).

Note: multiplicity needs to be explicitly set since this is not done at the eGovernment Core Vocabulary level.

Description

The eGovernment Core Vocabularies leave the multiplicities completely open [0..*]; while the application profiles need to be precise. In any case, one can optionally add the annotation if the attribute is considered Mandatory, Recommended, or Optional. These annotations must be aligned with the multiplicity indicators defined above.

Example

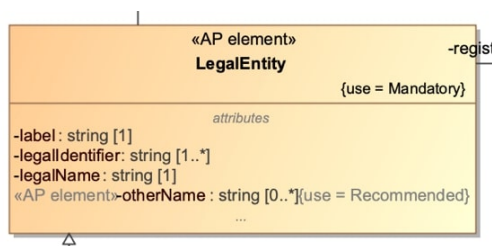


Figure 35. Example of attribute multiplicity

Associations

The guidelines formulated for the eGovernment Core Vocabularies [Associations](#) can be followed here as well. With an additional optional guideline.

Guideline 48: Specify association multiplicity and use

Summary

All associations SHALL contain a multiplicity indication.

In addition, any association CAN contain an annotation that denominates whether it is:

- **Mandatory:** a receiver SHALL be able to process the information for that property; a sender SHALL provide the information for that property. This is aligned with multiplicities (1..1 or 1..*).
- **Recommended:** a receiver SHALL be able to process the information for that property; a sender SHOULD provide the information for that property if it is available. This is aligned with multiplicities (0..1, 0..*).
- **Optional:** a receiver SHALL be able to process the information for that property; a sender MAY provide the information for that property but is not obliged to do so. This is aligned with multiplicities (0..1, 0..*).

Note: multiplicity needs to be explicitly set since this is not done at the eGovernment Core Vocabulary level.

Description

The eGovernment Core Vocabularies leave the multiplicities completely open [0..*]; while the application profiles need to be precise. Besides, one can optionally add the annotation if the attribute is considered Mandatory, Recommended or Optional. These annotations must be aligned with the multiplicity indicators defined above.

Example

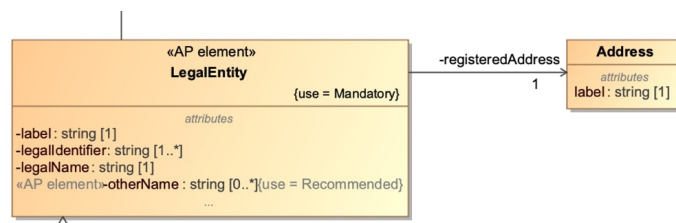


Figure 36. Example of legal entity association use with otherName

Generalisations

The guidelines formulated for the eGovernment Core Vocabularies [Generalisations](#) can be followed here as well.

Aggregations

The guidelines formulated for the eGovernment Core Vocabularies [Aggregations](#) can be followed here as well. With an additional guideline listed below.

Guideline 49: Specify composition multiplicity

Summary

As multiplicity is not specified at the eGovernment Core Vocabulary level, all compositions SHALL contain a multiplicity indication.

Description

The eGovernment Core Vocabularies leave the multiplicities completely open [0..*]; while the application profiles need to be precise. Compositions follow the same rules as for associations.

Example

N/A

DataTypes

The guidelines formulated for the eGovernment Core Vocabularies [DataTypes](#) can be followed here as well.

[17] Lindland, Odd & Sindre, Guttorm & Sølvsberg, Arne. (1994). Understanding Quality in Conceptual Modelling. IEEE Software. 11. 42-49

[18] The Elements of UML 2.0 style, Scott W. Ambler, Cambridge University Press, 2005

[19] <https://download.eclipse.org/modeling/mdt/uml2/javadoc/3.0.0/org/eclipse/uml2/uml/AggregationKind.html>

[20] Monique Snoeck, UML Class Diagrams for Software Engineering, edX, 2020

Chapter 5. RDF guidelines

5.1. Vocabulary annotations in RDFS

To solve the closed and open world mismatch, the proposal is made to separate the different concerns into different files:

- An RDFS file (required) with human-oriented semantics containing labels, definitions, scope and usage note that will serve two other models
- One (optional) in RDFS/OWL, aimed at inferring new facts
- One in SHACL (required) that defines the classes with their attributes and relationships, and the constraints on occurrences and values used

The focus of this section is on providing guidelines for core source RDFS files with human-oriented semantics (see Figure 1 below).

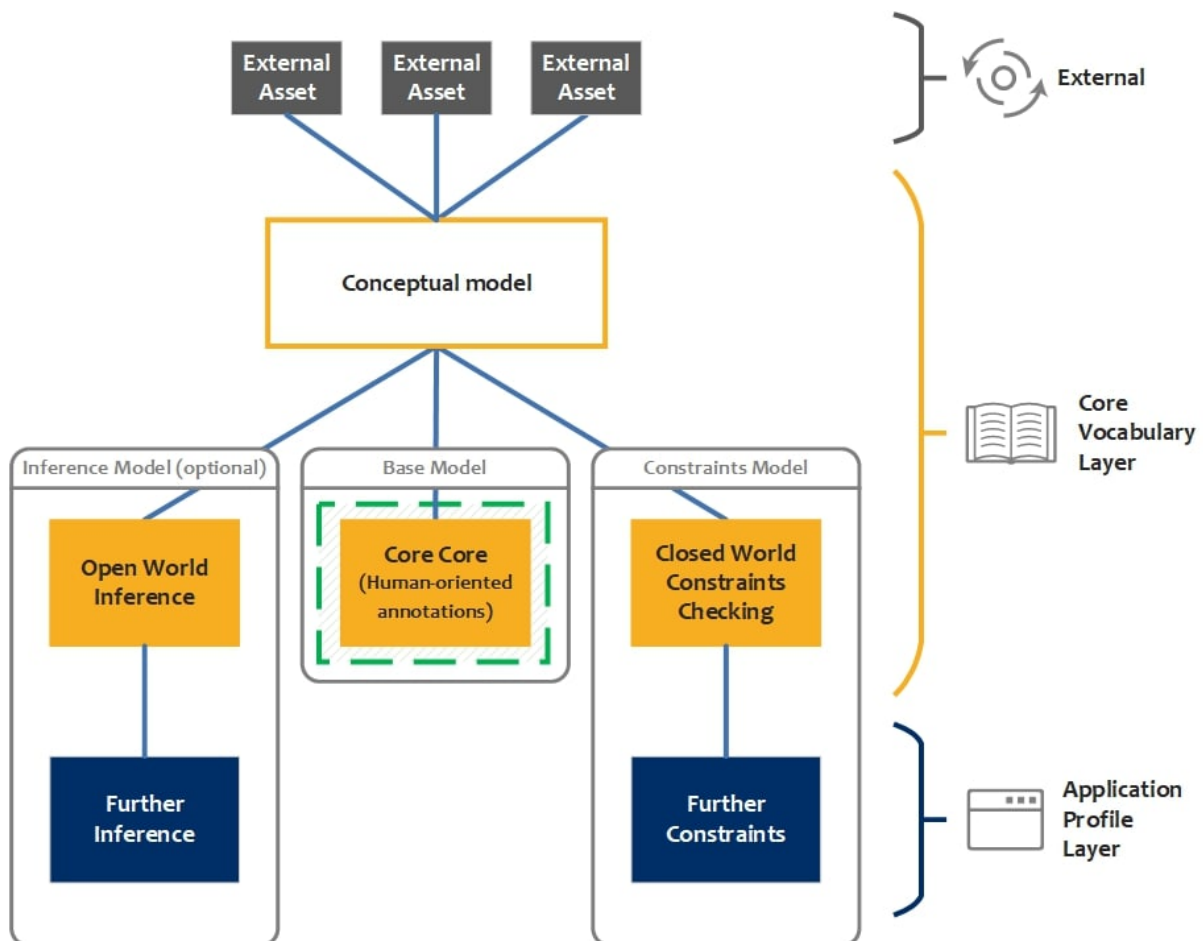


Figure 37. Situating RDFS for human consumption

The subsections below are a breakdown of the RDFS file and contain detailed related guidelines.

- [Environment set up](#): the guidelines for setting up the environment while creating an RDFS file
- [Vocabulary](#): the guidelines for vocabularies such as declaring the namespaces, appropriately

importing the other vocabularies and adding metadata in an RDFS file

5.1.1. Environment Setup

The guidelines for setting up the environment while creating an RDFS file are described below.

Guideline 50: Set up the environment

Summary

The namespace and the base URI of the vocabulary SHALL be defined according to the rules set in [Naming and versioning](#).

Each vocabulary should be uniquely identifiable.

Description

Important components are the domain and the name of the vocabulary with sa(semantic asset)/cv(core vocabulary) in between.

Example

Namespace: https://data.europe.eu/semanticassets/ns/cbv_v1.0.0

Base URI: https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#

Aggregated xample

```
@prefix cb: <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#> . ①
@prefix owl: <http://www.w3.org/2002/07/owl#> .
<https://data.europe.eu/semanticassets/ns/cbv_v1.0.0> ①
  a owl:Ontology;.
```

① Guideline 50

Guideline 51: Set up the environment II

Summary

Prefixes for all used vocabularies, including own one, SHALL be declared.

Description

The form of a prefix is followed by the base URI. Declaring the prefixes simplifies the reading of the vocabulary; otherwise, long URIs are exposed to the readers.

Example

```
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix cb: <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#> .
```

Aggregated example

```
@prefix cb: <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#> . ①
@prefix adms: <http://www.w3.org/ns/adms#> . ②
@prefix cc: <http://creativecommons.org/ns#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix locn: <http://www.w3.org/ns/locn#> .
@prefix org: <http://www.w3.org/ns/org#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix person: <http://www.w3.org/ns/person#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix schema: <http://schema.org/> .
@prefix sioc: <http://rdfs.org/sioc/ns#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix vann: <http://purl.org/vocab/vann/> .
@prefix wdrs: <http://www.w3.org/2007/05/powder-s#> .
@prefix xhtm: <http://www.w3.org/1999/xhtml#> .
@prefix xhv: <http://www.w3.org/1999/xhtml/vocab#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#> ①
  a owl:Ontology;
```

① Guideline 50

② Guideline 51

5.1.2. Vocabulary

Description

The following guidelines describe the way vocabularies need to be applied and used.

Guideline 52: The vocabulary

Summary

An ontology SHALL be declared using the namespace of the vocabulary.

Every subject needs to have type indication, hence also the vocabulary itself.

Description

The form of a prefix is followed by the base URI.

Example

```
<https://data.europe.eu/semanticassets/ns/cbv_v1.0.0> a owl:Ontology.
```

Aggregated example

```
@prefix cb: <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#> . ①  
@prefix adms: <http://www.w3.org/ns/adms#> . ②  
<https://data.europe.eu/semanticassets/ns/cbv_v1.0.0> ①  
  a owl:Ontology;. ③
```

① Guideline 50

② Guideline 51

③ Guideline 52

Guideline 53: The vocabulary II

Summary

Only one ontology SHALL be declared per RDF(S) file.

Description

RDF tools are sensitive to this and throw warnings if multiple exist.

Example

Aggragated example*

```
@prefix cb: <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#> . ①
@prefix adms: <http://www.w3.org/ns/adms#> . ②

<https://data.europe.eu/semanticassets/ns/cbv_v1.0.0> ①
  a owl:Ontology;. ③
```

① Guideline 50

② Guideline 51

③ Guideline 52

Guideline 54: The vocabulary III

Summary

External vocabularies, if appropriate, SHALL be imported.

Description

The rationale for this guideline is to be found in Section [Reuse of external vocabularies](#).

Example

```
<http://semic.eu/sa/cv/cbv_v1.0.0> a owl:Ontology;  
owl:imports <http://purl.org/dc/terms/>;  
owl:imports <http://xmlns.com/foaf/0.1/> .
```

Aggregated example

```
<https://data.europe.eu/semanticassets/ns/cbv_v1.0.0> ①  
@prefix adms: <http://www.w3.org/ns/adms#> . ②  
a owl:Ontology; ③ ④  
owl:imports dterms;; ⑤  
owl:imports <http://www.w3.org/2004/02/skos/core>;  
owl:imports <http://www.w3.org/ns/locn>;  
owl:imports org;;  
owl:imports foaf;;  
.
```

① Guideline 50

② Guideline 51

③ Guideline 52

④ Guideline 53

⑤ Guideline 54

Guideline 55: The vocabulary IV

Summary

The ontology SHALL have a label `rdfs:label` and a description `rdfs:comment`.

Description

Every vocabulary construct should be annotated so it can be easily understood and used; also the vocabulary itself.

Example

```
<https://data.europe.eu/semanticassets/ns/cbv_v1.0.0> a owl:Ontology;  
  rdfs:label "SEMIC Core Business Vocabulary"@en;  
  rdfs:comment "This file specifies the set of RDF classes and properties used for  
Legal Entities"@en;
```

Aggregated example

```
@prefix cb: <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#> . ①  
@prefix adms: <http://www.w3.org/ns/adms#> . ②  
...  
<https://data.europe.eu/semanticassets/ns/cbv_v1.0.0> ①  
  a owl:Ontology; ③ ④  
  rdfs:label "SEMIC Core Business Vocabulary"@en; ⑥  
  rdfs:comment "This file specifies the set of RDF classes and properties used for  
Legal Entities"@en; ⑥  
  owl:imports <http://purl.org/dc/terms/>; ⑤  
  owl:imports <http://www.w3.org/2004/02/skos/core>;  
  owl:imports <http://www.w3.org/ns/locn>;  
  owl:imports <http://www.w3.org/ns/org#>;  
  owl:imports <http://xmlns.com/foaf/0.1/>;  
  .
```

① Guideline 50

② Guideline 51

③ Guideline 52

④ Guideline 53

⑤ Guideline 54

⑥ Guideline 55

Guideline 56: The vocabulary V

Summary

Metadata SHALL be assigned to the vocabulary (see Metadata).

Description

This is established good practice. This helps to find the vocabulary at semantic asset portals and to evaluate if a vocabulary suits the needs.

Example

```
@prefix adms: <http://www.w3.org/ns/adms#> .
...
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
<https://data.europe.eu/semanticassets/ns/cbv_v1.0.0>
  a owl:Ontology;
  dcterms:license <https://joinup.ec.europa.eu/category/licence/isa-open-metadata-
licence-v11>;
### metadata proposed by adms(-ap)
  dcterms:title "Business eGovernment Core Vocabulary"@en;
  dcterms:description ""The Core Business Vocabulary provides a minimum set of
classes and properties for describing a registered legal entity (business). Its
limited scope does not include sole traders, or relationships between registered
legal entities. This vocabulary is closely integrated with the Location and Person
eGovernment Core Vocabularies.""@en;
  dcterms:modified "2020-31-03"^^xsd:date;
  adms:status <http://purl.org/adms/status/Completed>;
### =====
  rdfs:comment "This file specifies the set of RDF classes and properties used in
the Business eGovernment Core Vocabulary"@en;
  rdfs:label "Business eGovernment Core Vocabulary"@en;
  owl:imports dcterms;;
  owl:imports <http://www.w3.org/2004/02/skos/core>;
  owl:imports <http://www.w3.org/ns/locn>;
  owl:imports org;;
  owl:imports foaf;; .
```

Aggregated example


```

@prefix cb: <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#> . ①
@prefix adms: <http://www.w3.org/ns/adms#> . ②
...
<https://data.europe.eu/semanticassets/ns/cbv_v1.0.0> ①
  a owl:Ontology; ③ ④
  dcterms:abstract ""The Core Business Vocabulary provides a minimum set of
classes and properties for describing a registered legal entity (business). Its
limited scope does not include sole traders, or relationships between registered
legal entities. This vocabulary is closely integrated with the Location and Person
eGovernment Core Vocabularies.""@en; ⑦
  dcterms:license <https://joinup.ec.europa.eu/category/licence/isa-open-metadata-
licence-v11>; ⑦
  dcterms:modified "2020-31-03"^^xsd:date; ⑦
  rdfs:comment "This file specifies the set of RDF classes and properties used in
the Business eGovernment Core Vocabulary"@en; ⑥
  rdfs:label "Business eGovernment Core Vocabulary"@en; ⑥
  owl:imports dcterms;; ⑤
  owl:imports <http://www.w3.org/2004/02/skos/core>;
  owl:imports <http://www.w3.org/ns/locn>;
  owl:imports org;;
  owl:imports foaf;; .

```

- ① Guideline 50
- ② Guideline 51
- ③ Guideline 52
- ④ Guideline 53
- ⑤ Guideline 54
- ⑥ Guideline 55
- ⑦ Guideline 56

5.1.3. Vocabulary content

The content has declarations of the vocabulary itself, as mentioned in the guidelines below. The guidelines are provided initially for all artefacts (Classes and Properties) and then for Classes and Properties separately in an RDFS file.

- [All Artefacts \(Classes and Properties\)](#)
- [Classes](#)
- [Properties](#)

All Artefacts (Classes and Properties)

Guideline 57: All Artefacts (Classes and Properties)

Summary

All classes and properties SHALL be declared in the vocabulary's namespace.

Description

This is a design choice to take explicit ownership of the domain.

Example

```
cb:LegalEntity a rdfs:Class, owl:Class;
  rdfs:label "Legal Entity"@en;
  rdfs:comment ""This is the key class for the Business eGovernment Core
Vocabulary and represents a business that is legally registered.""@en;
  skos:scopeNote ""In many countries there is a single registry although in others,
such as Spain and Germany, multiple registries exist. A Legal Entity is able to
trade, is legally liable for its actions, accounts, tax affairs etc. It is a sub
class of org:FormalOrganization which covers a wider range of entities, such as
charities.""@en;
  rdfs:isDefinedBy <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0>;
  rdfs:subClassOf org:FormalOrganization;
.
```

The class **LegalEntity** has as URI https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#LegalEntity **cb:LegalEntity** meaning part of the vocabularies' namespace.

Aggregated example

```

@prefix cb: <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#> . ①
@prefix adms: <http://www.w3.org/ns/adms#> . ②
...
<https://data.europe.eu/semanticassets/ns/cbv_v1.0.0> ①
  a owl:Ontology; ③ ④
  dcterms:abstract ""The Core Business Vocabulary provides a minimum set of
classes and properties for describing a registered legal entity (business). Its
limited scope does not include sole traders, or relationships between registered
legal entities. This vocabulary is closely integrated with the Location and Person
eGovernment Core Vocabularies.""@en; ⑦
  dcterms:license <https://joinup.ec.europa.eu/category/licence/isa-open-metadata-
licence-v11>; ⑦
  dcterms:modified "2020-31-03"^^xsd:date; ⑦
  rdfs:comment "This file specifies the set of RDF classes and properties used in
the Business eGovernment Core Vocabulary"@en; ⑥
  rdfs:label "Business eGovernment Core Vocabulary"@en; ⑥
  owl:imports dcterms;; ⑤
  owl:imports <http://www.w3.org/2004/02/skos/core>;
  owl:imports <http://www.w3.org/ns/locn>;
  owl:imports org;;
  owl:imports foaf;;
  .
cb:LegalEntity ⑧
  a rdfs:Class;
  a owl:Class;
  .

```

① Guideline 50

② Guideline 51

③ Guideline 52

④ Guideline 53

⑤ Guideline 54

⑥ Guideline 55

⑦ Guideline 56

⑧ Guideline 57

Guideline 58: All Artefacts (Classes and Properties) II

Summary

All classes and properties SHALL have a `rdf:type`.

Description

All subjects need to be typed. This enhances the usability of the model in dedicated RDF editing environments.

Example

```
cb:LegalEntity rdf:type rdfs:Class, owl:Class;.
```

Aggregated example

```
@prefix cb: <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#> . ①
@prefix adms: <http://www.w3.org/ns/adms#> . ②
...
<https://data.europe.eu/semanticassets/ns/cbv_v1.0.0> ①
  a owl:Ontology; ③ ④
  dcterms:abstract ""The Core Business Vocabulary provides a minimum set of
classes and properties for describing a registered legal entity (business). Its
limited scope does not include sole traders, or relationships between registered
legal entities. This vocabulary is closely integrated with the Location and Person
eGovernment Core Vocabularies.""@en; ⑦
  dcterms:license <https://joinup.ec.europa.eu/category/licence/isa-open-metadata-
licence-v11>; ⑦
  dcterms:modified "2020-31-03"^^xsd:date; ⑦
  rdfs:comment "This file specifies the set of RDF classes and properties used in
the Business eGovernment Core Vocabulary"@en; ⑥
  rdfs:label "Business eGovernment Core Vocabulary"@en; ⑥
  owl:imports dcterms;; ⑤
  owl:imports <http://www.w3.org/2004/02/skos/core>;
  owl:imports <http://www.w3.org/ns/locn>;
  owl:imports org;;
  owl:imports foaf;;
  .
cb:LegalEntity ⑧
  a rdfs:Class; ⑨
  a owl:Class;
  .
```

① Guideline 50

② Guideline 51

③ Guideline 52

④ Guideline 53

⑤ Guideline 54

⑥ Guideline 55

⑦ Guideline 56

⑧ Guideline 57

⑨ Guideline 58

Guideline 59: All Artefacts (Classes and Properties) III

Summary

All classes and properties SHALL have labels.

- A class needs to have one `rdfs:label` in English;
- More are allowed for other languages, but only one `rdfs:label` per language.

Description

Everything needs to be human-understandable and usable.

Example

```
cb:LegalEntity a rdfs:Class, owl:Class;  
  rdfs:label "Legal Entity"@en;
```

Aggregated example

```
@prefix cb: <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#> . ①  
@prefix adms: <http://www.w3.org/ns/adms#> . ②  
...  
<https://data.europe.eu/semanticassets/ns/cbv_v1.0.0> ①  
  a owl:Ontology; ③ ④  
  dcterms:abstract ""The Core Business Vocabulary provides a minimum set of  
  classes and properties for describing a registered legal entity (business). Its  
  limited scope does not include sole traders, or relationships between registered  
  legal entities. This vocabulary is closely integrated with the Location and Person  
  eGovernment Core Vocabularies.""@en; ⑦  
  dcterms:license <https://joinup.ec.europa.eu/category/licence/isa-open-metadata-  
  licence-v11>; ⑦  
  dcterms:modified "2020-31-03"^^xsd:date; ⑦  
  rdfs:comment "This file specifies the set of RDF classes and properties used in  
  the Business eGovernment Core Vocabulary"@en; ⑥  
  rdfs:label "Business eGovernment Core Vocabulary"@en; ⑥  
  owl:imports dcterms;; ⑤  
  owl:imports <http://www.w3.org/2004/02/skos/core>;  
  owl:imports <http://www.w3.org/ns/locn>;  
  owl:imports org;;  
  owl:imports foaf;;  
  .  
cb:LegalEntity ⑧  
  a rdfs:Class; ⑨  
  a owl:Class;  
  rdfs:label "Legal Entity"@en; ⑩  
  .
```

① Guideline 50

② Guideline 51

③ Guideline 52

④ Guideline 53

⑤ Guideline 54

⑥ Guideline 55

⑦ Guideline 56

⑧ Guideline 57

⑨ Guideline 58

⑩ Guideline 59

Guideline 60: All Artefacts (Classes and Properties) IV

Summary

All classes and properties SHALL have definitions/descriptions.

- They need to have one `rdfs:comment` in English describing the meaning of the class or property;
- More are allowed for other languages, but only one `rdfs:comment` per language.

Description

Everything needs to be human-understandable and usable.

Example

```
cb:LegalEntity a rdfs:Class, owl:Class;  
  rdfs:label "Legal Entity"@en;  
  rdfs:comment ""This is the key class for the Business eGovernment Core  
Vocabulary and represents a business that is legally registered.""@en; .
```

Aggregated example


```

@prefix cb: <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#> . ①
@prefix adms: <http://www.w3.org/ns/adms#> . ②
...
<https://data.europe.eu/semanticassets/ns/cbv_v1.0.0> ①
  a owl:Ontology; ③ ④
  dcterms:abstract ""The Core Business Vocabulary provides a minimum set of
classes and properties for describing a registered legal entity (business). Its
limited scope does not include sole traders, or relationships between registered
legal entities. This vocabulary is closely integrated with the Location and Person
eGovernment Core Vocabularies.""@en; ⑦
  dcterms:license <https://joinup.ec.europa.eu/category/licence/isa-open-metadata-
licence-v11>; ⑦
  dcterms:modified "2020-31-03"^^xsd:date; ⑦
  rdfs:comment "This file specifies the set of RDF classes and properties used in
the Business eGovernment Core Vocabulary"@en; ⑥
  rdfs:label "Business eGovernment Core Vocabulary"@en; ⑥
  owl:imports dcterms;; ⑤
  owl:imports <http://www.w3.org/2004/02/skos/core>;
  owl:imports <http://www.w3.org/ns/locn>;
  owl:imports org;;
  owl:imports foaf;;
  .
cb:LegalEntity ⑧
  a rdfs:Class; ⑨
  a owl:Class;
  rdfs:comment ""This is the key class for the Business eGovernment Core
Vocabulary and represents a business that is legally registered.""@en; ⑪
  rdfs:label "Legal Entity"@en; ⑩
  .

```

① Guideline 50

② Guideline 51

③ Guideline 52

④ Guideline 53

⑤ Guideline 54

⑥ Guideline 55

⑦ Guideline 56

⑧ Guideline 57

⑨ Guideline 58

⑩ Guideline 59

⑪ Guideline 60

Guideline 61: All Artefacts (Classes and Properties) V

Summary

All classes and properties CAN have optional usage note or scope note.

- If there is one, `skos:scopeNote`, it SHALL be in English with the content on ‘how’ and ‘when’ to use the class or property;
- More are allowed for other languages, but only one `skos:scopeNote` per language.

Description

It is advised practice to include (next to the definition) further guidance on when and how to use the construct.

Example

```
cb:LegalEntity a rdfs:Class, owl:Class;  
  rdfs:label "Legal Entity"@en;  
  skos:scopeNote ""In many countries there is a single registry although in others,  
such as Spain and Germany, multiple registries exist. A Legal Entity is able to  
trade, is legally liable for its actions, accounts, tax affairs etc. It is a sub  
class of org:FormalOrganization which covers a wider range of entities, such as  
charities.""@en;.
```

Aggregated example

```

@prefix cb: <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#> . ①
@prefix adms: <http://www.w3.org/ns/adms#> . ②
...
<https://data.europe.eu/semanticassets/ns/cbv_v1.0.0> ①
  a owl:Ontology; ③ ④
  dcterms:abstract ""The Core Business Vocabulary provides a minimum set of
classes and properties for describing a registered legal entity (business). ... the
Location and Person eGovernment Core Vocabularies.""@en; ⑦
  dcterms:license <https://joinup.ec.europa.eu/category/licence/isa-open-metadata-
licence-v11>; ⑦
  dcterms:modified "2020-31-03"^^xsd:date; ⑦
  rdfs:comment "This file specifies the set of RDF classes and properties used in
the Business eGovernment Core Vocabulary"@en; ⑥
  rdfs:label "Business eGovernment Core Vocabulary"@en; ⑥
  owl:imports dcterms;; ⑤
  owl:imports <http://www.w3.org/2004/02/skos/core>;
  owl:imports <http://www.w3.org/ns/locn>;
  owl:imports org;;
  owl:imports foaf;;
  .
cb:LegalEntity ⑧
  a rdfs:Class; ⑨
  a owl:Class;
  rdfs:comment ""This is the key class for the Business eGovernment Core
Vocabulary and represents a business that is legally registered.""@en; ⑪
  rdfs:label "Legal Entity"@en; ⑩
  skos:scopeNote ""In many countries there is a single registry although in
others, such as Spain and Germany, multiple registries exist. A Legal Entity is
able to trade, is legally liable for its actions, accounts, tax affairs etc. It is
a subclass of org:FormalOrganization which covers a wider range of entities, such
as charities.""@en; ⑫
  .

```

① Guideline 50

② Guideline 51

③ Guideline 52

④ Guideline 53

⑤ Guideline 54

⑥ Guideline 55

⑦ Guideline 56

⑧ Guideline 57

⑨ Guideline 58

⑩ Guideline 59

⑪ Guideline 60

⑫ Guideline 61

Guideline 62: All Artefacts (Classes and Properties) VI

Summary

All classes and properties CAN have optional multiple alternative labels (`skos:altLabel`).

- Every `skos:altLabel` needs to have a language indicator; **more than one** alternative label per language **is allowed**.

Description

This addresses the need to capture and use synonyms.

Example

```
cb:LegalEntity a rdfs:Class, owl:Class;  
  rdfs:label "Legal Entity"@en;  
  skos:altLabel "Legal Organisation"@en .
```

Aggregated example

```

@prefix cb: <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#> . ①
@prefix adms: <http://www.w3.org/ns/adms#> . ②
...
<https://data.europe.eu/semanticassets/ns/cbv_v1.0.0> ①
  a owl:Ontology; ③ ④
  dcterms:abstract ""The Core Business Vocabulary provides a minimum set of
classes and properties for describing a registered legal entity (business). Its
limited scope does not include sole traders, or relationships between registered
legal entities. This vocabulary is closely integrated with the Location and Person
eGovernment Core Vocabularies.""@en; ⑦
  dcterms:license <https://joinup.ec.europa.eu/category/licence/isa-open-metadata-
licence-v11>; ⑦
  dcterms:modified "2020-31-03"^^xsd:date; ⑦
  rdfs:comment "This file specifies the set of RDF classes and properties used in
the Business eGovernment Core Vocabulary"@en; ⑥
  rdfs:label "Business eGovernment Core Vocabulary"@en; ⑥
  owl:imports dcterms;; ⑤
  owl:imports <http://www.w3.org/2004/02/skos/core>;
  owl:imports <http://www.w3.org/ns/locn>;
  owl:imports org;;
  owl:imports foaf;;
  .
cb:LegalEntity ⑧
  a rdfs:Class; ⑨
  a owl:Class;
  rdfs:comment ""This is the key class for the Business eGovernment Core
Vocabulary and represents a business that is legally registered.""@en; ⑪
  rdfs:label "Legal Entity"@en; ⑩
  skos:altLabel "Legal Organisation"@en; ⑬
  skos:scopeNote ""In many countries there is a single registry although in
others, such as Spain and Germany, multiple registries exist. A Legal Entity is
able to trade, is legally liable for its actions, accounts, tax affairs etc. It is
a subclass of org:FormalOrganization which covers a wider range of entities, such
as charities.""@en; ⑫
  .

```

① Guideline 50

② Guideline 51

③ Guideline 52

④ Guideline 53

⑤ Guideline 54

⑥ Guideline 55

⑦ Guideline 56

⑧ Guideline 57

⑨ Guideline 58

⑩ Guideline 59

⑪ Guideline 60

⑫ Guideline 61

⑬ Guideline 62

Guideline 63: All Artefacts (Classes and Properties) VII

Summary

All classes and properties SHALL be linked to their vocabulary with `rdfs:isDefinedBy`.

Description

An explicit link between the construct and the vocabulary in which it has been defined is informative when used out of context.

Example

```
cb:LegalEntity a rdfs:Class, owl:Class;  
  rdfs:label "Legal Entity"@en;  
  rdfs:isDefinedBy <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0>;  
  .
```

Aggregated example

```

@prefix cb: <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#> . ①
@prefix adms: <http://www.w3.org/ns/adms#> . ②
...
<https://data.europe.eu/semanticassets/ns/cbv_v1.0.0> ①
  a owl:Ontology; ③ ④
  dcterms:abstract ""The Core Business Vocabulary provides a minimum set of
classes and properties for describing a registered legal entity (business). Its
limited scope does not include sole traders, or relationships between registered
legal entities. This vocabulary is closely integrated with the Location and Person
eGovernment Core Vocabularies.""@en; ⑦
  dcterms:license <https://joinup.ec.europa.eu/category/licence/isa-open-metadata-
licence-v11>; ⑦
  dcterms:modified "2020-31-03"^^xsd:date; ⑦
  rdfs:comment "This file specifies the set of RDF classes and properties used in
the Business eGovernment Core Vocabulary"@en; ⑥
  rdfs:label "Business eGovernment Core Vocabulary"@en; ⑥
  owl:imports dcterms;; ⑤
  owl:imports <http://www.w3.org/2004/02/skos/core>;
  owl:imports <http://www.w3.org/ns/locn>;
  owl:imports org;;
  owl:imports foaf;;
  .
cb:LegalEntity ⑧
  a rdfs:Class; ⑨
  a owl:Class;
  rdfs:comment ""This is the key class for the Business eGovernment Core
Vocabulary and represents a business that is legally registered.""@en; ⑪
  rdfs:isDefinedBy <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0>; ⑭
  rdfs:label "Legal Entity"@en; ⑩
  skos:altLabel "Legal Organisation"@en; ⑬
  skos:scopeNote ""In many countries there is a single registry although in
others, such as Spain and Germany, multiple registries exist. A Legal Entity is
able to trade, is legally liable for its actions, accounts, tax affairs etc. It is
a subclass of org:FormalOrganization which covers a wider range of entities, such
as charities.""@en; ⑫
  .

```

① Guideline 50

② Guideline 51

③ Guideline 52

④ Guideline 53

⑤ Guideline 54

⑥ Guideline 55

⑦ Guideline 56

⑧ Guideline 57

⑨ Guideline 58

⑩ Guideline 59

⑪ Guideline 60

⑫ Guideline 61

⑬ Guideline 62

⑭ Guideline 63

Classes

Guideline 64: Classes

Summary

Class names SHALL be in UpperCamelCase.

Description

This is an established practice.

Example

```
cb:LegalEntity a rdfs:Class, owl:Class;  
  rdfs:label "Legal Entity"@en;  
.
```

Aggregated example

```

@prefix cb: <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#> . ①
@prefix adms: <http://www.w3.org/ns/adms#> . ②
...
<https://data.europe.eu/semanticassets/ns/cbv_v1.0.0> ①
  a owl:Ontology; ③ ④
  dcterms:abstract ""The Core Business Vocabulary provides a minimum set of
classes and properties for describing a registered legal entity (business). Its
limited scope does not include sole traders, or relationships between registered
legal entities. This vocabulary is closely integrated with the Location and Person
eGovernment Core Vocabularies.""@en; ⑦
  dcterms:license <https://joinup.ec.europa.eu/category/licence/isa-open-metadata-
licence-v11>; ⑦
  dcterms:modified "2020-31-03"^^xsd:date; ⑦
  rdfs:comment "This file specifies the set of RDF classes and properties used in
the Business eGovernment Core Vocabulary"@en; ⑥
  rdfs:label "Business eGovernment Core Vocabulary"@en; ⑥
  owl:imports dcterms;; ⑤
  owl:imports <http://www.w3.org/2004/02/skos/core>;
  owl:imports <http://www.w3.org/ns/locn>;
  owl:imports org;;
  owl:imports foaf;;
  .
cb:LegalEntity ⑧ ⑮
  a rdfs:Class; ⑨
  a owl:Class;
  rdfs:comment ""This is the key class for the Business eGovernment Core
Vocabulary and represents a business that is legally registered.""@en; ⑪
  rdfs:isDefinedBy <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0>; ⑭
  rdfs:label "Legal Entity"@en; ⑩
  skos:altLabel "Legal Organisation"@en; ⑬
  skos:scopeNote ""In many countries there is a single registry although in
others, such as Spain and Germany, multiple registries exist. A Legal Entity is
able to trade, is legally liable for its actions, accounts, tax affairs etc. It is
a subclass of org:FormalOrganization which covers a wider range of entities, such
as charities.""@en; ⑫
  .

```

① Guideline 50

② Guideline 51

③ Guideline 52

④ Guideline 53

⑤ Guideline 54

⑥ Guideline 55

⑦ Guideline 56

⑧ Guideline 57

⑨ Guideline 58

⑩ Guideline 59

⑪ Guideline 60

⑫ Guideline 61

⑬ Guideline 62

⑭ Guideline 63

⑮ Guideline 64

Guideline 65: Classes II

Summary

The `rdfs:subClassOf` relation between Classes CAN be used. This is important for constraint checking of SHACL files since `rdfs:subClassOf` relationships are taken into account for the triggering of shapes.

Example

```
cb:LegalEntity
a rdfs:Class, owl:Class;
  rdfs:label "Legal Entity"@en;
  rdfs:subClassOf org:FormalOrganization;
.
```

Aggregated example

```

@prefix cb: <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#> . ①
@prefix adms: <http://www.w3.org/ns/adms#> . ②
...
<https://data.europe.eu/semanticassets/ns/cbv_v1.0.0> ①
  a owl:Ontology; ③ ④
  dcterms:abstract ""The Core Business Vocabulary provides a minimum set of
classes and properties for describing a registered legal entity (business). Its
limited scope does not include sole traders, or relationships between registered
legal entities. This vocabulary is closely integrated with the Location and Person
eGovernment Core Vocabularies.""@en; ⑦
  dcterms:license <https://joinup.ec.europa.eu/category/licence/isa-open-metadata-
licence-v11>; ⑦
  dcterms:modified "2020-31-03"^^xsd:date; ⑦
  rdfs:comment "This file specifies the set of RDF classes and properties used in
the Business eGovernment Core Vocabulary"@en; ⑥
  rdfs:label "Business eGovernment Core Vocabulary"@en; ⑥
  owl:imports dcterms;; ⑤
  owl:imports <http://www.w3.org/2004/02/skos/core>;
  owl:imports <http://www.w3.org/ns/locn>;
  owl:imports foaf;;
  .
cb:LegalEntity ⑧ ⑮
  a rdfs:Class; ⑨
  a owl:Class;
  rdfs:comment ""This is the key class for the Business eGovernment Core
Vocabulary and represents a business that is legally registered.""@en; ⑪
  rdfs:isDefinedBy <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0>; ⑭
  rdfs:label "Legal Entity"@en; ⑩
  skos:altLabel "Legal Organisation"@en; ⑬
  rdfs:subClassOf org:FormalOrganization; ⑯
  skos:scopeNote ""In many countries there is a single registry although in
others, such as Spain and Germany, multiple registries exist. A Legal Entity is
able to trade, is legally liable for its actions, accounts, tax affairs etc. It is
a subclass of org:FormalOrganization which covers a wider range of entities, such
as charities.""@en; ⑫
  .

```

① Guideline 50

② Guideline 51

③ Guideline 52

④ Guideline 53

⑤ Guideline 54

⑥ Guideline 55

⑦ Guideline 56

⑧ Guideline 57

⑨ Guideline 58

⑩ Guideline 59

⑪ Guideline 60

⑫ Guideline 61

⑬ Guideline 62

⑭ Guideline 63

⑮ Guideline 64

⑯ Guideline 65

Properties

Guideline 66: Properties

Summary

If a property has type `rdf:Property`, a more specific type CAN be mentioned (`DatatypeProperty` or `ObjectProperty`).

Description

This is more informative to the RDF/OWL parser.

Example

```
cb:legalName
  a rdf:Property;
  a owl:DatatypeProperty;
  rdfs:comment "The legal name of the business."@en;
  rdfs:isDefinedBy <http://www.w3.org/ns/legal>;
  rdfs:label "legal name"@en;
  skos:scopeNote ""A business might have more than one legal name, particularly
in countries with more than one official language. In such cases the language of
the string should be identified.""@en;
.
cb:registeredAddress
  a rdf:Property;
  a owl:ObjectProperty;
  rdfs:comment "Links a Legal Entity to its registered address."@en;
  rdfs:isDefinedBy <http://www.w3.org/ns/legal>;
  rdfs:label "registered address"@en;
  skos:scopeNote "This may or may not be the actual address at which the legal
entity does its business, it is commonly the address of their lawyer or
accountant, but it is the address to which formal communications can be sent.
legal:registeredAddress is a sub property of locn:address that has a domain of
legal:LegalEntity and a range of locn:Address."@en;.
```

Aggregated example

```
@prefix cb: <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#> . ①
@prefix adms: <http://www.w3.org/ns/adms#> . ②
...
<https://data.europe.eu/semanticassets/ns/cbv_v1.0.0> ①
  a owl:Ontology; ③ ④
  dcterms:abstract ""The Core Business Vocabulary provides a minimum set of
classes and properties for describing a registered legal entity (business). Its
limited scope does not include sole traders, or relationships between registered
legal entities. This vocabulary is closely integrated with the Location and Person
eGovernment Core Vocabularies.""@en; ⑦
  dcterms:license <https://joinup.ec.europa.eu/category/licence/isa-open-metadata-
licence-v11>; ⑦
  dcterms:modified "2020-31-03"^^xsd:date; ⑦
  rdfs:comment "This file specifies the set of RDF classes and properties used in
the Business eGovernment Core Vocabulary"@en; ⑥
```



```

rdfs:label "Business eGovernment Core Vocabulary"@en; ⑥
owl:imports dcterms;; ⑤
owl:imports <http://www.w3.org/2004/02/skos/core>;
owl:imports <http://www.w3.org/ns/locn>;
owl:imports org;;
owl:imports foaf;;
.

cb:LegalEntity ⑧ ⑮
  a rdfs:Class; ⑨
  a owl:Class;
  rdfs:comment ""This is the key class for the Business eGovernment Core
Vocabulary and represents a business that is legally registered.""@en; ⑪
  rdfs:isDefinedBy <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0>; ⑭
  rdfs:label "Legal Entity"@en; ⑩
  skos:altLabel "Legal Organisation"@en; ⑬
  rdfs:subClassOf org:FormalOrganization; ⑯
  skos:scopeNote ""In many countries there is a single registry although in
others, such as Spain and Germany, multiple registries exist. A Legal Entity is
able to trade, is legally liable for its actions, accounts, tax affairs etc. It is
a subclass of org:FormalOrganization which covers a wider range of entities, such
as charities.""@en; ⑫
.

cb:companyActivity ⑧
  a rdf:Property; ⑨
  a owl:ObjectProperty; ⑰
  rdfs:comment "The economical activity of a company."@en; ⑪
  rdfs:isDefinedBy <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0>; ⑭
  rdfs:label "company activity"@en; ⑩
  skos:scopeNote ""The activity of a company should be recorded using a
controlled vocabulary expressed as a SKOS concept scheme (see <a
xmlns="http://www.w3.org/ns/legal#" href="#"#skos:Concept">Code</a>. Several
such vocabularies exist, many of which map to the UN's ISIC codes. Where a
particular controlled vocabulary is in use within a given context, such as SIC
codes in the UK, it is acceptable to use these, however, the preferred choice for
European interoperability is NACE.""@en; ⑫
.

```

① Guideline 50

② Guideline 51

③ Guideline 52

④ Guideline 53

⑤ Guideline 54

⑥ Guideline 55

⑦ Guideline 56

⑧ Guideline 57

⑨ Guideline 58

⑩ Guideline 59

⑪ Guideline 60

⑫ Guideline 61

⑬ Guideline 62

⑭ Guideline 63

⑮ Guideline 64

⑯ Guideline 65

⑰ Guideline 66

Guideline 67: Properties II

Summary

Property names SHALL be in lowerCamelCase.

Description

This is an established practice.

Example

```
cb:registeredAddress
  a rdf:Property;
  a owl:ObjectProperty;.
```

Aggregated example

```
@prefix cb: <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#> . ①
@prefix adms: <http://www.w3.org/ns/adms#> . ②
...
<https://data.europe.eu/semanticassets/ns/cbv_v1.0.0> ①
  a owl:Ontology; ③ ④
  dcterms:abstract ""The Core Business Vocabulary provides a minimum set of
classes and properties for describing a registered legal entity (business). Its
limited scope does not include sole traders, or relationships between registered
legal entities. This vocabulary is closely integrated with the Location and Person
eGovernment Core Vocabularies.""@en; ⑦
  dcterms:license <https://joinup.ec.europa.eu/category/licence/isa-open-metadata-
licence-v11>; ⑦
  dcterms:modified "2020-31-03"^^xsd:date; ⑦
  rdfs:comment "This file specifies the set of RDF classes and properties used in
the Business eGovernment Core Vocabulary"@en; ⑥
  rdfs:label "Business eGovernment Core Vocabulary"@en; ⑥
  owl:imports dcterms;; ⑤
  owl:imports <http://www.w3.org/2004/02/skos/core>;
  owl:imports <http://www.w3.org/ns/locn>;
  owl:imports org;;
  owl:imports foaf;;
  .
cb:LegalEntity ⑧ ⑮
  a rdfs:Class; ⑨
  a owl:Class;
  rdfs:comment ""This is the key class for the Business eGovernment Core
Vocabulary and represents a business that is legally registered.""@en; ⑪
  rdfs:isDefinedBy <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0>; ⑭
  rdfs:label "Legal Entity"@en; ⑩
  skos:altLabel "Legal Organisation"@en; ⑬
  rdfs:subClassOf org:FormalOrganization; ⑯
  skos:scopeNote ""In many countries there is a single registry although in
others, such as Spain and Germany, multiple registries exist. A Legal Entity is
able to trade, is legally liable for its actions, accounts, tax affairs etc. It is
```

a subclass of `org:FormalOrganization` which covers a wider range of entities, such as charities."@en; ⑫

.

`cb:companyActivity` ⑧ ⑱

`a rdf:Property;` ⑨

`a owl:ObjectProperty;` ⑰

`rdfs:comment` "The economical activity of a company."@en; ⑪

`rdfs:isDefinedBy` <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0>; ⑭

`rdfs:label` "company activity"@en; ⑩

`skos:scopeNote` ""The activity of a company should be recorded using a controlled vocabulary expressed as a SKOS concept scheme (see Code. Several such vocabularies exist, many of which map to the UN's ISIC codes. Where a particular controlled vocabulary is in use within a given context, such as SIC codes in the UK, it is acceptable to use these, however, the preferred choice for European interoperability is NACE.""@en; ⑫

.

① Guideline 50

② Guideline 51

③ Guideline 52

④ Guideline 53

⑤ Guideline 54

⑥ Guideline 55

⑦ Guideline 56

⑧ Guideline 57

⑨ Guideline 58

⑩ Guideline 59

⑪ Guideline 60

⑫ Guideline 61

⑬ Guideline 62

⑭ Guideline 63

⑮ Guideline 64

⑯ Guideline 65

⑰ Guideline 66

⑱ Guideline 67

Guideline 68: Properties III

Summary

`rdfs:domain` and `rdfs:range` SHOULD not be used.

Description

- If one wants to attach a property to a class in an object oriented way, our first choice is to use a SHACL Shape to express this containment; when not using SHACL we advise to use the `domainIncludes` property from <http://purl.org/dc/dcam/>.
- If one wants to express a constraint on the allowed values of a property, e.g. the value must be a `skos:Concept`, the preferred solution is to express this constraint in a SHACL Shape; or use `rangeIncludes` from <http://purl.org/dc/dcam/> when not using SHACL.
- If one wants to explicitly use `rdfs:domain` and `rdfs:range` for inferring new facts, move these constructs to the semantics RDFS/OWL file.

Guideline 69: Properties IV

Summary

A warning SHOULD be issued by a validating processor when one encounters domain and/or range in a eGovernment Core Vocabulary RDFS file.

Guideline 70: Properties V

Summary

If a domain or range is encountered on an imported vocabulary, a warning SHOULD be issued that the changed semantics need to align with those of the original.

Description

The original vocabulary has already established the inferences which can be deducted; overruling those can lead to semantic incompatibilities. It is bad practice to redefine the semantics of a vocabulary that you don't own.

Example

```
<http://www.w3.org/ns/org#purpose> a rdf:Property;  rdfs:label "purpose"@en;  
rdfs:domain cv:PublicOrganisation .
```

The original domain as defined by w3c's org vocabulary is **org:Organisation**. The overruling in this case means that any entity that uses **org:purpose** becomes a **cv:PublicOrganisation**, which leads to potential problems in cases where both vocabularies (org and core public organisation) are used/imported, since every instance with a purpose will be a PublicOrganisation.

5.2. Vocabulary semantics in RDFS/OWL

To solve the closed and open world mismatch, the proposal is made to separate the different concerns into different files:

- An RDFS file (required) with human-oriented semantics containing labels, definitions, scope and usage note that will serve two other models
- One (optional) in RDFS/OWL, aimed at inferring new facts
- One in SHACL (required) that defines the classes with their attributes and relationships, and the constraints on occurrences and values used

The focus of this section is on the optional RDFS/OWL file with additional constructs purely aimed at reasoning (see Figure 1). The presence of these additional constructs explains why this file is optional. Most linked data applications do not reason because, either there is no apparent need seen for reasoning, or there is a fear of the additional computing resources required. In this instances situation, there is no need to spend time and community money on constructs for this aim.

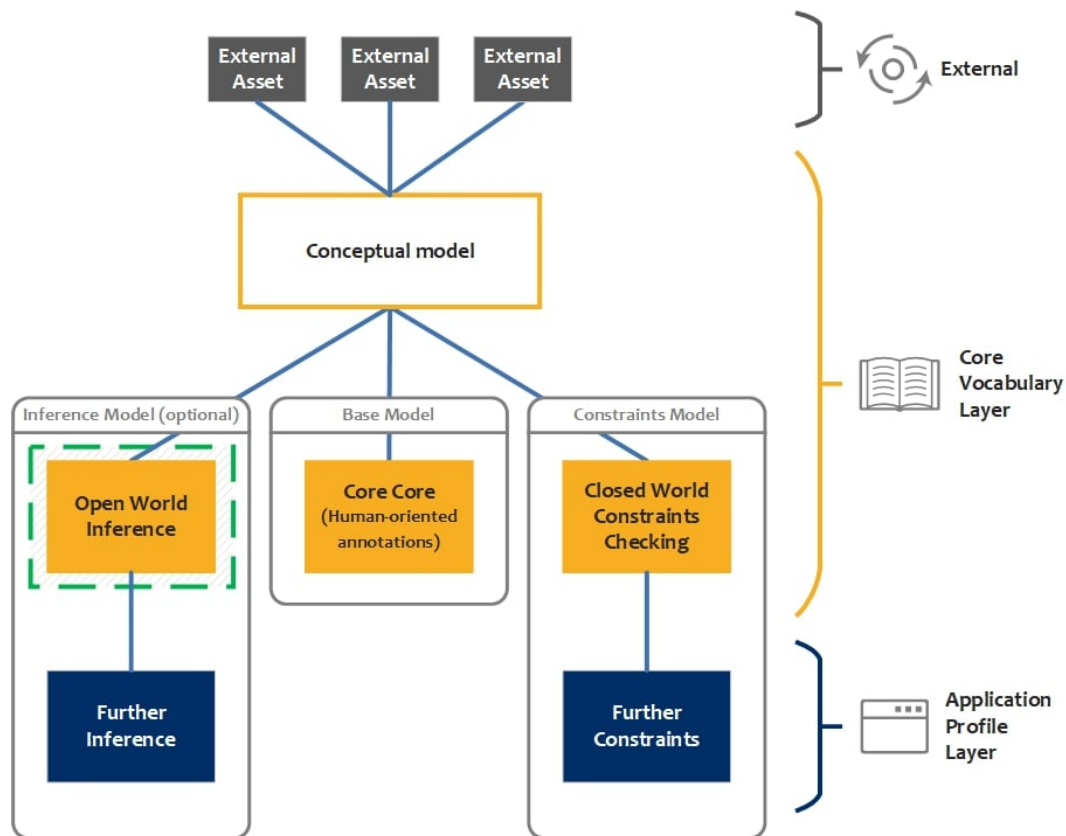


Figure 38. Situating RDFS for machine semantics

5.2.1. Environment Setup

The guidelines for setting up the environment while creating an RDFS file are described below.

Guideline 71: Set up the environment

Summary

The namespace of the semantic version of the vocabulary SHALL be defined according to the rules set in [Naming and versioning](#).

Description

Each vocabulary should be uniquely identifiable.

Example

https://data.europe.eu/semanticassets/ns/cbv-semantic_v-1.0.0

Aggregated xample

```
@prefix owl: <http://www.w3.org/2002/07/owl#> .
<https://data.europe.eu/semanticassets/ns/cbv-semantic_v1.0.0> ①
  a owl:Ontology;
  .
```

① Guideline 71

Guideline 72: Set up the environment II

Summary

A prefix SHALL be declared for the semantics version and for the version with the core. The use of prefixes enhances readability.

Description

This takes the form of the namespace URI followed by a #, the base URI. An example in turtle syntax is given here.

Example

```
@prefix cb-sem: <https://data.europe.eu/semanticassets/ns/cbv-semantics_v1.0.0#> .  
@prefix cb: <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#> .
```

Aggregated xample

```
@prefix cb-sem: <https://data.europe.eu/semanticassets/ns/cbv-semantics_v1.0.0#> .  
②  
@prefix cb: <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#> . ②  
@prefix owl: <http://www.w3.org/2002/07/owl#> .  
<https://data.europe.eu/semanticassets/ns/cbv-semantics_v1.0.0> ①  
  a owl:Ontology;  
  .
```

① Guideline 71

② Guideline 72

5.2.2. Vocabulary

The following guidelines describe the way vocabularies need to be applied and used.

Guideline 73: Vocabulary

Summary

An ontology using the vocabularies' namespace SHALL be declared as in: "<URI> a owl:Ontology."

Example

```
<https://data.europe.eu/semanticassets/ns/cbv-semantics_v1.0.0> a owl:Ontology.
```

Aggregated example

```
@prefix cb-sem: <https://data.europe.eu/semanticassets/ns/cbv-semantics_v1.0.0#> .  
②  
@prefix cb: <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#> . ②  
@prefix owl: <http://www.w3.org/2002/07/owl#> .  
<https://data.europe.eu/semanticassets/ns/cbv-semantics_v1.0.0> ①  
  a owl:Ontology; ③  
  .
```

① Guideline 71

② Guideline 72

③ Guideline 73

Guideline 74: Vocabulary II

Summary

Only one ontology SHALL be declared per RDF(S) file.

Description

RDF tools are sensitive to this and throw warnings if multiple exist.

Example

```
<https://data.europe.eu/semanticassets/ns/cbv-semantic_v1.0.0#> a owl:Ontology.
```

Aggregated example

```
@prefix cb-sem: <https://data.europe.eu/semanticassets/ns/cbv-semantic_v1.0.0#> .  
②  
@prefix cb: <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#> . ②  
@prefix owl: <http://www.w3.org/2002/07/owl#> .  
<https://data.europe.eu/semanticassets/ns/cbv-semantic_v1.0.0#> ①  
  a owl:Ontology; ③ ④  
.
```

① Guideline 71

② Guideline 72

③ Guideline 73

④ Guideline 74

Guideline 75: Vocabulary III

Summary

Source vocabulary SHALL be imported.

Description

Import is needed since one is adding machine semantics to the human semantics. The alternative (copy and paste) leads to errors and semantic hijacking.

Example

```
<data.europe.eu/semanticassets/ns/cbv-semantic_v1.0.0> a owl:Ontology;  
owl:imports <data.europe.eu/semanticassets/ns /cbv_v1.0.0>;
```

Aggregated example

```
@prefix cb-sem: <https://data.europe.eu/semanticassets/ns/cbv-semantic_v1.0.0#> .  
②  
@prefix cb: <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#> . ②  
@prefix owl: <http://www.w3.org/2002/07/owl#> .  
<https://data.europe.eu/semanticassets/ns/cbv-semantic_v1.0.0> ①  
  a owl:Ontology; ③ ④  
  owl:imports <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0>; ⑤  
  .
```

① Guideline 71

② Guideline 72

③ Guideline 73

④ Guideline 74

⑤ Guideline 75

Guideline 76: Vocabulary IV

Summary

External vocabularies SHOULD be imported if needed and appropriate.

Description

The rationale for this guideline is to be found in Section #reuse-of-external-vocabularies[Reuse of external vocabularies].

Example

```
<https://data.europe.eu/semanticassets/ns/cbv-semantic_v1.0.0> a owl:Ontology;  
owl:imports <https:// data.europe.eu/semanticassets/ns/cbv_v1.0.0>;  
owl:imports <http://purl.org/dc/terms/>;  
owl:imports <http://xmlns.com/foaf/0.1/> .
```

Aggregated example

```
@prefix cb-sem: <https://data.europe.eu/semanticassets/ns/cbv-semantic_v1.0.0#> .  
②  
@prefix cb: <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#> . ②  
@prefix owl: <http://www.w3.org/2002/07/owl#> .  
@prefix dterms: <http://purl.org/dc/terms/> .  
@prefix foaf: <http://xmlns.com/foaf/0.1/> .  
  
<https://data.europe.eu/semanticassets/ns/cbv-semantic_v1.0.0> ①  
  a owl:Ontology;③ ④  
  owl:imports <https:// data.europe.eu/semanticassets/ns/cbv_v1.0.0>; ⑤  
  owl:imports dterms;; ⑥  
  owl:imports foaf;;  
  .
```

① Guideline 71

② Guideline 72

③ Guideline 73

④ Guideline 74

⑤ Guideline 75

⑥ Guideline 76

Guideline 77: Vocabulary V

Summary

Metadata SHOULD be assigned to the vocabulary (see Metadata).

Description

This is established good practice. This helps to find the vocabulary at semantic asset portals and to evaluate if a vocabulary suits the needs.

Example

```
<https://data.europe.eu/semanticassets/ns/cbv-semantic_v1.0.0> a owl:Ontology;
owl:imports <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0>;
owl:imports <http://purl.org/dc/terms/>;
owl:imports <http://xmlns.com/foaf/0.1/> ;
### metadata proposed by adms(-ap)
  dcterms:title "Business eGovernment Core Vocabulary Semantics"@en;
  dcterms:description ""The Core Business Vocabulary provides the minimum set of
classes and properties for describing a registered legal entity (business) and
their machine oriented semantics e.g. the inferences that are useful to make using
the model""@en;
  dcterms:modified "2020-31-03"^^xsd:date;
  adms:status <http://purl.org/adms/status/Completed>;
###
```

Aggregated example

```
@prefix cb-sem: <https://data.europe.eu/semanticassets/ns/cbv-semantic_v1.0.0#> .
②
@prefix cb: <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#> . ②
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix adms: <http://www.w3.org/ns/adms#> .

<https://data.europe.eu/semanticassets/ns/cbv-semantic_v1.0.0> ①
  a owl:Ontology; ③ ④
  owl:imports <http://data.europe.eu/semanticassets/ns/cbv_v1.0.0>; ⑤
  owl:imports dcterms;; ⑥
  owl:imports foaf;; ⑥
  dcterms:title "Business eGovernment Core Vocabulary Semantics"@en; ⑦
  dcterms:description ""The Core Business Vocabulary provides the minimum set of
classes and properties for describing a registered legal entity (business) and
their machine oriented semantics e.g. the inferences that are useful to make using
the model""@en; ⑦
  dcterms:modified "2020-31-03"^^xsd:date; ⑦
  adms:status <http://purl.org/adms/status/Completed>; ⑦
  .
```

① Guideline 71

② Guideline 72

③ Guideline 73

④ Guideline 74

⑤ Guideline 75

⑥ Guideline 76

⑦ Guideline 77

Guideline 78: Vocabulary VI

Summary

Disjointness between classes (`owl:disjointWith`) CAN be indicated. This surely helps in detecting inconsistencies when doing inferencing.

Description

This is crucial for detecting inconsistencies in the vocabulary by means of automated reasoners.

Example

```
@prefix cb-sem: <https://data.europe.eu/semanticassets/ns/cbv-semantic_v1.0.0#> .
@prefix cb: <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix locn: <http://www.w3.org/ns/locn#> .
<https://data.europe.eu/semanticassets/ns/cbv-semantic_v1.0.0> a owl:Ontology;
owl:imports <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#>;
cb:LegalEntity owl:disjointWith locn:Address.
```

Aggregated example

```
@prefix cb-sem: <https://data.europe.eu/semanticassets/ns/cbv-semantic_v1.0.0#> . ②
@prefix cb: <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#> . ②
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix adms: <http://www.w3.org/ns/adms#> .

<https://data.europe.eu/semanticassets/ns/cbv-semantic_v1.0.0> ①
  a owl:Ontology; ③ ④
  owl:imports <http://data.europe.eu/semanticassets/ns/cbv_v1.0.0#>; ⑤
  owl:imports dcterms;; ⑥
  owl:imports foaf;; ⑥
  dcterms:title "Business eGovernment Core Vocabulary Semantics"@en; ⑦
  dcterms:description ""The Core Business Vocabulary provides the minimum set of
classes and properties for describing a registered legal entity (business) and
their machine oriented semantics e.g. the inferences that are useful to make using
the model""@en; ⑦
  dcterms:modified "2020-31-03"^^xsd:date; ⑦
  adms:status <http://purl.org/adms/status/Completed>; ⑦

<https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#LegalEntity>
  owl:disjointWith <http://www.w3.org/ns/locn#Address>; ⑧
.
```

① Guideline 71

② Guideline 72

- ③ Guideline 73
- ④ Guideline 74
- ⑤ Guideline 75
- ⑥ Guideline 76
- ⑦ Guideline 77
- ⑧ Guideline 78

5.3. Application Profiles

Commonly an application profile focuses on defining constraints and should be covered with SHACL (see Figure 1 below). However, an additional RDFS/OWL file could be needed in two cases:

1. If the application profile wants to introduce additional classes and properties not available in the eGovernment Core Vocabulary or other application profile one is deriving from
2. If additional semantic constructs are needed for generating further derivations or inferences

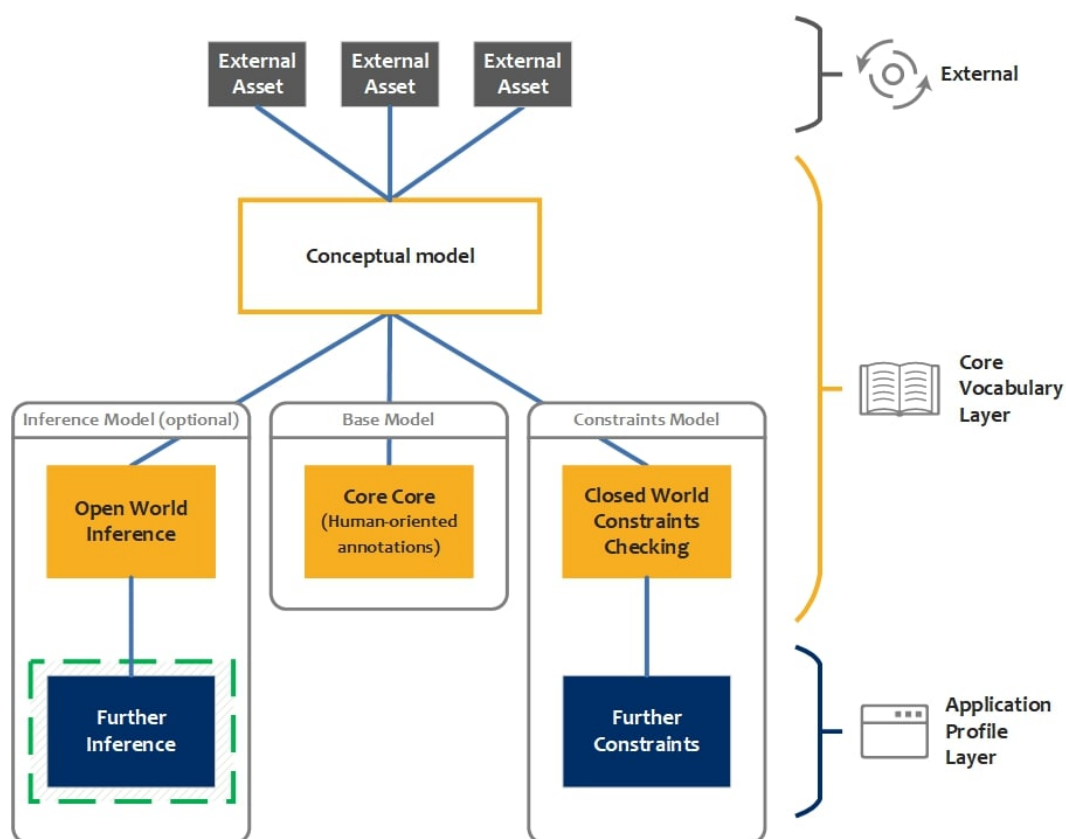


Figure 39. Situating of Application Profiles in RDFS

Such an application profile follows the same rules as eGovernment Core Vocabulary; therefore, the identical guidelines apply:

- [eGovernment Core Vocabularies Core annotations in RDFS](#)

- [eGovernment Core Vocabularies semantics in RDFS/OWL](#)

Chapter 6. SHACL guidelines

6.1. SHACL Approach

A SHACL file captures the object-oriented aspects of data modelling. It contains the constraints of the model (relations, values, classes, etc.) in the form of “shapes”. This section provides guidelines on the proper formulation of the SHACL files.

Transitioning from UML class diagrams (see UML Class Diagrams Style Guide), this section provides guidelines on the proper formulation of the SHACL files. To make the guidelines more specific, a distinction is made between two different constraints files, and they are:

- [eGovernment Core Vocabularies constraints in SHACL](#): provide guidelines for expressing the constraints of the model at the level of eGovernment Core Vocabularies;
- [Application Profiles in SHACL](#): offer guidelines for defining additional or more specific constraints at the level of application profiles.

6.1.1. SHACL Constraints

To solve the closed and open world mismatch, the proposal is made to separate the various concerns into different files:

- An RDFS file (required) with human-oriented semantics containing labels, definitions, scope and usage note that will serve two other models
- One (optional) in RDFS/OWL, aimed at inferring new facts
- One in SHACL (required) that defines the classes with their attributes and relationships, and the constraints on occurrences and values used

The focus of this section is to provide guidelines for object-oriented shapes at the level of eGovernment Core Vocabularies (see Figure below).

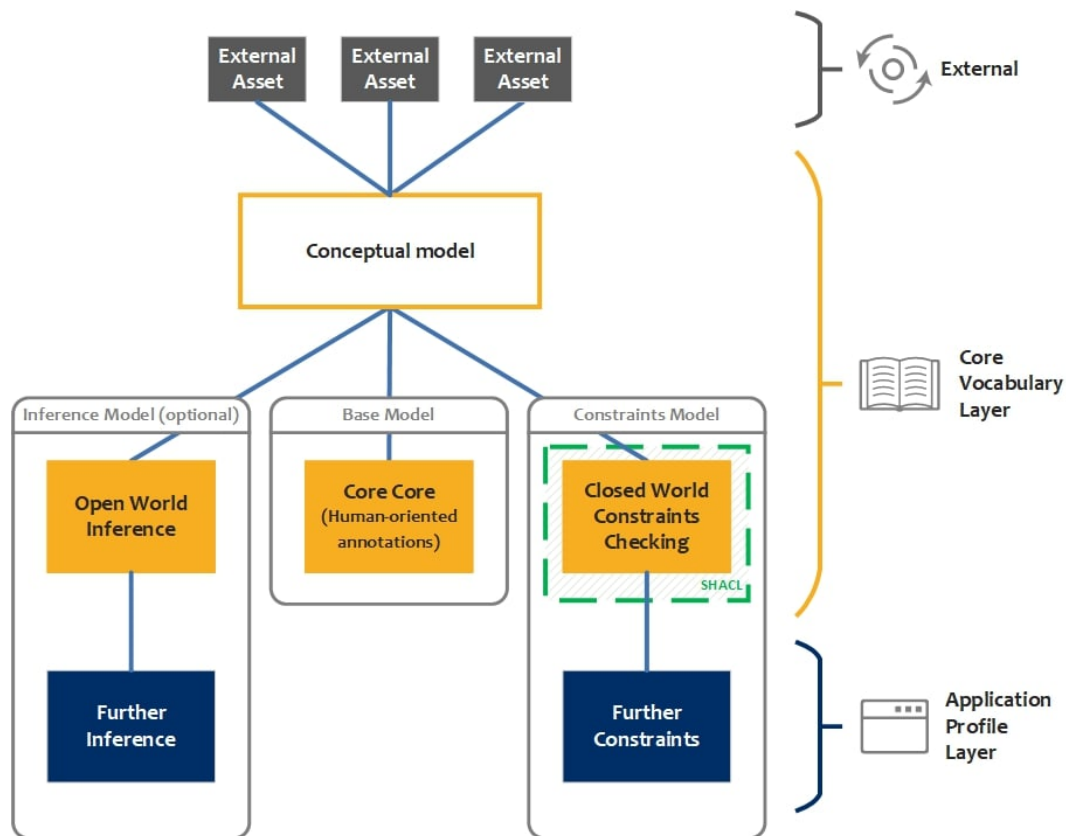


Figure 40. Situating SHACL for constraint checking

6.1.2. Generic

Since the objective is to keep the potential reuse of the vocabulary as open as possible, the cardinalities at the eGovernment Core Vocabulary level should be completely open. The cardinalities will then be specified on the application profiles level. One exception to this approach is that every instance of all classes SHALL have one `rdfs:label`.

6.1.3. SHACL Level

When the setup of the environment, on the SHACL level, needs to be in place, the following guidelines should be applied.

Guideline 79: SHACL Level

Summary

Every ontology SHALL be declared following the naming guidelines in Namespaces.

Description

Every vocabulary should be uniquely identifiable.

Example

The following example contains the definition of the constraints for the core business vocabulary.

The ‘core’ eGovernment Core Business Vocabulary has as URI https://data.europe.eu/semanticassets/ns/cbv_v1.0.0.

The constraints file gets as identifier https://data.europe.eu/semanticassets/ns/cbv-constraints_v1.0.0.

```
<https://data.europe.eu/semanticassets/ns /cbv-constraints_v1.0.0> a owl:Ontology;  
.
```

Aggregated example

```
@prefix owl: <http://www.w3.org/2002/07/owl#> .  
<https://data.europe.eu/semanticassets/ns /cbv-constraints_v1.0.0> ①  
  a owl:Ontology; ①  
.
```

① Guideline 79

Guideline 80: SHACL Level II

Summary

The related Core eGovernment Core Vocabulary SHALL be imported.

Description

The SHACL file adds the class containment and the properties of attributes and associations to the already in RDFS defined human semantics.

Example

```
<https://data.europe.eu/semanticassets/ns/cbv-constraints_v1.0.0>  
  a owl:Ontology;  
  owl:imports <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0>;.
```

Aggregated example

```
@prefix owl: <http://www.w3.org/2002/07/owl#> .  
<https://data.europe.eu/semanticassets/ns/cbv-constraints_v1.0.0> ①  
  a owl:Ontology; ①  
  owl:imports <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0>; ②  
.
```

① Guideline 79

② Guideline 80

Guideline 81: SHACL Level III

Summary

Semantic Asset Metadata SHALL be assigned as described in Metadata.

Description

This helps to find the constraint SHACL file of the vocabulary at semantic asset portals and to evaluate if it suits the needs.

Example

```
dcterms:title "Business eGovernment Core Vocabulary constraint SHACL"@en;  
dcterms:description ""The Core Business Vocabulary constraint SHACL provides  
the classes with their attributes and relationships, and the constraints on  
occurrences and values used.""@en;
```

Aggregated example

```
@prefix owl: <http://www.w3.org/2002/07/owl#> .  
@prefix dcterms: <http://purl.org/dc/terms/> .  
<https://data.europe.eu/semanticassets/ns/cbv-constraints_v1.0.0> ①  
  a owl:Ontology; ①  
  owl:imports <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0>; ②  
  dcterms:title "Business eGovernment Core Vocabulary constraint SHACL"@en; ③  
  dcterms:description ""The Core Business Vocabulary constraint SHACL provides  
the classes with their attributes and relationships, and the constraints on  
occurrences and values used.""@en; ③  
  .
```

① Guideline 79

② Guideline 80

③ Guideline 81

Guideline 82: SHACL Level IV

Summary

Prefixes for all namespaces SHOULD be defined.

Description

The use of prefixes enhances readability.

Example

```
@prefix cb-c: <https://data.europe.eu/semanticassets/ns/cbv-constraints_v1.0.0#> .
@prefix cb: <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
```

Aggregated example

```
@prefix cb-c: <https://data.europe.eu/semanticassets/ns/cbv-constraints_v1.0.0#> .
④
@prefix cb: <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix dcterms: <http://purl.org/dc/terms/> .

<https://data.europe.eu/semanticassets/ns/cbv-constraints_v1.0.0> ①
  a owl:Ontology; ①
  owl:imports <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#>; ②
  dcterms:title "Business eGovernment Core Vocabulary constraint SHACL"@en; ③
  dcterms:description ""The Core Business Vocabulary constraint SHACL provides
the classes with their attributes and relationships, and the constraints on
occurrences and values used.""@en; ③
  .
```

① Guideline 79

② Guideline 80

③ Guideline 81

④ Guideline 82

6.1.4. NodeShape

Every Class in UML translates in a SHACL NodeShape that describes the structure of the class and its constraints. The following guidelines explain how this NodeShape need to be formulated.

Guideline 83: NodeShape

Summary

Every Class SHALL have a corresponding NodeShape with `sh:targetClass` as the respective Class identifier.

Description

The choice has been made not to use implicit class targets since they conflate the concept of shape and class as the same.

Example

For the class `LegalEntity` a corresponding node shape is defined with name `LegalEntityShape`. Both are connected by the `sh:targetClass` property on the NodeShape.

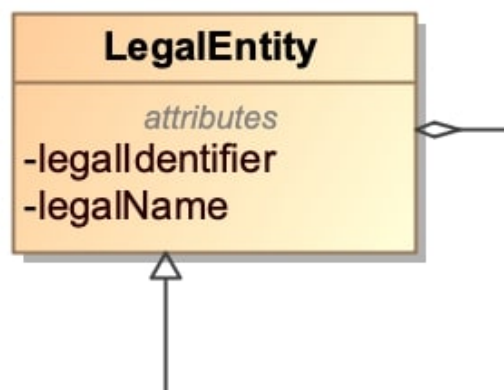


Figure 41. Example class

```
cb-c:LegalEntityShape
  a sh:NodeShape;
  *sh:targetClass cb:LegalEntity;.*
```

Aggregated example

```

@prefix cb-c: <https://data.europe.eu/semanticassets/ns/cbv-constraints_v1.0.0#> .
④
@prefix cb: <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix dcterms: <http://purl.org/dc/terms/> .

<https://data.europe.eu/semanticassets/ns/cbv-constraints_v1.0.0> ①
  a owl:Ontology; ①
  owl:imports <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#>; ②
  dcterms:title "Business eGovernment Core Vocabulary constraint SHACL"@en; ③
  dcterms:description ""The Core Business Vocabulary constraint SHACL provides
the classes with their attributes and relationships, and the constraints on
occurrences and values used.""@en; ③
  .
cb-c:LegalEntityShape
  a sh:NodeShape; ⑤
  sh:targetClass cb:LegalEntity; ⑤
  .

```

① Guideline 79

② Guideline 80

③ Guideline 81

④ Guideline 82

⑤ Guideline 83

Guideline 84: NodeShape II

Summary

The local name of the NodeShape SHALL be the ClassName + Shape.

Description

The choice has been made not to use implicit class targets since they conflate the concept of shape and class as the same.

Example

Classname= `LegalEntity` NodeShapename= `LegalEntityShape`

Aggregated example

```
@prefix cb-c: <https://data.europe.eu/semanticassets/ns/cbv-constraints_v1.0.0#> .  
④  
@prefix cb: <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#> .  
@prefix owl: <http://www.w3.org/2002/07/owl#> .  
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .  
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .  
@prefix sh: <http://www.w3.org/ns/shacl#> .  
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .  
@prefix dcterms: <http://purl.org/dc/terms/> .  
  
<https://data.europe.eu/semanticassets/ns/cbv-constraints_v1.0.0> ①  
  a owl:Ontology; ①  
  owl:imports <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0>; ②  
  dcterms:title "Business eGovernment Core Vocabulary constraint SHACL"@en; ③  
  dcterms:description ""The Core Business Vocabulary constraint SHACL provides  
the classes with their attributes and relationships, and the constraints on  
occurrences and values used.""@en; ③  
  .  
cb-c:LegalEntityShape ⑥  
  a sh:NodeShape; ⑤  
  sh:targetClass cb:LegalEntity; ⑤  
  .
```

① Guideline 79

② Guideline 80

③ Guideline 81

④ Guideline 82

⑤ Guideline 83

⑥ Guideline 84

Guideline 85: NodeShape III

Summary

Every NodeShape SHOULD contain `rdfs:label` and `rdfs:comment`, and they SHOULD only have one value per language tag.

Description

Human-oriented documentation is of importance.

Example

```
cb-c:LegalEntityShape
a sh:NodeShape;
rdfs:comment "Shape for checking instances of the class Legal Entity"@en;
rdfs:label "Legal Entity Shape"@en;
sh:targetClass cb:LegalEntity;
```

Aggregated example

```
@prefix cb-c: <https://data.europe.eu/semanticassets/ns/cbv-constraints_v1.0.0#> .
④
@prefix cb: <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix dcterms: <http://purl.org/dc/terms/> .

<https://data.europe.eu/semanticassets/ns/cbv-constraints_v1.0.0> ①
  a owl:Ontology; ①
  owl:imports <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#>; ②
  dcterms:title "Business eGovernment Core Vocabulary constraint SHACL"@en; ③
  dcterms:description ""The Core Business Vocabulary constraint SHACL provides
the classes with their attributes and relationships, and the constraints on
occurrences and values used.""@en; ③
  .
cb-c:LegalEntityShape ⑥
  a sh:NodeShape; ⑤
  sh:targetClass cb:LegalEntity; ⑤
  rdfs:comment "Shape for checking instances of the class Legal Entity"@en; ⑦
  rdfs:label "Legal Entity Shape"@en; ⑦
  .
```

① Guideline 79

② Guideline 80

③ Guideline 81

④ Guideline 82

⑤ Guideline 83

⑥ Guideline 84

⑦ Guideline 85

6.1.5. PropertyShape

Classes contain attributes and have relationships. Each attribute and relationship contains occurrence indicators and constraints on the allowed values. These requirements are captured in SHACL PropertyShapes. The following guidelines describe how these PropertyShapes need to be formulated.

Guideline 86: PropertyShape

Summary

For every attribute and relationship related to a Class:

- A **sh:property** SHALL be added to the **NodeShape** pertaining to the Class;
- The value of this **sh:property** SHALL be a **PropertyShape** with a **sh:path** property having the respective attribute or relationship as value.

Description

These are examples of the essential constraints. For the full range of possibilities, please refer to the SHACL standard.^[21] and the book “Validating RDF data”.^[22]

Example

Starting from following UML Class diagram:

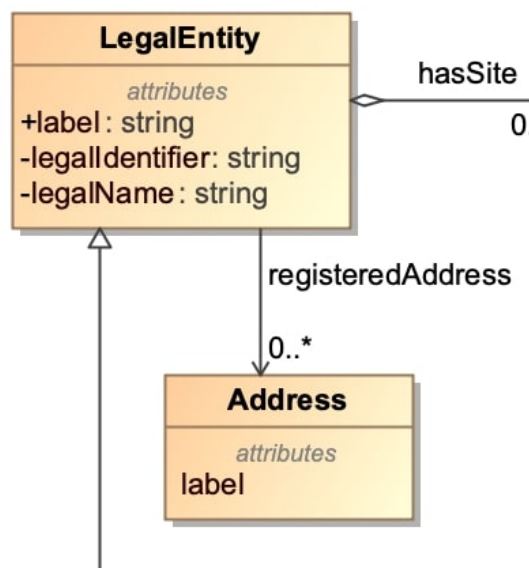


Figure 42. Example attribute and relationship

An example of the translation for attribute `legalIdentifier` is given below:

```
cb-c:LegalEntityShape
  a sh:NodeShape;
  rdfs:comment "Shape for checking instances of the class Legal Entity"@en;
  rdfs:label "Legal Entity Shape"@en;
  sh:targetClass cb:LegalEntity;
  sh:property cb-c:LegalEntity-legalIdentifier.

cb-c:LegalEntity-legalIdentifier
  a sh:PropertyShape;
  sh:path cb:legalIdentifier;.
```

Once a path is indicated, additional constraints can be added. In the case of attributes, it is possible to indicate that the values must be of a specific data type.

```
cb-c:LegalEntity-legalIdentifier
  a sh:PropertyShape;
  sh:path cb:legalIdentifier;
  sh:datatype xsd:string;.
```

An example that complies to this definition is given here.

```
:CompanyZ a cb:LegalEntity;
cb:legalIdentifier "Company Z SA";.
```

In the case of relationships, it is possible to indicate that the values must be instances of the specific class.

```
cb-c:LegalEntity-registeredAddress
  a sh:PropertyShape;
  sh:path cb:registeredAddress;
  sh:class cb:Address;.
```

An example that complies to this definition is given here.

```
:CompanyZ a cb:LegalEntity;
cb:legalIdentifier "Company Z";
cb:registeredAddress :adres125.
:adres125 a cb:Address.
```

Aggregated example

```

@prefix cb-c: <https://data.europe.eu/semanticassets/ns/cbv-constraints_v1.0.0#> .
④
@prefix cb: <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix dcterms: <http://purl.org/dc/terms/> .

<https://data.europe.eu/semanticassets/ns/cbv-constraints_v1.0.0> ①
  a owl:Ontology; ①
  owl:imports <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#>; ②
  dcterms:title "Business eGovernment Core Vocabulary constraint SHACL"@en; ③
  dcterms:description ""The Core Business Vocabulary constraint SHACL provides
the classes with their attributes and relationships, and the constraints on
occurrences and values used.""@en; ③
  .
cb-c:LegalEntityShape ⑥
  a sh:NodeShape; ⑤
  sh:targetClass cb:LegalEntity; ⑤
  rdfs:comment "Shape for checking instances of the class Legal Entity"@en; ⑦
  rdfs:label "Legal Entity Shape"@en; ⑦
  sh:property cb-c:LegalEntity-companyActivity; ⑧
  .

cb-c:LegalEntity-companyActivity
  a sh:PropertyShape; ⑧
  sh:path cb:companyActivity; ⑧
  sh:class <http://www.w3.org/2004/02/skos/core#Concept>;
  sh:description ""The activity of a company should be recorded using a
controlled vocabulary expressed
as a SKOS concept scheme (see <a
xmlns="https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#"
href="#"skos:Concept">Code</a>. Several such vocabularies exist, many of which
map to the UN's ISIC codes. Where a particular controlled vocabulary is in use
within a given context, such as SIC codes in the UK, it is acceptable to use
these, however, the preferred choice for European interoperability is NACE.""@en;
  sh:name "company activity"@en;
  sh:nodeKind sh:IRI;
  .

```

① Guideline 79

② Guideline 80

③ Guideline 81

④ Guideline 82

⑤ Guideline 83

⑥ Guideline 84

⑦ Guideline 85

⑧ Guideline 86

Guideline 87: PropertyShape II

Summary

Every class SHALL have in its template at least one `rdfs:label` with only one value per language tag.

- A class needs to have at least one `rdfs:label` in English.^[23];
- More are allowed for other languages, but only one `rdfs:label` per language.

Description

The guideline represents the formalisation of the general rule that every instance of a class shall have a `rdfs:label`. `rdfs:label` can be used in human-oriented presentations instead of the more opaque URI. In just this exceptional case, `sh:minCount` and `sh:maxCount` should be used.

Example

```
:x    a sh:PropertyShape;  
      sh:path rdfs:label;  
      sh:datatype rdf:langString;  
      sh:name "required rdfs:label"@en;  
      sh:description "every instance of this class needs to have at least 1 label.  
Only one label per language is allowed."@en;  
      sh:message "At least one label is needed."@en;  
      sh:minCount 1;  
      sh:uniqueLang true;.
```

Aggregated example

```

@prefix cb-c: <https://data.europe.eu/semanticassets/ns/cbv-constraints_v1.0.0#> .
④
@prefix cb: <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix dcterms: <http://purl.org/dc/terms/> .

<https://data.europe.eu/semanticassets/ns/cbv-constraints_v1.0.0> ①
  a owl:Ontology; ①
  owl:imports <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#>; ②
  dcterms:title "Business eGovernment Core Vocabulary constraint SHACL"@en; ③
  dcterms:description ""The Core Business Vocabulary constraint SHACL provides
the classes with their attributes and relationships, and the constraints on
occurrences and values used.""@en; ③
  .
cb-c:LegalEntityShape ⑥
  a sh:NodeShape; ⑤
  sh:targetClass cb:LegalEntity; ⑤
  rdfs:comment "Shape for checking instances of the class Legal Entity"@en; ⑦
  rdfs:label "Legal Entity Shape"@en; ⑦
  sh:property cb-c:LegalEntity-companyActivity; ⑧
  .

cb-c:LegalEntity-companyActivity
  a sh:PropertyShape; ⑧
  sh:path cb:companyActivity; ⑧
  sh:class <http://www.w3.org/2004/02/skos/core#Concept>;
  sh:description ""The activity of a company should be recorded using a
controlled vocabulary expressed
as a SKOS concept scheme (see <a
xmlns="https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#"
href="#"skos:Concept">Code</a>. Several such vocabularies exist, many of which
map to the UN's ISIC codes. Where a particular controlled vocabulary is in use
within a given context, such as SIC codes in the UK, it is acceptable to use
these, however, the preferred choice for European interoperability is NACE.""@en;
  sh:name "company activity"@en;
  sh:nodeKind sh:IRI;
  .

cb-c:rdfsLabel ⑨
  a sh:PropertyShape;
  sh:path rdfs:label;

```

① Guideline 79

② Guideline 80

③ Guideline 81

④ Guideline 82

⑤ Guideline 83

⑥ Guideline 84

⑦ Guideline 85

⑧ Guideline 86

⑨ Guideline 87

Guideline 88: PropertyShape III

Summary

All property shapes SHALL have a URI, i.e. they may not be blank nodes.

Description

If the eGovernment Core Vocabulary SHACL file is imported into the constraints file of a deriving application profile, then additional or more restricted property constraints can be added to these identifiers.

Example

```
cb-c:LegalEntityShape
  a sh:NodeShape;
  sh:targetClass cb:LegalEntity;
  rdfs:comment "Shape for checking instances of Legal Entity"@en;
  rdfs:label "Legal entity shape"@en;
  sh:property cb-c:LegalEntity-companyActivity;
  .
cb-c:LegalEntity-companyActivity
  a sh:PropertyShape;
  .
```

Aggregated example

```

@prefix cb-c: <https://data.europe.eu/semanticassets/ns/cbv-constraints_v1.0.0#> .
④
@prefix cb: <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix dcterms: <http://purl.org/dc/terms/> .

<https://data.europe.eu/semanticassets/ns/cbv-constraints_v1.0.0> ①
  a owl:Ontology; ①
  owl:imports <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#>; ②
  dcterms:title "Business eGovernment Core Vocabulary constraint SHACL"@en; ③
  dcterms:description ""The Core Business Vocabulary constraint SHACL provides
the classes with their attributes and relationships, and the constraints on
occurrences and values used.""@en; ③
  .
cb-c:LegalEntityShape ⑥
  a sh:NodeShape; ⑤
  sh:targetClass cb:LegalEntity; ⑤
  rdfs:comment "Shape for checking instances of the class Legal Entity"@en; ⑦
  rdfs:label "Legal Entity Shape"@en; ⑦
  sh:property cb-c:LegalEntity-companyActivity; ⑧ ⑩
  sh:property cb-c:rdfsLabel;
  .

cb-c:LegalEntity-companyActivity
  a sh:PropertyShape; ⑧ ⑩
  sh:path cb:companyActivity; ⑧
  sh:class <http://www.w3.org/2004/02/skos/core#Concept>;
  sh:description ""The activity of a company should be recorded using a
controlled vocabulary expressed
as a SKOS concept scheme (see <a
xmlns="https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#"
href="#skos:Concept">Code</a>. Several such vocabularies exist, many of which
map to the UN's ISIC codes. Where a particular controlled vocabulary is in use
within a given context, such as SIC codes in the UK, it is acceptable to use
these, however, the preferred choice for European interoperability is NACE.""@en;
  sh:name "company activity"@en;
  sh:nodeKind sh:IRI;
  .

cb-c:rdfsLabel ⑨
  a sh:PropertyShape;
  sh:path rdfs:label;
  .

```

① Guideline 79

② Guideline 80

③ Guideline 81

④ Guideline 82

⑤ Guideline 83

⑥ Guideline 84

⑦ Guideline 85

⑧ Guideline 86

⑨ Guideline 87

⑩ Guideline 88

Guideline 89: PropertyShape IV

Summary

Each property SHALL have a `sh:name` (in the context of the target where it appears) to provide human-oriented labels. This is the preferred alternative to overwriting `rdfs:labels` coming from external foreign vocabularies to fit the context better.

Description

This allows localised labels.

Example

```
cb-c:LegalEntity-dct_alternative
a sh:PropertyShape;
sh:path dct:alternative;
sh:name "alternative company name"@en;
sh:nodeKind sh:Literal;.
```

Aggregated example

```
@prefix cb-c: <https://data.europe.eu/semanticassets/ns/cbv-constraints_v1.0.0#> .
④
@prefix cb: <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix dcterms: <http://purl.org/dc/terms/> .

<https://data.europe.eu/semanticassets/ns/cbv-constraints_v1.0.0> ①
a owl:Ontology; ①
owl:imports <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#>; ②
dcterms:title "Business eGovernment Core Vocabulary constraint SHACL"@en; ③
dcterms:description ""The Core Business Vocabulary constraint SHACL provides
the classes with their attributes and relationships, and the constraints on
occurrences and values used.""@en; ③
.
cb-c:LegalEntityShape ⑥
a sh:NodeShape; ⑤
sh:targetClass cb:LegalEntity; ⑤
rdfs:comment "Shape for checking instances of the class Legal Entity"@en; ⑦
rdfs:label "Legal Entity Shape"@en; ⑦
sh:property cb-c:LegalEntity-companyActivity;⑧ ⑩
sh:property cb-c:rdfsLabel;
.
cb-c:LegalEntity-companyActivity
a sh:PropertyShape; ⑧ ⑩
sh:path cb:companyActivity;⑧
```



```

sh:class <http://www.w3.org/2004/02/skos/core#Concept>;
sh:description ""The activity of a company should be recorded using a
controlled vocabulary expressed
as a SKOS concept scheme (see <a
xmlns="https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#"
href="#skos:Concept">Code</a>. Several such vocabularies exist, many of which
map to the UN's ISIC codes. Where a particular controlled vocabulary is in use
within a given context, such as SIC codes in the UK, it is acceptable to use
these, however, the preferred choice for European interoperability is NACE.""@en;
sh:name "company activity"@en; ⑪
sh:nodeKind sh:IRI;
.

cb-c:rdfsLabel ⑨
a sh:PropertyShape;
sh:path rdfs:label;
sh:description "Every instance should get a rdfs:label"@en;
sh:name "rdfs label"@en; ⑪
sh:datatype rdf:langString;
sh:minCount 1;
sh:message "This instance of a clas does not have rdfs:label"@en;
sh:uniqueLanguage true.

```

① Guideline 79

② Guideline 80

③ Guideline 81

④ Guideline 82

⑤ Guideline 83

⑥ Guideline 84

⑦ Guideline 85

⑧ Guideline 86

⑨ Guideline 87

⑩ Guideline 88

⑪ Guideline 89

Guideline 90: PropertyShape V

Summary

Each property SHOULD have `sh:description` (in the context of the target where it appears) to provide further human-oriented details. This is the preferred solution over adding additional usage notes to fit the context.

Description

This allows localised definitions.

Example

```
cb-c:LegalEntity-dct_alternative
  a sh:PropertyShape;
  sh:path dct:alternative;
  sh:description "Some jurisdictions recognise concepts such as a trading name or
alternative forms of a legal entity's name. The Alternative Name property can be
used to record such names but should not be used to record translations of the
primary legal name. Where more than one legal name exists and where they have
equal standing but are expressed in different languages, identify the language
used in each of the multiple legal names."@en;
  sh:name "alternative company name"@en;
  sh:nodeKind sh:Literal;.
```

Aggregated example

```
@prefix cb-c: <https://data.europe.eu/semanticassets/ns/cbv-constraints_v1.0.0#> .
④
@prefix cb: <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix dcterms: <http://purl.org/dc/terms/> .

<https://data.europe.eu/semanticassets/ns/cbv-constraints_v1.0.0#> ①
  a owl:Ontology; ①
  owl:imports <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#>; ②
  dcterms:title "Business eGovernment Core Vocabulary constraint SHACL"@en; ③
  dcterms:description ""The Core Business Vocabulary constraint SHACL provides
the classes with their attributes and relationships, and the constraints on
occurrences and values used.""@en; ③
  .
cb-c:LegalEntityShape ⑥
  a sh:NodeShape; ⑤
  sh:targetClass cb:LegalEntity; ⑤
  rdfs:comment "Shape for checking instances of the class Legal Entity"@en; ⑦
  rdfs:label "Legal Entity Shape"@en; ⑦
  sh:property cb-c:LegalEntity-companyActivity;⑧ ⑩
```

```

sh:property cb-c:rdfsLabel;
.

cb-c:LegalEntity-companyActivity
  a sh:PropertyShape; ⑧ ⑩
  sh:path cb:companyActivity;⑧
  sh:class <http://www.w3.org/2004/02/skos/core#Concept>;
  sh:description ""The activity of a company should be recorded using a
controlled vocabulary expressed
as a SKOS concept scheme (see <a
xmlns="https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#"
href="#skos:Concept">Code</a>. Several such vocabularies exist, many of which
map to the UN's ISIC codes. Where a particular controlled vocabulary is in use
within a given context, such as SIC codes in the UK, it is acceptable to use
these, however, the preferred choice for European interoperability is NACE.""@en;
⑫
  sh:name "company activity"@en; ⑪
  sh:nodeKind sh:IRI;
.

cb-c:rdfsLabel ⑨
  a sh:PropertyShape;
  sh:path rdfs:label;
  sh:description "Every instance should get a rdfs:label"@en; ⑫
  sh:name "rdfs label"@en; ⑪
  sh:datatype rdf:langString;
  sh:minCount 1;
  sh:message "This instance of a clas does not have rdfs:label"@en;
  sh:uniqueLanguage true.

```

① Guideline 79

② Guideline 80

③ Guideline 81

④ Guideline 82

⑤ Guideline 83

⑥ Guideline 84

⑦ Guideline 85

⑧ Guideline 86

⑨ Guideline 87

⑩ Guideline 88

⑪ Guideline 89

⑫ Guideline 90

Guideline 91: PropertyShape VI

Summary

`sh:minCount` and `sh:maxCount` SHOULD NOT be indicated . If not stated, `sh:minCount` means 0 and `sh:maxCount` means no limit. These are the default values.

Description

As noted earlier, the occurrences are indicated on the level of the application profiles.

Example

N/A

Guideline 92: PropertyShape VII

Summary

`sh:message` CAN be used to give the best possible error message in case the constraint is violated.

Description

Useful for generating more specific error messages.

Example

```
cb-c:rdfsLabel
a sh:PropertyShape;
sh:path rdfs:label;
sh:description "Every instance should get a rdfs:label"@en;
sh:name "rdfs label"@en;
sh:datatype rdf:langString;
sh:minCount 1;
sh:maxCount 1;
sh:uniqueLang true;
sh:targetClass rdfs:Class, owl:Class, rdf:Property, owl:DatatypeProperty,
owl:ObjectProperty;
sh:message "This instance of a class does not have a rdfs:label"@en;
.
```

Aggregated example

```
@prefix cb-c: <https://data.europe.eu/semanticassets/ns/cbv-constraints_v1.0.0#> .
④
@prefix cb: <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix dcterms: <http://purl.org/dc/terms/> .

<https://data.europe.eu/semanticassets/ns/cbv-constraints_v1.0.0> ①
a owl:Ontology; ①
owl:imports <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#>; ②
dcterms:title "Business eGovernment Core Vocabulary constraint SHACL"@en; ③
dcterms:description ""The Core Business Vocabulary constraint SHACL provides
the classes with their attributes and relationships, and the constraints on
occurrences and values used.""@en; ③
.
cb-c:LegalEntityShape ⑥
a sh:NodeShape; ⑤
sh:targetClass cb:LegalEntity; ⑤
rdfs:comment "Shape for checking instances of the class Legal Entity"@en; ⑦
rdfs:label "Legal Entity Shape"@en; ⑦
```

```
sh:property cb-c:LegalEntity-companyActivity; ⑧ ⑩
```

```
sh:property cb-c:rdfsLabel;
```

```
.
```

```
cb-c:LegalEntity-companyActivity
```

```
  a sh:PropertyShape; ⑧ ⑩
```

```
  sh:path cb:companyActivity; ⑧
```

```
  sh:class <http://www.w3.org/2004/02/skos/core#Concept>;
```

```
  sh:description ""The activity of a company should be recorded using a  
controlled vocabulary expressed
```

```
as a SKOS concept scheme (see <a
```

```
xmlns="https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#">
```

```
href="#skos:Concept">Code</a>. Several such vocabularies exist, many of which  
map to the UN's ISIC codes. Where a particular controlled vocabulary is in use
```

```
within a given context, such as SIC codes in the UK, it is acceptable to use
```

```
these, however, the preferred choice for European interoperability is NACE.""@en;
```

```
⑫
```

```
  sh:name "company activity"@en; ⑪
```

```
  sh:nodeKind sh:IRI;
```

```
.
```

```
cb-c:rdfsLabel ⑨
```

```
  a sh:PropertyShape;
```

```
  sh:path rdfs:label;
```

```
  sh:description "Every instance should get a rdfs:label"@en; ⑫
```

```
  sh:name "rdfs label"@en; ⑪
```

```
  sh:datatype rdf:langString;
```

```
  sh:minCount 1;
```

```
  sh:message "This instance of a clas does not have rdfs:label"@en; ⑬
```

```
  sh:uniqueLanguage true.
```

① Guideline 79

② Guideline 80

③ Guideline 81

④ Guideline 82

⑤ Guideline 83

⑥ Guideline 84

⑦ Guideline 85

⑧ Guideline 86

⑨ Guideline 87

⑩ Guideline 88

⑪ Guideline 89

⑫ Guideline 90

⑬ Guideline 92

6.2. SHACL and Application Profiles

The guidelines in the section should be used when defining additional or more specific constraints at the level of application profiles (see Figure below). The application profile can import the constraints file of the eGovernment Core Vocabulary and enhance it with additional constraints.

As indicated earlier: occurrences need to be set at this level.

Since all imported Property Shapes have identifiers, it is possible to attach/add further constraints and, if required, to deactivate those.

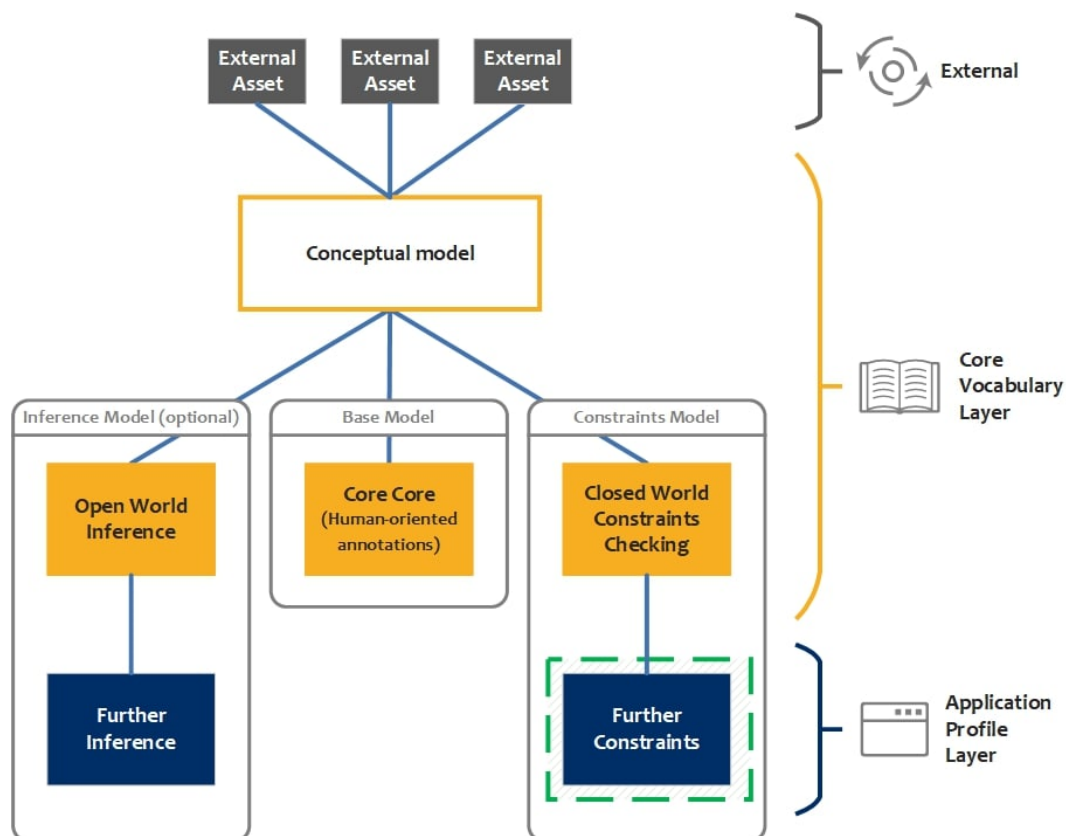


Figure 43. Situating SHACL constraint checking for Application Profiles

6.2.1. SHACL Level Application Profile

The following guidelines describe how the setup environment needs to be formulated while creating SHACL files for the Application Profiles.

Guideline 93: SHACL Level

Summary

Every ontology SHALL be declared following the naming guidelines in [Naming and versioning](#).

Description

Every vocabulary should be uniquely identifiable.

Example

```
<https://data.europe.eu/semanticassets/ns/cbv-ap_v1.0.0>  
  rdf:type owl:Ontology; .
```

Aggregated example

```
@prefix owl: <http://www.w3.org/2002/07/owl#> .  
<https://data.europe.eu/semanticassets/ns/cbv-ap_v1.0.0> ①  
  a owl:Ontology; ①  
  .
```

① Guideline 93

Guideline 94: SHACL Level II

Summary

The related SHACL constraint file of the eGovernment Core Vocabulary SHOULD be imported. Alternatively, an existing application profile CAN be imported, if it is being extended.

Description

The existing structures need to be further refined, extended or restricted. Importing is preferred over copy and paste, which leads to errors and semantic hijacking.

Example

In the following case, one starts an application profile based on the eGovernment Business eGovernment Core Vocabulary. This application profile will extend the constraints already defined in the SHACL shape file of the eGovernment Core Vocabulary itself, hence the import `owl:imports` is done.

```
<https://data.europe.eu/semanticassets/ns/cbv-ap_v1.0.0>
  rdf:type owl:Ontology;
  owl:imports <https://data.europe.eu/semanticassets/ns/cbv-constraints_v1.0.0>;.
```

Aggregated example

```
@prefix owl: <http://www.w3.org/2002/07/owl#> .
<https://data.europe.eu/semanticassets/ns/cbv-ap_v1.0.0> ①
  a owl:Ontology; ①
  owl:imports <https:// data.europe.eu/semanticassets/ns/cbv-constraints_v1.0.0>;
②
.
```

① Guideline 93

② Guideline 94

Guideline 95: SHACL Level III

Summary

Prefixes for all namespaces SHOULD be defined.

Description

The use of prefixes leads to more naturally readable statements.

Example

```
@prefix cb-c: <https://data.europe.eu/semanticassets/ns/cbv-constraints_v1.0.0#> .
@prefix cb: <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#> .
@prefix cb-ap: <https://data.europe.eu/semanticassets/ns/cbv-ap_v1.0.0#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
```

Aggregated example

```
@prefix cb-ap: <https://data.europe.eu/semanticassets/ns/cbv-ap_v1.0.0#> . ③
@prefix cb-c: <https://data.europe.eu/semanticassets/ns/cbv-constraints_v1.0.0#> .
@prefix cb: <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
<https://data.europe.eu/semanticassets/ns/cbv-ap_v1.0.0#> ①
  a owl:Ontology; ①
  owl:imports <https://data.europe.eu/semanticassets/ns/cbv-constraints_v1.0.0#>;
②
.
```

① Guideline 93

② Guideline 94

③ Guideline 95

Guideline 96: SHACL Level IV

Summary

Semantic Asset Metadata SHALL be assigned as described in *Metadata*.

Description

This helps to find the constraint SHACL file of the vocabulary at semantic asset portals and to evaluate if it suits the needs.

Example

```
dcterms:title "Business eGovernment Core Vocabulary constraint SHACL AP"@en;  
dcterms:description ""The Core Business Vocabulary constraint SHACL Application  
Profile provides the classes with their attributes and relationships, and the  
constraints on occurrences and values used.""@en;
```

Aggregated example

```
prefix cb-ap: <https://data.europe.eu/semanticassets/ns/cbv-ap_v1.0.0#> . ③  
@prefix cb-c: <https://data.europe.eu/semanticassets/ns/cbv-constraints_v1.0.0#> .  
@prefix cb: <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#> .  
@prefix owl: <http://www.w3.org/2002/07/owl#> .  
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .  
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .  
@prefix sh: <http://www.w3.org/ns/shacl#> .  
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .  
@prefix owl: <http://www.w3.org/2002/07/owl#> .  
@prefix dcterms: <http://purl.org/dc/terms/> .  
<https://data.europe.eu/semanticassets/ns/cbv-ap_v1.0.0> ①  
  a owl:Ontology; ①  
  owl:imports <https://data.europe.eu/semanticassets/ns/cbv-constraints_v1.0.0>;  
②  
  dcterms:title "Business eGovernment Core Vocabulary constraint SHACL AP"@en; ④  
  dcterms:description ""The Core Business Vocabulary constraint SHACL Application  
Profile provides the classes with their attributes and relationships, and the  
constraints on occurrences and values used.""@en; ④  
  .
```

① Guideline 93

② Guideline 94

③ Guideline 95

④ Guideline 96

6.2.2. NodeShape and PropertyShape

The Class NodeShape guidelines and PropertyShape guidelines are similar to those formulated for the eGovernment Core Vocabularies, except for occurrence indicators.

Guideline 97: SHACL Level

Summary

`sh:minCount` and `sh:maxCount` SHOULD be indicated. ^[24]

Description

As noted earlier, the occurrences are indicated on the level of the application profiles.

Example

```
cb-c:LegalEntity-companyStatus
  sh:in (
    <http://vocab.belgif.be/auth/companyStatus/active>
    <http://vocab.belgif.be/auth/companyStatus/defaulted>
    <http://vocab.belgif.be/auth/companyStatus/stopped>
  );
  sh:minCount 1;
  sh:maxCount 1;
```

Aggregated example

```
@prefix cb-ap: <https://data.europe.eu/semanticassets/ns/cbv-ap_v1.0.0#> . ③
@prefix cb-c: <https://data.europe.eu/semanticassets/ns/cbv-constraints_v1.0.0#> .
@prefix cb: <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
<https://data.europe.eu/semanticassets/ns/cbv-ap_v1.0.0> ①
  a owl:Ontology; ①
  owl:imports <https://data.europe.eu/semanticassets/ns/cbv-constraints_v1.0.0#>;
  ②
  dcterms:title "Business eGovernment Core Vocabulary constraint SHACL AP"@en; ④
  dcterms:description ""The Core Business Vocabulary constraint SHACL Application
Profile provides the classes with their attributes and relationships, and the
constraints on occurrences and values used.""@en; ④
  .
cb-c:LegalEntity-companyStatus
  sh:in (
    <http://vocab.belgif.be/auth/companyStatus/active>
    <http://vocab.belgif.be/auth/companyStatus/defaulted>
    <http://vocab.belgif.be/auth/companyStatus/stopped>
  );
  sh:minCount 1; ⑤
  sh:maxCount 1; ⑤
  .
```

① Guideline 93

② Guideline 94

③ Guideline 95

④ Guideline 96

⑤ Guideline 97

6.2.3. PropertyShape Refinement

This subsection describes, through an example, the way the PropertyShape refinements can be implemented by introducing additional constraints and properties for the PropertyShape.

Guideline 98: PropertyShape Refinement

Summary

PropertyShape refinements CAN be implemented by introducing additional constraints and properties for the PropertyShape.

Description

Occurrences and values are indicated on the level of the application profiles.

Example

Assuming this is the model on the eGovernment Core Vocabulary level (see Figure below).

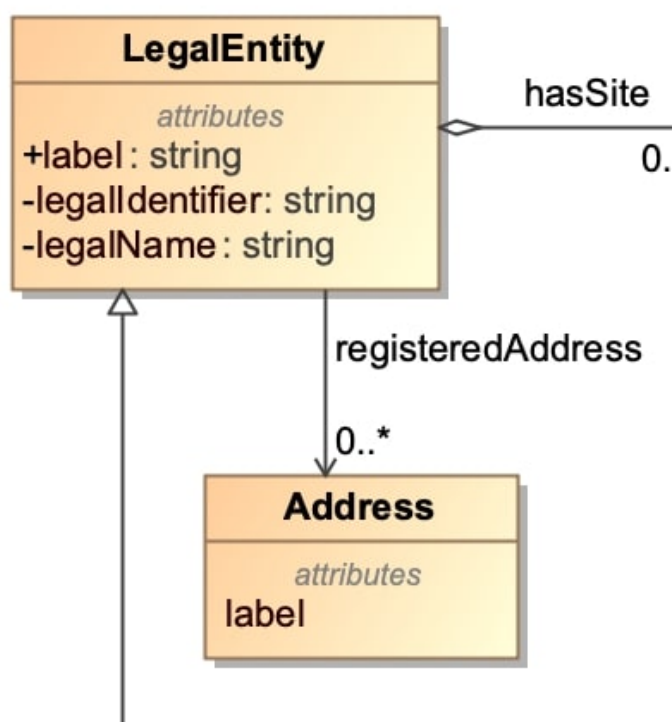


Figure 44. Model from eGovernment Core Vocabulary

This model can be further refined in the deriving application profile (see Figure below).

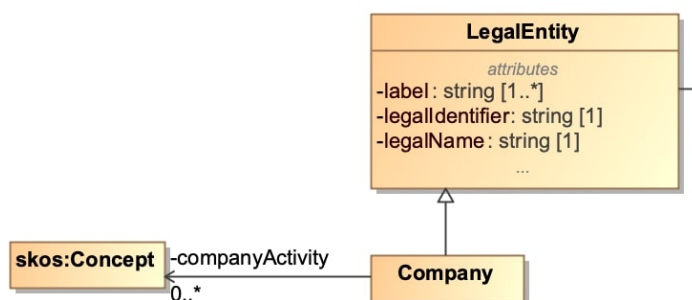


Figure 45. Refined model of the Application Profile

Then, the `cb-c:LegalEntity-legalIdentifier` can be refined (since it is imported) by adding

the following additional constraints.

```
cb-c:LegalEntity-legalIdentifier
  sh:minOccurs 1;
  sh:maxOccurs 1;.
```

And, for adding further properties such as `companyActivity`, one can extend the existing `CompanyShape`

```
cb-c:CompanyShape
sh:property cb-ap:Company-companyActivity.

cb-ap:Company-companyActivity a sh:PropertyShape;
  sh:path cb:companyActivity;
  sh:class skos:Concept;
.
```

Aggregated example


```

@prefix cb-ap: <https://data.europe.eu/semanticassets/ns/cbv-ap_v1.0.0#> . ③
@prefix cb-c: <https://data.europe.eu/semanticassets/ns/cbv-constraints_v1.0.0#> .
@prefix cb: <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
<https://data.europe.eu/semanticassets/ns/cbv-ap_v1.0.0> ①
  a owl:Ontology; ①
  owl:imports <https://data.europe.eu/semanticassets/ns/cbv-constraints_v1.0.0#>;
②
  dcterms:title "Business eGovernment Core Vocabulary constraint SHACL AP"@en; ④
  dcterms:description ""The Core Business Vocabulary constraint SHACL Application
Profile provides the classes with their attributes and relationships, and the
constraints on occurrences and values used.""@en; ④
.
cb-c:LegalEntity-companyStatus
  sh:in (
    <http://vocab.belgif.be/auth/companyStatus/active>
    <http://vocab.belgif.be/auth/companyStatus/defaulted>
    <http://vocab.belgif.be/auth/companyStatus/stopped>
  );
  sh:minCount 1; ⑤ ⑥
  sh:maxCount 1; ⑤ ⑥
.

```

① Guideline 93

② Guideline 94

③ Guideline 95

④ Guideline 96

⑤ Guideline 97

⑥ Guideline 98

[21] <https://www.w3.org/TR/shacl/>

[22] <https://book.validatingrdf.com/>

[23] This constraint is implemented in SHACL-SPARQL.

[24] Except for the required `rdfs:label`.

Chapter 7. XSD guidelines

7.1. XSD Design

W3C XML Schema Definition commonly referred to as ‘XSD’ is the means to check the validity of the structure and the vocabulary of an XML document. Transitioning from UML class diagrams (see [Class Diagrams Style Guide](#)).

An XSD captures the object-oriented aspects of the modelling (similarly to SHACL). It establishes element/attribute declarations and simple/complex types to express a set of rules to which XML documents must conform ^[25]. For illustration purposes, these subsections use the Core Criterion and Core Evidence Vocabulary v2.0.0 artefacts available in the ‘branch CV-2.0.0’ of the SEMIC GitHub CCCEV repository ^[26].

The figure below shows how the general architecture approach proposed to the development of an XSD file defining an eGovernment Core Vocabulary:

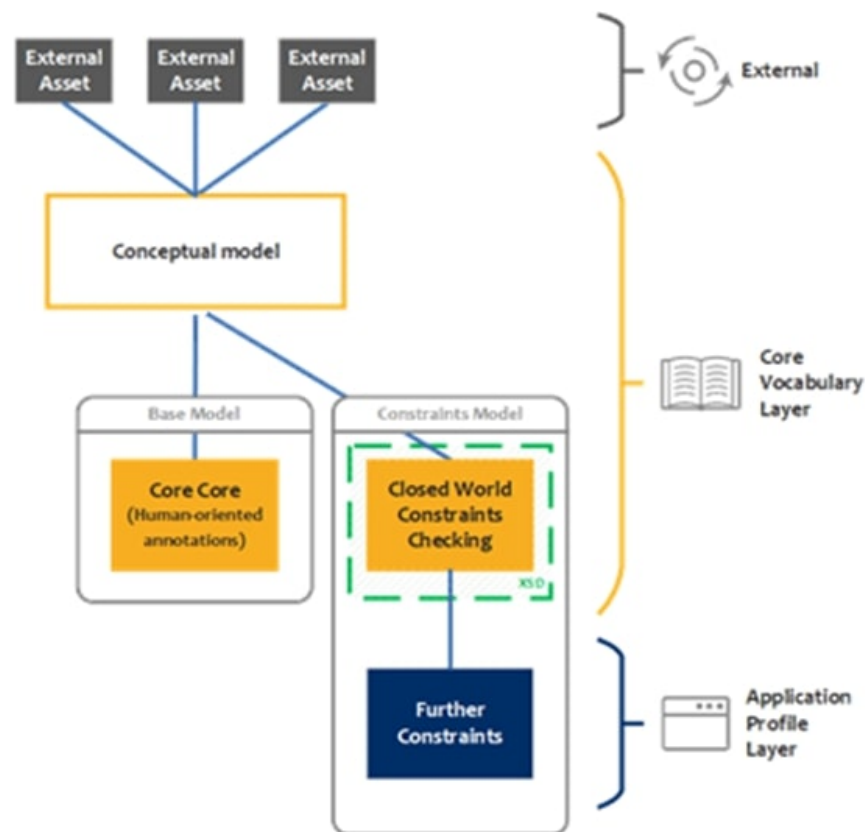


Figure 46. General architecture approach proposed

Based on the general architecture approach, the following guidelines describe how the setup environment needs to be formulated while creating XSD files, following the modelling approach described throughout this document.

Guideline 99: Specify version and encoding

Summary

As it is an XML requirement, versions and related encoding SHALL be identified.

Description

All XSD Schemas defining an eGovernment vocabulary or a library of information elements (e.g. common reusable components) SHALL identify the version of the XML specification and the encoding used. This piece of XML code SHALL be the first line of the file. Global comments, too, SHALL be located after this element.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
  <!-- Global comments SHALL go here --->
  <xsd:schema xmlns="https://semic.org/sa/cv/cccev-2.0.0#"
  ...
</xsd:schema>
```

Aggregated example

```
<?xml version="1.0" encoding="UTF-8"?> ①
<!-- This is an example XSD --> ①
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="https://semic.org/sa/cv/cccev-2.0.0#"
  targetNamespace="https://semic.org/sa/cv/cccev-2.0.0#"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified"
  version="2.0.0">
</xsd:schema>
```

① Guideline 99

Guideline 100: Specify namespace and version

Summary

The root element of the XSD Schema SHALL specify default namespace, targetNamespace and version. This is a basic common XSD design practice, also compatible with other standard designs (e.g. UN/CEFACT, UBL, HR-Open Standards).

Description

The root element of the XSD Schema SHALL specify the following attributes:

- the default and target Namespaces SHALL be specified. Both, default and target namespaces SHALL be identical;
- the attributeFormDefault CAN be “unqualified”;
- the attribute version SHALL be used to identify the version of the vocabulary;
- a prefix SHALL be specified to identify any element imported from other external vocabularies and libraries of commonly reused components.

Example

```
<xsd:schema xmlns="https://semic.org/sa/cv/cccev-2.0.0#"
  xmlns:dcap="http://data.europa.eu/r5r/"
  xmlns:cac="https://semic.org/sa/cv/common/cac-2.0.0#"
  xmlns:cbc="https://semic.org/sa/cv/common/cbc-2.0.0#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  targetNamespace="https://semic.org/sa/cv/cccev-2.0.0#" elementFormDefault=
"qualified"
  attributeFormDefault="unqualified" version="2.0.0">
...
</xsd:schema>
```

Aggregated example

```

<?xml version="1.0" encoding="UTF-8"?>①
<!-- This is an example XSD -->①
<xsd:schema xmlns="https://semic.org/sa/cv/cccev-2.0.0#" ②
  xmlns:dcap-ap="http://data.europa.eu/r5r/" ②
  xmlns:cac="https://semic.org/sa/cv/common/cac-2.0.0#"
  xmlns:cbc="https://semic.org/sa/cv/common/cbc-2.0.0#"
  xmlns:ccts=
"urn:un:unece:uncefact:data:specification:CoreComponentTypeSchemaModule:2"
  xmlns:an="https://semic.org/sa/cv/common/semic-annot-2.0.0#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:foaf="http://xmlns.com/foaf/0.1/" targetNamespace=
"https://semic.org/sa/cv/cccev-2.0.0#" ②
  elementFormDefault="qualified"
  attributeFormDefault="unqualified"
  version="2.0.0"> ②
</xsd:schema>

```

① Guideline 99

② Guideline 100

Guideline 101: Re-use external vocabularies

Summary

Any vocabulary defined by SEMIC or by other authorities CAN be reused provided that they were defined as W3C XSD Schemas. This encompasses the reuse of both properties and classes.

Description

The recommendation is to reuse vocabularies defined by SEMIC or ISA. If there is the need of reusing other vocabularies defined externally to ISA which were never implemented as XSD (e.g. DCAT-AP, FOAF, SPDX, etc.), the recommendation is that the XSD implementation SHOULD be developed according to SEMIC style guide (this document).

The definition of new XSD vocabularies or libraries of components SHALL define their own namespace. If the vocabulary was already designed using another modelling paradigm and syntax (e.g. RDF and Turtle), the namespace for the XSD implementation SHOULD use the namespace used for that serialisation.

These reused vocabularies SHALL be ‘imported’, never ‘included’. For this, the schema location SHALL be provided, either as a URL or as file path and name. If the URL is provided, the resource (the imported XSD) must be available and accessible when using the Core Vocabulary that imports it. If the resource is imported from a file system, the path CAN (strongly encouraged) be ‘relative’ to the location of the Core Vocabulary that imports the reused XSD.

The imported vocabularies SHALL be also assigned to a prefix in the root element. Each element of the Core Vocabulary referring to one of those elements SHALL then use the prefix to unambiguously specify that the element belongs to a specific namespace.

Example

```

<xsd:schema xmlns="https://semic.org/sa/cv/cccev-2.0.0#"
  xmlns:dcap="http://data.europa.eu/r5r/" xmlns:cac=
"https://semic.org/sa/cv/common/cac-2.0.0#"
  xmlns:cbc="https://semic.org/sa/cv/common/cbc-2.0.0#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  targetNamespace="https://semic.org/sa/cv/cccev-2.0.0#" elementFormDefault=
"qualified"
  attributeFormDefault="unqualified" version="2.0.0"><br>
  <xsd:import namespace="https://semic.org/sa/cv/common/cac-2.0.0#"
    schemaLocation="../../../../common/2.0.0/xml/xsdrt/CV-
CommonAggregateComponents.xsd"/>
  <xsd:import namespace="https://semic.org/sa/cv/common/cbc-2.0.0#"
    schemaLocation="../../../../common/2.0.0/xml/xsdrt/CV-
CommonBasicComponents.xsd"/>
  <xsd:import namespace="http://data.europa.eu/r5r/"
    schemaLocation="../../../../dcap/2.0.0/xml/dcap.xsd"/>
  <xsd:import namespace="http://xmlns.com/foaf/0.1/"
    schemaLocation="../../../../foaf/xml/foaf.xsd"/>
  <xsd:import namespace="http://www.w3.org/2002/07/owl#"
    schemaLocation="../../../../owl/2.0.0/xml/owl.xsd"/>

```

Aggregated example

```

<?xml version="1.0" encoding="UTF-8"?>①
<!-- This is an example XSD -->①
<xsd:schema xmlns="https://semic.org/sa/cv/cccev-2.0.0#" ②
  xmlns:dcap="http://data.europa.eu/r5r/" ②
  xmlns:cac="https://semic.org/sa/cv/common/cac-2.0.0#"
  xmlns:cbc="https://semic.org/sa/cv/common/cbc-2.0.0#"
  xmlns:ccts=
"urn:un:unece:uncefact:data:specification:CoreComponentTypeSchemaModule:2"
  xmlns:an="https://semic.org/sa/cv/common/semic-annot-2.0.0#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:foaf="http://xmlns.com/foaf/0.1/" targetNamespace=
"https://semic.org/sa/cv/cccev-2.0.0#" ②
  elementFormDefault="qualified"
  attributeFormDefault="unqualified"
  version="2.0.0"> ②
  <xsd:import namespace="https://semic.org/sa/cv/common/cac-2.0.0#"
    schemaLocation="../../../common/2.0.0/xml/xsd/CV-
CommonAggregateComponents.xsd"/> ③
    <xsd:import namespace="https://semic.org/sa/cv/common/cbc-2.0.0#"
      schemaLocation="../../../common/2.0.0/xml/xsd/CV-
CommonBasicComponents.xsd"/>
      <xsd:import namespace=
"urn:un:unece:uncefact:data:specification:CoreComponentTypeSchemaModule:2"
        schemaLocation="../../../common/2.0.0/xml/xsd/CCTS_CCT_SchemaModule-
2.3.xsd"/>
        <xsd:import namespace="http://data.europa.eu/r5r/"
          schemaLocation="../../../dcap/2.0.0/xml/dcap.xsd"/>
          <xsd:import namespace="http://www.w3.org/2002/07/owl#"
            schemaLocation="../../../owl/2.0.0/xml/owl.xsd"/> ③
            <xsd:import namespace="http://xmlns.com/foaf/0.1/"
              schemaLocation="../../../foaf/xml/foaf.xsd"/>
              </xsd:schema>

```

① Guideline 99

② Guideline 100

③ Guideline 101

Guideline 102: Define UML classes as complex types

Summary

In order to increase the semantic and technical interoperability by always using the same information elements from the same namespaces, all classes from UML SHALL be defined as complex element types.

Description

The following requirements are to be observed:

- `xsd:sequence` SHALL be used for the contained elements (do not use `xsd:all`);
- Class ‘attributes’: attributes SHOULD be reserved only for metadata about the element, not as value placeholders. For values use sequence and/or groups of elements inside the class.

Example

```
<xsd:complexType name="ConceptType">
  <xsd:complexContent>
    <xsd:extension base="owl:ThingType">
      <xsd:sequence>
        <xsd:element ref="cbc:id" minOccurs="0" maxOccurs="unbounded"> </xsd:element>
        <xsd:element ref="cbc:name" minOccurs="0" maxOccurs="1"> </xsd:element>
        <xsd:element ref="cbc:description" minOccurs="0" maxOccurs="unbounded">
</xsd:element>
        <xsd:element ref="cbc:type" minOccurs="0" maxOccurs="1"> </xsd:element>
        <xsd:element ref="cbc:qName" minOccurs="0" maxOccurs="unbounded">
</xsd:element>
        <xsd:element ref="constraint" minOccurs="0" maxOccurs="unbounded">
</xsd:element>
        <xsd:element ref="value" minOccurs="0" maxOccurs="unbounded">
          <!--we have two predicates that are the same with different domain/range-->
        </xsd:element>
        <xsd:element ref="concept" minOccurs="0" maxOccurs="unbounded">
</xsd:element>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

Aggregated example

```
<?xml version="1.0" encoding="UTF-8"?> <!-- guideline 99 -->
<!-- This is an example XSD -->①
<xsd:schema xmlns="https://semic.org/sa/cv/cccev-2.0.0#" ②
  xmlns:dcat-ap="http://data.europa.eu/r5r/" ②
  xmlns:cac="https://semic.org/sa/cv/common/cac-2.0.0#"
  xmlns:cbc="https://semic.org/sa/cv/common/cbc-2.0.0#"
  xmlns:ccts=
```

```

"urn:un:unece:uncefact:data:specification:CoreComponentTypeSchemaModule:2"
  xmlns:an="https://semic.org/sa/cv/common/semic-annot-2.0.0#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:foaf="http://xmlns.com/foaf/0.1/" targetNamespace=
"https://semic.org/sa/cv/cccev-2.0.0#" ②
  elementFormDefault="qualified"
  attributeFormDefault="unqualified"
  version="2.0.0"> ②
  <xsd:import namespace="https://semic.org/sa/cv/common/cac-2.0.0#"
    schemaLocation="../../../../common/2.0.0/xml/xsd/CV-
CommonAggregateComponents.xsd"/> ③
  <xsd:import namespace="https://semic.org/sa/cv/common/cbc-2.0.0#"
    schemaLocation="../../../../common/2.0.0/xml/xsd/CV-
CommonBasicComponents.xsd"/>
  <xsd:import namespace=
"urn:un:unece:uncefact:data:specification:CoreComponentTypeSchemaModule:2"
    schemaLocation="../../../../common/2.0.0/xml/xsd/CCTS_CCT_SchemaModule-
2.3.xsd"/>
  <xsd:import namespace="http://data.europa.eu/r5r/"
    schemaLocation="../../../../dcap-ap/2.0.0/xml/dcap-ap.xsd"/>
  <xsd:import namespace="http://www.w3.org/2002/07/owl#"
    schemaLocation="../../../../owl/2.0.0/xml/owl.xsd"/> ③
  <xsd:import namespace="http://xmlns.com/foaf/0.1/"
    schemaLocation="../../../../foaf/xml/foaf.xsd"/><br>
<!-- Declaration of elements --><br>
<!-- Data properties declarations -->
<xsd:element name="associatedRoleType" type="cbc:CodeType"/>
<!-- Object properties -->
<xsd:element name="concept" type="ConceptType"/>
<!-- Class declarations -->
<xsd:element name="Concept" type="ConceptType"/>
<!-- Element definition -->
<xsd:complexType name="ConceptType"> ④
  <xsd:complexContent>
    <xsd:extension base="owl:ThingType"> ③
      <xsd:sequence> ④
        <xsd:element ref="cbc:id" minOccurs="0" maxOccurs="unbounded">
</xsd:element>
        <xsd:element ref="cbc:name" minOccurs="0" maxOccurs="1"> </xsd:element>
        <xsd:element ref="cbc:description" minOccurs="0" maxOccurs="unbounded">
</xsd:element>
        <xsd:element ref="cbc:type" minOccurs="0" maxOccurs="1"> </xsd:element>
        <xsd:element ref="cbc:qName" minOccurs="0" maxOccurs="unbounded">
</xsd:element>
        <xsd:element ref="constraint" minOccurs="0" maxOccurs="unbounded">
</xsd:element>
        <xsd:element ref="value" minOccurs="0" maxOccurs="unbounded">
          <!--we have two predicates that are the same with different
domain/range-->
          </xsd:element>

```

```
        <xsd:element ref="concept" minOccurs="0" maxOccurs="unbounded">
    </xsd:element>
    </xsd:sequence>
  </xsd:extension>
</xsd:complexContent>
</xsd:complexType>
</xsd:schema>
```

- ① Guideline 99
- ② Guideline 100
- ③ Guideline 101
- ④ Guideline 102

Guideline 103: Facilitate reusing elements

Summary

Any mechanism that facilitates the modularisation and reuse of properties and classes defined **SHOULD** be used.

The main rationale behind this guideline can be summarised in the following:

- prevention of design errors by concentrating the definition in one single point of maintenance, e.g. avoid data definition duplicity and data-typing inconsistency;
- modularisation for reuse;
- simplification of the documentation;
- disambiguation of the semantics of specialised concepts;
- application of sound design principles.

Description

The reusability of properties and classes is mainly based on the inheritance types, as briefly described in the following:

- **Inheritance:** The use of generic and specialised classes **SHOULD** be considered always when defining eGovernment Core Vocabularies. A good example for this is the definition of the (quite abstract) class ‘Requirement’ in the CCCEV. All classes defined in an eGovernment Core Vocabulary **SHOULD** inherit the *owl:Thing* class. This is not always possible, e.g. when the defined class inherits from another class defined in an external (reused) vocabulary.
- **Multiple inheritance:** Since multiple inheritance is not possible in XML, two approaches **CAN** be used to ‘simulate’ some aspects which are characteristic in the multiple inheritance it:
 - **Groups of elements and attributes:** one **CAN** define groups of attributes or of elements and include the group in every class that needs to use the same attributes or elements. An example of this can be seen in the library of data-types defined for its reuse by any SEMIC vocabulary, where each data type, treated as a class (a complex element), reuses the Assurance Level attributes defined for eIDAS and similar contexts of uses .
 - **Indirect inheritance:** one **CAN** design hierarchies of classes and subclasses where the bottom classes inherit all the properties of the parent super-classes. This is an “incremental” and coherent way of expressing multiple concepts that are related in their nature but with slight nuances (properties) amongst themselves.

Example

Example I: groups of attributes:

```

<xsd:complexType name="IdentifierType">
  <xsd:simpleContent>
    <xsd:extension base="udt:IdentifierType">
      <xsd:attributeGroup ref="AssuranceLevelAttributes"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

```

Example II: ‘cascaded’ inheritance:

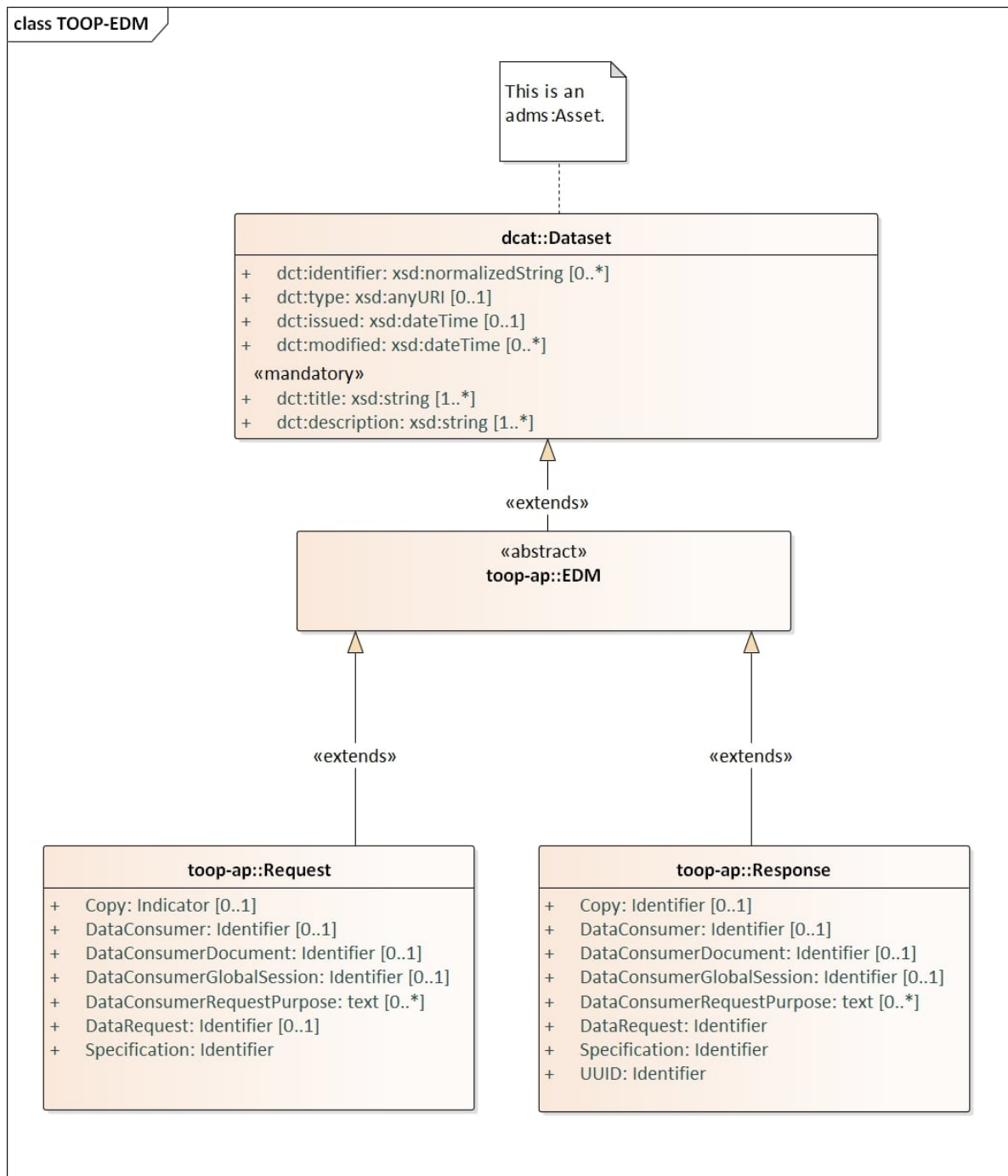


Figure 47. General architecture approach proposed

```

<xsd:complexType name="EDMType">
  <xsd:complexContent>
    <xsd:extension base="dcat-ap:DatasetType">
      <xsd:sequence>
...
<xsd:complexType name="TOOPRequestType">
  <xsd:complexContent>
    <xsd:extension base="EDMType">
      <xsd:sequence>
...
<xsd:complexType name="TOOPResponseType">
  <xsd:complexContent>
    <xsd:extension base="EDMType">
      <xsd:sequence>
...

```

Aggregated example

```

<?xml version="1.0" encoding="UTF-8"?>①
<!-- This is an example XSD -->①
<xsd:schema xmlns="https://semic.org/sa/cv/cccev-2.0.0#" ②
  xmlns:dcat-ap="http://data.europa.eu/r5r/" ②
  xmlns:cac="https://semic.org/sa/cv/common/cac-2.0.0#"
  xmlns:cbc="https://semic.org/sa/cv/common/cbc-2.0.0#"
  xmlns:cts=
"urn:un:unece:uncefact:data:specification:CoreComponentTypeSchemaModule:2"
  xmlns:an="https://semic.org/sa/cv/common/semic-annot-2.0.0#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:foaf="http://xmlns.com/foaf/0.1/" targetNamespace=
"https://semic.org/sa/cv/cccev-2.0.0#" ②
  elementFormDefault="qualified"
  attributeFormDefault="unqualified"
  version="2.0.0"> ②
  <xsd:import namespace="https://semic.org/sa/cv/common/cac-2.0.0#"
    schemaLocation="../../../../common/2.0.0/xml/xsd/CV-
CommonAggregateComponents.xsd"/> ③
  <xsd:import namespace="https://semic.org/sa/cv/common/cbc-2.0.0#"
    schemaLocation="../../../../common/2.0.0/xml/xsd/CV-
CommonBasicComponents.xsd"/>
  <xsd:import namespace=
"urn:un:unece:uncefact:data:specification:CoreComponentTypeSchemaModule:2"
    schemaLocation="../../../../common/2.0.0/xml/xsd/CCTS_CCT_SchemaModule-
2.3.xsd"/>
  <xsd:import namespace="http://data.europa.eu/r5r/"
    schemaLocation="../../../../dcat-ap/2.0.0/xml/dcat-ap.xsd"/>
  <xsd:import namespace="http://www.w3.org/2002/07/owl#"
    schemaLocation="../../../../owl/2.0.0/xml/owl.xsd"/> ③

```

```

<xsd:import namespace="http://xmlns.com/foaf/0.1/"
  schemaLocation="../../../foaf/xml/foaf.xsd"/><br>
<!-- Declaration of elements --><br>
<!-- Data properties declarations -->
<xsd:element name="associatedRoleType" type="cbc:CodeType"/>
<!-- Object properties -->
<xsd:element name="concept" type="ConceptType"/>
<!-- Class declarations -->
<xsd:element name="Concept" type="ConceptType"/>
<!-- Element definition -->
<xsd:complexType name="ConceptType"> ④
  <xsd:complexContent>
    <xsd:extension base="owl:ThingType"> ③ ⑤
      <xsd:sequence> ④
        <xsd:element ref="cbc:id" minOccurs="0" maxOccurs="unbounded">
</xsd:element>
        <xsd:element ref="cbc:name" minOccurs="0" maxOccurs="1"> </xsd:element>
        <xsd:element ref="cbc:description" minOccurs="0" maxOccurs="unbounded">
</xsd:element>
        <xsd:element ref="cbc:type" minOccurs="0" maxOccurs="1"> </xsd:element>
        <xsd:element ref="cbc:qName" minOccurs="0" maxOccurs="unbounded">
</xsd:element>
        <xsd:element ref="constraint" minOccurs="0" maxOccurs="unbounded">
</xsd:element>
        <xsd:element ref="value" minOccurs="0" maxOccurs="unbounded">
        <!--we have two predicates that are the same with different domain/range--
>
        </xsd:element>
        <xsd:element ref="concept" minOccurs="0" maxOccurs="unbounded">
</xsd:element>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
</xsd:schema>

```

- ① Guideline 99
- ② Guideline 100
- ③ Guideline 101
- ④ Guideline 102
- ⑤ Guideline 103

Guideline 104: Use Core Component Types for CBC definition

Summary

Known standard business languages use the ISO 15000 Core Component Types (CCT) Specification: open standards such as UN/CEFACT, OASIS UBL, HR-Open Standards, other not-so-open specifications used in the private sector. This facilitates the mapping and conversion of data among these standards and the SEMIC eGovernment Core Vocabularies. Therefore, CCT for the definition of cross-vocabulary reusable Common Basic Components (CBC) SHOULD be used

Description

This style guide proposes the use of the CCT with two main goals:

- **SEMIC eGovernment Core Vocabularies library of Common Basic Components:** The CCT SHOULD be used as a basis for its extension with particular EU attributes (e.g. eIDAS Assurance Level attributes). The goal is to build a library of commonly used ‘basic’ components that are regularly needed in any other vocabulary. See examples below.
Such a ‘library’ of common basic components was started when developing the CCCEV, and is available ^[27] for its consideration and reuse by any other vocabulary.
- **Vocabulary-specific CBC:** This ‘CBC library’ SHOULD NOT include basic components that are particular of a Core Vocabulary. These basic components SHOULD be defined in the XSD of the Core Vocabulary. The fact that the other Core Vocabulary basic component start to be reused by several other vocabularies would indicate that the component does not contribute to define the true nature of the concepts of the Core Vocabulary where it was originally defined (and would need to be moved into the common CBC library).

Example

Example I: SEMIC CBC:

```
cbc:alias, cbc:identifier, cbc:code, cbc:amount, cbc:quantity, cbc:measure,  
cbc:name, cbc:description, etc.
```

Example II: Vocabulary-specific CBC:

```
cccev:bias or cccev:weight (data property of cccev:CriterionType),  
cccev:confidentialityLevel (data property of the cccev:EvidenceType), etc.
```

Aggregated example

```
<?xml version="1.0" encoding="UTF-8"?>①  
<!-- This is an example XSD -->①  
<xsd:schema xmlns="https://semic.org/sa/cv/cccev-2.0.0#" ②  
  xmlns:dcat-ap="http://data.europa.eu/r5r/" ②
```



```

xmlns:cac="https://semic.org/sa/cv/common/cac-2.0.0#"
xmlns:cbc="https://semic.org/sa/cv/common/cbc-2.0.0#"
xmlns:ccts=
"urn:un:unece:uncefact:data:specification:CoreComponentTypeSchemaModule:2"
xmlns:an="https://semic.org/sa/cv/common/semic-annot-2.0.0#"
xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:foaf="http://xmlns.com/foaf/0.1/" targetNamespace=
"https://semic.org/sa/cv/cccev-2.0.0#" ②
  elementFormDefault="qualified"
  attributeFormDefault="unqualified"
  version="2.0.0"> ②
    <xsd:import namespace="https://semic.org/sa/cv/common/cac-2.0.0#"
      schemaLocation="../../../../common/2.0.0/xml/xsd/CV-
CommonAggregateComponents.xsd"/> ③
    <xsd:import namespace="https://semic.org/sa/cv/common/cbc-2.0.0#"
      schemaLocation="../../../../common/2.0.0/xml/xsd/CV-
CommonBasicComponents.xsd"/> ⑥
    <xsd:import namespace=
"urn:un:unece:uncefact:data:specification:CoreComponentTypeSchemaModule:2"
      schemaLocation="../../../../common/2.0.0/xml/xsd/CCTS_CCT_SchemaModule-
2.3.xsd"/>
    <xsd:import namespace="http://data.europa.eu/r5r/"
      schemaLocation="../../../../dcap-ap/2.0.0/xml/dcap-ap.xsd"/>
    <xsd:import namespace="http://www.w3.org/2002/07/owl#"
      schemaLocation="../../../../owl/2.0.0/xml/owl.xsd"/> ③
    <xsd:import namespace="http://xmlns.com/foaf/0.1/"
      schemaLocation="../../../../foaf/xml/foaf.xsd"/><br>
    <!-- Declaration of elements --><br>
    <!-- Data properties declarations -->
    <xsd:element name="associatedRoleType" type="cbc:CodeType"/> <--6--><br>
    <!-- Object properties -->
    <xsd:element name="concept" type="ConceptType"/>
    <!-- Class declarations -->
    <xsd:element name="Concept" type="ConceptType"/>
    <!-- Element definition -->
    <xsd:complexType name="ConceptType"> ④
      <xsd:complexContent>
        <xsd:extension base="owl:ThingType"> ③ ⑤
          <xsd:sequence> <-- 4 -->
            <xsd:element ref="cbc:id" minOccurs="0" maxOccurs="unbounded">
</xsd:element>
            <xsd:element ref="cbc:name" minOccurs="0" maxOccurs="1"> </xsd:element>
            <xsd:element ref="cbc:description" minOccurs="0" maxOccurs="unbounded">
</xsd:element>
            <xsd:element ref="cbc:type" minOccurs="0" maxOccurs="1"> </xsd:element>
            <xsd:element ref="cbc:qName" minOccurs="0" maxOccurs="unbounded">
</xsd:element>
            <xsd:element ref="constraint" minOccurs="0" maxOccurs="unbounded">
</xsd:element>
            <xsd:element ref="value" minOccurs="0" maxOccurs="unbounded">

```

```
<!--we have two predicates that are the same with different domain/range-->
</xsd:element>
<xsd:element ref="concept" minOccurs="0" maxOccurs="unbounded">
</xsd:element>
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
</xsd:schema>
```

- ① Guideline 99
- ② Guideline 100
- ③ Guideline 101
- ④ Guideline 102
- ⑤ Guideline 103
- ⑥ Guideline 105

Guideline 105: Use CCT to identify external resource locations

Summary

Note that:

- Data type and data validation coherence: dissimilar data types require different design, implementation and validation approaches
- Respect of the original specification: If the XSD version of the vocabulary does not exist, the argument that by using the SEMIC CBC the original definition is not respected does not stand soundly, provided that the primitive data type and/or the specification of the identifier and location of the resource are functionally equivalent.

Therefore, The CCT attributes to identify the resource location SHOULD be used when mapping external vocabularies.

Description

One of the problems of reusing vocabularies defined in external namespaces is that they use naming, design, and implementation rules different to the ones established in this SEMIC style guide. This leads inevitably to situations that require additional technical and governance decisions. Consider, for an example, the reuse of DCAT-AP within the CCCEV and the situations hereby described:

- **Choice of the appropriate primitive data types:** vocabularies modelled based on RDF and Linked Open Data principles approach the use of primitive data differently to what is usual in XML designs. Thus, for example, in RDF identifiers are usually modelled as *xsd:string* or *xsd:normalizedString*. In RDF, controlled vocabularies use SKOS. The RDF world also leaves the possibility open to not indicate the type of resource a class is associated to (for which the *rdfs:resource* element is used, thus pointing to any type of data presumably available at a URL from which both data and metadata may be obtained).

Inspired by the development of the CCCEV ways of solving this situation when the external vocabulary being has not been implemented yet as an XSD Schema, are being presented in the following:

Respecting the original definition and mapping the primitive data types SHOULD be in place. A work-around of the definition of ‘nearly-XML primitive’ data types SHOULD be in place. For example, use *xsd:anyURI* or *xsd:anyType* as possible equivalent data types for the *rdfs:resource*.

A mapping of each primitive (or nearly-primitive) data type to one of the proposed SEMIC Common Basic Component (CBC) is recommended.

The third option are the recommendations found within this style guide document.

- **Development of vocabulary-specific supplementary software:** The RDF paradigm allows to model more “open worlds” than the XML designs, which are more focused on the data types, formats and constraints. This means that when converting an RDF-based vocabulary into an XSD artefact, specific data types should be selected. This is one of the reasons why the use of *xsd:anyType* to map the *rdfs:resource* SHOULD NOT be considered.

Example

The CCCEV development provides a SEMIC CBC-based implementation of the DCAT-AP XSD vocabulary (and of every vocabulary it imports). See the file ‘dcat-ap_semic-cbc.xsd’ under the branch v2.0.0 of the CCCEV GitHub: <https://github.com/SEMICeu/CCCEV/tree/CV-2.0.0/dcat-ap/2.0.0/xml>.

Guideline 106: Make use of Common Aggregate Components library

Summary

Highly reusable classes SHOULD be defined in a separated Common Aggregate Components (CAC) library, as this ensures the cross-vocabulary and cross-domain semantic interoperability and complies with sharing and reuse principle.

Description

This is the case, for example of the classes Period or Address, which are frequently needed in many different situations. However, when reusing these classes, attention needs to be given in the following:

- The prefix ‘cac’ (standing for Common Aggregate Components’) CAN be used in order to identify the classes defined in the CAC SEMIC XSD file. This facilitates the identification and use of the imported classes and properties
- Definition of properties in the ‘cac:’ namespace that are particular to Core Vocabulary or Application Profile SHOULD NOT take place. For example, a property named ‘certificateValidityPeriod’ SHOULD NOT be defined in the namespace represented with the ‘cac:’ prefix.

Example

```
<xsd:element name="DatumValue" type="cac:ValueType"/>
<xsd:element name="PeriodValue" type="cac:PeriodType"/>
```

Guideline 107: Represent the central concept as abstract class

Summary

The concept of 'abstract' class CAN be applied for the representation of the central concept of any Vocabulary, as it ensures the following:

- Usage of the extensibility principle
- Ability to be cross-syntax compatible
- Enablement of cross-domain interoperability
- Flexibility in design and implementation.

Description

This recommendation is tightly related to the concept of 'core'. Being 'core' implies that the vocabulary is domain-agnostic, and therefore the central concept being defined therein is abstract enough so it can be further on specialised for domain-specific needs. It also implies that the 'abstract' class is the one that carry the majority of the properties that is truly 'core'. The design and instantiation of 'abstract' classes poses the challenges that are described below, along with recommendations on how to address them:

- Classes in XML can be declared abstract, however they SHOULD NOT be declared abstract in an eGovernment Core Vocabulary. Declaring classes as 'abstract' would impose a restriction that could be inconvenient at XML instance production time.
- How to specify that a requirement inside a requirement is of a specific subclass type. The solution to this is to specify it at XML instantiation-time, using the *xsi:type* attribute of any XML object ('xsi' stands for XML Schema Instance). See the example below (taken from the CCCEV XSD). This case is encompassed under the notion of 'polymorphism' because one class declaration is used for multiple different sub-class instantiations ^[28].

Example

Example I: Subclassing of 'pseudo-abstract' classes

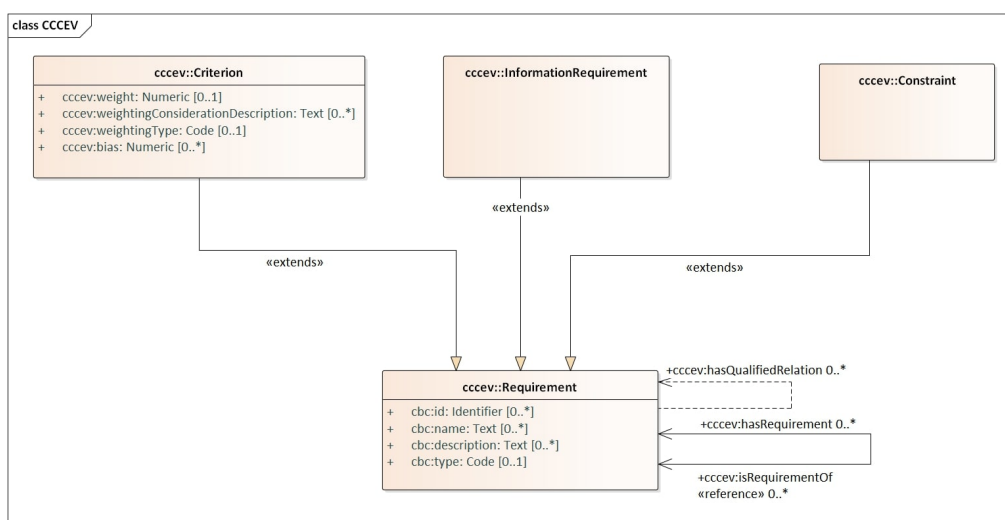


Figure 48. CCCEV::Requirement as 'pseudo-abstract'

Example II: Use of xsi:type

```
Requirement xsi:type="cccev:CriterionType" xmlns:espd-  
voc="http://semic.eu/sa/espd/voc#" xmlns:espd-pi="http://semic.eu/sa/espd/pi">  
  <cbc:id schemeID="ESPD" schemeVersionID="2.1.1">005eb9ed-1347-4ca3-bb29-  
9bc0db64e1ab</cbc:id>  
  <cbc:name languageID="en">Participation in a criminal organisation</cbc:name>  
  <cbc:description languageID="eng">Has the economic operator  
...</cbc:description>  
  <cbc:type AssuranceLevelValue="High" listID="CriteriaTypeCode" listAgencyID=  
"EU-COM-GROW"  
listVersionID="2.1.1"  
>CRITERION.EXCLUSION.CONVICTIONS.PARTICIPATION_IN_CRIMINAL_ORGANISATION</cbc:type>  
  <cccev:requirement xsi:type="cccev:InformationRequirementType">  
    <cbc:id>7c637c0c-7703-4389-ba52-02997a055bd7</cbc:id>
```

Guideline 108: Keep file names brief and concise

Summary

File names SHOULD be brief, self-contained, unambiguously and uniquely identifiable, hence when naming XSD files, abbreviations, hyphens and identifiers CAN be used.

Description

The name of the file used to define the XSD Schema CAN be composed of:

- The abbreviated name of the vocabulary, e.g. 'cccev'
- A hyphen, '-'
- The version identifier, e.g. 'v2.0.0'
- The extension of the W3C file, i.e. '.xsd'

Example

cccev-v2.0.0.xsd

Guideline 109: Use lowerCamelBack for attributes and predicates

Summary

UML attributes and predicates SHALL use the lowerCamelBack syntax and avoid appending the word Type at the end (except for a code), as this ensures:

- The facilitation of the distinction between properties and classes
- That data properties are easily differentiated from the object properties based on the prefix and the naming rule (e.g. codes and identifiers)
- Coherence with the UML and RDF and JSON serialisations, while trying to preserve the XML user/developer experience

Description

The basic rules for the naming of the UML class attributes and predicates, are described in the following:

- The name of any property SHALL follow the lowerCamelBack syntax
- The name of a property SHALL NOT append the word Type at the end, except if it is a code
- Any data property CAN start with verb (but it is not recommended). This naming is classic in XSD design, contrary to the naming of property in RDF. Codes defining the nature of a class SHOULD reuse the component *cbc:Type*. Otherwise the property name SHOULD be qualified with a prepended noun or adjective. If the appendix 'Type', added to a qualified property pointing to a code, causes confusion or ambiguity, then the word 'Classification' to the qualifier SHOULD be appended (as in *cccev:evidenceTypeClassification*, instead of *cccev:EvidenceTypeType*, for example).

Example

```
<!-- Data properties declarations -->
<xsd:element name="associatedRoleType" type="cbc:CodeType"/><br>
<!-- Object properties -->
<xsd:element name="concept" type="ConceptType"/>
```

Aggregated example

```
<?xml version="1.0" encoding="UTF-8"?>①
<!-- This is an example XSD -->①
<xsd:schema xmlns="https://semic.org/sa/cv/cccev-2.0.0#" ②
  xmlns:dcap="http://data.europa.eu/r5r/" ②
  xmlns:cac="https://semic.org/sa/cv/common/cac-2.0.0#"
  xmlns:cbc="https://semic.org/sa/cv/common/cbc-2.0.0#"
  xmlns:ccts=
"urn:un:unece:uncefact:data:specification:CoreComponentTypeSchemaModule:2"
  xmlns:an="https://semic.org/sa/cv/common/semic-annot-2.0.0#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:foaf="http://xmlns.com/foaf/0.1/" targetNamespace=
```

```

"https://semic.org/sa/cv/cccev-2.0.0#" ②
  elementFormDefault="qualified"
  attributeFormDefault="unqualified"
  version="2.0.0"> ②
  <xsd:import namespace="https://semic.org/sa/cv/common/cac-2.0.0#"
    schemaLocation="../../../../common/2.0.0/xml/xsd/CV-
CommonAggregateComponents.xsd"/> ③
  <xsd:import namespace="https://semic.org/sa/cv/common/cbc-2.0.0#"
    schemaLocation="../../../../common/2.0.0/xml/xsd/CV-
CommonBasicComponents.xsd"/>
  <xsd:import namespace=
"urn:un:unece:uncefact:data:specification:CoreComponentTypeSchemaModule:2"
    schemaLocation="../../../../common/2.0.0/xml/xsd/CCTS_CCT_SchemaModule-
2.3.xsd"/>
  <xsd:import namespace="http://data.europa.eu/r5r/"
    schemaLocation="../../../../dcap/2.0.0/xml/dcap-ap.xsd"/>
  <xsd:import namespace="http://www.w3.org/2002/07/owl#"
    schemaLocation="../../../../owl/2.0.0/xml/owl.xsd"/> <-- 3 -->
  <xsd:import namespace="http://xmlns.com/foaf/0.1/"
    schemaLocation="../../../../foaf/xml/foaf.xsd"/><br>
  <!-- Declaration of elements --><br>
  <!-- Data properties declarations -->
  <xsd:element name="associatedRoleType" type="cbc:CodeType"/>⑥ ⑦
  <!-- Object properties -->
  <xsd:element name="concept" type="ConceptType"/> ⑦
  <!-- Class declarations -->
  <xsd:element name="Concept" type="ConceptType"/>
  <!-- Element definition -->
  <xsd:complexType name="ConceptType"> ④
    <xsd:complexContent>
      <xsd:extension base="owl:ThingType"> ③ ⑤
        <xsd:sequence> ④
          <xsd:element ref="cbc:id" minOccurs="0" maxOccurs="unbounded">
</xsd:element>
          <xsd:element ref="cbc:name" minOccurs="0" maxOccurs="1"> </xsd:element>
          <xsd:element ref="cbc:description" minOccurs="0" maxOccurs="unbounded">
</xsd:element>
          <xsd:element ref="cbc:type" minOccurs="0" maxOccurs="1"> </xsd:element>
          <xsd:element ref="cbc:qName" minOccurs="0" maxOccurs="unbounded">
</xsd:element>
          <xsd:element ref="constraint" minOccurs="0" maxOccurs="unbounded">
</xsd:element>
          <xsd:element ref="value" minOccurs="0" maxOccurs="unbounded">
            <!--we have two predicates that are the same with different domain/range--
>
          </xsd:element>
          <xsd:element ref="concept" minOccurs="0" maxOccurs="unbounded">
</xsd:element>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>

```



```
</xsd:complexType>  
</xsd:schema>
```

- ① Guideline 99
- ② Guideline 100
- ③ Guideline 101
- ④ Guideline 102
- ⑤ Guideline 103
- ⑥ Guideline 105
- ⑦ Guideline 109

Guideline 110: Use UpperCamelBack plus type for classes

Summary

UML classes SHALL use UpperCamelBack syntax and the word “Type” at the end of the term. This ensures:

- Facilitation of the distinction between properties and classes
- Coherence with the UML and RDF and JSON serialisations, while trying to preserve the XML user/developer experience as much as possible.

Description

The rules for the naming of the UML classes when defined as XML simple or complex element types, are described in the following :

- The name of any class defined in the vocabulary (simple or complex element) SHALL follow the UpperCamelBack syntax
- All classes SHALL append the word “Type” at the end of the term used in its definition (e.g. ‘CriterionType’, ‘InformationRequirementType’, ‘ConstraintType’, etc.)
- No spaces or non-ASCII codes are allowed in the name of a Class, either in the UML representation nor in the XSD implementation. Only ASCII letters, numbers, SHOULD be used
- The use of the characters ‘-’ and ‘_’ SHOULD NOT be used. An exception to this rule CAN be the use of middle-hyphen when the hyphen is currently used in the language used to term the class (e.g. Latin words normally use in English, like Ex-AnteDeclaration).

Example

```
<!-- Class declarations -->
<xsd:element name="Concept" type="ConceptType"/>
```

Aggregated example

```
<?xml version="1.0" encoding="UTF-8"?>①
<!-- This is an example XSD -->①
<xsd:schema xmlns="https://semic.org/sa/cv/cccev-2.0.0#" ②
  xmlns:dcap="http://data.europa.eu/r5r/" ②
  xmlns:cac="https://semic.org/sa/cv/common/cac-2.0.0#"
  xmlns:cbc="https://semic.org/sa/cv/common/cbc-2.0.0#"
  xmlns:ccts=
"urn:un:unece:uncefact:data:specification:CoreComponentTypeSchemaModule:2"
  xmlns:an="https://semic.org/sa/cv/common/semic-annot-2.0.0#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:foaf="http://xmlns.com/foaf/0.1/" targetNamespace=
"https://semic.org/sa/cv/cccev-2.0.0#" ②
  elementFormDefault="qualified"
  attributeFormDefault="unqualified"
  version="2.0.0"> ②
```

```

<xsd:import namespace="https://semic.org/sa/cv/common/cac-2.0.0#"
  schemaLocation="../../../common/2.0.0/xml/xsd/CV-
CommonAggregateComponents.xsd"/> ③
<xsd:import namespace="https://semic.org/sa/cv/common/cbc-2.0.0#"
  schemaLocation="../../../common/2.0.0/xml/xsd/CV-
CommonBasicComponents.xsd"/>
<xsd:import namespace=
"urn:un:unece:uncefact:data:specification:CoreComponentTypeSchemaModule:2"
  schemaLocation="../../../common/2.0.0/xml/xsd/CCTS_CCT_SchemaModule-
2.3.xsd"/>
<xsd:import namespace="http://data.europa.eu/r5r/"
  schemaLocation="../../../dcat-ap/2.0.0/xml/dcat-ap.xsd"/>
<xsd:import namespace="http://www.w3.org/2002/07/owl#"
  schemaLocation="../../../owl/2.0.0/xml/owl.xsd"/> ③
<xsd:import namespace="http://xmlns.com/foaf/0.1/"
  schemaLocation="../../../foaf/xml/foaf.xsd"/><br>
<!-- Declaration of elements --><br>
<!-- Data properties declarations -->
<xsd:element name="associatedRoleType" type="cbc:CodeType"/>⑥ ⑦
<!-- Object properties -->
<xsd:element name="concept" type="ConceptType"/> ⑦
<!-- Class declarations -->
<xsd:element name="Concept" type="ConceptType"/> ⑧
<!-- Element definition -->
<xsd:complexType name="ConceptType"> ④
  <xsd:complexContent>
    <xsd:extension base="owl:ThingType"> ③ ⑤
      <xsd:sequence> ④
        <xsd:element ref="cbc:id" minOccurs="0" maxOccurs="unbounded">
</xsd:element>
        <xsd:element ref="cbc:name" minOccurs="0" maxOccurs="1"> </xsd:element>
        <xsd:element ref="cbc:description" minOccurs="0" maxOccurs="unbounded">
</xsd:element>
        <xsd:element ref="cbc:type" minOccurs="0" maxOccurs="1"> </xsd:element>
        <xsd:element ref="cbc:qName" minOccurs="0" maxOccurs="unbounded">
</xsd:element>
        <xsd:element ref="constraint" minOccurs="0" maxOccurs="unbounded">
</xsd:element>
        <xsd:element ref="value" minOccurs="0" maxOccurs="unbounded">
        <!--we have two predicates that are the same with different domain/range--
>
        </xsd:element>
        <xsd:element ref="concept" minOccurs="0" maxOccurs="unbounded">
</xsd:element>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
</xsd:schema>

```

② Guideline 100

③ Guideline 101

④ Guideline 102

⑤ Guideline 103

⑥ Guideline 105

⑦ Guideline 109

⑧ Guideline 110

Guideline 111: Provide the XSD manifestation

Summary

Any eGovernment Core Vocabulary XSD Schema SHOULD provide a manifestation of the XSD (i.e. a copy of the XSD file) where each property and class SHOULD be documented. This ensures:

- Human readability and understanding of the semantics and usage of the elements
- Registration and population of properties and classes in registry/repository services for the automated discoverability, sharing and reuse (e.g OASIS RegRep-compliant registries of components)
- Automated processing of the elements based on the annotations, e.g. for the production of supporting human-readable documentation

Note that the practice of providing a manifestation is based on well-established standards, like UN/CEFACT and OASIS UBL, which provide two different sets of XSD files: a run-time copy and a fully documented version ^[29].

Description

This SEMIC style guide recommends that the use of the *xsd:annotation* element jointly with the CCCTS components defined in the UN/CEFACT CCT-CCTS 1.1 specification ^[30], SHOULD be followed as much as possible.

The development of the CCCEV showed that the SEMIC vocabularies implementing this SEMIC style guide SHOULD reuse a set of additional elements for the documentation of the elements (both properties and classes) of an eGovernment Core Vocabulary ^[31].

Example

```
<xsd:annotation>
  <xsd:documentation>
    <an:Class>
      <ccts:ComponentType>Class</ccts:ComponentType>
      <ccts:DictionaryEntryName>cccev:Concept. Details</ccts:DictionaryEntryName>
      <ccts:Definition>
        A reference to an entity, i.e. a class or a property, which is defined in
        a known ontology or vocabulary.
        Additional information:
        A concept defined in the context of a Requirement must be unambiguous so
        there is no possibility of misinterpretation. Hence the use of qualified names (a
        pair {namespace, local name} is strongly recommended for the identification of the
        entity being referred to.
      </ccts:Definition>
      <ccts:ObjectClass>cccev:Concept</ccts:ObjectClass>
      <PrimitiveClass>owl:ThingType</PrimitiveClass>
    </an:Class>
  </xsd:documentation>
</xsd:annotation>
```

Aggregated example

```
<?xml version="1.0" encoding="UTF-8"?>①
<!-- This is an example XSD -->①
<xsd:schema xmlns="https://semic.org/sa/cv/cccev-2.0.0#" ②
  xmlns:dcap="http://data.europa.eu/r5r/" ②
  xmlns:cac="https://semic.org/sa/cv/common/cac-2.0.0#"
  xmlns:cbc="https://semic.org/sa/cv/common/cbc-2.0.0#"
  xmlns:ccts=
"urn:un:unece:uncefact:data:specification:CoreComponentTypeSchemaModule:2"
  xmlns:an="https://semic.org/sa/cv/common/semic-annot-2.0.0#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:foaf="http://xmlns.com/foaf/0.1/" targetNamespace=
"https://semic.org/sa/cv/cccev-2.0.0#" ②
  elementFormDefault="qualified"
  attributeFormDefault="unqualified"
  version="2.0.0"> ②
  <xsd:import namespace="https://semic.org/sa/cv/common/cac-2.0.0#"
    schemaLocation="../../../common/2.0.0/xml/xsd/CV-
CommonAggregateComponents.xsd"/> ③
  <xsd:import namespace="https://semic.org/sa/cv/common/cbc-2.0.0#"
    schemaLocation="../../../common/2.0.0/xml/xsd/CV-
CommonBasicComponents.xsd"/>
  <xsd:import namespace=
"urn:un:unece:uncefact:data:specification:CoreComponentTypeSchemaModule:2"
    schemaLocation="../../../common/2.0.0/xml/xsd/CCTS_CCT_SchemaModule-
2.3.xsd"/>
  <xsd:import namespace="http://data.europa.eu/r5r/"
    schemaLocation="../../../dcap/2.0.0/xml/dcap.xsd"/>
  <xsd:import namespace="http://www.w3.org/2002/07/owl#"
    schemaLocation="../../../owl/2.0.0/xml/owl.xsd"/> ③
  <xsd:import namespace="http://xmlns.com/foaf/0.1/"
    schemaLocation="../../../foaf/xml/foaf.xsd"/><br>
  <!-- Declaration of elements --><br>
  <!-- Data properties declarations -->
  <xsd:element name="associatedRoleType" type="cbc:CodeType"/>⑥ ⑦
  <!-- Object properties -->
  <xsd:element name="concept" type="ConceptType"/> ⑦
  <!-- Class declarations -->
  <xsd:element name="Concept" type="ConceptType"/> ⑧
  <!-- Element definition -->
  <xsd:complexType name="ConceptType"> ④
    <xsd:complexContent>
      <xsd:extension base="owl:ThingType"> ③ ⑤
        <xsd:annotation> ⑨
          <xsd:documentation>
            <an:Class>
              <ccts:ComponentType>Class</ccts:ComponentType>
              <ccts:DictionaryEntryName>cccev:Concept. Details</ccts:DictionaryEntryName>
              <ccts:Definition>
```

A reference to an entity, i.e. a class or a property, which is defined in a known ontology or vocabulary.
Additional information:
...strongly recommended for the identification of the entity being referred to.

```
</ccts:Definition>
<ccts:ObjectClass>cccev:Concept</ccts:ObjectClass>
<PrimitiveClass>owl:ThingType</PrimitiveClass>
</an:Class>
</xsd:documentation>
</xsd:annotation>
<xsd:sequence> ④
  <xsd:element ref="cbc:id" minOccurs="0" maxOccurs="unbounded">
</xsd:element>
  <xsd:element ref="cbc:name" minOccurs="0" maxOccurs="1"> </xsd:element>
  <xsd:element ref="cbc:description" minOccurs="0" maxOccurs="unbounded">
</xsd:element>
  <xsd:element ref="cbc:type" minOccurs="0" maxOccurs="1"> </xsd:element>
  <xsd:element ref="cbc:qName" minOccurs="0" maxOccurs="unbounded">
</xsd:element>
  <xsd:element ref="constraint" minOccurs="0" maxOccurs="unbounded">
</xsd:element>
  <xsd:element ref="value" minOccurs="0" maxOccurs="unbounded">
    <!--we have two predicates that are the same with different domain/range--
>
  </xsd:element>
  <xsd:element ref="concept" minOccurs="0" maxOccurs="unbounded">
</xsd:element>
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
</xsd:schema>
```

- ① Guideline 99
- ② Guideline 100
- ③ Guideline 101
- ④ Guideline 102
- ⑤ Guideline 103
- ⑥ Guideline 105
- ⑦ Guideline 109
- ⑧ Guideline 110
- ⑨ Guideline 111

7.2. Application Profiles

The focus of this section is to provide guidelines for XML Schemas at the level of Application Profile. There are two possible ways to develop an XSD file for an application profile.

1. By generating it automatically from the UML Class diagram model using the appropriate tooling.
2. By using the XSD 1.1 Override feature. It allows the definition of a new version of a schema component, with the same name, that replaces the original definitions in the schema.

The following guidelines describes how restrictions in the Application profile should be formulated.

Guideline 112: Externalise rules

Summary

Externalised rules are easier (and cheaper) to maintain than an XSD Schema. Modifying the XSD Schema may have deeper impacts on the existing software applications than the addition or modification of rules maintained externally to the XSD Schema, hence application profiles SHOULD externalise (i.e. separate from the eGovernment Core Vocabulary) the maximum possible the definition and validation of the restrictions.

Description

There are at least two ways of applying the guideline to the Application Profiles:

- XSD Schema profiling: a new XSD Schema CAN be produced where the restrictions are applied, e.g. removal of elements that are not required by the application, restriction of the cardinalities, etc.
- Semantic and technical validation: this SEMIC style guide recommends the use of ISO Schematron for the definition of validation rules for the control of cardinalities, format patterns, co-occurrence of elements (e.g. disjoints or union of elements), and other validation aspects that are not desired in the XSD Schema.

The preferred option SHOULD be the second one (use of Schematron): the maintenance effort is lesser and can be applied to the original eGovernment Core Vocabulary without having to tamper with the conceptual model nor the XSD design.

Example

The following example shows that this application profile schema overrides the Core Criterion and Core Evidence Vocabulary and 1) requests always a weight for any given criterion, 2) does not use some properties defined in the original XSD Schema, and 3) requests always the specification of the type of weighing used.

The important aspect of the profile is that it does not contradict the original XSD definition. This implies that such a profile is also compatible with any other implementation that is conformant to the original Core Vocabulary.

```
<xsd:complexType name="CriterionType">
  <xsd:complexContent>
    <xsd:extension base="RequirementType">
      <xsd:sequence>
        <xsd:element ref="weight"/>
        <xsd:element ref="weightingType"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

Guideline 113: Externalise restrictions

Summary

Externalised rules are easier (and cheaper) to maintain than an XSD Schema. Modifying the XSD Schema may have deeper impacts on the existing software applications than the addition or modification of rules maintained externally to the XSD Schema, hence the design of the Vocabulary SHOULD define the lesser restrictions possible and keep flexible. Restrictions SHOULD be defined and validated by means external to the XSD Schema.

Description

The design of the Vocabulary SHOULD define the lesser restrictions possible and keep flexible. Restrictions SHOULD be defined and validated by means external to the XSD Schema (see guideline on Application Profiles and Validation, below).

The main reason for this is that restrictions are to be imposed by specific domains and contexts of use. Otherwise the reusability and customisation of the ‘core’ vocabulary could be threatened.

Regarding restrictions, this style guide recommends the following principles when designing an eGovernment Core Vocabulary:

- Keep the cardinality open and flexible: 0..1, 0..n SHOULD be the commonly used cardinalities. Clear exceptions CAN be done, e.g. the CCCEV states that an *EvidenceType* cannot exist if it does not referred to in an *EvidenceTypeList* (this cardinality normally defines a kind of ‘composition’ relationship: the life-cycle of an *EvidenceTypeType* object depends on the life-cycle of the list(s) to which it belongs: when all lists are destroyed the types of evidence objects should be also destroyed)
- Cardinality of identifiers: The cardinality of identifiers SHOULD be always 0..n. This allows for setting maps of identifiers defined by multiple agents or agencies, and/or using diverse identification schemes and assurance level values, methods, etc.. Restricting this cardinality to 1 or 0..1 SHOULD be the mission of an Application Profile
- Cardinality of codes: Codes are used to classify concepts in different categories. A category is a synonym of ‘class’ in the sense that they define groups of concepts that share similar natures and for which there is no need of specifying the attributes of the concept. Two different situations need to be differentiated in relation to codes:
 - When a code contributes to perfect the semantics of the class (i.e. to better explain the nature of the class), the cardinality SHOULD BE 0..1. In this case, there is no need of ‘qualifying’ the name assigned to the property pointing to the code. The property used for this purpose can be simply `cbc:type`, since the class provides the context to understand that the type refers to the nature of the object of instantiated. The explanation for this cardinality is simple: an object cannot be of two or more different natures simultaneously.
 - There is an exception, though, to this rule: when the nature of an object is described by means of a taxonomy. In this case there may be the need of specifying the codes of the ‘parent’, ‘children branches’ and the ‘leaf’.
- When a code contributes to explain the nature of a property, the cardinality MAY be multiple, and the name of the property pointing to a concept defined in a controlled vocabulary SHOULD be ‘qualified’, i.e. a prefix SHOULD be pre-pended to the work

Type, e.g. *cccev:datasetType* in the class *cccev:EvidenceType*

- Cardinality of text-based properties: the cardinality of names, description and other text data types should be multiple. This allows for the expression of the text in multiple different languages. The *cbc:Text* component provides attributes for this specific purpose
- Reference to instance of classes: IdRef, Key and KeyRef SHOULD be avoided. The UML diagram, though, helps to see where references to object via unique identifiers could be necessary at XML instantiation time (stereotypes << reference >> in the UML diagram). These restrictions should be left to Application Profiles and custom vocabularies
- Choices: The XML built-in choice structure allows only one of the elements contained in the structure to be present within the containing element (it declares that all the elements within the structure are ‘disjoint’ amongst themselves, they cannot ‘co-occur’) This style guide does not recommend the use of choices in the definition of an eGovernment Core Vocabulary, since the Core Vocabulary can be used for unknown purposes. Co-occurrence and disjointness SHOULD be defined and controlled externally, e.g. via rules specified in ISO Schematron schemata, in rules engines or *ad hoc* software developments.

Example

See the CCCEV UML diagrams and XSD file for concrete examples: <https://github.com/SEMICEu/CCCEV/tree/CV-2.0.0/cccev>.

See also the XML example https://github.com/SEMICEu/CCCEV/blob/CV-2.0.0/use_cases/testing/Requirement-Polymorphism-example-KeyKeyRef.xml, provided to illustrate how to implement Key/KeyRef restrictions to control the assignment of identifiers to particular XML object instances. The example illustrates this case for ‘elements’ *cbc:id*, since this style guide does not recommend the use of element attributes ‘id’ (instance identifiers are not to be considered ‘metadata’ about the element, they may be ‘business domain-specific’ identifiers).

7.3. XSD Annotations

Any Vocabulary intended to be read by humans should offer two different sets of XSD schemata:

- **Runtime:** meant to facilitate implementing the schemata.
- **Annotated:** meant to provide human users with extended information.

Even though XSD schemata can be self-explanatory in most cases, Core Vocabularies should offer normative schemata providing a more comprehensive description of the elements, such as their definition, range, and multiplicity. However, this is inefficient when implementing the schemata and may burden users looking for very specific sections in the code. Hence, this established practice ^[32] allows users to conveniently choose the set they need.

For more information regarding annotations, see [CCCEV annotated schemata](#).

[25] Source consulted: https://www.unece.org/fileadmin/DAM/cefact/xml/XML-Naming-And-Design-Rules-V2_1.pdf

- [26] See folder ‘cccev’ in the CCCEV GitHub repository: <https://github.com/SEMICEu/CCCEV/tree/CV-2.0.0>.
- [27] See the file ‘CV-CommonBasicComponents.xsd’ in the the branch v2.0.0 of the CCCEV Github folder ‘common’: <https://github.com/SEMICEu/CCCEV/tree/CV-2.0.0/common/2.0.0/xml/xsd>.
- [28] See the example XML files provided for the ESPD use case for a better understanding of the use of the xsi:type: https://github.com/SEMICEu/CCCEV/tree/CV-2.0.0/use_cases/espd-cv.
- [29] See folders ‘xsd’ and ‘xsdrtr’ in the branch v2.0.0 of the CCCEV GitHub repository: <https://github.com/SEMICEu/CCCEV/tree/CV-2.0.0/cccev/2.0.0/xml>
- [30] See the UN/CEFACT CCT-CCTS specification used by the CCCEV: https://github.com/SEMICEu/CCCEV/blob/CV-2.0.0/common/2.0.0/xml/xsd/CCTS_CCT_SchemaModule-2.3.xsd
- [31] This proposal is implemented in the file <https://github.com/SEMICEu/CCCEV/blob/CV-2.0.0/common/2.0.0/xml/xsd/CCCEV-Annotation.xsd>
- [32] See: [CCCEV XML folder](#) or [UBL XSD Schemas](#)

Chapter 8. Publishing

8.1. Artefacts

In this section, the guidelines for publication of the artefacts are documented.

The types of artefacts are:

1. [Machine-readable Files](#)
2. [Human-oriented Documentation](#)

Machine-readable Files

The guidelines for publishing the machine-readable files for [eGovernment Core Vocabularies](#) and [Application Profiles](#) are indicated below.

eGovernment Core Vocabularies

Guideline 114: Make basic artefacts public

Summary

The following artefacts SHALL be made available via the web portal:

- The core RDFS file in turtle and JSON-LD formats;
- The SHACL file with the class models and simple constraints in turtle format.

Description

These artefacts are the minimum atomic parts needed to implement Vocabulary.

Example

RDFS file: `cbv_v1.0.0.ttl`, `cbv_v1.0.0.jsonld`.

SHACL file: `cbv-constraints_v1.0.0.ttl`, `cbv-constraints_v1.0.0.xsd`.

Guideline 115: Make additional artefacts public

Summary

The following artefacts CAN be made available via the web portal:

- The RDFS file with additional semantics to infer new knowledge in turtle format
- The XMI file that could be exported from the UML Class Diagram model

Description

These additional artefacts help to better understand the Vocabulary.

Example

RDFS file: cbv-semantics_v1.0.0.ttl

XMI file: cbv-constraints_v1.0.0.xmi

Application Profiles

Guideline 116: Make basic artefacts public

Summary

The following artefacts SHALL be made available via the web portal:

- The core RDFS file in turtle and JSON-LD formats;
- The SHACL file with the class models and simple constraints in turtle format.

Description

These artefacts are the minimum atomic parts needed to implement Vocabulary.

Example

RDFS file: cbv-AP_v1.0.0.ttl, cbv_v1.0.0.jsonld.

SHACL file: cbv-AP-constraints_v1.0.0.ttl, cbv-constraints_v1.0.0.xsd.

Guideline 117: Make additional artefacts public

Summary

The following artefacts CAN be made available via the web portal:

- The RDFS file with additional semantics to infer new knowledge in turtle format
- The XMI file that could be exported from the UML Class Diagram model

Description

These additional artefacts help to better understand the Vocabulary.

Example

RDFS file: cbv-AP-semantics_v1.0.0.ttl

XMI file: cbv-AP-constraints_v1.0.0.xmi

Human-oriented documentation

This subsection provides guidelines for human-oriented documentation:

Guideline 118: Publish an introductory page

Summary

Each eGovernment Core Vocabulary and application profile SHALL have an introductory page on a web portal that provides a high-level overview and offers links to

- The documentation on the standard as described in the following section;
- The UML class diagram that gives a visual representation of the model in file formats PNG or SVG
- The machine-readable files as enumerated in section [Machine-readable Files](#)

Description

This is necessary to help potential users to understand the Vocabulary.

Example

Introductory [web page](#) for the Core Business Vocabulary in ISA² portal: <https://joinup.ec.europa.eu/solution/e-government-core-vocabularies/core-business-vocabulary>

Guideline 119: Use the standard template

Summary

The specifications documentation SHALL be using the standard HTML template:

- [Template](#)
- [Example](#)

Description

Respecting the standard ensures an easier reading of the documentation and improves user experience.

Example

- [Template](#)
- [Example](#)

Guideline 120: Publish an HTML glossary

Summary

The glossary SHALL consist of an HTML document providing an overview of the classes, attributes and associations used. For each one of them, the following information SHOULD be indicated:

- Label
- Definition
- Scope or usage note (optional)

Description

This improves user experience by making the content swiftly accessible.

Example

The eProcurement Ontology [Glossary](#)

8.2. LOD Resources

Publishing a vocabulary as a Linked Open Data resource means to make it accessible, both human and machine-readable, in a standard way, thus supporting the semantic web. The publication as full LOD implies that the data is compliant with the highest grade of *5-Star* ^[33] open data evaluation:

- The data is available on the Web under an open license
- The data is available following a clear structure

- The data is available using a non-proprietary open format
- The data uses URIs to denote concepts
- The data is linked to other resources to provide context

The following subsections describe how to set up a system to ensure such a publication.

Publishing a Vocabulary/Application Profile Using Persistent URIs

When publishing LOD data, it is crucial to have a structured and persistent system of URIs in place. Such URIs make it easy for the users to understand the structure of the vocabulary and retrieve information as they expect. The guidelines for using persistent URIs while publishing a vocabulary or application profile are stated below.

Guideline 121: Use one URI per artefact

Summary

All artefacts of the model (classes, attributes, associations and others) SHALL have a URI assigned to them.

Description

N/A

Example

N/A

Guideline 122: Use the standard base URIs

Summary

The following base URIs SHOULD be used:

- For SEMIC interoperability solutions such URIs SHOULD use base URI <https://semic.eu>
- For European controlled vocabularies such URIs SHOULD be maintained by the Publications Office of the EU. Their base URI is <http://data.europa.eu/>

Description

N/A

Example

N/A

Dereferencing the Vocabulary

Dereferencing means that if one retrieves the URL of the vocabulary or the application profile, they get the related information back.

In what format the information is returned, depends on content negotiation. A web client can use in the header part of the HTTP request (the Accept field) to indicate the content type(s) that is acceptable and expected as the response.

Guideline 123: Dereferencing the Vocabulary I

Summary

In case of a core RDFS vocabulary, the vocabulary server SHOULD return:

- A human-aimed HTML representation, being the **default** response;
- The RDFS representation.

Description

N/A

Example

HTML representation

1. With ^[34], a command-line tool for getting data via URL, the corresponding command is :

```
curl "https://data.europe.eu/semanticassets/ns/cbv_v1.0.0"
```

This file can also be directly retrieved with ‘.html’ file extension (https://data.europe.eu/semanticassets/ns/cbv_v1.0.0.html).

RDFS representation

1. In turtle using MIME type text/turtle:

```
curl "https://data.europe.eu/semanticassets/ns/cbv_v1.0.0" -H 'Accept: text/turtle'
```

Also, directly retrievable with ‘.ttl’ extension (https://data.europe.eu/semanticassets/ns/cbv_v1.0.0.ttl).

2. In json-ld using MIME type application/ld+json:

```
curl "https://data.europe.eu/semanticassets/ns/cbv_v1.0.0" -H 'Accept: application/ld+json'
```

Also, directly retrievable with ‘.jsonld’ extension (https://data.europe.eu/semanticassets/ns/cbv_v1.0.0.jsonld).

Guideline 124: Dereferencing the Vocabulary II

Summary

In case of a semantics-based RDFS vocabulary, the vocabulary server SHOULD return:

- A human-aimed HTML representation, being the **default** response;
- The RDFS representation.

Description

N/A

Example

HTML representation

1. With ^[34], a command-line tool for getting data via URL, the corresponding command is :

```
curl "https://data.europe.eu/semanticassets/ns/cbv-semantics_v1.0.0"
```

This file can also be directly retrieved with ‘.html’ file extension (https://data.europe.eu/semanticassets/ns/cbv-semantics_v1.0.0.html).

RDFS representation

1. In turtle using MIME type text/turtle: ---

```
curl "https://data.europe.eu/semanticassets/ns/cbv-semantics_v1.0.0" -H 'Accept: text/turtle'
```

Also, directly retrievable with ‘.ttl’ extension (https://data.europe.eu/semanticassets/ns/cbv-semantics_v1.0.0.ttl).

2. In json-ld using MIME type application/ld+json:

```
curl "https://data.europe.eu/semanticassets/ns/cbv-semantics_v1.0.0" -H 'Accept: application/ld+json'
```

Also, directly retrievable with ‘.jsonld’ extension (https://data.europe.eu/semanticassets/ns/cbv-semantics_v1.0.0.jsonld).

Guideline 125: Dereferencing the Vocabulary III

Summary

In case of a constraints model, the vocabulary server SHOULD return:

- A human-aimed HTML representation, being the **default** response;
- The SHACL representation in turtle using MIME type text/turtle;
- The XML schema in xsd using MIME type application/xml.

Description

N/A

Example

SHACL representation

1. In turtle using MIME type text/turtle: With ^[34], a command-line tool for getting data via URL, the corresponding command is :

```
curl "https://data.europe.eu/semanticassets/ns/cbv-constraints_v1.0.0" -H 'Accept: text/turtle'
```

Direct retrievable with ‘ttl’ extension (https://data.europe.eu/semanticassets/ns/cbv-constraints_v1.0.0.ttl).

XML Schema

1. In XSD using MIME type application/XML:

```
curl "https://data.europe.eu/semanticassets/ns/cbv-constraints_v1.0.0" -H 'Accept: application/xml'
```

Direct retrievable with ‘xsd’ extension (https://data.europe.eu/semanticassets/ns/cbv-constraints_v1.0.0.xsd).

Guideline 126: Dereferencing the Vocabulary IV

Summary

Every response of the server SHOULD contain links to:

- The other representations/formats;
- The external assets that have been imported.

Description

N/A

Example

N/A

Guideline 127: Dereferencing the Vocabulary V

Summary

The HTML representation SHOULD have landing points based on the used fragment identifier. The other representations/formats.

Further technical details can be found at “*Best Practice Recipes for Publishing RDF Vocabularies*”^[35].

Description

N/A

Example

The URL https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#LegalEntity should end at the HTML fragment describing the LegalEntity Class in the document.

The URL https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#registeredAddress should result in the HTML fragment describing the registeredAddress property in the document.

Providing human-oriented documentation

For a LOD ontology, it is essential to provide definitions and usage notes for all elements. Absence of such documentation will hinder external partners in their reuse.

Guideline 128: Make easy-to-browse documentation

Summary

The documentation SHOULD be easy to browse and be able to explain all classes and relationships with their:

- labels
- descriptions
- usage notes

Such information CAN be conveyed by pointing to the free-text documentation (deliverable/report/PDF) which describes the model.

Description

N/A

Example

The proper documentation in the form of web pages with easy navigation is the preferable option. Details on such documentation are provided in section [Human-oriented documentation](#).

[33] <https://5stardata.info/en/>

[34] <https://curl.haxx.se/>

[35] <https://www.w3.org/TR/swbp-vocab-pub/#negotiation>

Chapter 9. Samples

9.1. Core RDFS

```
@prefix cb: <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#> .
@prefix adms: <http://www.w3.org/ns/adms#> .
@prefix cc: <http://creativecommons.org/ns#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix locn: <http://www.w3.org/ns/locn#> .
@prefix org: <http://www.w3.org/ns/org#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix person: <http://www.w3.org/ns/person#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix schema: <http://schema.org/> .
@prefix sioc: <http://rdfs.org/sioc/ns#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix vann: <http://purl.org/vocab/vann/> .
@prefix wdrs: <http://www.w3.org/2007/05/powder-s#> .
@prefix xhtml: <http://www.w3.org/1999/xhtml> .
@prefix xhv: <http://www.w3.org/1999/xhtml/vocab#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<https://data.europe.eu/semanticassets/ns/cbv_v1.0.0>
  a owl:Ontology;
  dcterms:title "Business eGovernment Core Vocabulary"@en;
  dcterms:description ""The Business Core Vocabulary provides a minimum set of classes
and properties for describing a registered legal entity (business). Its limited scope
does not include sole traders, or relationships between registered legal entities.
This vocabulary is closely integrated with the Location and Person eGovernment Core
Vocabularies.""@en;
  dcterms:license <https://joinup.ec.europa.eu/category/licence/isa-open-metadata-
licence-v11>;
  dcterms:modified "2020-31-03"^^xsd:date;
  adms:status <http://purl.org/adms/status/Completed>;
  rdfs:comment "This file specifies the set of RDF classes and properties used in the
Business eGovernment Core Vocabulary"@en;
  rdfs:label "Business eGovernment Core Vocabulary"@en;
  owl:imports dcterms;;
  owl:imports <http://www.w3.org/2004/02/skos/core>;
  owl:imports <http://www.w3.org/ns/locn>;
  owl:imports org;;
  owl:imports foaf;;
  .
cb:LegalEntity
  a rdfs:Class;
  a owl:Class;
  rdfs:comment ""This is the key class for the Business eGovernment Core Vocabulary
```

```

and represents a business that is legally registered."""@en;
  rdfs:isDefinedBy <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0>;
  rdfs:label "Legal Entity"@en;
  skos:altLabel "Legal Organisation"@en;
  rdfs:subClassOf org:FormalOrganization;
  skos:scopeNote ""In many countries there is a single registry although in others,
such as Spain and Germany, multiple registries exist. A Legal Entity is able to trade,
is legally liable for its actions, accounts, tax affairs etc. It is a subclass of
org:FormalOrganization which covers a wider range of entities, such as
charities.""@en;
.
cb:companyActivity
  a rdf:Property;
  a owl:ObjectProperty;
  rdfs:comment "The economical activity of a company."@en;
  rdfs:isDefinedBy <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0>;
  rdfs:label "company activity"@en;
  skos:scopeNote ""The activity of a company should be recorded using a controlled
vocabulary expressed as a SKOS concept scheme (see <a
xmlns="http://www.w3.org/ns/legal#" href="#"skos:Concept">Code</a>. Several such
vocabularies exist, many of which map to the UN's ISIC codes. Where a particular
controlled vocabulary is in use within a given context, such as SIC codes in the UK,
it is acceptable to use these, however, the preferred choice for European
interoperability is NACE.""@en;
.
cb:companyStatus
  a rdf:Property;
  a owl:ObjectProperty;
  rdfs:comment "The legal status of a company."@en;
  rdfs:isDefinedBy <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0>;
  rdfs:label "company status"@en;
  skos:scopeNote ""Terms like insolvent, bankrupt and in receivership are likely to
have different legal implications in different jurisdictions. Best Practice for
recording various other status levels is to use the relevant jurisdiction's terms and
to do so in a consistent manner using a skos:Concept as described in the <a
xmlns="http://www.w3.org/ns/legal#" href="#"skos:Concept">Code</a> Class. The term
"Normal Activity" appears to have cross-border usefulness and this should be used in
preference to terms like trading or operating.""@en;
.
cb:companyType
  a rdf:Property;
  a owl:ObjectProperty;
  rdfs:comment "the legal type of the company."@en;
  rdfs:isDefinedBy <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0>;
  rdfs:label "company type"@en;
  skos:scopeNote ""This property records the type of company. Familiar types are SA,
PLC, LLC, GmbH etc. Each
jurisdiction will have a limited set of recognised company types and these should be
used in a consistent manner using a skos:Concept as described in the <a
xmlns="http://www.w3.org/ns/legal#" href="#"skos:Concept">Code</a> Class.""@en;
.

```



```

cb:legalEntity
  a rdf:Property;
  a owl:ObjectProperty;
  rdfs:comment "The legal entity relationship can be used to link any resource to an
instance of a Legal Entity Class."@en;
  rdfs:isDefinedBy <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0>;
  rdfs:label "legal entity"@en;
  skos:scopeNote "This property can be used when an organisation includes one or more
legal entities. The Dublin Core term isPartOf is a suitable inverse of this
relationship."@en;
.
cb:legalIdentifier
  a rdf:Property;
  a owl:DatatypeProperty;
  rdfs:comment ""The legal identifier is a fundamental relationship between a legal
entity and the authority with which it is registered and that confers legal status
upon it.""@en;
  rdfs:isDefinedBy <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0>;
  rdfs:label "legal identifier"@en;
  skos:scopeNote "cbv:legalIdentifier is a sub property of dcterms:identifier.
cbv:legalIdentifier has a defined domain of cbv:LegalEntity"@en;
.
cb:legalName
  a rdf:Property;
  a owl:DatatypeProperty;
  rdfs:comment "The legal name of the business."@en;
  rdfs:isDefinedBy <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0>;
  rdfs:label "legal name"@en;
  skos:scopeNote ""A business might have more than one legal name, particularly in
countries with more than one official language. In such cases the language of the
string should be identified.""@en;
.
cb:registeredAddress
  a rdf:Property;
  a owl:ObjectProperty;
  rdfs:comment "Links a Legal Entity to its registered address."@en;
  rdfs:isDefinedBy <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0>;
  rdfs:label "registered address"@en;
  skos:scopeNote "This may or may not be the actual address at which the legal entity
does its business, it is commonly the address of their lawyer or accountant, but it is
the address to which formal communications can be sent. cbv:registeredAddress is a sub
property of locn:address that has a domain of cbv:LegalEntity and a range of
locn:Address."@en;
.

```

9.2. SHACL

```

@prefix cb: <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#> .
@prefix cb-c: <https://data.europe.eu/semanticassets/ns/cbv-constraints_v1.0.0#> .

```

```

@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<http://semic.eu/sa/cv/cbv-constraints_v1.0.0>
  a owl:Ontology ;
  dcterms:description "The Core Business Vocabulary constraint SHACL provides the
classes with their attributes and relationships, and the constraints on occurrences
and values used."@en ;
  dcterms:title "Business eGovernment Core Vocabulary constraint SHACL"@en ;
  owl:imports sh: ;
  owl:imports <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0> ;
.
cb-c:Class-rdfs_label
  a sh:PropertyShape ;
  sh:path rdfs:label ;
  sh:datatype rdf:langString ;
  sh:description "Every instance should get a rdfs:label"@en ;
  sh:message "This instance does not have a rdfs:label"@en ;
  sh:minCount 1 ;
  sh:name "rdfs label"@en ;
  sh:uniqueLang true ;
.
cb-c:ClassShape
  a sh:NodeShape ;
  rdfs:comment "Shape for checking instances of any Class"@en ;
  rdfs:label "Class shape"@en ;
  sh:property cb-c:Class-rdfs_label ;
  sh:sparql [
    sh:message "Every instance of a class must have a rdfs:label in english"@en ;
    sh:prefixes [
      sh:declare [
        sh:namespace "http://www.w3.org/2000/01/rdf-schema#"^^xsd:anyURI ;
        sh:prefix "rdfs" ;
      ] ;
    ] ;
    sh:select """SELECT $this
WHERE {$this a ?type;
      NOT EXISTS { $this rdfs:label ?label.
        FILTER((langMatches(lang(?label),\"en\")))}
}""";
  ] ;
  sh:targetClass cb:Company ;
  sh:targetClass cb:LegalEntity ;
.
cb-c:Company-companyActivity
  a sh:PropertyShape ;
  sh:path cb:companyActivity ;

```

```

sh:class <http://www.w3.org/2004/02/skos/core#Concept> ;
sh:description ""The activity of a company should be recorded using a controlled
vocabulary expressed
as a SKOS concept scheme (see <a
xmlns="https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#"
href="#skos:Concept">Code</a>. Several such vocabularies exist, many of which map to
the UN's ISIC codes. Where a particular controlled vocabulary is in use
within a given context, such as SIC codes in the UK, it is acceptable to use these,
however, the preferred choice for European interoperability is NACE.""@en ;
sh:name "company activity"@en ;
sh:nodeKind sh:IRI ;
.
cb-c:Company-companyStatus
a sh:PropertyShape ;
sh:path cb:companyStatus ;
sh:class <http://www.w3.org/2004/02/skos/core#Concept> ;
sh:description "Terms like insolvent, bankrupt and in receivership are likely to
have different legal implications in different jurisdictions. Best Practice for
recording various other status levels is to use the relevant jurisdiction's terms and
to do so in a consistent manner using a skos:Concept as described in the <a
xmlns="https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#"
href="#skos:Concept">Code</a> Class. The term \"Normal Activity\" appears to have
cross-border usefulness and this should be used in preference to terms like trading or
operating."@en ;
sh:name "company status"@en ;
sh:nodeKind sh:IRI ;
.
cb-c:Company-companyType
a sh:PropertyShape ;
sh:path cb:companyType ;
sh:class <http://www.w3.org/2004/02/skos/core#Concept> ;
sh:description "This property records the type of company. Familiar types are SA,
PLC, LLC, GmbH etc. Each jurisdiction will have a limited set of recognised company
types and these should be used in a consistent manner using a skos:Concept as
described in the <a xmlns="https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#"
href="#skos:Concept">Code</a> Class."@en ;
sh:name "company type"@en ;
sh:nodeKind sh:IRI ;
.
cb-c:CompanyShape
a sh:NodeShape ;
rdfs:comment "Shape for checking instances of Company"@en ;
rdfs:label "Company shape"@en ;
sh:property cb-c:Company-companyActivity ;
sh:property cb-c:Company-companyStatus ;
sh:property cb-c:Company-companyType ;
sh:targetClass cb:Company ;
.
cb-c:LegalEntity-dct_alternative
a sh:PropertyShape ;
sh:path dct:alternative ;

```

```

    sh:description "Some jurisdictions recognise concepts such as a trading name or
alternative forms of a legal entity's name. The Alternative Name property can be used
to record such names but should not be used to record translations of the primary
legal name. Where more than one legal name exists and where they have equal standing
but are expressed in different languages, identify the language used in each of the
multiple legal names."@en ;
    sh:name "dctterms alternative"@en ;
    sh:nodeKind sh:Literal ;
.
cb-c:LegalEntity-dct_identifier
    a sh:PropertyShape ;
    sh:path dctterms:identifier ;
    sh:description "The identifier relation links a resource to any formally issued
identifier for that resource other than one that confers legal status upon it."@en ;
    sh:name "dctterms identifier"@en ;
    sh:nodeKind sh:Literal ;
.
cb-c:LegalEntity-legalIdentifier
    a sh:PropertyShape ;
    sh:path cb:legalIdentifier ;
    sh:description "The legal identifier is the identifier under which a legal entity is
registered at an authority and that confers legal status upon it."@en ;
    sh:name "legal identifier"@en ;
    sh:nodeKind sh:Literal ;
.
cb-c:LegalEntity-legalName
    a sh:PropertyShape ;
    sh:path cb:legalName ;
    sh:description ""The legal name of the business. A business might have more than
one
legal name, particularly in countries with more than one official language. In such
cases the
language of the string should be identified.""@en ;
    sh:name "legal name"@en ;
    sh:nodeKind sh:Literal ;
.
cb-c:LegalEntity-registeredAddress
    a sh:PropertyShape ;
    sh:path cb:registeredAddress ;
    sh:class <http://www.w3.org/ns/locn#Address> ;
    sh:description "To link a resource to the address where an official communication
needs to be sent to."@en ;
    sh:name "registered address"@en ;
    sh:nodeKind sh:IRI ;
.
cb-c:LegalEntityShape
    a sh:NodeShape ;
    rdfs:comment "Shape for checking instances of Legal Entity"@en ;
    rdfs:label "Legal entity shape"@en ;
    sh:property cb-c:LegalEntity-dct_alternative ;
    sh:property cb-c:LegalEntity-dct_identifier ;

```

```

sh:property cb-c:LegalEntity-legalIdentifier ;
sh:property cb-c:LegalEntity-legalName ;
sh:property cb-c:LegalEntity-registeredAddress ;
sh:targetClass cb:LegalEntity ;
.

```

9.3. Semantics RDFS

```

@prefix cb-sem: <https://data.europe.eu/semanticassets/ns/cbv-semantic_v1.0.0#> .
@prefix cb: <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix adms: <http://www.w3.org/ns/adms#> .

<https://data.europe.eu/semanticassets/ns/cbv-semantic_v1.0.0>
  a owl:Ontology;
  owl:imports <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#>;
  owl:imports dcterms;
  owl:imports foaf;
  dcterms:title "Business eGovernment Core Vocabulary Semantics"@en;
  dcterms:description ""The Core Business Vocabulary provides the minimum set of
classes and properties for describing a registered legal entity (business) and their
machine oriented semantics e.g. the inferences that are useful to make using the
model""@en;
  dcterms:modified "2020-31-03"^^xsd:date;
  adms:status <http://purl.org/adms/status/Completed>;
.
cb:LegalEntity
  owl:disjointWith <http://www.w3.org/ns/locn#Address>;
.
cb:companyActivity
  rdfs:domain cb:Company;
  rdfs:subPropertyOf <http://www.w3.org/ns/org#classification>;
.
cb:companyStatus
  rdfs:domain cb:Company;
  rdfs:subPropertyOf <http://www.w3.org/ns/org#classification>;
.
cb:companyType
  rdfs:domain cb:Company;
;
  rdfs:subPropertyOf <http://www.w3.org/ns/org#classification>;
.
cb:legalEntity
  rdfs:range cb:LegalEntity

```

```
.
cb:legalIdentifier
  a owl:InverseFunctionalProperty;
  rdfs:range rdfs:Literal;
  rdfs:subPropertyOf <http://purl.org/dc/terms/identifier>;
.
cb:legalName
  rdfs:range rdfs:Literal;
.
cb:registeredAddress
  rdfs:range <http://www.w3.org/ns/locn#Address>;
  rdfs:subPropertyOf <http://www.w3.org/ns/locn#address>;
.
```

9.4. SHACL Application Profiles

```

@prefix cb-ap: <https://data.europe.eu/semanticassets/ns/cbv-ap_v1.0.0#> .
@prefix cb-c: <https://data.europe.eu/semanticassets/ns/cbv-constraints_v1.0.0#> .
@prefix cb: <https://data.europe.eu/semanticassets/ns/cbv_v1.0.0#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix dcterms: <http://purl.org/dc/terms/> .

<https://data.europe.eu/semanticassets/ns/cbv-ap_v1.0.0>
  a owl:Ontology;
  owl:imports <https://data.europe.eu/semanticassets/ns/cbv-constraints_v1.0.0>;
  dcterms:title "Business eGovernment Core Vocabulary constraint SHACL AP"@en;
  dcterms:description ""The Core Business Vocabulary constraint SHACL Application
Profile provides the classes with their attributes and relationships, and the
constraints on occurrences and values used.""@en;
.
cb-c:LegalEntity-companyActivity
  sh:description "Use the belgian NACE codes at <a
href=\"https://vocab.belgif.be/auth/nace2008\">https://vocab.belgif.be/auth/nace2008</
a>."@en;
  sh:pattern "^http://vocab.belgif.be/auth/nace2008/.*";
.
cb-c:LegalEntity-companyStatus
  sh:in (
    <http://vocab.belgif.be/auth/companyStatus/active>
    <http://vocab.belgif.be/auth/companyStatus/defaulted>
    <http://vocab.belgif.be/auth/companyStatus/stopped>
  );
  sh:minCount 1;
  sh:maxCount 1;
.
cb-c:LegalEntity-companyType
  sh:pattern "^http://vocab.belgif.be/auth/cbe-org1800/.+";
.
cb-c:LegalEntity-legalIdentifier
  sh:description "In Belgium use the KBO number as legal identifier."@en;
  sh:pattern "^0\\d{3}\\.\d{3}\\.\d{3}";
.

```

9.5. XSD Vocabularies

9.6. XSD Constraints and AP