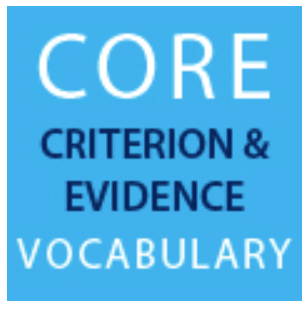
CCCEV V.2.0.0



Specification

**Change Control**

|  |  |
| --- | --- |
| **Modification** | **Details** |
| **Version 2.0.0 [Work in progress]** | |
| **First revision** |  |

[TODO: Disclaimer]

TABLE OF CONTENTS

[1. Introduction 6](#_Toc39343149)

[1.1. General Context 6](#_Toc39343150)

[1.1.1. History 6](#_Toc39343151)

[1.1.2. Use Cases 6](#_Toc39343152)

[1.2. Objective and Scope of the document 6](#_Toc39343153)

[1.3. Methodological approach 6](#_Toc39343154)

[1.4. Structure of this document 7](#_Toc39343155)

[2. Namespaces 7](#_Toc39343156)

[3. The Vocabulary (simplified overview) 8](#_Toc39343157)

[4. UML class attributes, associations and associated classes 8](#_Toc39343158)

[5. Reuse of common components and vocabularies (complete overview) 9](#_Toc39343159)

[6. CCCEV classes and properties: indexes 9](#_Toc39343160)

[6.1. Index of classes 9](#_Toc39343161)

[6.2. Index of properties 10](#_Toc39343162)

[1. CCCEV classes and properties: descriptive tables 10](#_Toc39343163)

[1.1. Class: Requirement 12](#_Toc39343164)

[1.1.1. Property: cbc:id 13](#_Toc39343165)

[1.1.2. Property: cbc:name 13](#_Toc39343166)

[1.1.3. Property: cbc:description 14](#_Toc39343167)

[1.1.4. Property: cbc:type 15](#_Toc39343168)

[1.1.5. Property: cccev:hasQualifiedRelation 15](#_Toc39343169)

[1.1.6. Property: cccev:hasRequirement 16](#_Toc39343170)

[1.1.7. Property: cccev:hasSupportingEvidence 17](#_Toc39343171)

[1.2. Class: Criterion 18](#_Toc39343172)

[1.2.1. Property: cccev:weigth 18](#_Toc39343173)

[2. SEMIC Common Aggregate Components used in the CCCEV 19](#_Toc39343174)

[3. SEMIC Common Basic Components (CBC) used in the CCCEV 19](#_Toc39343175)

[4. Annex I: Acronyms and abbreviations 20](#_Toc39343176)

[5. Annex II: XML examples 22](#_Toc39343177)

[6. References 23](#_Toc39343178)

Table of Figures

[Figure 1: the CCCEV, simplified overview 6](#_Toc39336169)

# Introduction

## General Context

### History

[TODO:

* Refer to the origins in ESPD and e-Certis2
* Refer to how UBL-2.2 adopted the model

]

### Use Cases

[TODO:

* ESPD and e-Certis, eProcurement Criteria and Constraints
* Single Digital Gateway, Evidences as Information Requirements
* Public Service eAuthorisation, RPaM eAuthorisation Criteria and Evidences
* Ask Cécile for her use cases
* Ask Costas for is use cases
* Benchmarking
* Other

]

## Objective and Scope of the document

[TODO:

* Definition of classes, properties and basic constraints (cardinality)
* This specification does not include the Semantic and Rules layers of the SEMIC design of ontologies (refer to the Style Guide).
* This is not an implementation guide. Many different implementations are possible, depending of the context of use and domain implementation rules. Refer to the forthcoming handbook.

]

## Methodological approach

[TODO:

* Refer to the style guide naming and design rules and the handbook for examples of implementations in any of the serialisation types.
* Rule for the predicate syntax (Convention: trimming of the verb in the XML and JSON serialisations);
* Reuse of Common Aggregated Components (CAC) and Common Basic Components (CBC)
* Use of stereotypes:
  + «extends» and «equivalence»
  + «abstract»
  + «reference»

]

## Structure of this document

# Namespaces

A full set of alphabetically ordered namespaces and prefixes used by the CCCEV vocabulary (or by the e-Government vocabularies imported in the CCCEV) is shown in the table below.

[TODO: complete the table]

|  |  |  |
| --- | --- | --- |
| Prefix | Namespace | Reference |
| cccev |  |  |
| cct |  |  |
| ccts |  |  |
| rdf |  |  |
| owl |  |  |
| skos |  |  |
| rdfs |  |  |
| xsd |  |  |
| xml |  |  |

# The Vocabulary (simplified overview)

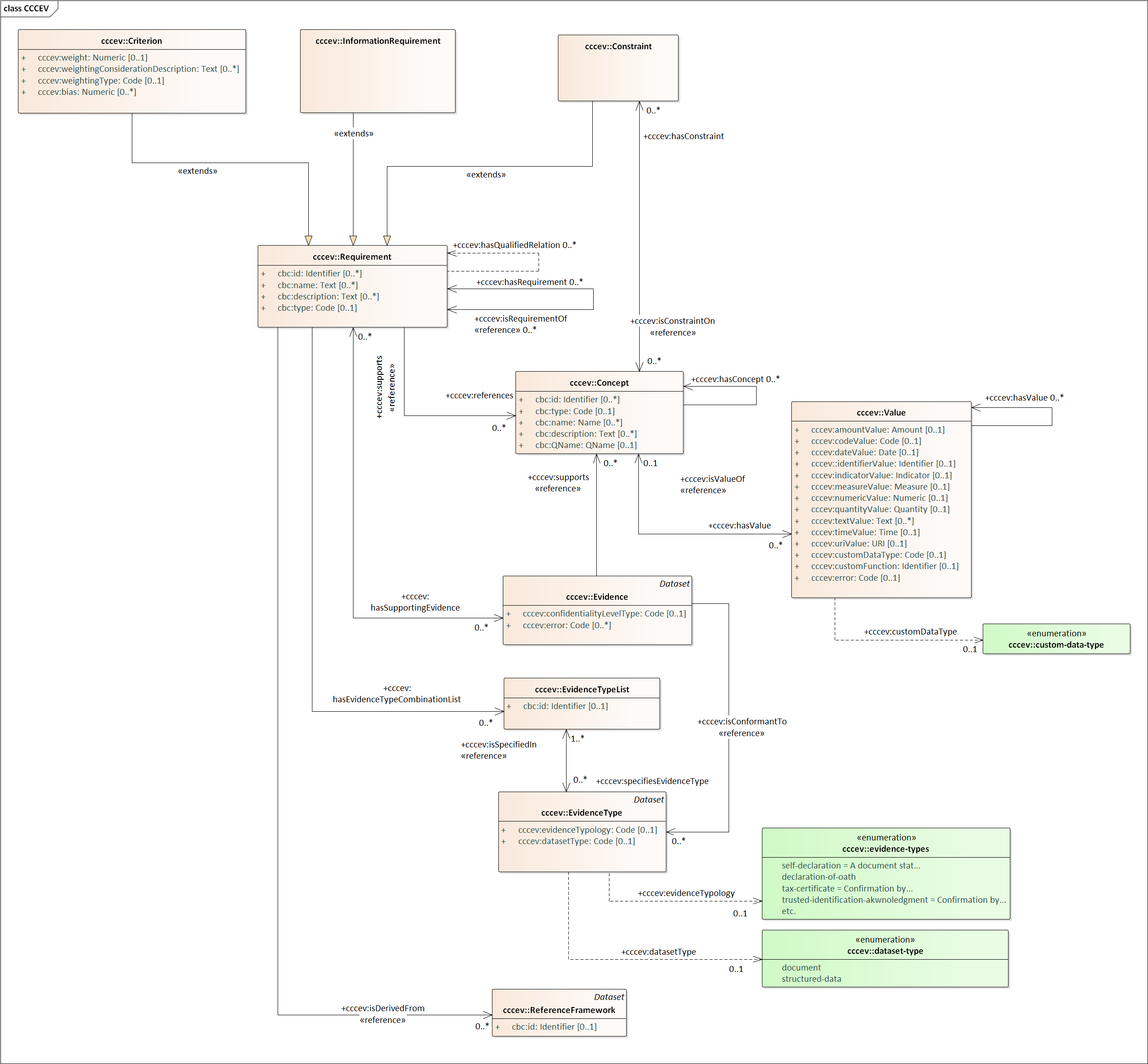


Figure 1: the CCCEV, simplified overview

# UML class attributes, associations and associated classes

[TODO:

* Data properties: class attributes represent classes which extend primitive data types that need to be accompanied by its own metadata properties (provide examples defined by the ISO 15000-t CCT);
* Object properties….define, illustrated.
* The CBC library:
  + Describe what it is. Explain that it is a library for all SEMIC vocabularies
  + Refer to the section on CBCs further away in this document
  + Explain that every *cbc* includes the CCT and the eIDAS attributes
* Associations are ‘predicates’ in the RDF world but surrogates (representative) of classes in the XSD design.
* Associated classes are ‘object properties’ in OWL, in the XSD design they are classes or the object properties (the ‘predicates’) pointing at classes.

]

# Reuse of common components and vocabularies (complete overview)

[TODO:

* Drop the layered architecture shown in the Style Guides.
* Depict the specific component libraries and vocabularies reused by the SEMIC
* Drop

]

# CCCEV classes and properties: indexes

## Index of classes

The table below lists alphabetically ordered the classes defined by the CCCEV and reused from the SEMIC’s Common Aggregated Components library. Each term is a link to the table where the class is defined in this very document. For details on the classes defined in other vocabularies or ontologies please refer to the authoritative source where they are defined. See the section *2 Namespaces* for the reference to those sources.

|  |
| --- |
| CCCEV Classes |
| Criterion, Concept, Constraint, Evidence, EvidenceType, EvidenceTypeList, InformationRequirement, PeriodValue, ReferenceFramework, [Requirement](#ClassRequirement), Value. |

|  |
| --- |
| Classes defined in the Common Aggregate Component library |
| Period |

## Index of properties

The table below lists alphabetically ordered the properties defined by the CCCEV and reused from the SEMIC’s Common Basic Components library. Each term is a link to the table where the property is defined in this very document. For details on the properties defined in other vocabularies or ontologies please refer to the authoritative source where they are defined. See the section *2 Namespaces* for the reference to those sources.

|  |
| --- |
| CCCEV Properties |
| amountValue, bias, codeValue, confidentialityLevelType, datasetType, dateValue, error, evidenceTypology, identifierValue, indicatorValue, measureValue, numericValue, quantityValue, textValue, timeValue, weight, weigthingConsiderationDescription, weightingType, |

|  |
| --- |
| Properties defined in the Common Basic Component library |
| id, name, description, endDate, endTime, durationMeasure, QName, startDate, startTime, type. |

# CCCEV classes and properties: descriptive tables

This section describes each class and property of the CCCEV. The description is presented in tables where the class or property is identified and defined. The tables are organised as follows:

* **Class tables**:
  + **Class**: the prefixed qualified name of the class representing a concept (or a set of concepts)[[1]](#footnote-1). See the section *2 Namespaces*  to check the URI of the prefix;
  + **URI**: the URI and the term assigned to the entity as defined in the CCCEV namespace. This term represents the class in any artefact, e.g. in the conceptual representation (e.g. UML diagrams), the logic design (e.g. RDF-based TBoxes, XSD and JSON schemas, E/R designs, etc.), the implementation (e.g. Knowledge Bases, XML and JSON instances, RDBMS or NoSQL dabatabases, etc.), or in validation solutions (e.g. SHCL shapes, Schematron and SVRL schemas and reports, other).
  + **Definition**: a generic explanation of the concept represented by the term. The definition may be accompanied by additional information refining the explanation by means of domain-specific interpretations, examples or references to other sources of information about the concept. When the source of the definition is not indicated, it means that it has been provided by the authors of the specification.
  + **Subclass of**: the base class (or classes) from which this class inherits;
  + **Superclass of**: the descendant specialisations of this class;
  + **Comments**: additional information related to the interpretation, design or usage of the class.
* **Properties tables**:
  + **Property Type**: specifies whether the property is a ‘data property’ (where the target entity, or range, is a ‘Literal’; or an ‘object property’, where the range is one or more classes associated to a domain via this property.
  + **URI**: the URI and the term assigned to predicate of the property as defined in the CCCEV namespace. In the case of object properties, the term assigned to the predicate follows the rule “verb in third singular person of the present tense | [adjective | adverb | noun] | [range class name]”. I the XML and JSON serialisations, the verb at the beginning of the predicate is trimmed off. See section *1.3 Methodological approach* for details on the naming and design rules;
  + **Range**: the value of the property (e.g. the instance of the literal(s) or the class(es) associated to the domain via the property)[[2]](#footnote-2);
  + **Subclass of**: the literal or class used as a base for the addition of data or object properties. This is used, for example, in the OWL2, XSD and JSON-LD serialisations for the enrichment of basic literals, as in the case of identifiers, codes, amounts, or quantities where CCT and eIDAS attributes are added to the basic entities to specify scheme identifiers, list identifiers, currency codes, unit codes, etc. For more details, see sections *4 Class attributes, associations and associated classes*, and *15 SEMIC Common Basic Components (CBC) used in the CCCEV*;
  + **Superproperty of**: descendant specialisations of this property;
  + **Inverse of**: a relation between two properties (e.g. *P1* and *P2*) according to which, given a subject (source) instance *x* and an object (target) instance *y* linked through the property *P1*, there is a symmetric property *P2* that links the instance *y* to the instance *x*. For more details see the definition provided by W3C[[3]](#footnote-3).
  + **Stereotype**: in the context of the SEMIC vocabularies, property stereotypes are used to instruct about how to treat the property when creating the syntax-specific design. Thus, for the XML or JSON serialisations, the stereotype «reference» indicates that the range of the property must be the unique identifier of a class instance that is somehow accessible at run-time.
  + **Cardinality**: the default repeatability and optionality of the property range value. Notice that the cardinalities defined for the CCCEV are all “flexible”, in the sense that they do not impose mandatory elements, and for some elements they allow multiple instances by default (e.g. multiple cardinality for identifiers and texts, so identifier maps and multiple translations can be implemented). See section *1.3 Methodological approach* for details on the naming and design rules;
  + **Comments**: additional information related to the interpretation, design or usage of the property.

## Class: Requirement

|  |  |
| --- | --- |
| Class | cccev:Requirement |
| URI: | <https://semic.org/sa/cv/cccev-2.0.0#Requirement> |
| Definition: | A condition demanded.  Additional Information:  Generic class representing any type of prerequisite that may be desired, needed or imposed as an obligation. The European Directive on services in the internal market[[4]](#footnote-4) defines requirement as any obligation, prohibition, condition or limit provided for in the laws, regulations or administrative provisions of the Member States or in consequence of case-law, administrative practice, the rules of professional bodies, or the collective rules of professional associations or other professional organisations, adopted in the exercise of their legal autonomy [..]”.  Source: based on the Collins English Dictionary, accessed through TheFreeDictionary.com hub. |
| Subclass of: | *owl:Thing* |
| Superclass of: | *cccev:Criterion*, *cccev:InformationRequirement*, *cccev:Constraint* |
| Comments: | This class is to be considered as an “abstract” class. In principle, instances of the class Requirement should not be created. Instead, more semantically-restricted or domain-related subclasses should be used, as “Criterion”, “Information Requirement”, “Constraint”, “Rule”, “Legal Obligation”, “Functional Requirement”, “Use Case Precondition”, etc. |

### Property: cbc:id

|  |  |
| --- | --- |
| Property | cbc:id |
| Property type: | Object property |
| URI: | <https://semic.org/sa/cv/common/cbc-2.0.0#id> |
| Definition: | A unique value, in the context of use, assigned to the instance of this property or to one of its subproperties. |
| Range: | *cbc:Identifier* |
| Cardinality: | 0..n |
| Comments: | **Usage note**: the multiple cardinality is intended for the establishment of ‘maps’ of co-related identifiers.  **Design notes**: the attributes of the cbc:Identifier class cater for the possibility of identifying the scheme, the agency responsible for the definition of the scheme, the URI were the identifier is accessible from, the URI where the scheme is defined, the version of the scheme, etc. It also allows to map different assurance levels issued by different authorities.  The primitive value behind a *cbc:Identifier* class is an *xsd:normalizedString*. |

### Property: cbc:name

|  |  |
| --- | --- |
| Property | cbc:name |
| Property type: | Object property |
| URI: | <https://semic.org/sa/cv/common/cbc-2.0.0#name> |
| Definition: | The preferred label used to identify the instance of this class or of one of its subclasses. |
| Range: | *cbc:Name* |
| Cardinality: | 0..n |
| Comments: | **Usage note**: The multiple cardinality is intended to allow the expression of the name by means of different linguistic systems.  **Design notes** the primitive value behind a *cbc:Name* class is an *xsd:string*, which has been extended with an optional attribute ‘*languageId’*, used to identify the linguistic system used for this name. It also includes all the assurance level-related attributes. See the class *cbc:Description* for more details.  **Implementation note**: the attribute *languageId* is namely reserved for the XML and JSON serialisations, since the RDF-based serialisations use the appendix ‘*@<languageId>*’. |

### Property: cbc:description

|  |  |
| --- | --- |
| Property | cbc:description |
| Property type: | Object property |
| URI: | <https://semic.org/sa/cv/common/cbc-2.0.0#description> |
| Definition: | A short explanation about the nature, attributes, uses or any other additional information that helps clarify the understanding of the concept being instantiated. |
| Range: | cbc:Description |
| Cardinality: | 0..n |
| Comments: | **Usage note**: The multiple cardinality is intended to allow the expression of the name by means of different linguistic systems.  **Design notes** the primitive value behind a *cbc:Description* class is an *xsd:string*, which has been extended with an optional attribute ‘*languageId’*, used to identify the linguistic system used for this name. It also includes all the assurance level-related attributes. See the class *cbc:Description* for more details.  **Implementation note**: the attribute *languageId* of the class *cbc:Description* is namely reserved for the XML and JSON serialisations, since the RDF-based serialisations use the appendix ‘*@<languageId>*’. |

### Property: cbc:type

|  |  |
| --- | --- |
| Property | cbc:type |
| Property type: | Object property |
| URI: | <https://semic.org/sa/cv/common/cbc-2.0.0#type> |
| Definition: | The identifier representing the division within a system of classification to which the instance of this class belongs.  Additional information  The classification system is normally a controlled vocabulary (e.g. a codelist or a taxonomy) the use of which is commonly agreed by a community of users in the context of a business or problem domain.  Source: based on sense #2 of the Merriam-Webster English dictionary. |
| Range: | *cbc:Code* |
| Cardinality: | 0..1 |
| Comments: | **Usage note**: A code with the maximum cardinality of 1 contributes to define the nature of the class. Codes with cardinality greater than 1 are normally qualified (i.e. the name of the code is always preceded with an adjective or a noun phrase), and usually describe the nature of an attribute of the class.  **Design note:** the primitive value behind a *cbc:Code* class is an *xsd:normalizedString*, which has been extended with the CCT and the assurance level attributes. See the class *cbc:Code*  for more details.  **Implementation note**: the recommendation is to use SKOS or SKOS-XL based controlled vocabularies for the RDF-based serialisations and OASIS Genericode 1.0 codelists for the XML serialisations. |

### Property: cccev:hasQualifiedRelation

|  |  |
| --- | --- |
| Property | cbc:hasQualifiedRelation |
| Property type: | Object property |
| URI: | <https://semic.org/sa/cv/common/cbc-2.0.0#hasQualifiedRelation> |
| Definition: | A described and/or categorised relation to the instance of another Requirement class or subclass. |
| Range: | *dcat:Relationship* |
| Subproperty of: | *dcat:qualifiedRelation* |
| Cardinality: | 0..n |
| Comments: | **Representation note**: the dotted line, in the simplified overview, indicates that the Requirement class is related to itself through a class defined in a namespace different to the CCCEV namespace (DCAT-AP, in this case). The simplified view is meant to show only the classes and properties defined within the limits of the SEMIC’s namespaces.  **Usage note**: See the class *dcat:Relationship* in the DCAT-AP and DCAT specifications[[5]](#footnote-5) for details on the meaning and use of this property.  **Implementation note**: the CCCEV code repository provides examples on how to use this property. See for instance [this one](https://github.com/SEMICeu/CCCEV/blob/CV-2.0.0/use_cases/e-certis/CriteriaCollection-example-of-ReferenceFramework.xml), implemented as an XML instance[[6]](#footnote-6). Notice that in the XML serialisation, the property is termed as ‘*qualifiedRelation*’ |

### Property: cccev:hasRequirement

|  |  |
| --- | --- |
| Property | cbc:hasRequirement |
| Property type: | Object property |
| URI: | <https://semic.org/sa/cv/common/cbc-2.0.0#hasRequirement> |
| Definition: | A sub-requirement with more granular detail(s) or of different nature.  Additional Information  An example of granular sub-requirements can be a criterion defining more specific sub-criteria. Different examples can be found in different domains and situations. Thus, in the domain of e-Procurement EU exclusion criteria (a.k.a. ‘grounds for exclusion’), national criteria are implemented as sub-criteria ‘hierarchically descending’ from an EU parent criterion. Neural networks can also be implemented as criteria and sub-criteria linked via the property *cccev:hasRequirement* pointing to a requirement of type cccev:Criterion.  Examples of sub requirements of different nature can a criterion that defines different types of information requirements (e.g. evidences to support the responses to the criterion) and establishing constraints (e.g. input parameters, thresholds, or other types of restrictions). |
| Inverse of: | *cccev:isRequirementOf* |
| Range: | *cccev:Requirement* |
| Inverse of: | *cccev:isRequirementOf* |
| Cardinality: | 0..n |
| Comments: | **Implementation note**: the CCCEV code repository provides examples on how to use this property. See for instance [any ESPD instance](https://github.com/SEMICeu/CCCEV/blob/CV-2.0.0/use_cases/espd-cv/ESPD-response-example_GeneralYearlyTurnover-AggregatedValues.xml). |

### Property: cccev:hasSupportingEvidence

|  |  |
| --- | --- |
| Property | cccev:hasSupportingEvidence |
| Property type: | Object property |
| URI: | <https://semic.org/sa/cv/common/cbc-2.0.0#hasRequirement> |
| Definition: | A sub-requirement with more granular detail(s) or of different nature.  Additional Information  An example of granular sub-requirements can be a criterion defining more specific sub-criteria. Different examples can be found in different domains and situations. Thus, in the domain of e-Procurement EU exclusion criteria (a.k.a. ‘grounds for exclusion’), national criteria are implemented as sub-criteria ‘hierarchically descending’ from an EU parent criterion. Neural networks can also be implemented as criteria and sub-criteria linked via the property *cccev:hasRequirement* pointing to a requirement of type cccev:Criterion.  Examples of sub requirements of different nature can a criterion that defines different types of information requirements (e.g. evidences to support the responses to the criterion) and establishing constraints (e.g. input parameters, thresholds, or other types of restrictions). |
| Inverse of: | *cccev:isRequirementOf* |
| Range: | *cccev:Requirement* |
| Inverse of: | *cccev:isRequirementOf* |
| Cardinality: | 0..n |
| Comments: | **Implementation note**: the CCCEV code repository provides examples on how to use this property. See for instance [any ESPD instance](https://github.com/SEMICeu/CCCEV/blob/CV-2.0.0/use_cases/espd-cv/ESPD-response-example_GeneralYearlyTurnover-AggregatedValues.xml). |

## Class: Criterion

|  |  |
| --- | --- |
| Class | cccev:Criterion |
| URI: | <https://semic.org/sa/cv/cccev-2.0.0#Criterion> |
| Definition: | A condition demanded for evaluation or assessment.  Additional Information:  [TODO: examples, etc.]  Source: based on the definition provided by the ePO[[7]](#footnote-7) and CAV[[8]](#footnote-8). |
| Subclass of: | *owl:Thing* |
| Comments: | **Implementation note**: See the [ESPD examples](https://github.com/SEMICeu/CCCEV/tree/CV-2.0.0/use_cases/espd-cv) in the Github folder for Use Cases. |

### Property: cccev:weigth

|  |  |
| --- | --- |
| Property | cccev:weight |
| Property type: | Object property |
| URI: | <https://semic.org/sa/cv/cccev-2.0.0#weight> |
| Definition: | Relative importance assigned to a given criterion.  Source: ePO |
| Range: | *cbc:Numeric* |
| Cardinality: | 0..1 |
| Comments: | **Usage note**: use a decimal value between 0 and 1. |

# SEMIC Common Aggregate Components used in the CCCEV

# SEMIC Common Basic Components (CBC) used in the CCCEV

# Annex I: Acronyms and abbreviations

|  |  |  |
| --- | --- | --- |
| Acronym | Definition | Reference |
| CAV | Common Assessment Vocabulary |  |
| CAC | Common Aggregate Components | CCCEV specification (this document) |
| CAgV | Core Agent Vocabulary | Being proposed in SEMIC. See the Core Person Vocabulary issues workspace ([Issue #6](https://github.com/SEMICeu/Core-Person-Vocabulary/issues/6)) |
| CBC | Common Basic Components | CCCEV specification (this document) |
| CCCEV |  |  |
| CCT | Core Component Types |  |
| CCTS | Core Component Type Specification |  |
| DCAT | Dataset Catalogue |  |
| DCAT-AP | Dataset Catalogue-Application Profile |  |
| ePO | eProcurement Ontology | [OP’s GitHub](https://joinup.ec.europa.eu/solution/eprocurement-ontology) |
| OWL |  |  |
| RDF | Resource Description Framework |  |
| RDFS | Resource Description Framework Schema |  |
| SKOS |  |  |
| SKOS-XL |  |  |
|  |  |  |
| XML |  |  |
| XSD |  |  |

# Annex II: XML examples

* Refer to the code repository where the examples are in the CCCEV
* Refer to the ESP GitHub and latest version (short history and reference to DG GROW and OP).
* List and comment the different examples provided and why they only contain one Criterion.

# References

1. A qualified name is a name subject to namespace interpretation. Syntactically, they are either prefixed names or unprefixed names (see <https://www.w3.org/TR/xml-names/#dt-qualname>, section ‘2.1 Basic Concepts’). A prefixed qualified name follows the pattern “prefix:entity”. [↑](#footnote-ref-1)
2. See definition in W3C <https://www.w3.org/TR/rdf-schema/#ch_range> [↑](#footnote-ref-2)
3. See the definition provided for the property relation *rdfs:inverseOf* in <https://www.w3.org/TR/owl-ref/#inverseOf-def>. [↑](#footnote-ref-3)
4. See Directive 2006/123/EU, Article 4(7), ‘Requirements’.: <https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32006L0123&from=EN>. [↑](#footnote-ref-4)
5. The latest W3C DCAT specification can be accessed at <https://www.w3.org/TR/vocab-dcat-2/>. The latest EU DCAT-AP specification is accessible through the Joinup platform: <https://joinup.ec.europa.eu/solution/dcat-application-profile-data-portals-europe/release/200>. [↑](#footnote-ref-5)
6. Example of a ‘Conviction Criterion’ used to illustrate how to structure the e-Certis Reference Framework sub-model. Last update 20200419: <https://github.com/SEMICeu/CCCEV/blob/CV-2.0.0/use_cases/e-certis/CriteriaCollection-example-of-ReferenceFramework.xml>. [↑](#footnote-ref-6)
7. ePO, eProcurement Ontology, Publications Office of the European Union: <https://joinup.ec.europa.eu/solution/eprocurement-ontology>. [↑](#footnote-ref-7)
8. Core Assessment Vocabulary (CAV), ISA2’s CAMSS: <https://joinup.ec.europa.eu/solution/core-assessment-vocabulary-cav>. [↑](#footnote-ref-8)