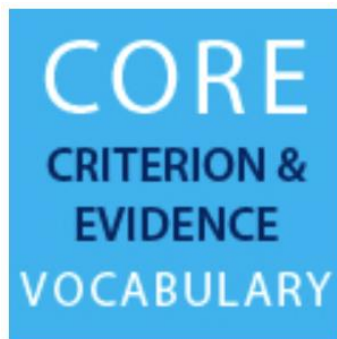


CCCEV V.2.0.0



Specification

Change Control

Modification		Details
Version 2.0.0		[Work in progress]
First revision		

[TODO: Disclaimer]

TABLE OF CONTENTS

1. INTRODUCTION.....	6
1.1. General Context	6
1.1.1. History	6
1.1.2. Use Cases.....	6
1.2. Objective and Scope of the document.....	6
1.3. Methodological approach	6
1.4. Structure of this document.....	7
2. NAMESPACES.....	7
3. THE VOCABULARY (SIMPLIFIED OVERVIEW).....	8
4. UML CLASS ATTRIBUTES, ASSOCIATIONS AND ASSOCIATED CLASSES.....	8
5. REUSE OF COMMON COMPONENTS AND VOCABULARIES (COMPLETE OVERVIEW).....	10
6. CCCEV CLASSES AND PROPERTIES: INDEXES.....	11
6.1. Index of classes	11
6.2. Index of properties.....	11
1. CCCEV CLASSES AND PROPERTIES: DESCRIPTIVE TABLES.....	12
1.1. Class: Requirement	14
1.1.1. Property: cbc:id	14
1.1.2. Property: cbc:name	14
1.1.3. Property: cbc:description	15
1.1.4. Property: cbc:type	16
1.1.5. Property: cccev:hasQualifiedRelation	18
1.1.6. Property: cccev:hasRequirement	19
1.1.7. Property: cccev:hasSupportingEvidence	20
1.2. Class: Criterion	21
1.2.1. Property: cccev:weighth.....	22
2. SEMIC COMMON AGGREGATE COMPONENTS USED IN THE CCCEV.....	29
3. SEMIC COMMON BASIC COMPONENTS (CBC) USED IN THE CCCEV.....	29
4. ANNEX I: ACRONYMS AND ABBREVIATIONS	30
5. ANNEX II: XML EXAMPLES.....	32
6. REFERENCES.....	33

TABLE OF FIGURES

Figure 1: the CCCEV, simplified overview8

1. INTRODUCTION

1.1. General Context

1.1.1. History

[TODO:

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

]

1.1.2. 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

]

1.2. 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.

]

1.3. 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»

]

1.4. Structure of this document

2. 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		

3. THE VOCABULARY (SIMPLIFIED UML)

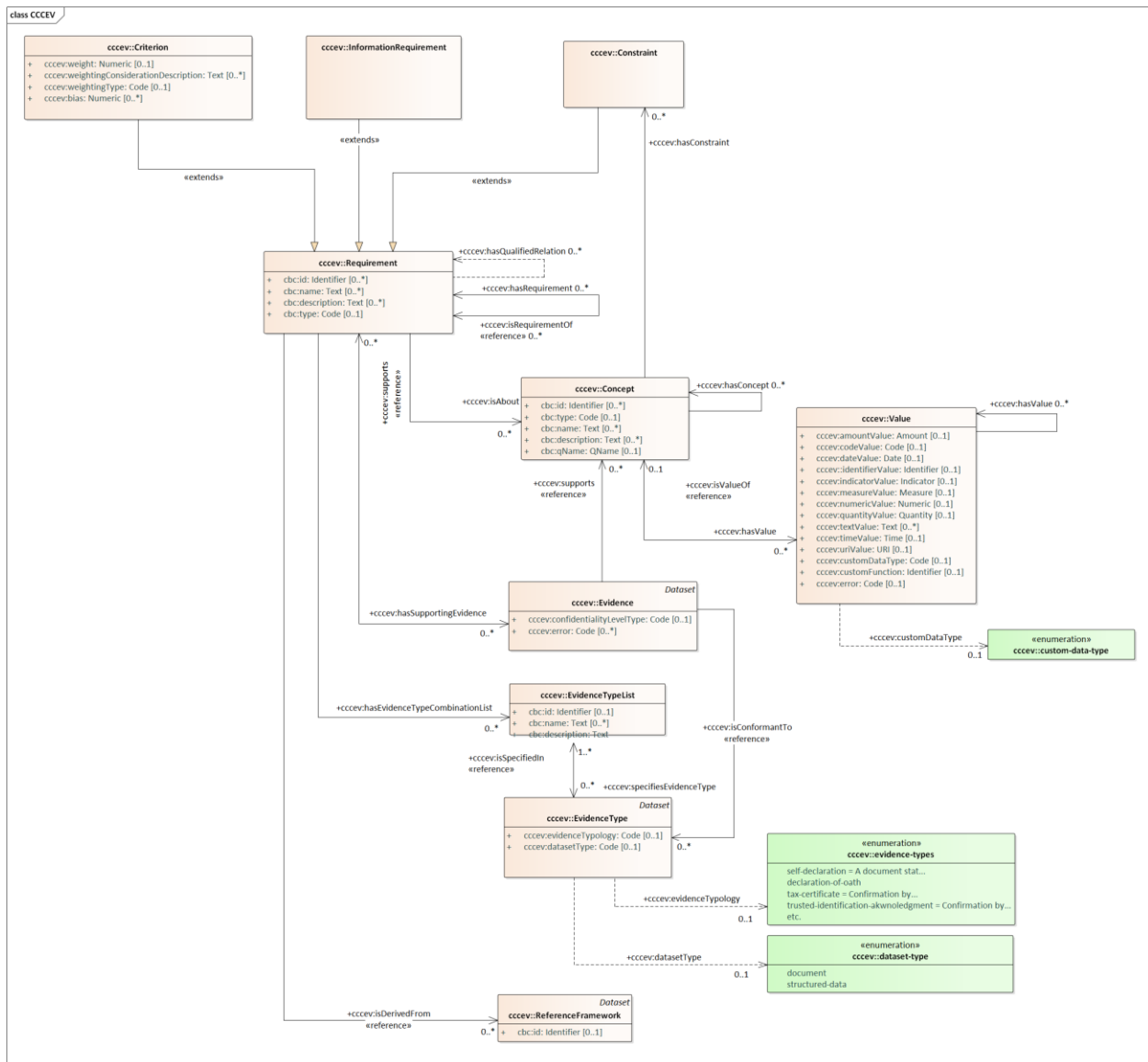


Figure 1: UML of the CCCEV, simplified overview

4. 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.

]

5. REUSE OF COMMON COMPONENTS AND VOCABULARIES (EXTENDED UML)

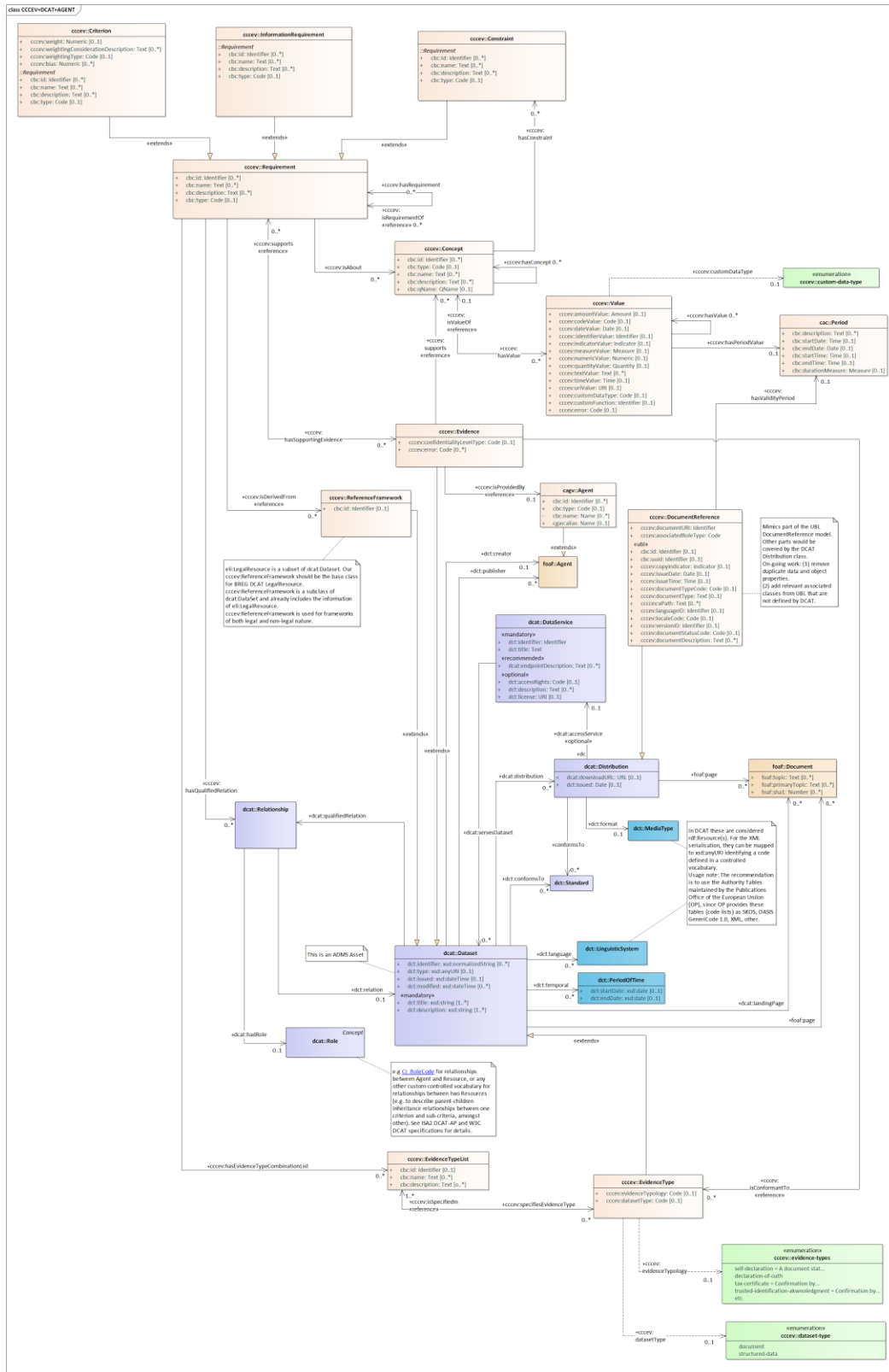


Figure 2: UML of the CCCEV, extended view

[TODO:

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

]

6. CCCEV CLASSES AND PROPERTIES: INDEXES

6.1. 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](#), Value.

Classes defined in the Common Aggregate Component library

Period

6.2. 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, weighingConsiderationDescription, weightingType,

Properties defined in the Common Basic Component library

id, name, description, endDate, endTime, durationMeasure, QName, startDate, startTime, type.

1. 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. 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 databases, 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;

¹ A qualified name is a name subject to namespace interpretation. Syntactically, they are either prefixed names or unprefix 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".

- **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]”. In 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;
 - **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 UML class attributes, associations and associated classes*, and *43 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 W3C³.
 - **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

² See definition in W3C https://www.w3.org/TR/rdf-schema/#ch_range

³ See the definition provided for the property relation *rdfs:inverseOf* in <https://www.w3.org/TR/owl-ref/#inverseOf-def>.

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.

1.1. Common Basic Properties

The following properties are reused by several domain classes. They have been grouped under this section and referred to from the following sections describing the classes in order to render the specification less verbose and to reduce error-proneness.

1.1.1. 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:	<i>cbc:Identifier</i>
Cardinality:	0..n
Comments:	<p>Usage note: the multiple cardinality is intended for the establishment of ‘maps’ of co-related identifiers.</p> <p>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 where 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.</p> <p>The primitive value behind a <i>cbc:Identifier</i> class is an <i>xsd:normalizedString</i>.</p>

1.1.2. 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:	<i>cbc:Name</i>
Cardinality:	0..n
Comments:	<p>Usage note: The multiple cardinality is intended to allow the expression of the name by means of different linguistic systems.</p> <p>Design notes the primitive value behind a <i>cbc:Name</i> class is an <i>xsd:string</i>, which has been extended with an optional attribute '<i>languageId</i>', used to identify the linguistic system used for this name. It also includes all the assurance level-related attributes. See the class <i>cbc:Description</i> for more details.</p> <p>Implementation note: the attribute <i>languageId</i> is namely reserved for the XML and JSON serialisations, since the RDF-based serialisations use the appendix '@<<i>languageId</i>>'. </p>

1.1.3. Property: *cbc:description*

Property	<i>cbc:description</i>
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:	<i>cbc:Description</i>
Cardinality:	0..n
Comments:	<p>Usage note: The multiple cardinality is intended to allow the expression of the name by means of different linguistic systems.</p> <p>Design notes the primitive value behind a <i>cbc:Description</i> class is an <i>xsd:string</i>, which has been extended with an optional attribute '<i>languageId</i>', used to identify the linguistic system used for this name.</p>

	<p>It also includes all the assurance level-related attributes. See the class <i>cbc:Description</i> for more details.</p> <p>Implementation note: the attribute <i>languageId</i> of the class <i>cbc:Description</i> is namely reserved for the XML and JSON serialisations, since the RDF-based serialisations use the appendix '@<languageId>'.</p>
--	--

1.1.4. Property: cbc:type

Property	cbc:type
Property type:	Object property
URI:	https://semic.org/sa/cv/common/cbc-2.0.0#type
Definition:	<p>The identifier representing the division within a system of classification to which the instance of this class belongs.</p> <p>Additional information</p> <p>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.</p> <p>Source: based on sense #2 of the Merriam-Webster English dictionary.</p>
Range:	<i>cbc:Code</i>
Cardinality:	0..1
Comments:	<p>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.</p> <p>Design note: the primitive value behind a <i>cbc:Code</i> class is an <i>xsd:normalizedString</i>, which has been extended with the CCT and the assurance level attributes. See the class <i>cbc:Code</i> for more details.</p> <p>Implementation note: the recommendation is to use SKOS or SKOS-XL based controlled vocabularies for the RDF-based serialisations and OASIS Genericcode 1.0 codelists for the XML serialisations.</p>

1.1.5. Property cbc:qName

Property	cbc:QName
----------	-----------

Property type:	Object property
URI:	https://semic.org/sa/cv/common/cbc-2.0.0#qName
Definition:	
Range:	<i>cbc:QName</i>
Cardinality:	0..1
Comments:	

1.2. Class: Requirement

Class	cccev:Requirement
URI:	https://semic.org/sa/cv/cccev-2.0.0#Requirement
Definition:	<p>A condition demanded.</p> <p>Additional Information:</p> <p>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 market⁴ 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 [...]".</p> <p>Source: based on the Collins English Dictionary, accessed through TheFreeDictionary.com hub.</p>
Subclass of:	<i>owl:Thing</i>
Superclass of:	<i>cccev:Criterion</i> , <i>cccev:InformationRequirement</i> , <i>cccev:Constraint</i>
Comments:	This class is to be considered as an “abstract” class. In principle, instances of the class Requirement should not be created. Instead, more

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>.

	semantically-restricted or domain-related subclasses should be used, as “Criterion”, “Information Requirement”, “Constraint”, “Rule”, “Legal Obligation”, “Functional Requirement”, “Use Case Precondition”, etc.
--	---

1.2.1. Basic properties

This class uses the following “Common Basic Properties”:

Common properties
cbc:id , cbc:name , cbc:description , cbc:type .

1.2.2. Property: cccev:hasEvidenceTypeCombinationList

Property	cccev: hasEvidenceTypeCombinationList
Property type:	Object property
URI:	https://semic.org/sa/cv/common/cbc-2.0.0#hasQualifiedRelation
Definition:	
Range:	<i>dcat:EvidenceTypeList</i>
Cardinality:	0..n
Comments:	Usage note:

1.2.3. Property: cccev:hasQualifiedRelation

Property	cccev: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:	<i>dcat:Relationship</i>
Subproperty of:	<i>dcat:qualifiedRelation</i>
Cardinality:	0..n

Comments:	<p>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.</p> <p>Usage note: See the class <i>dcat:Relationship</i> in the DCAT-AP and DCAT specifications⁵ for details on the meaning and use of this property.</p> <p>Implementation note: the CCCEV code repository provides examples on how to use this property. See for instance this one, implemented as an XML instance⁶. Notice that in the XML serialisation, the property is termed as <i>'qualifiedRelation'</i></p>
------------------	---

1.2.4. Property: cccev:hasRequirement

Property	cccev:hasRequirement
Property type:	Object property
URI:	https://semic.org/sa/cv/common/cbc-2.0.0#hasRequirement
Definition:	<p>A sub-requirement with more granular detail(s) or of different nature.</p> <p>Additional Information</p> <p>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 <i>cccev:hasRequirement</i> pointing to a requirement of type <i>cccev:Criterion</i>.</p> <p>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).</p>

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>.

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.

Inverse of:	<i>cccev:isRequirementOf</i>
Range:	<i>cccev:Requirement</i>
Cardinality:	0..n
Comments:	Implementation note: the CCCEV code repository provides examples on how to use this property. See for instance any ESPD instance .

1.2.5. Property: cccev:hasSupportingEvidence

Property	cccev:hasSupportingEvidence
Property type:	Object property
URI:	https://semic.org/sa/cv/common/cbc-2.0.0#hasSupportingEvidence
Definition:	
Inverse of:	<i>cccev:supports</i>
Range:	<i>cccev:Evidence</i>
Cardinality:	0..n
Comments:	Implementation note: the CCCEV code repository provides examples on how to use this property. See for instance any ESPD instance .

1.2.6. Property: cccev:isAbout

Property	cccev:references
Property type:	Object property
URI:	https://semic.org/sa/cv/common/cbc-2.0.0#references
Definition:	
Range:	<i>cccev:Concept</i>
Cardinality:	0..n

1.2.7. Property: cccev:isRequirementOf

Property	cccev: isRequirementOf
Property type:	Object property
URI:	https://semic.org/sa/cv/common/cbc-2.0.0# isRequirementOf
Definition:	
Inverse of:	<i>cccev:hasRequirement</i>
Range:	<i>cccev:Requirement</i>
Cardinality:	0..n

1.2.8. Property: cccev:isDerivedFrom

Property	cccev: isDerivedFrom
Property type:	Object property
URI:	https://semic.org/sa/cv/common/cbc-2.0.0# isDerivedFrom
Definition:	
Range:	<i>cccev:ReferenceFramework</i>
Cardinality:	0..n

1.3. Class: Criterion

Class	cccev:Criterion
URI:	https://semic.org/sa/cv/cccev-2.0.0#Criterion
Definition:	<p>A condition for evaluation or assessment.</p> <p>Additional Information:</p> <p>In general, criteria are used for comparison, filtering and selection. Criteria usually set minimum constraints (e.g. limits, intervals, thresholds, etc.) that need to be met in order to pass the required conditions or to fulfil them in a certain degree or quality. The evaluation of the fulfilment is usually supported with the provision of evidences.</p>

	Source: based on the definition and practices from ESPD and e-Certis ⁷ , ePO ⁸ and CAV ⁹ .
Subclass of:	<i>cccev:Requirement</i>
Comments:	<p>Design note: See the ePO⁸ for an example of domain-specific extensions of Criterion. In eProcurement three subclasses of Criterion are defined, exclusion grounds, selection criteria and award criteria. The extension of the selection criteria, aggregates a domain-specific concept, the Lot.</p> <p>Implementation note: A Criterion can either use sub-criteria or a complex Concept to define structures of detailed questions and constraints. See the ESPD examples in the Github folder for Use Cases for an example of implementation of criteria in the domain of procurement¹⁰.</p>

1.3.1. Property: cccev:weight

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:	<i>cbc:Numeric</i>
Cardinality:	0..1
Comments:	Usage note: use a decimal value between 0 and 1.

1.3.2. Property: cccev:weightingConsiderationDescription

Property	cccev:weight
----------	--------------

⁷ ESPD, European Single Procurement Document. Internal market, Industry, Entrepreneurship and SMEs: https://ec.europa.eu/growth/single-market/public-procurement/digital/espd_en; see also the GitHub repository with the Exchange Data Model: <https://github.com/ESPD/ESPD-EDM>.

⁸ ePO, eProcurement Ontology, Publications Office of the European Union: <https://joinup.ec.europa.eu/solution/eprocurement-ontology>.

⁹ Core Assessment Vocabulary (CAV), ISA²'s CAMSS: <https://joinup.ec.europa.eu/solution/core-assessment-vocabulary-cav>.

¹⁰ Examples of implementation of Criteria in XML for the ESPD Use Case: https://github.com/SEMICeu/CCCEV/tree/CV-2.0.0/use_cases/espd-cv.

Property type:	Object property
URI:	https://semic.org/sa/cv/cccev-2.0.0#weightingConsiderationDescription
Definition:	<p>The explanation of how the weighting of a Criterion is to be used at evaluation time.</p> <p>Additional Information:</p> <p>Normally used for transparency matters.</p> <p>Source: ESPD and ePO</p>
Range:	<i>cbc:Description</i>
Cardinality:	0..n
Comments:	Usage note: Multiple cardinality intended for the expression of the description in multiple languages and/or for the provision of several paragraphs.

1.3.3. Property: cccev:weightingType

Property	cccev:weightingType
Property type:	Object property
URI:	https://semic.org/sa/cv/cccev-2.0.0#weightingType
Definition:	<p>The numeric form used to weight.</p> <p>Additional Information:</p> <p>Typically these forms are decimal or percentage. However some business domains provide long lists of other types of weighing. See for example the codelist 'number-weight' used in eForms¹¹.</p> <p>Source: based on ESPD and EU Vocabularies¹².</p>
Range:	<i>cbc:Description</i>

¹¹ eForms: https://ec.europa.eu/growth/single-market/public-procurement/digital/eforms_en.

¹² EU Vocabularies, Publications Office of the European Union: <https://op.europa.eu/en/web/eu-vocabularies>.

Cardinality:	0..n
Comments:	Usage note: Multiple cardinality intended for the expression of the description in multiple languages and/or for the provision of several paragraphs.

1.3.4. Property: cccev:bias

Property	cccev:bias
Property type:	Object property
URI:	https://semic.org/sa/cv/cccev-2.0.0#bias
Definition:	<p>Additional parameter used to adjust the evaluation of the criterion composed of multiple sub-criteria.</p> <p>Additional Information:</p> <p>In neural networks, the bias is equivalent to an intercept added in a linear equation used to shift the activation function to either right or left. It can be also used to manually adjust the pass/fail calculation of a weighted criterion.</p>
Range:	<i>cbc:Numeric</i>
Cardinality:	0..1
Comments:	Design note: The properties weight and bias allow the use of the class Criterion (or a subclass of it) to build the inputs of a neuron in a neural network or for the implementation of binary perceptron classifiers.

1.3.5. Property: cccev:bias

Property	cccev:bias
Property type:	Object property
URI:	https://semic.org/sa/cv/cccev-2.0.0#bias
Definition:	<p>Additional parameter used to adjust the evaluation of the criterion composed of multiple sub-criteria.</p> <p>Additional Information:</p> <p>In neural networks, the bias is equivalent to an intercept added in a linear equation used to shift the activation function to either right or left. It can</p>

	be also used to manually adjust the pass/fail calculation of a weighted criterion.
Range:	<i>cbc:Numeric</i>
Cardinality:	0..1
Comments:	Design note: The properties weight and bias allow the use of the class Criterion (or a subclass of it) to build the inputs of a neuron in a neural network or for the implementation of binary perceptron classifiers.

1.3.6. Inherited properties

The following properties are inherited from the base class [cccev:Requirement](#).

Properties inherited from cccev:Requirement	
	cbc:id , cbc:name , cbc:description , cbc:type , cccev:isAbout , cccev:hasRequirement , cccev:isRequirementOf , cccev:hasSupportingEvidence , cccev:isDerivedFrom , cccev:hasQualifiedRelation , cccev:hasEvidenceTypeCombinationList .

1.4. Class: Information Requirement

Class	cccev:InformationRequirement
URI:	https://semic.org/sa/cv/cccev-2.0.0#InformationRequirement
Definition:	<p>A request for data that is proof of evidence or can be taken as such.</p> <p>Additional Information:</p> <p>The response to an information requirement is an evidence when the issuer of the response is an authoritative source (a.k.a. ‘source of truth’, e.g. a Civil Registry providing data about a natural person for the provision of public service through the Single Digital Gateway). In other cases the responses might not be issued by an authoritative source but the issuer supports the responses with evidences (or commits to support them timely, e.g. a self-declaration or a declaration of oath).</p>
Subclass of:	cccev:Requirement
Comments:	Usage note: The information requirement can require structured data or documents of any form. For structured data, the requirement can use ‘Concepts’ to specify the structure and type of the data expected in the response. For both structured and unstructured data, the information

	requirement can indicate the expected type of evidence, its format, source, and other properties related to the evidence.
--	---

1.4.1. *Inherited properties*

The following properties are inherited from the base class [cccev:Requirement](#).

Properties inherited from cccev:Requirement
cbc:id , cbc:name , cbc:description , cbc:type , cccev:isAbout , cccev:hasRequirement , cccev:isRequirementOf , cccev:hasSupportingEvidence , cccev:isDerivedFrom , cccev:hasQualifiedRelation , cccev:hasEvidenceTypeCombinationList .

1.5. Class: Constraint

Class	cccev: Constraint
URI:	https://semic.org/sa/cv/cccev-2.0.0#Constraint
Definition:	
Subclass of:	cccev:Requirement
Comments:	Usage note: Constraints can apply to any other type of Requirement, e.g. Criterion and InformationRequirement, via the property <i>cccev:hasRequirement</i> ; or to Concepts, via the property <i>cccev:isAbout</i> . It could also be applied to itself thus building structures of constraints and sub-constraints.

1.5.1. *Inherited properties*

The following properties are inherited from the base class [cccev:Requirement](#).

Properties inherited from cccev:Requirement
cbc:id , cbc:name , cbc:description , cbc:type , cccev:isAbout , cccev:hasRequirement , cccev:isRequirementOf , cccev:hasSupportingEvidence , cccev:isDerivedFrom , cccev:hasQualifiedRelation , cccev:hasEvidenceTypeCombinationList .

1.6. Class: Concept

Class	cccev: Concept
-------	----------------

URI:	https://semic.org/sa/cv/cccev-2.0.0#Concept
Definition:	
Subclass of:	<i>owl:Thing</i>
Comments:	Usage note:

1.6.1. Basic properties

This class uses also the following “Common Basic Properties”:

Common properties
cbc:id , cbc:type , cbc:name , cbc:description , cbc:qName .

1.6.2. Property: cccev:hasConcept

Property	cccev:hasConcept
Property type:	Object property
URI:	https://semic.org/sa/cv/cccev-2.0.0#hasConcept
Definition:	
Range:	<i>cccev:Concept</i>
Cardinality:	0..n
Comments:	Usage note:

1.6.3. Property: cccev:hasConstraint

Property	cccev:hasConstraint
Property type:	Object property
URI:	https://semic.org/sa/cv/cccev-2.0.0#hasConstraint
Definition:	
Range:	<i>cccev:Constraint</i>
Cardinality:	0..n
Comments:	Usage note:

1.6.4. Property: cccev:hasValue

Property	cccev:hasValue
Property type:	Object property
URI:	https://semic.org/sa/cv/cccev-2.0.0#hasValue
Definition:	
Range:	<i>cccev:Value</i>
Cardinality:	0..n
Comments:	Usage note:

1.7. Class: EvidenceTypeList

Class	cccev: EvidenceTypeList
URI:	https://semic.org/sa/cv/cccev-2.0.0#EvidenceTypeList
Definition:	
Subclass of:	<i>owl:Thing</i>
Comments:	Usage note:

1.7.1. Basic properties

This class uses also the following “Common Basic Properties”:

Common properties
cbc:id , cbc:type , cbc:name , cbc:description , cbc:qName .

1.7.2. Property: cccev:hasConcept

Property	cccev:hasConcept
Property type:	Object property
URI:	https://semic.org/sa/cv/cccev-2.0.0#hasConcept
Definition:	

Range:	<i>cccev:Concept</i>
Cardinality:	0..n
Comments:	Usage note:

2. SEMIC COMMON AGGREGATE COMPONENTS USED IN THE CCCEV

3. SEMIC COMMON BASIC COMPONENTS (CBC) USED IN THE CCCEV

4. 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)
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
OWL		
RDF	Resource Description Framework	
RDFS	Resource Description Framework Schema	
SKOS		
SKOS-XL		
XML		
XSD		

5. 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.

6. REFERENCES