

ePO - eProcurement Ontology 2.0.0

[Report]

Abstract

In 2016 the [Publications Office of the European Union](#) (Publications Office or OP) was commissioned by the [ISA Programme](#) to organise and support the development of an Ontology of the Public e-Procurement (ePO).

The ultimate objective of the ontology is to put forth a commonly agreed OWL ontology that will conceptualise, formally encode and make available in an open, structured and machine-readable format data about public procurement, covering it from end to end, i.e. from notification, through tendering to awarding, ordering, invoicing and payment. With this goal in mind the OP engaged a Working Group (WG) of experts with the mission of building consensus on the analysis results and deliverables developed by the OP's teams.

By mid-2017 a version 1.00 of the Ontology was delivered and submitted to the Working group for discussion. All the materials related to this version were made public and are accessible through the ISA2 [Joinup](#) platform; including the discussions held in the Working Group (WG).

A new stage of the development, identified as ePO v2.0.0, was initiated in January 2018. The goal of this new version is to improve and extend the previous version.

This document describes the objectives, the methodological approach and the deliverables produced in this new version.

Table of Content

| | |
|--|----|
| ePO - eProcurement Ontology 2.0.0 [Report] | 1 |
| Abstract | 1 |
| I. Context | 5 |
| About ePO v1.00 | 5 |
| ePO v.20.0. objectives | 7 |
| Scope | 7 |
| Tasks in-scope | 8 |
| Methodological approach | 9 |
| ePO governance | 11 |
| II. Information Requirements | 12 |
| The Data Element Dictionary (DED v2.0.0) | 12 |
| III. Conceptual Data Model (CM) | 16 |
| CM Overview | 17 |
| How to read the diagrams | 17 |
| Procurement Project | 18 |
| Procurement Procedure | 20 |
| Accelerated Procedure | 21 |
| Procurement Terms | 22 |
| Lots | 23 |
| Technique | 23 |
| Procuring Entity | 24 |
| Economic Operator | 25 |
| Contract | 26 |
| Tender | 27 |
| Evaluation Result | 27 |
| Contract | 28 |
| Criterion | 30 |
| Award Criterion | 32 |
| Exclusion Grounds | 33 |
| Selection Criterion | 34 |
| Documents | 35 |
| Contract Award Notice | 36 |
| Data Types | 37 |
| IV. Design and Implementation | 38 |

| | |
|--|----|
| ePo 1.0 RDF-XML vocabulary | 38 |
| ePO v2.0.0 OWL-DL syntax | 39 |
| Codes and Identifiers | 41 |
| Alignment to CCTS | 41 |
| V. Proof of Concept..... | 50 |
| Activity 1: Use Cases | 51 |
| Activity 2: User Stories | 52 |
| Activity 3: Competency Questions..... | 53 |
| Activity 4: Review CQs..... | 54 |
| Activity 5: Input Data Set..... | 55 |
| Activity 6: ETL process..... | 56 |
| Technical approach..... | 57 |
| Activity 7: Populate Graph store | 60 |
| Activity 8: SPARQL Queries | 62 |
| Notes..... | 65 |

I. Context

Given the increasing importance of data standards for e-Procurement, a number of initiatives driven by the public sector, the industry and academia have been launched in the recent years. Some have grown organically, while others are the result of standardisation work. The vocabularies and the semantics that they are introducing, the phases of public procurement they are covering, and the technologies they are using vary greatly amongst themselves. These differences hamper data interoperability and their reuse.

This situation creates the need for a common representation of the knowledge about the eProcurement domain as it is understood and practiced in the European Union. For this to happen a common vocabulary, axioms and rules are needed.

In 2016 the [Publications Office of the European Union](#) (Publications Office or OP) was commissioned by the [ISA Programme](#) to organise and support the development of such a standard.

This document describes the efforts made by the European Commission to develop the development of an "**eProcurement Ontology**" (henceforth referred to as the ePO).

About ePO v1.00

One fundamental document for the development of the eOP is its [Project Charter](#). This document defines the principles, scope and the foreseen time-line and resources needed to develop the Ontology.

In this document the ultimate objective was stated as "*to put forth a commonly agreed OWL Ontology that will conceptualise, formally encode and make available in an open, structured and machine-readable format data about public procurement, covering it from end to end, i.e. from notification, through tendering to awarding, ordering, invoicing and payment*".

The figure below represents the eProcurement Value-Chain as commonly envisaged in Europe (source: OP).

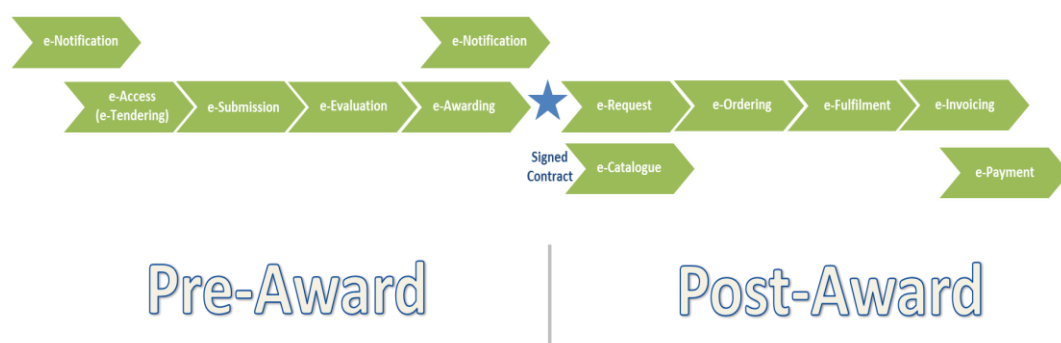


Figure 1. eProcurement Value-Chain, an EU vision

The scope of the Charter was set to cover the "specification showing the conceptual model and its representation in OWL, and the deployment of the ontology and related code lists and classifications on the [Metadata Registry](#)".

The original aim was to produce the final Ontology within twelve months including a public review of at least two months; and a set of three deliverables was identified as the main outcome of the project:

| Deliverable Name | Deliverable Description |
|---------------------------------------|---|
| e-Procurement Conceptual Model | Conceptual model of the e-procurement ontology specifying the relevant entities, attributes and relationships. This deliverable will be developed in <i>an incremental way</i> , with several drafts being created and published for discussion in the working group. |
| Specification of the conceptual model | The specification will provide the definition of the concepts and relationships and eventual synonyms. |
| e-Procurement Ontology | OWL expression of the ontology. |

Figure 2. ePO Main Deliverables

For the development of these objectives the OP team proposed a methodological approach based on the recommendations and good practices "Ontology Development 101: A Guide to Creating Your First Ontology"^[1]"e-Government Core Vocabularies handbook"^[2].

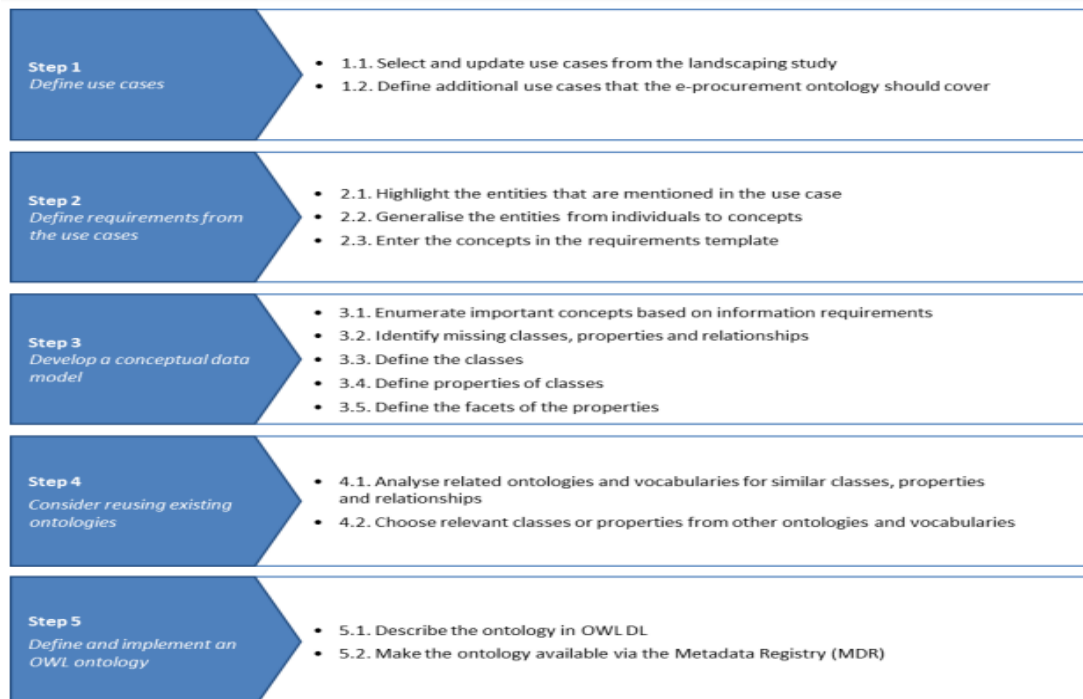


Figure 3. ePO v1.00 - Development methodological approach (source [D02.01 Specification and Methodology](#))

Both the Working Group (WG) and the Publications Office (OP) put hands on the work and a version 1.0 of the eProcurement Ontology (ePO) was delivered in the planned period of 12 months.

ePO v.20.0. objectives

The experience of the version 1.0 proved that the goal of developing a whole ontology on eProcurement that is aligned to the EU legislation and practices was too ambitious to be completed in 12 months.

Two other relevant conclusions were drawn for that experience:

1. The concepts of the ontology needed of common agreed terms and definitions that directed the design and implementation;
2. The development of the ontology requires a "phased" approach based on the Use Cases defined version 1.0 on the one hand, but also focused on at least one of the processes of the eProcurement value chain as understood in the European Union (EU). See figure 4 above.

Hence the proposal of second version of the ePO ontology, named ePO 2.0.0.

The main objective of the ePO v2.0.0 is to **take leverage of the results produced in version 1.00** and to **extend and hone the OWL Ontology**. To reach these objectives the owners of the project have set the following strategic objectives:

1. Focus on only one important policy area, e.g. "**Transparency**";
2. Extend and perfect a small set of phases of the eProcurement, if possible only one, e.g. **e-Notification** and **e-Access**.
3. Develop and test exhaustively the phase chosen before undertaking the evolution of the ePO into other phases of the eProcurement;
4. While developing the selected phase, elicit and define information requirements and data elements that will be used in other phases, even if the selected phases - i.e. eNotification and eAccess-do not use them;
5. Select a rich source of information from where to extract data in order to populate and test the ontology, e.g. the TED portal.

Scope

Hence **the scope of the ePO v2.0.0 was set to the eNotification and eAccess phases** of the Public eProcurement valuechain, represented as blue coloured in the figure below:

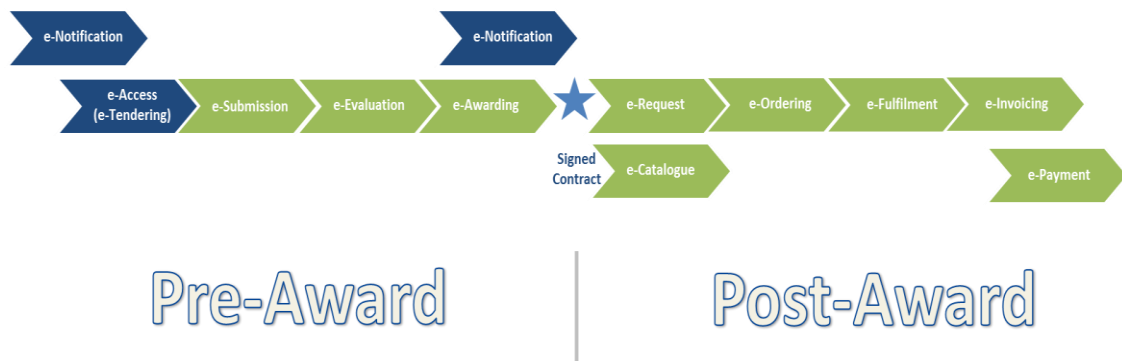


Figure 4. ePO v2.0.0 Scope of the ePO v2.0.0, eNotification and eAccess

Tasks in-scope

The Knowledge Map (K-Map) ^[3] method and tools below provides an abstract representation of the objectives, tasks, inputs and outputs of the tasks in the scope of the ePO v2.0.0.

Eac task (ellipses in blue) is used to organise the structure of the rest of this document into four main sections:

- Information elicitation: About the main inputs taken into consideration when identifying information requirement and artefacts used to this elicitation;
- Conceptual Data Model: About the analysis of the information requirements - and business rules- and the drafting of a simple graphic representation of the Ontology;
- OWL design and implementation: About the transformation of the Conceptual Data Model into a machine-readable format (TTL) that includes the vocabulary and the axioms of the ePO;
- ePO testing: About the Proof-of-Concept developed to test and refine the Conceptual Data Model and the OWL implementation.

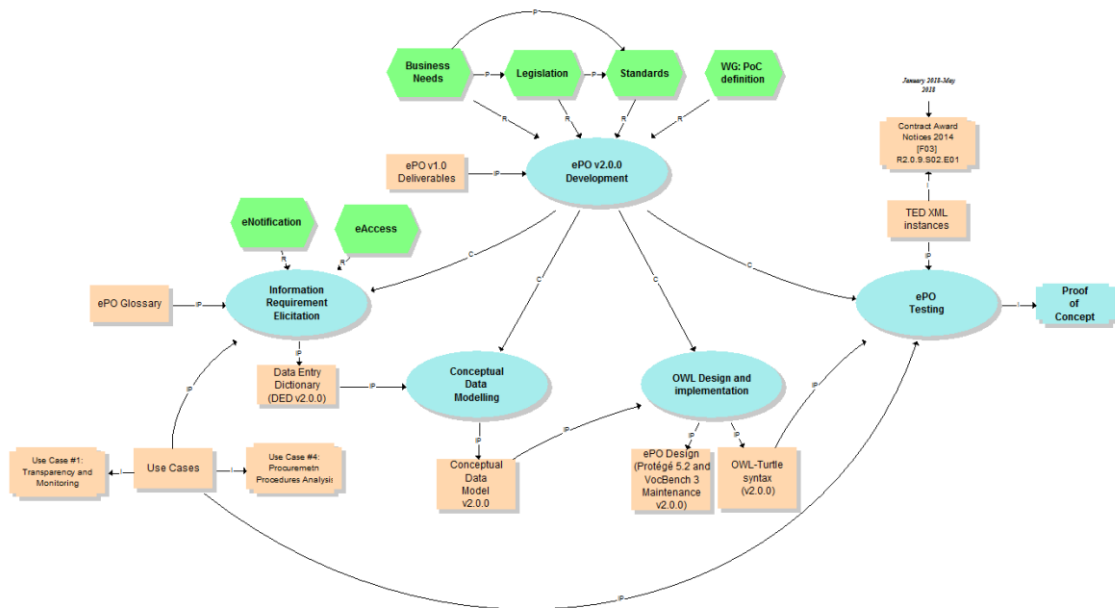


Figure 5. ePO Project v2.0.0 - Objectives

This other activity diagram, below, provides a more simplified view of the recurrent (and cyclic) activities of requirements elicitation, data element definition, analysis, design, implementation and testing:

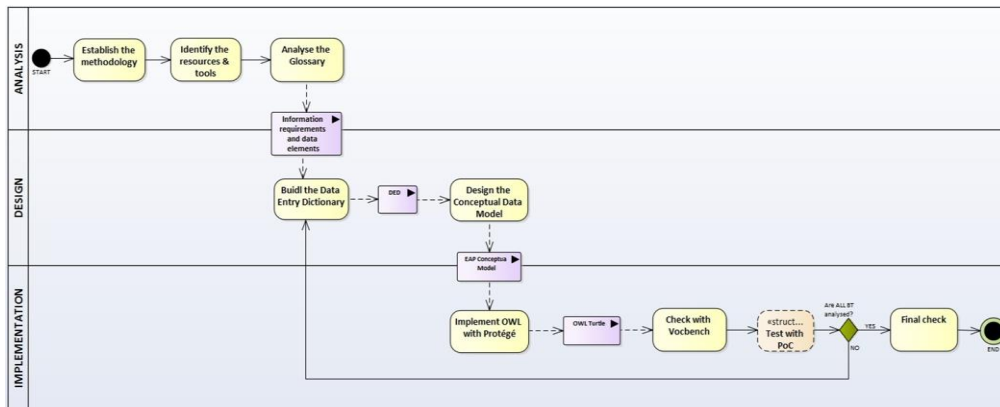


Figure 6. ePO Project v2.0.0 - development activities

Methodological approach

To reach the strategic objectives stated above this version 2.0.0 of the ePO proposes to evolve the previous v1.00 based on a few global principles:

1. **Business and Information requirements cannot contradict the EU and the Member States legislation;**
 - The ePO Ontology is soundly based on the EU Directives on Public Procurement [4], the ePO glossary, the new eForms Regulation (under consultation), the General Data Protection Regulation ([GDPR](#)), and other EU legal frameworks related to the Public Procurement;

2. Information Requirements should be identified through the exhaustive analysis of the business processes;

- This version 2.0.0 takes as inputs works like the [CEN/BII Workshops](#), the DG GROWTH's [ESPD](#)-related developments, the eSens developments on the VCD and the ESPD (see the ESPDint project), as well as international standards like UN/CEFACT and OASIS UBL. Most of them used an UN/CEFACT Universal Method Modelling (UMM) approach that represent the "value-chain Use Cases" to identify the information requirements exhaustively. The "value chain Use Cases" eNotification and eAccess are described in these works and cast lists of information requirements and business rules.

3. Use Cases around the policy area should be defined to (i) complement the identification of information requirements; and (ii) test the completeness, correctness and performance of the Ontology;

- The link:Use Cases defined in the ePO v1.00 are essential instruments to complete and test the information requirements elicited during the analysis of the business processes. In this version we propose to focus on the "[Use Case 1: Data Journalism](#)" and the [Use Case 4: Analysing eProcurement Procedures](#), as the eNotification and eAccess provide the largest part of the requirements necessary to implement the case.

4. Reuse of ontologies:

- There is no point in reinventing models that do already exist and are so generic that they can be reused. That is the case, for example, when the time comes to represent entities like natural person, legal person, address, etc. Hence, for generic ontologies, we proposed to reuse: W3C's rovo ([Registered Organization Vocabulary](#)), org ([Organization](#)), skos ([Simple Knowledge Organization System](#)) and vCard ([Virtual Contact File](#); originally proposed by the [Versit Consortium](#)); foaf ([Friend of a Friend](#)).
- On another hand, other lexical (non-ontological) resources and good practices have been taken as models to inspire the drafting of sub-vocabularies to be imported and reused by the ePO. This has been the case of the CCTS ([Core Component Type Specification](#), an [ebXML](#) model based on the [ISO 15000-5:2014](#) maintained by UN/CEFACT) for the data types Identifier, Amount, Quantity and Measure ^[5]; and [UBL-2.2](#) (OASIS [Universal Business Language](#), just the Period element, for the time being).
- Finally, the inputs from other ontology developments have been, and still are, appreciated and welcome. Thus the OCDS ([Open Contracting Partnership](#)) and PPROC ([Public Procurement Ontology](#)) vocabularies where studied and lessons were drawn from them.

5. The Ontology must always be tested and perfected using a sufficiently large sample of real data;

- The analysis and design of an Ontology cannot be declared as finished unless it is tested. The Use Cases are only one of the essential elements to reach this goal. However the testing cannot be trusted unless the data used are not (i) abundant, (ii) real data, and (iii) with a certain degree of quality. In the case of the **eNotification** and **eAccess** phases the use of the Tender Electronic Daily (TED) platform was proposed to retrieve, transform and load the data contained in Contract Notice. Contract Notices contain the richest data related to the procurement procedure and the contract awarded.

ePO governance

In order to develop these global deliverables the following Governance Structure, the following roles and responsibilities were established in ePO v1.00 and are still valid for the governance of the ePO v2.0.0:



Figure 7. ePO Project - Governance Structure

With the following **roles and responsibilities**:

| Who | What | When |
|--------------------------------|---|---|
| The Publications Office | Owns the project, provides oversight and supplies WG chair; endorses the final result at the end of the project. | Continuously |
| Working Group | Provides input, reviews and validate the ontology in synergy with the active developmental propositions of editors. | Continuously |
| Community | Participates in public review | At publication of draft for public review |

Figure 8. ePO Project - Roles and responsibilities

For more details on the members of each governance body see the document [D04.07 Report on policy support for eProcurement, eProcurement ontology](#); e.g. "Editors: are responsible for the operational work of defining and documenting the ePO".

II. Information Requirements

According to the development plan, the first task to undertake was the elicitation of information requirements, axioms and business rules. For this the version 2.0.0 of the ePO has taken into account the following inputs:

1. The *Use Case 1 defined in version 1.00*, [Data Journalism](#); and
2. The *Standards, business needs and Legislation* studied in the previous versions, plus the most recent developments;
3. The *EU eProcurement Glossary*, maintained by the ePO Working group; see [Glossary Management](#) and [Glossary](#).

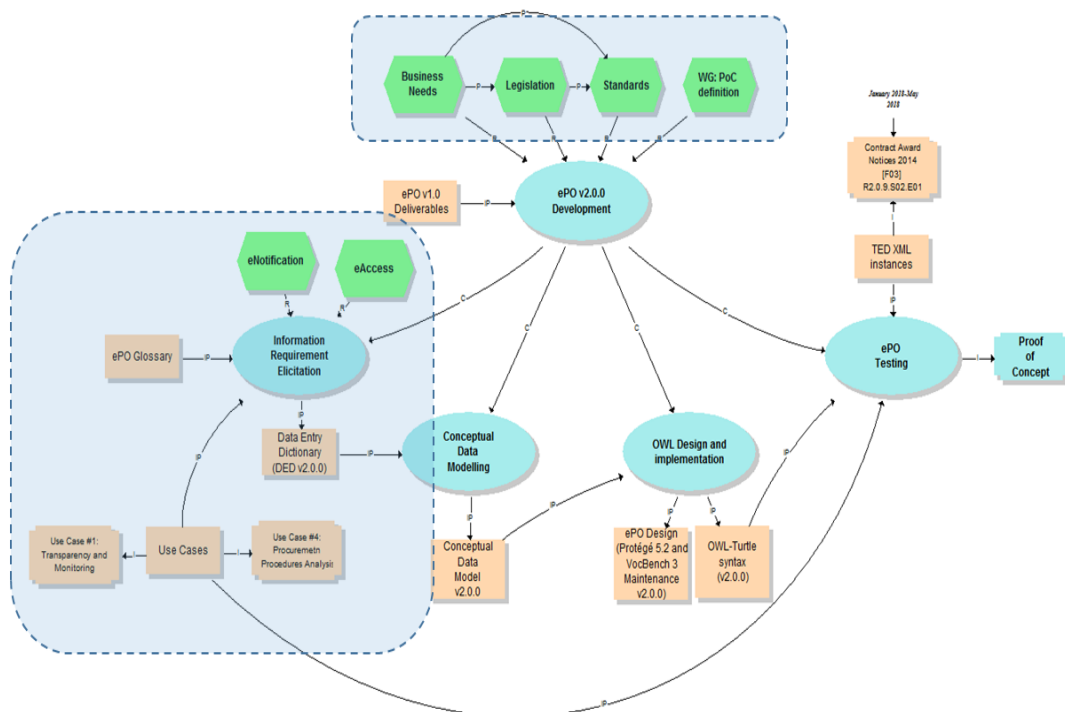


Figure 9. Information Requirements elicitation

The outcome of this task are namely two artefacts:

1. The *Information Requirements (IR v2.0.0)* already identified in [Information Requirements v1.00](#);
2. The *Data Element Dictionary (DED v2.0.0)*: a [spread-sheet](#) where the Use Case information requirements are broken down into more detailed information requirements, axioms, business rules and annotations resulting from the analysis.

The Data Element Dictionary (DED v2.0.0)

Information Analysts need to, based on each information requirement, make decisions like determining whether an element is an entity representing a *class* of objects, an *attribute* of the class of a basic data type e.g. a code, an identifier, a date, a time, a text, etc.), or a more complex *data object* associated to the class (e.g. another class).

One way of listing this information is using a **Data Element Dictionary**, a table (e.g. a [spread-sheet](#) with the following columns and rows:

| IR ID# | ePO Business Term | Concept Definition (these definitions come from the Glossary) | Examples | Comments | Inheritance | Range | Cardinality | In e-Forms | ePO v1.0C In PPROC | In OCDS In URL-2.2 | Anyoms | Avyom Objects | Business Rules |
|--------|---------------------------------------|---|---|--|---------------------------|--------------------|-------------|------------|--------------------|--|--------|---------------------------------|----------------|
| 1 | Accelerated Procedure | A procurement procedure where the time limit for receipt of tenders | | | | | | YES | | | | | |
| 2 | Accelerated Procedure Type | Identifies the reason why the procurement procedure is not ordinary, urgency, emergency | | | | Code | 1 | NO | | | | | |
| 3 | Accelerated Procedure Justification | Explanation about why the choice of an accelerated procedure is | | | | Text | 1..n | YES | | | | | |
| 4 | Access Tool | The tools and devices for electronic communication between cont | | | | | | NO | | | | | |
| 5 | Access Tool URI | The URL of the tools and devices. | | | | URI | 1 | YES | | | | | |
| 6 | Additional Information | Any other relevant data not covered elsewhere. | | | | Text | 0..1 | YES | | NO | | | |
| 7 | Buyer | Body that managed the the budget allocated for the procedure anc | | | org:Organization | | | | | YES | YES | | |
| 8 | Buyer Category | Category of the procuring agent in the procurement procedure. | | This make sense especially, but perhaps n | | Code | 0..1 | YES | | | | | |
| 9 | Buyer Role | Identifies the function of the buyer in the procurement procedure. | | Two roles identified so far: CPB, BuyerOnB | | Code | 1 | | | | | | |
| 10 | Buyer Profile | Web site where the buyer publishes information regarding procur | | | Buyer Profile | | 0..n | NO | | | | | |
| 11 | Buyer Profile | Web site where the buyer publishes information regarding procur | | | | | | NO | | | | | |
| 12 | Buyer Profile URL | World wide web uniform resource locator that identifies the locat | | | | URI | 1 | YES | | | | | |
| 13 | Criterion | [TODO: ENRIC] | | We define or own Criterion based on the ccev and espd vocabularies | | | | NO | | | | | |
| 14 | Criterion ID | Identifies the criterion unambiguously and uniquely across all pr | The UUID of a criterion as issued by e-Certis | | | Identifier | 0..1 | NO | | | | | |
| 15 | Criterion Name | A short text representing the group of properties inside a criterion. | | | | Text | 0..n | NO | | | | | |
| 16 | Criterion Description | A narrative description describing the content of the group of proq | | | | Text | 0..n | NO | | | | | |
| 17 | Weighting Consideration Description | [TODO: Enric, espd-based] | | | | Text | 0..n | NO | | | | | |
| 18 | Weight | [TODO: Enric, espd-based] | | | | Numeric | 0..1 | NO | | | | | |
| 19 | Criterion Evaluation Method Type | [TODO: Enric, espd-based] | | espd: | | Code | 0..1 | NO | | | | | |
| 20 | Criterion Type | [TODO: Enric, espd-based] | | espd-based (extension of) | | Code | 1 | NO | | | | | |
| 21 | is Criterion Objective | [TODO: Enric, espd-based] | | ocds-based ??? | | Indicator | 1 | NO | | | | | |
| 22 | Criterion Fulfilled Indicator | [TODO: Enric, espd-based] | | ccev-based (used at e-Evaluation time) | | Indicator | 0..1 | NO | | | | | |
| 23 | Legal Basis | [TODO: based on e-Forms] | | | | Legislation | 1..n | NO | | | | | |
| 24 | Property Group | [TODO: Enric, espd-based] | | | | Property Group | 1..n | NO | | | | | |
| 25 | Criterion Property | [TODO: ENRIC] | | espd: | | | | NO | | | | | |
| 26 | Criterion Property ID | Identifies unambiguously and uniquely a particular structure of d | | | | Identifier | 0..1 | NO | | | | | |
| 27 | Criterion Property Name | A short text representing the group of properties inside a criterion | | | | Text | 0..n | NO | | | | | |
| 28 | Criterion Property Description | A narrative description describing the content of the group of proq | | | | Text | 0..n | NO | | | | | |
| 29 | Criterion Property Type | A code categorizing the data structure category, how to use or pro | | espd-based (CAPTION) | | Code | 1 | NO | | | | | |
| 30 | Criterion Property Value Data Type | [TODO: Enric, espd-based] | | | | Code | 1 | NO | | | | | |
| 31 | Criterion Property Value | [TODO: Enric, espd-based] | | | | Criterion Property | 0..n | NO | | | | | |
| 32 | Criterion Property Evidence Template | A reference to a class of documents that can be required by the procuring entity. | | espd-based | | Evidence | 0..n | NO | | | | This can only be instantiated i | |
| 33 | Criterion Property Response | [TODO: ENRIC] | | espd: | | | | NO | | | | | |
| 34 | Criterion Property Response ID | [TODO: ENRIC] | | | | Identifier | 0..1 | NO | | | | | |
| 35 | Criterion Property Response Criterion | [TODO: ENRIC] | | ccev-based | | Criterion Property | 1 | NO | | | | | |
| 36 | Criterion Property Response Value | [TODO: ENRIC] | | | | Value | 1..n | NO | | | | | |
| 37 | Criterion Property Evidence | A reference to a class of documents that can be required by the procuring entity. | | espd-based | | Evidence | 0..n | NO | | | | This can only be instantiated i | |
| 38 | Economic Operator | Any natural or legal person or public entity, including any tempor | | espd-based | rovRegisteredOrganisation | | | NO | | YES | | | |
| 39 | Industry Classification Code | [TODO: NURIA] | Micro, Small, Medium, Lar | espd-based | | Code | 1 | SI | | YES: this element is reused from URL, but Compulsory use of the OP Cod | | | |
| 40 | Economic Operator Group | [TODO: NURIA] | | espd-based | rovRegisteredOrganisation | | | NO | | YES but differently. | | | |
| 41 | Economic Operator Group ID | [TODO: NURIA] | | | | Identifier | 0..1 | NO | | | | | |
| 42 | Economic Operator Group Type | [TODO: NURIA] | | | | Code | 1 | NO | | | | Compulsory use of the OP Cod | |
| 43 | Economic Operator Group Name | [TODO: NURIA] | | | | Text | 1 | NO | | | | If the group has already been r | |
| 44 | Economic Operator | Any natural or legal person or public entity, including any tempor | | | Economic Operator | | 1..* | NO | | | | | |

Figure 10. Information Requirements elicitation

Uses of the DED

The DED is normally used with three objectives:

1. To help analysts design the Ontology. The DED is a "logical artefact". It takes the "Conceptual Data Model" as an input and reflects the conceptual model and adds more technical details, such as all object and data properties of each class, their axioms and constraints;
2. To maintain the definitions of the data elements. The ePO Glossary contains mainly the definitions of the concepts used in the Ontology. The DED takes the

definitions of the ePO Glossary for the classes and adds definitions for each property of each class;

3. To **identify reference data linked to the data elements**, i.e. code lists and taxonomies;
4. To **automatise the production of different syntax bindings**. The DED is usually kept as a spread-sheet. This spread-sheet can be easily used to generate XML, OWL or other machine-readable renditions of the data model. Thus it could be used to generate automatically the OWL-TTL expression of the ePO Ontology. Specifications like UN/CEFACT and UBL use the DED to automatically generate XSD schemas fully annotated (documented) with the data element definitions, examples, etc. This does also facilitates the registration of these data elements in registries for their automatic discovery and cross-sector mapping (See [ISO 11179-6:2015 Registration](#) parts for more details on this).

Current status of the DED

The DED depends on the Glossary definitions and on the Conceptual Data Model, amongst other inputs for the elicitation of information requirements. The ePO Glossary is currently under revision by the members of the Working Group. This revision makes evident how the Conceptual Data Model can be improved. It also helps to identify elements that may be designed as object or data properties.

As the ePO Glossary is an ongoing work the DED cannot be considered finished. Additionally many of the DED properties will have to be defined based on their context, the class where they belong in. These definitions are being worked out during the discussions about the ePO Glossary with the Member States.

Content of the DED

The content of the DED is organised as follows:

Columns:

- **A - "IR#ID"**: Reserved to link each entry of the Dictionary (each element) with the general or concrete information requirement that generated the class, attribute or property;
- **B - "ePO Business Term"**: Contains a label in English ("the term") assigned by the analysts to each class or property of the Dictionary. Beware that: (i) *A term is a set of one or more words that represent a concept*; (ii) most of the concepts of the ePO Ontology are defined in the [ePO Glossary](#); and (iii) the analysts sometimes decide to shorten the text (the label) of the term by combining differently the words of the term or by eliminating some words (e.g. "Access Tool URI" instead of "URI of the Access Tool"). The reason for this is that at design and implementation time the name of the classes and properties need to be simple and yet self-explanatory. When this happens, the rewording shall be agreed with the Working Group.

- **C - "Concept Definition"**: The definition of each concept as it appears in the ePO Glossary.
- **D - "Examples"**: When considered useful to better illustrate the concept, this column contains examples. Concept definitions should not contain examples (hence the [ISO 11179-3:2015](#) defines a special field for documenting the examples for data elements that may be registered for automatic discovery and reuse).
- **E - "Comments"**: Notes and observations by the analysts that may be relevant at design time; e.g. "Buyer Category - This make sense especially, but perhaps not uniquely, in the case of Framework Agreements where the Buyer can have an "Added Category"; or "Buyer Role - Two roles identified so far: "Central Purchasing Body" and "Buyer On Behalf Of Other Procuring Entities", etc.
- **F - "Inheritance"**: Some classes can already be proposed at this phase to be considered (at design time) as possible base (parent) classes; e.g. the study and knowledge of the W3C Organization Ontology (identified with the prefix "org:") tells the analysts that the Buyer is a class that can inherit many of its attributes from the the "org:Organization" class.
- **G - "Range"**: Identifies the type of a data type or of an object type. The name of the column, "range", comes from the fact that these elements can be seen as the "object" of a *triple* composed of (i) a "subject", i.e. the class being analysed (the "domain"); (ii) a "predicate", i.e. the property that links the subject and the object; and (iii) this "object".
- **H - "Cardinality"**: Identifies the multiplicity and compulsority of an element inside a class. The possibilities are: 1, meaning "compulsory"; 1..n, meaning at least one instance is compulsory, but additional instances are also possible; 0..1, meaning optional and if used maximum one instance; 0..n, meaning optional and if used multiple instances are possible.
- **J to M - in e-Forms, v1.00, OCDS, etc.**: used by the analysts to check whether this elements was defined in one of the studied ontologies, standards or resources **related to the business domain** selected to be reused. Beware that other **generic** ontologies and vocabularies are also used or reused by ePO, e.g. W3C org (Organization), W3C rov (Registered Organizations), ISA2's Core Criterion and Evidence Vocabulary, Dublin Core, vCard, FOAF, etc.
- **N - "Axioms"**: Analysts while studying the data element MAY already identify certain elementary conditions to which the properties MAY be submitted, e.g. transitivity, disjointness, reciprocity, etc.;
- **O - "Axiom Objects"**: The object of the axiom; as in "Lots are disjoint with Groups of Lots" where "Groups of Lots" are the object of the disjoint axiom, meaning that a specific procurement procedure that is divided into Lots will refer to individual Lots or to Groups of Lots but not to both;

- **P - "Business Rules"**: Ontology constraints and axioms cannot control specific business rules, as when flexible cardinalities that in certain situations need to be further restricted (e.g.: "If Procurement Procedure is divided into lots then cardinality should be 1"); or to check the values of two or more fields that is present in different individuals (e.g., "If an economic group has already been registered the text of the group name should match exactly the text kept in the registry. If this name is used in different places the text of the name **MUST** be always, and exactly, the same in all those placeholders."

Rows:

- **"Pink rows"**: represents a class. The rows between one pink row and another are the content of the class;
- **"Transparent rows"**: represent a property of a class the range of which is an attribute (simple data type);
- **"Green rows"**: represents a property of class the range of which is another class of the Ontology.

III. Conceptual Data Model (CM)

The construction of the DED ran in parallel to the drafting of the Domain Conceptual Data Model. Up to 15 diagrams have been drafted. Fourteen of them cover specific "topics" in the scope of eNotification and eAccess, e.g. Procurement Project, Procurement Procedure, Procuring Entity, Economic Operator, Lots, Procurement Terms, etc. An additional "overview" diagram provides a general view of the whole model.

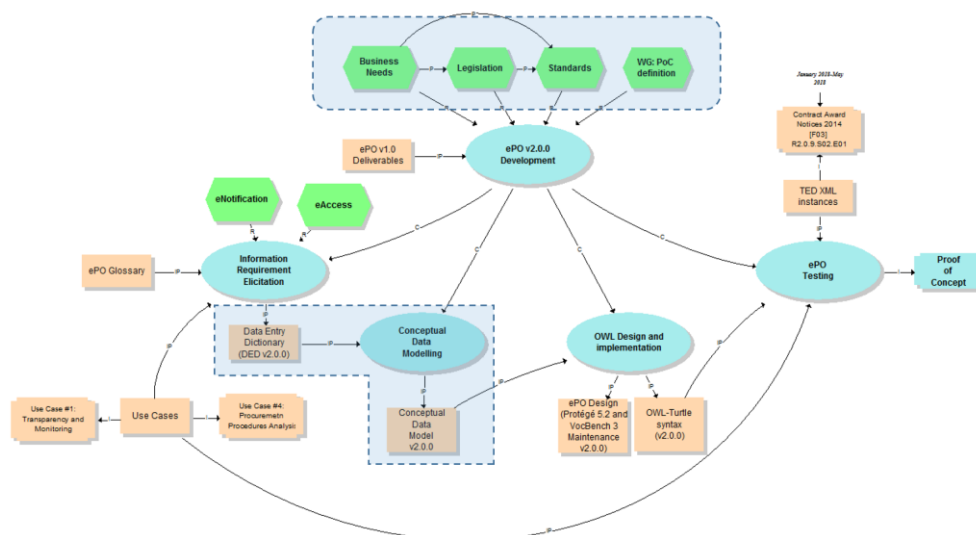


Figure 11. Development of the Domain Conceptual Data Model

The conceptual data model is available both as an [XMI format - version 2.1](#) and [Enterprise Architect](#) project file.

CM Overview

General view of the Classes and relations of the ePO v2.0.0 Ontology. Keep in mind that for this version the focus was put on e-Notification and e-Access.

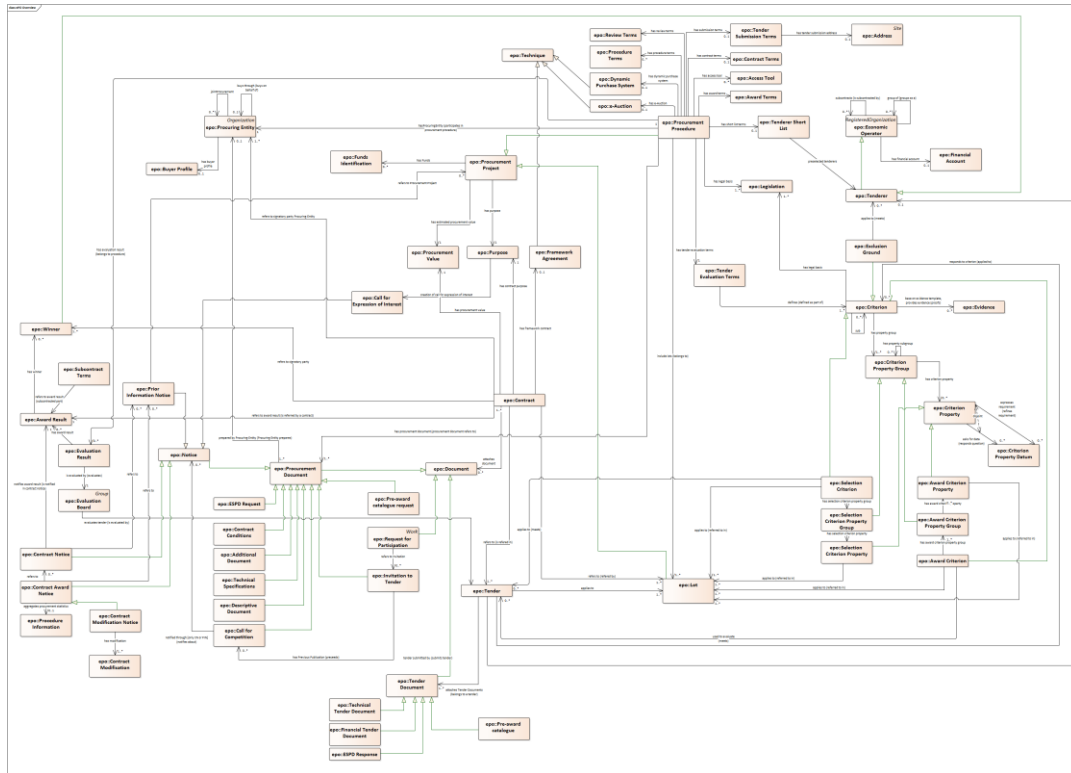


Figure 12. ePO Conceptual Data Model, overview

How to read the diagrams

The subsections below provide further details on key Classes of the ePO and about how these Classes relate to other Ontologies. A brief description for each model is provided in order to make it more understandable. The "legend" below should also facilitate the comprehension of the diagrams content.

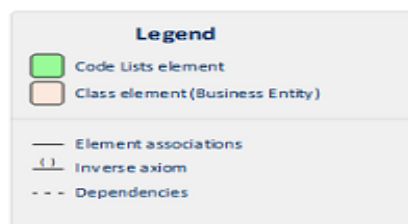


Figure 13. Legend

- Boxes in colour beige are Classes, i.e. main entities of the ontology, like "Procurement Procedure", "Procuring Entity", "Economic Operator", etc.;
- Boxes in colour green are "Code Lists", i.e. enumerations of disjoint concepts represented with a code);

- Classes associate other classes via "object properties", i.e. directed association arrows ("predicates", from the ontology perspective) that have a class at the origin (the subject of a triple, in the ontology) and another class at the end of the link (the "object" of the triple).
- "Data properties", i.e. links between the Class and more primitive/basic elements, are represented as attributes of the Classes, e.g. Text, Indicator, Date, etc.
- Associations between Classes are represented as unidirectional arrows to keep the diagrams simple. However, when the association is bi-directional it is indicated with two predicates and the second one is enclosed with parenthesis "()". In the OWL-TTL these are declared as "inverse" properties. Examples: "*Procurement Procedure includes lots (belongs to) Lots*", in the diagram, is to be read as:
 - "*Procurement Procedure includes one or more Lots*", and
 - "*One Lot belongs to one Procurement Procedure*";
- Classes may contain codes. In this representation, the ePO codes are not included inside the Class but are represented as associations of the Class to a specific enumeration element. The name of the code is built upon the verb "uses" and the name of the enumeration. Thus the triple used to say that a *Procurement Procedure is of type Open* is expressed like this in the OWL-TTL:
 - `:ProcurementProcedure :usesProcurementProcedureType epo-rd:ProcurementProcedureType`, where `:` is the default prefix representing the ePO ontology and `epo-rd:` is the prefix reserved for the namespace representing all the codes defined in ePO (eProcurement-specific, to be located in the OP's Metadata Registry ([MDR](#))).

See also the section "[Codes and Identifiers](#)", in chapter "[IV. Design and implementation](#)" for details on the implementation of these.

Procurement Project

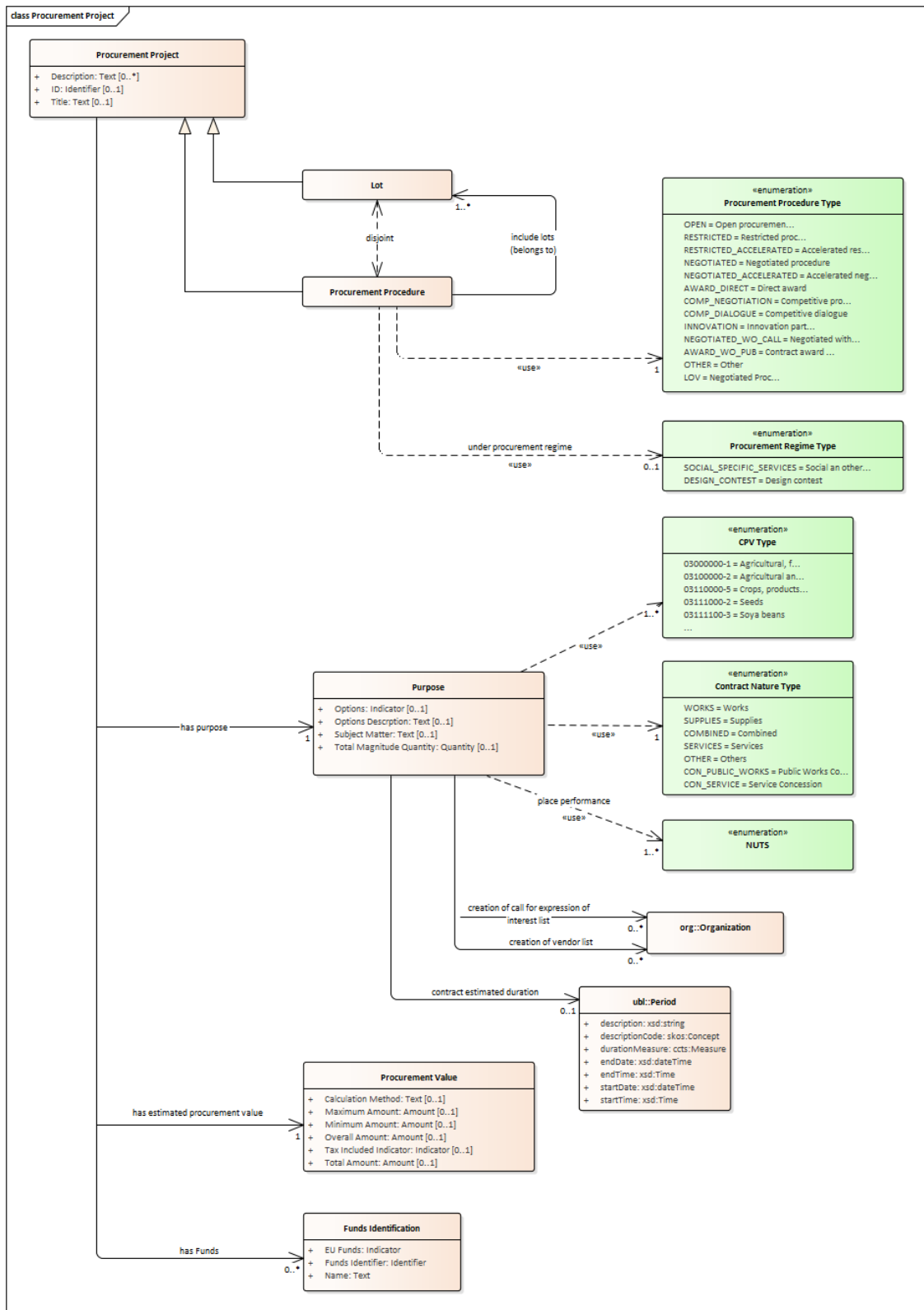


Figure 14. Procurement Project

1. Procurement is the acquisition by means of a public contract of works, supplies or services by one or more contracting authorities from economic operators chosen by those contracting authorities, whether or not the works, supplies or services are intended for a public purpose. (Directive 2014/24/EU, Article 1(2)).

- At its inception phase, the Procurement can be thought as a "Procurement Project".
- Procurement Projects are implemented through a Procurement Procedure or through the Lots of a Procurement Procedure.
- Procurement projects have a purpose which include aspects such as the subject-matter, the place of performance, contract nature, estimated duration, and other elements.
- The Procurement Project has an estimated value. These estimations are later on confirmed or finally established and reflected in the Contract and announced through the Contract Award Notice.
- The Procurement Project may use Techniques (see Technique Type).
- The Procurement Project may use Funds provided by the European Union.

Procurement Procedure

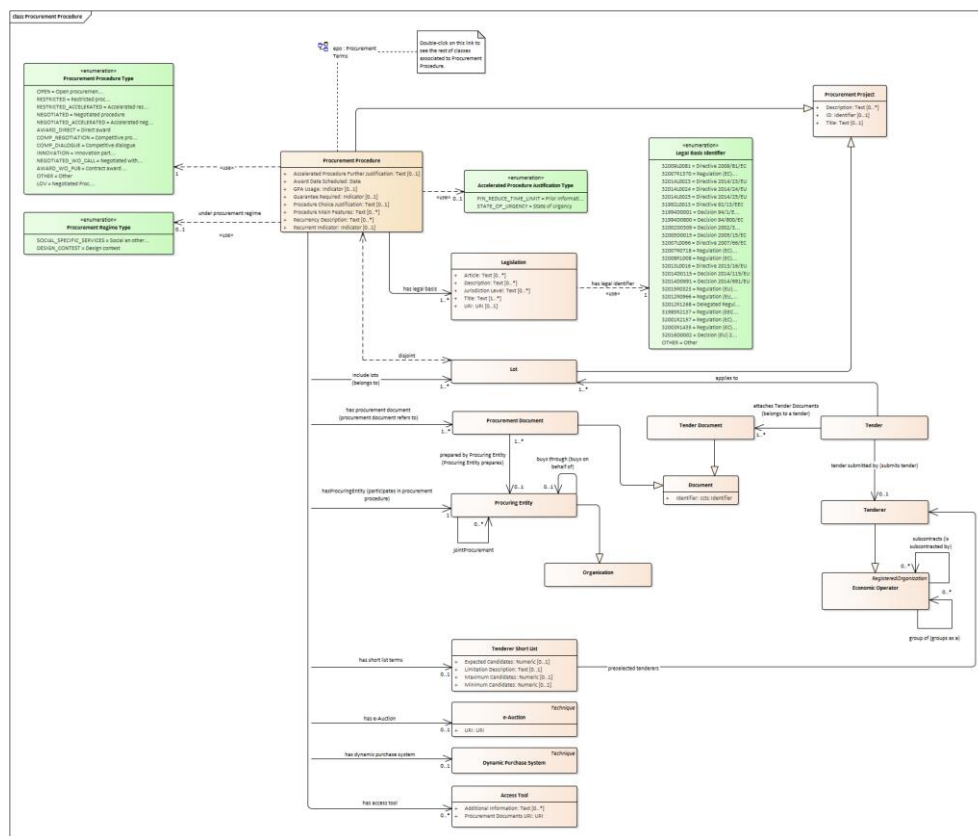


Figure 15. Procurement Procedure

- Procurement Procedures is a series of activities leading to the conclusion of a public contract.
- Pay attention to the fact that the Procuring Procedure is not directly linked to the Contract. Instead, this connection is made through the Procuring Entities involved in the Procedure. There are different reasons for this: e.g. if no Tenders are submitted

for a Procedure, no Contract is issued, which also entails that the link could not be established through the Tender. This also explains why Economic Operator is not directly related to the Procurement Procedure.

3. Different types of Procurement Procedures are carried out according to the EU Legislation (see Procurement Procedure Type).
4. Some Procurement Procedures apply specific legal regimes and instruments for the awarding of certain services or the acquisition of designs (see Procurement Regime Type).
5. Procurement Procedures are divided in one or more Lots (see diagram Lots).
6. Procurement Procedures usually generate, collect or refer to different documents. Two of the most relevant groups of documents are represented by the classes Procurement Document and Tender Document (see diagram Documents).
7. All Procurement Procedures are conducted by at least one Procuring Entity, in some cases Procuring Entities carry out join procurement (see diagram Procuring Entity).
8. Procurement Procedures may need to refer to certain types of organisations responsible for the management or control of a number of aspects of the procedure, e.g. environmental party, tax party.
9. In some types of Procurement Procedures (e.g. restricted, competitive with negotiation, other), Procuring Entities may limit the number of candidates accessing the award criteria phase. When this is the case, certain information must be notified by the Procuring Entity, e.g. expected maximum and minimum number of candidates, justification / description of the limitation, etc. (Tender Short List).

Accelerated Procedure

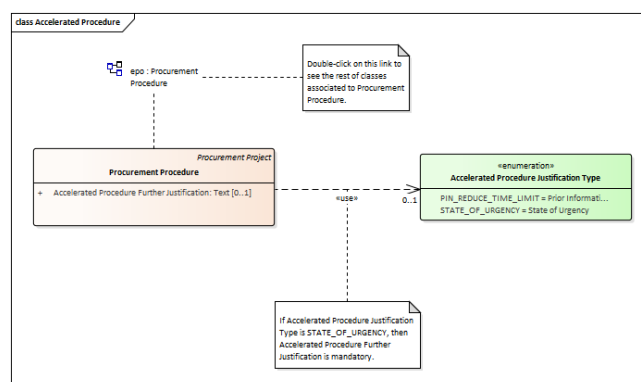


Figure 16. Accelerated Procedure

1. An accelerated procedure takes place when the time limits within the procedure are reduced.

2. Time limits can be reduced due to as state of urgency (Accelerated Procedure Justification Type) in which case a justification must be provided (Accelerated Procedure Further Justification).
3. They can also be reduced by a Prior Information Notice (PIN) published specifically for reducing the time limits.
4. For example see Directive 2014/24/EU Article 27(3) and 28(6).

Procurement Terms

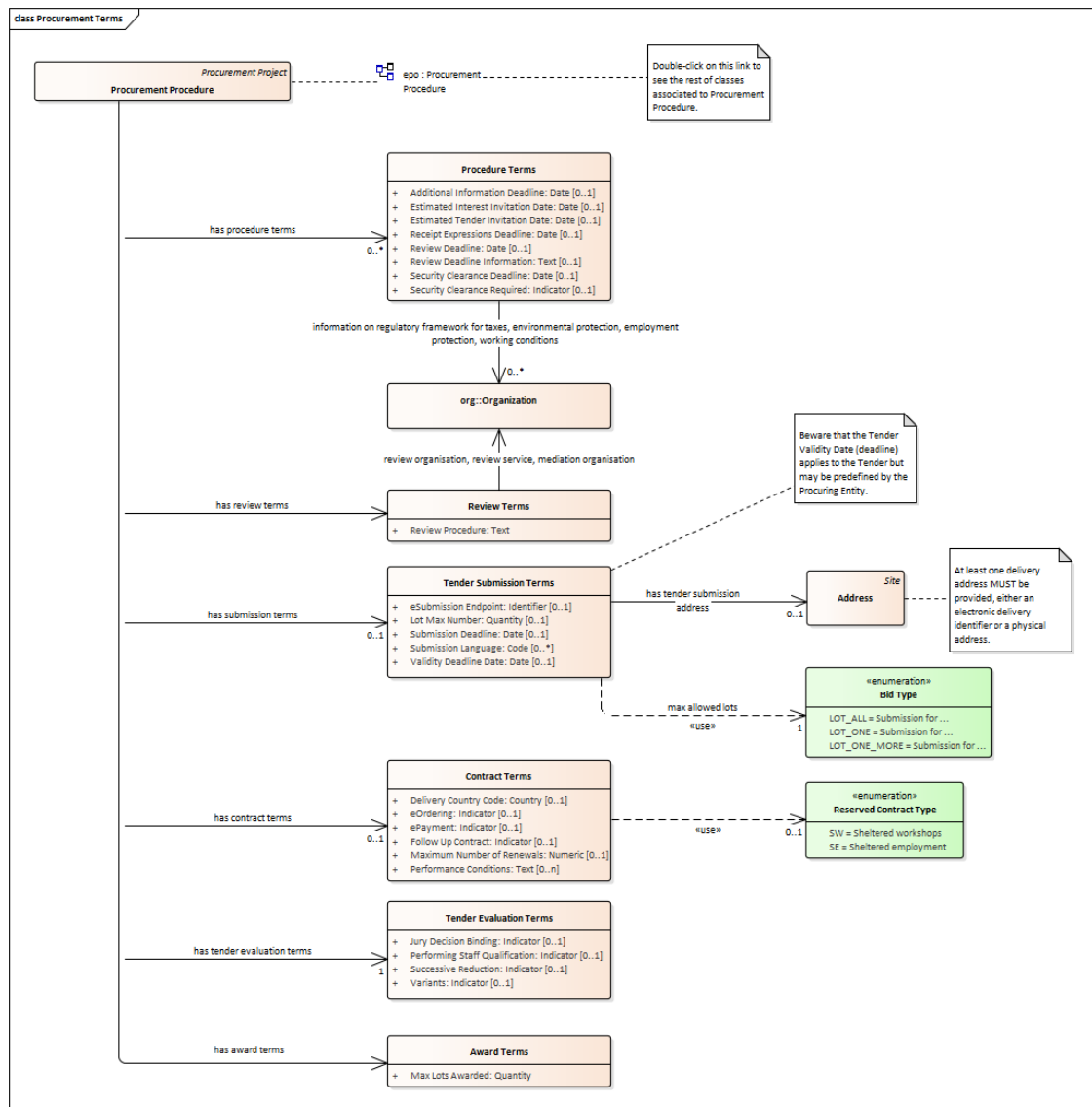


Figure 17. Procurement Terms

1. The Procurement Terms are "conditions or stipulations established by the Procuring Entity":
 - a. Procedure Terms: conditions and stipulations determining how the procurement procedure is executed.

- b. Review Terms: conditions and stipulations about the information and organisation responsible for the revision of a Procurement Procedure.
- c. Tender Submission Terms: conditions and stipulations about the Tender and its submission.
- d. Contract Terms: conditions and stipulations related to the implementation of the contract.
- e. Tender Evaluation Terms: conditions and stipulations to evaluate the tenders.
- f. Award Terms: conditions and stipulations to determine how the procurement procedure is awarded.

Lots

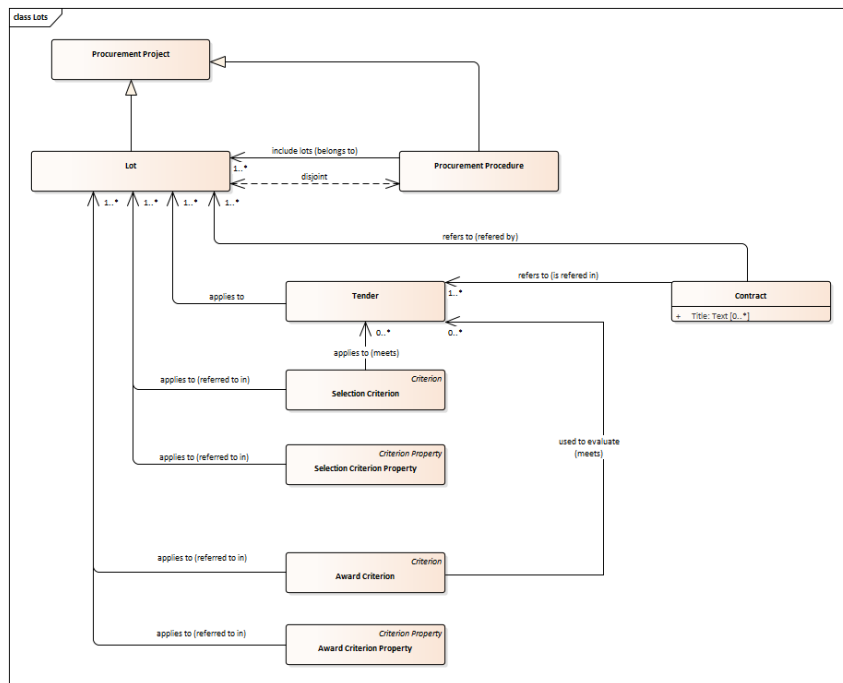


Figure 18. Lots

1. A Lot is one of the parts into which a Procurement Procedure is divided.
2. One or more lots may aim at one or more Contract.
3. When preparing the Procurement Projects, Lots may be grouped.
4. Tenderers prepare their Tender for one or more Lots.
5. The Procuring Entity apply Selection and Award Criteria to one or more Lots or Group of Lots.

Technique

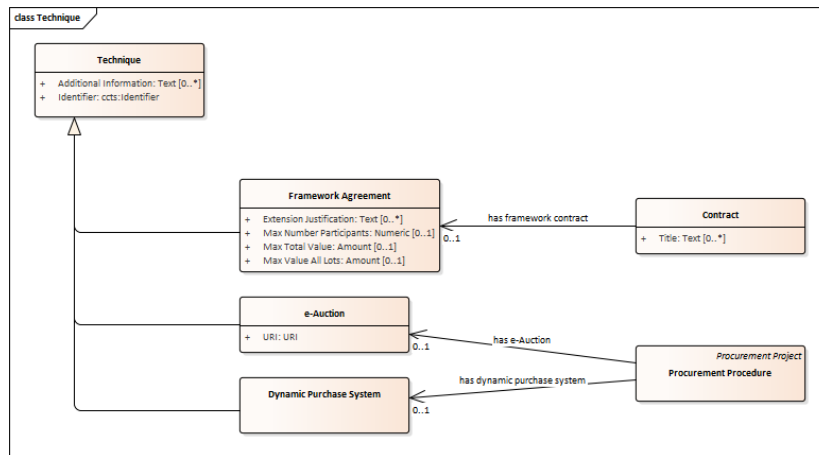


Figure 19. Technique

1. Techniques are specific methods of carrying out the procurement or a purchase. E.g. Framework Agreement, e-Auction or Dynamic Purchase System.
2. Each Technique has its own properties, thus Framework Agreement can be typified, has a duration, its own values, etc.

Procuring Entity

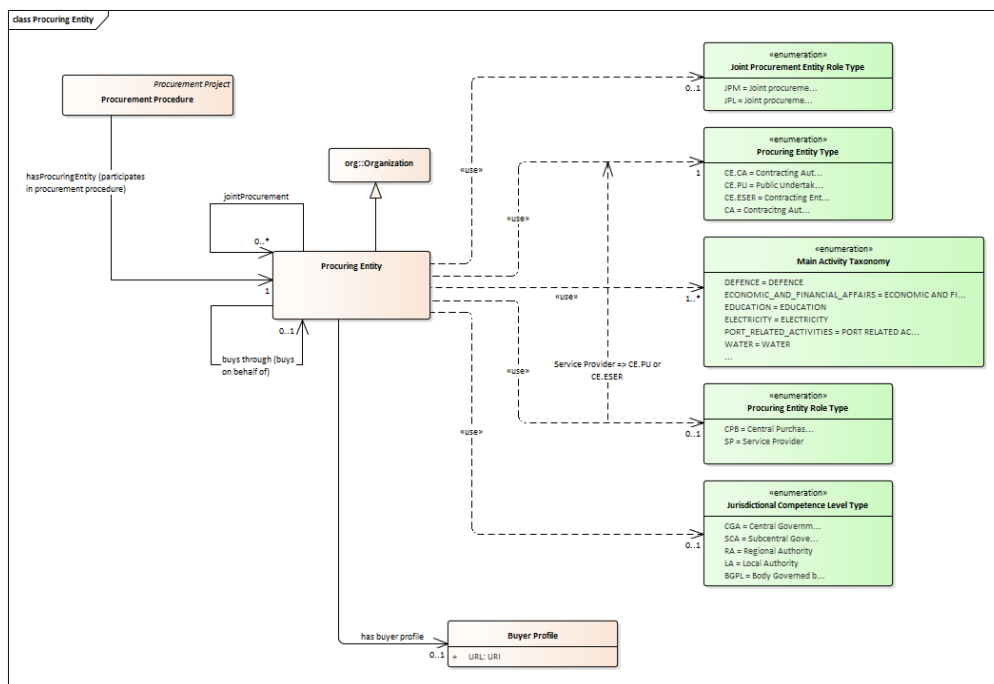


Figure 20. Procuring Entity

1. In any Procurement Procedure, there is at least one Procuring Entity;
2. Procuring Entities are “Organizations”, appropriately identified and described (IDs, Names, Addresses, Contact Points, etc.);
3. Depending on its nature and main activity a Procuring Entity may be identified simply as a Contracting Authority (general procurement) or as a Contracting Entity

pursuing the procurement of gas and heat, electricity, water, transport services, ports and airports, postal services and extraction of oil and gas and exploration for, or extraction of, coal or other solid fuels. A Contracting Entity may in turn be a Contracting Authority, a Public Undertaking or entities with special or exclusive rights (Procuring Entity Type code list);

4. For some Procurement Procedures, a Procuring Entity can join other Procuring Entities (Joint Procurement)
5. In these cases, the Procuring Entities participating in the Joint Procurement adopt one role (Procuring Entity Role Type code list), e.g. the lead of the group.
6. Procuring Entities are in general responsible for the both the management of the procurement procedure and the purchase. However in some cases procuring entities may buy on behalf of other procuring entities or through other procuring entities ("Procuring Entity Role Type").

Economic Operator

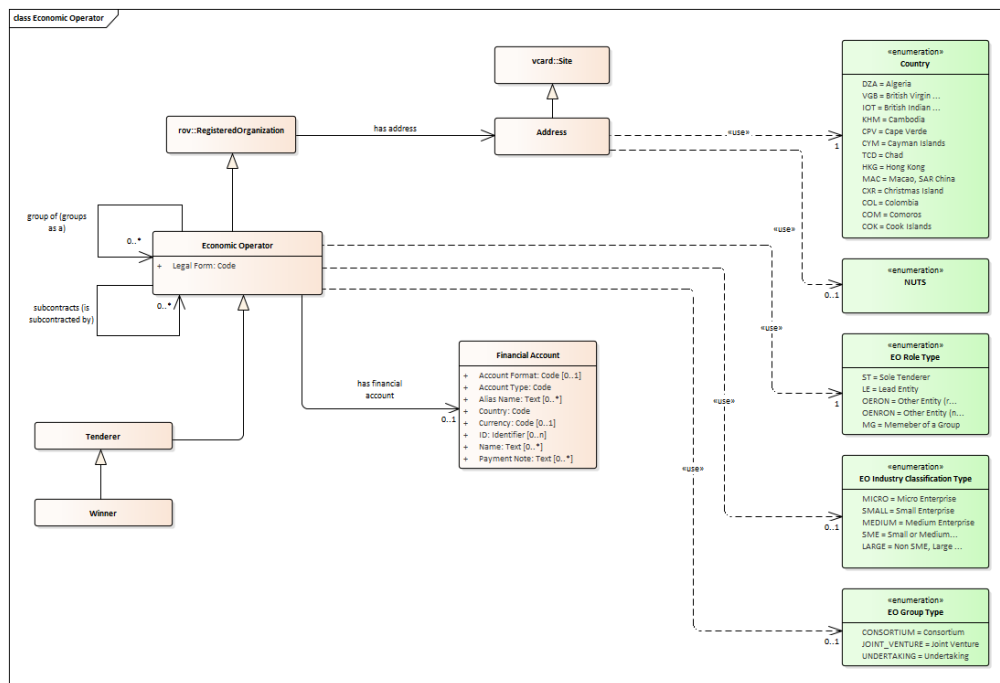


Figure 21. Economic Operator

1. An Economic Operator is an organisation.
2. Economic Operators can be Tenderers (the submitter of the Tender) or sub-contractors.
3. When the Economic Operators are members of a group (e.g. Consortia, Joint ventures, Undertaking (EO Group Type)), and they play different roles, e.g. group lead entity, member of the group, etc. (EO Role Type).
4. The Winner of a contract is a tenderer or group of Tenderers.

5. Tenderers may rely on other Economic Operator that are subcontractors but not tenderers.
6. When guarantees are required by the Procuring Entity, Economic Operators may have to provide Financial Account details (e.g. a bank account data).

Contract

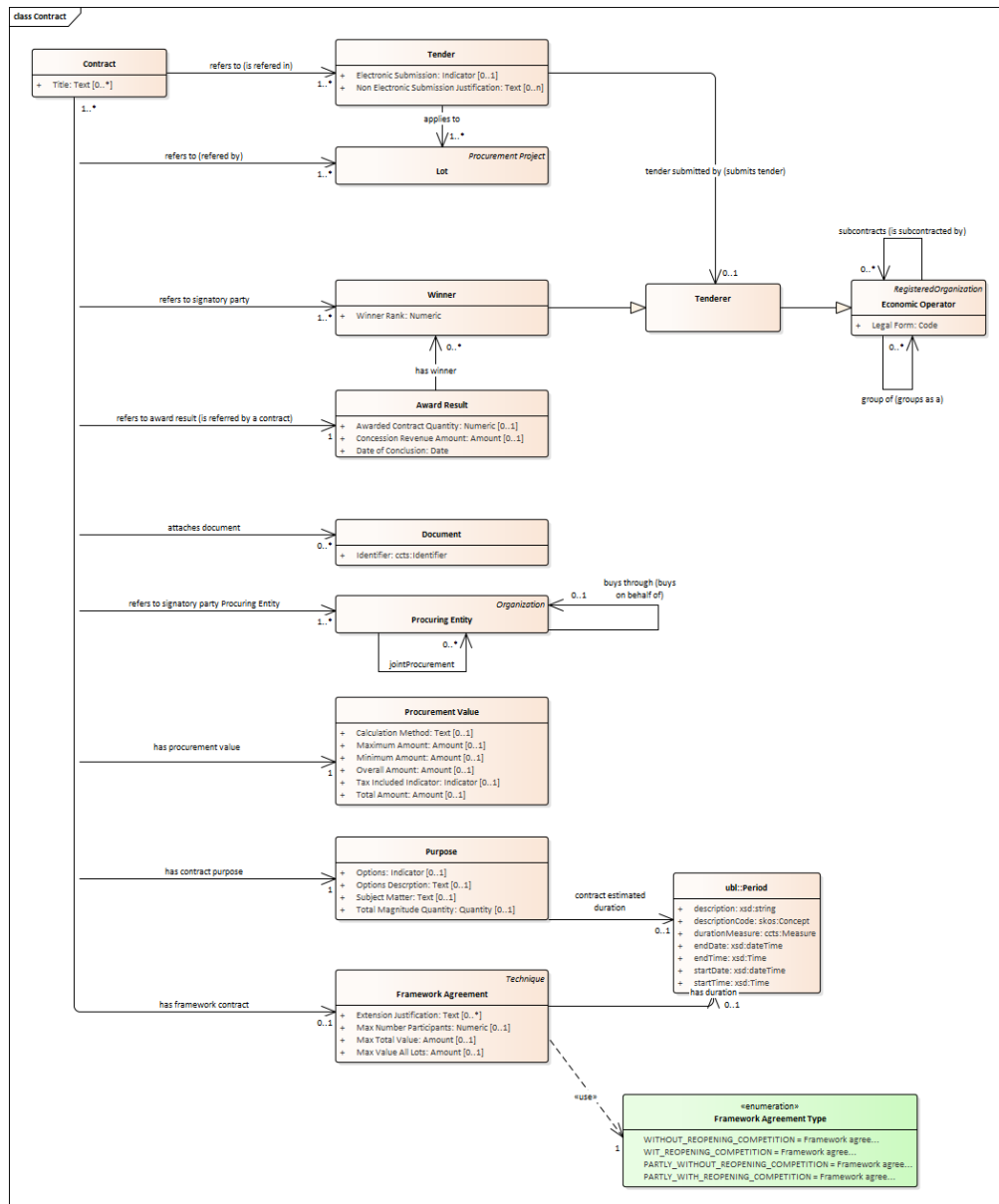


Figure 22. Contract

1. One of the activities that takes place in the Procurement Procedure life-cycle is the evaluation of Tenderers and Tenders, and the awarding of a contract to one or more Tenderer. The awarded Tenderer(s) are the "Winner(s)".
2. The Contract may attach other Procurement Documents and other types of Documents.

3. The object of the Contract and additional data that were stated in the Procurement Project are also placed in the contract Purpose (e.g. Subject Matter, Place of Performance, Total Magnitude Quantity, etc.).
4. Similarly, the values of the Procurement that were initially estimated in the Procurement Project are set in the Procurement Value class.
5. The Contract reflects also the Awarding Results (resulting from the evaluation) and the signatory parties (Procuring Entities and Winners).
6. In case the Procurement Procedure uses Framework Agreement as Technique, the contract refers to it.

Tender

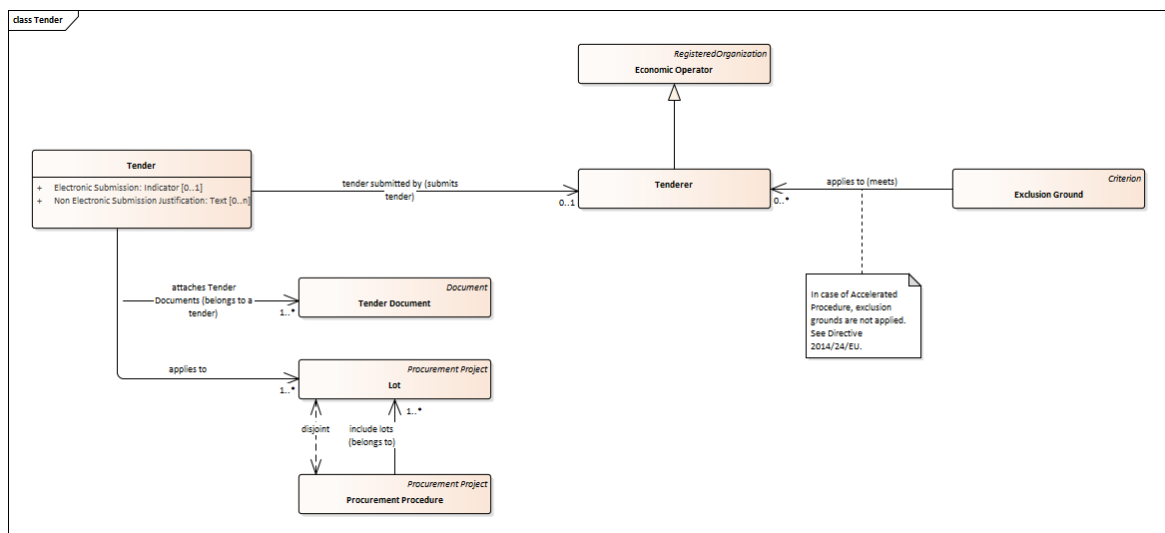


Figure 23. Tender

1. Tenders are submitted by Tenderers, who are Economic Operators.
2. One Tender may attach one or more "Tender Documents" (e.g. the Financial Tender, the Technical Tender, Technical annexes and specifications, etc.; see the Diagram "Documents");
3. In Procurement Procedures divided into Lots, one Economic Operator submits one Tender. The tender specifies to which Lots it applies.
4. Procurement Procedures are always considered to have at least one lot.

Evaluation Result

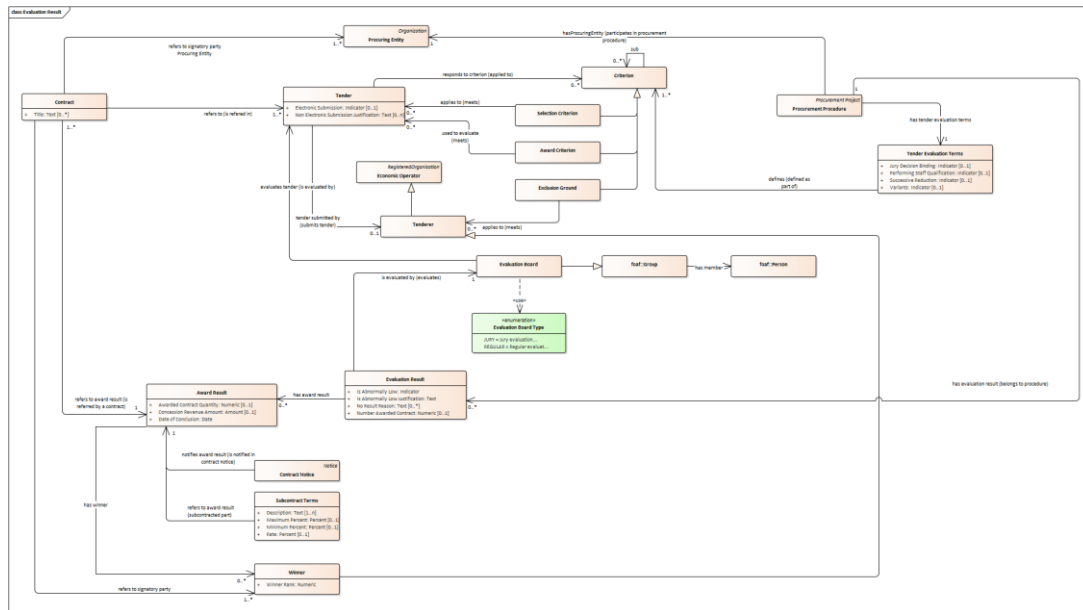


Figure 24. Evaluation Result

1. The Evaluation Result is presented in the form of a report showing the assessment of the tenders by the evaluation board.
2. The Evaluation board takes into consideration the Criterion and the Tender Evaluation Terms when assessing the tenders.
3. The awards result takes into consideration the evaluation result and awards the contract.
4. In the case of contest design competitions, the board is formed by a Jury, whose decision may be binding for the Procuring Entity (see Evaluation Board Type).

Contract

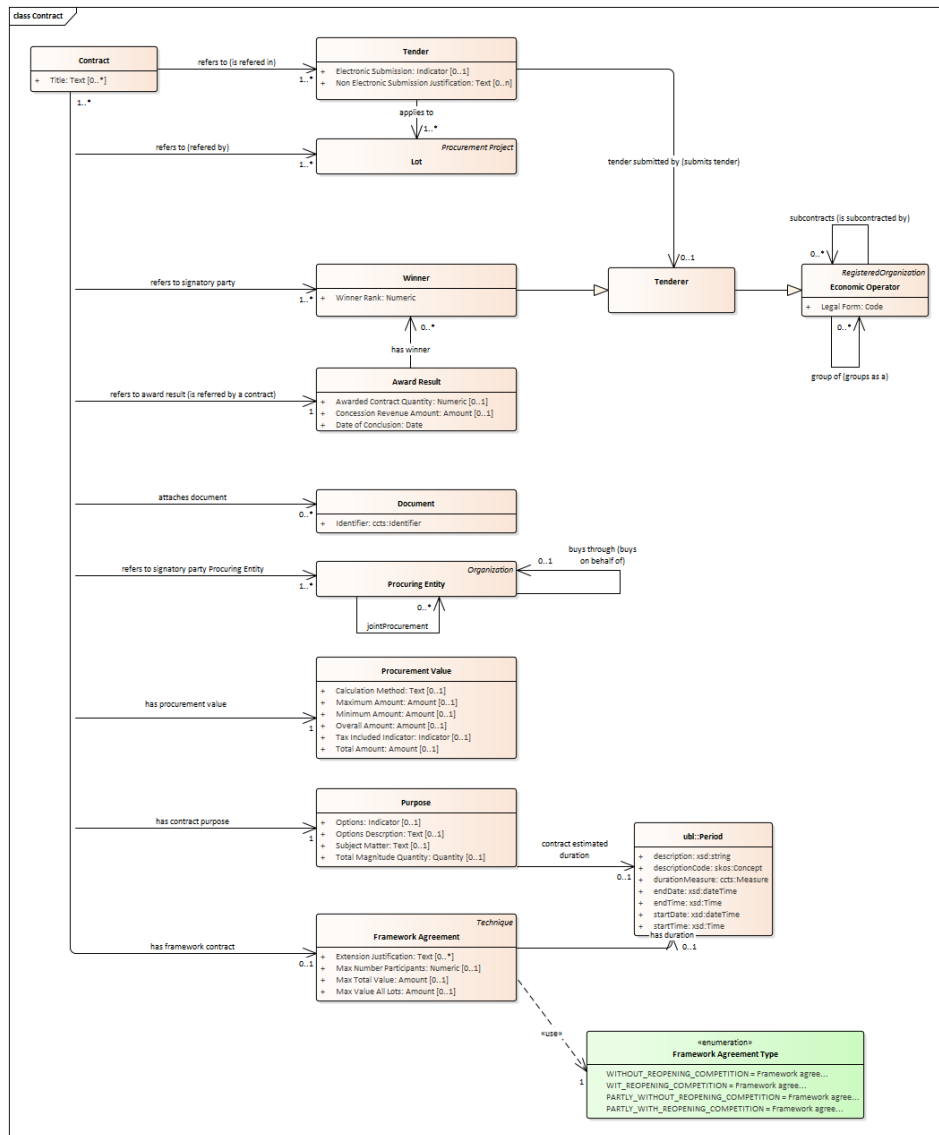


Figure 25. Contract

1. One of the activities that takes place in the Procurement Procedure life-cycle is the evaluation of Tenderers and Tenders, and the awarding of a contract to one or more Tenderer. The awarded Tenderer(s) are the "Winner(s)".
2. The Contract may attach other Procurement Documents and other types of Documents.
3. The object of the Contract and additional data that where stated in the Procurement Project are also placed in the contract Purpose (e.g. Subject Matter, Place of Performance, Total Magnitude Quantity, etc.).
4. Similarly, the values of the Procurement that where initially estimated in the Procurement Project are set in the Procurement Value class.
5. The Contract reflects also the Awarding Results (resulting from the evaluation) and the signatory parties (Procuring Entities and Winners).

- In case the Procurement Procedure uses Framework Agreement as Technique, the contract refers to it.

Criterion

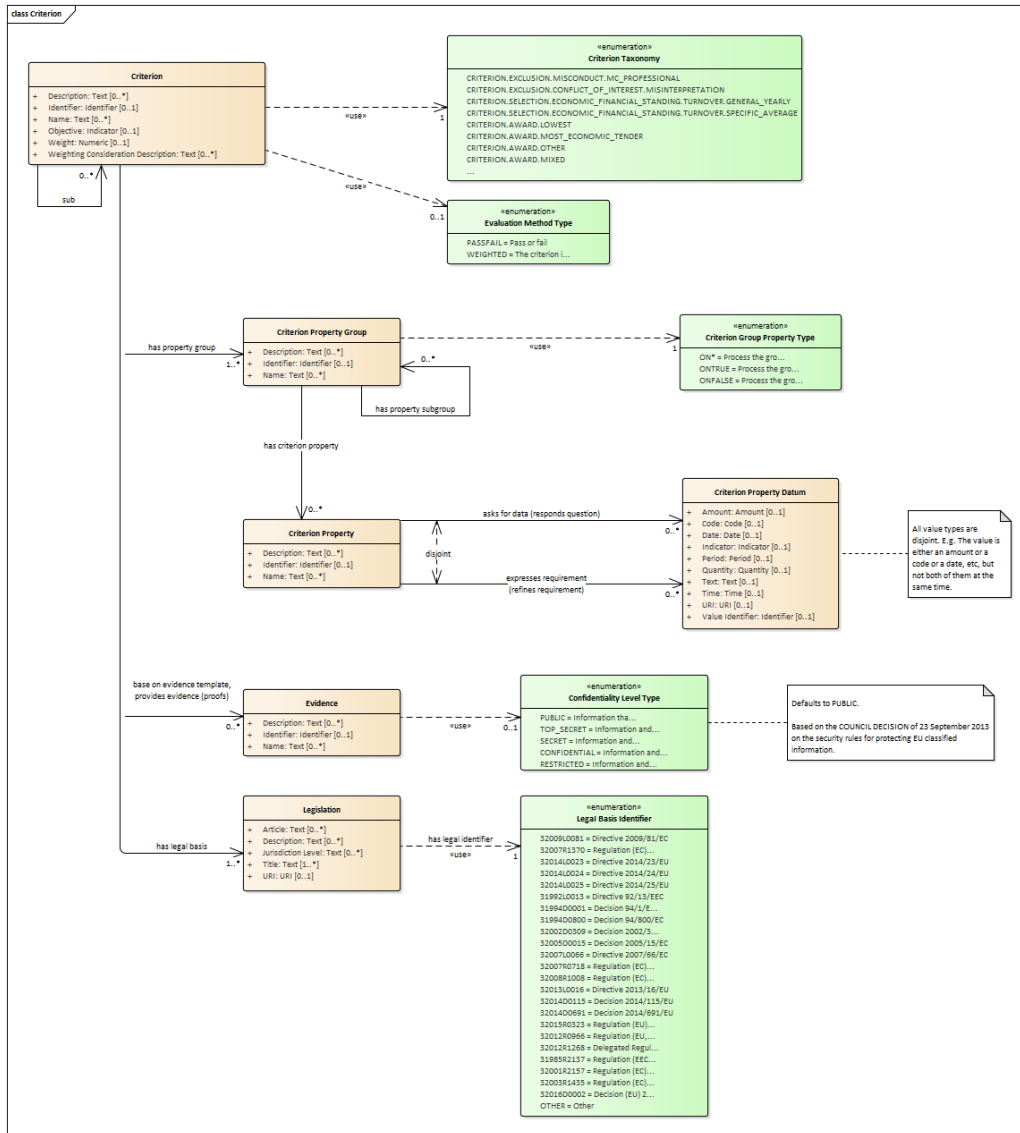


Figure 26. Criterion

- Criterion is a generic business-agnostic class. This eProcurement ontology (ePO) uses this as a base class to extend Award Criterion, Exclusion Grounds and Selection Criterion (see the rest of diagrams about criteria for details).
- A Criterion is a condition that needs to be answered for evaluation purposes. For example: General average turnover for the past three years.
- All Criteria are codified via a Criteria Taxonomy. Thus, the examples above have an associated code as exclusion, selection and award criteria (see Criteria Taxonomy). Exclusion, Selection and Award criteria do extend the classes and properties of Criterion.

4. In general, Criteria are evaluated using a pass/fail method, meaning that the Tenderer or the Tender meet or do not meet the Criterion. However, selection and award criteria may be weighted (see Evaluation Method Type).
5. A Criterion may contain sub-criteria. Thus, the exclusion criteria defined in the European Directives may be further detailed in national sub-criteria, e.g. national professional misconduct-related criteria.
6. The condition described in a Criterion may be broken down into simpler elements named "Criterion Property", which are always grouped into Criterion Property Groups.
7. A Criterion Property is a more specific information needed to measure a criterion. It is a question that usually goes hand in hand with a specific requirement. For example which follows on from the example given for criterion: Question: Amount?
Requirement: The text explaining what the procuring entity is interested in measuring i.e. minimum turnover.
8. Criterion Property Groups are organised structures or related criterion properties. Following on from the example of Criterion property. In the case of a yearly general turnover that needs to specify three turnovers for three specific years, a group of properties would be: turnover 1987, turnover 1988, turnover 1989.
9. One criterion property is normally associated to a value (Criterion Property Datum). The value may be an economic amount, a text, a date or a period, etc.
10. The responses to one Criterion may be supported by one or more evidences (property "provides evidence"). This evidence might have to be based on a template specified by the Procuring Entity (property "base on evidence template"). The fact that one individual of an evidence is linked to one Criterion does not preclude the possibility of linking this same individual (or instance) to other Criteria.
11. In the domain of public procurement, exclusion grounds, selection criteria and award criteria are normally based on a specific legal framework (see class Legislation).

Award Criterion

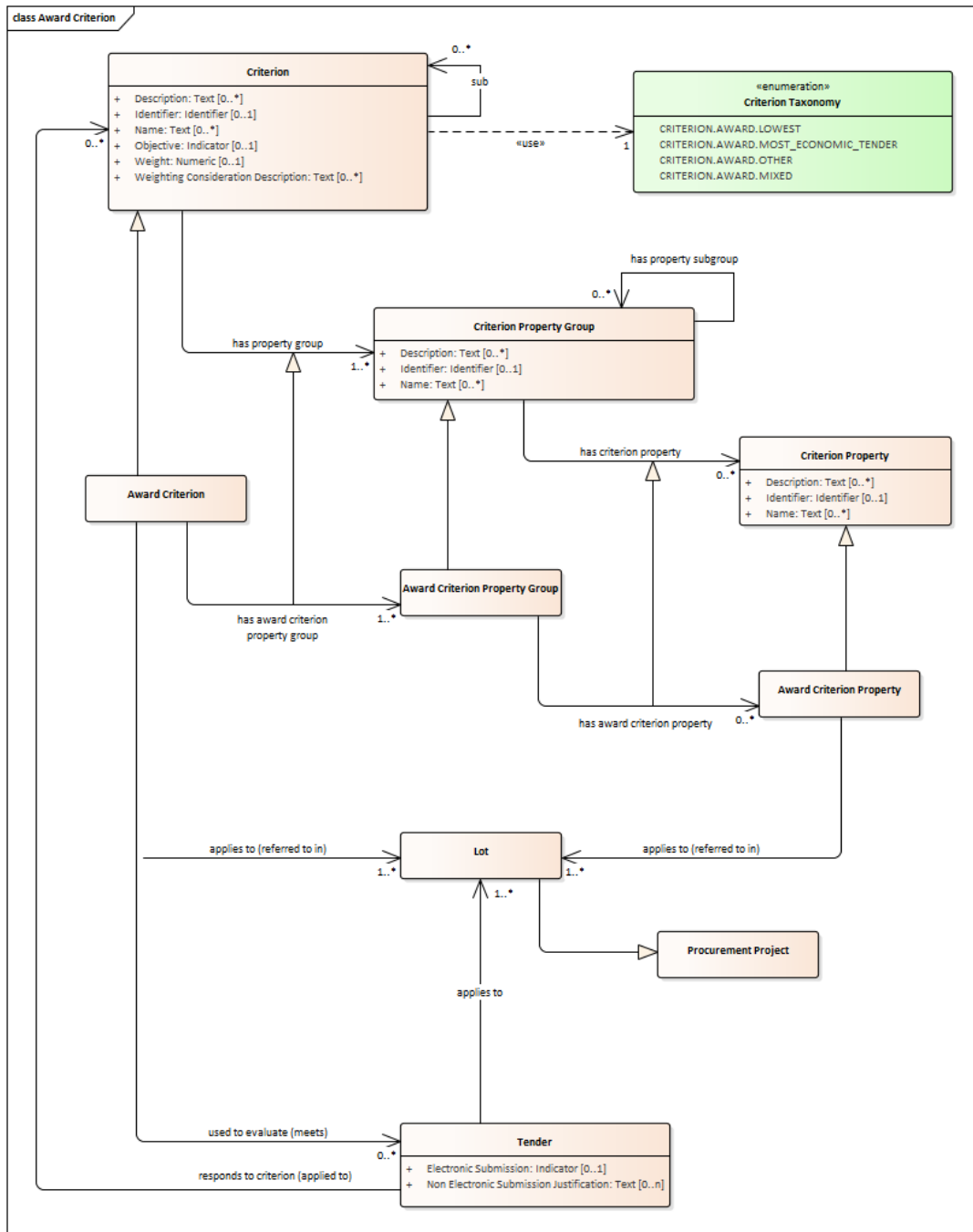


Figure 27. Award Criterion

1. Award Criteria are used to evaluate Tenders. They may include the best price-quality ratio, including qualitative, environmental and/or social aspects, linked to the subject-matter of the public contract in question.
2. Thus, an Award Criterion needs to be codified as lowest, most economic tender, mixed or other (for non-objective / qualitative criteria - see Criteria Taxonomy).
3. In two-phase procedures technical and financial criteria, used in the first phase for the selection, can be reused as weighted criteria to evaluate the Tenders.

4. Award Criterion is a class that specialises Criterion. The specialisation consists in providing a property to link the Criterion to Lot.
5. Award Criterion and Award Criterion Property, both need to link to Lot.
6. This is why the class Award Criterion needs to provide specialised sub-classes for the Criterion Property Group and Criterion Property, as well as the properties linking them.

Exclusion Grounds

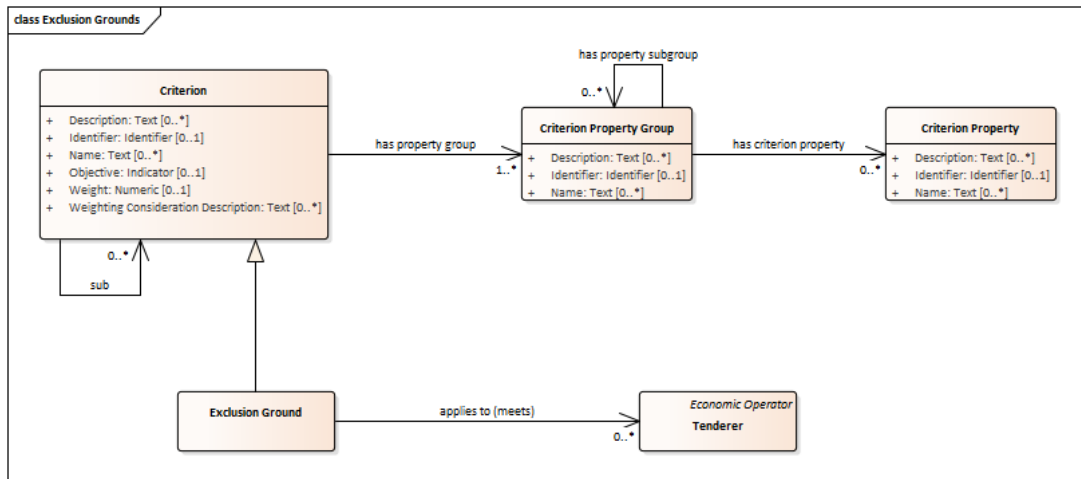


Figure 28. Exclusion Grounds

1. Tenderers may be excluded from participate in a Procurement Procedure, in case they bridge any of the legal criteria established in the Directives. This criteria are named Exclusion Grounds.
2. Exclusion Ground extends the generic Criterion class by adding a new property ("applies to") to refer to the Tenderers that are excluded in a procedure.
3. The ePO allows to determine the exact Exclusion Grounds were not met by the Tenderer for specific Procurement Procedure. To see how the Tenderer related to Procurement Procedure, please see the diagram "Evaluation Result".

Selection Criterion

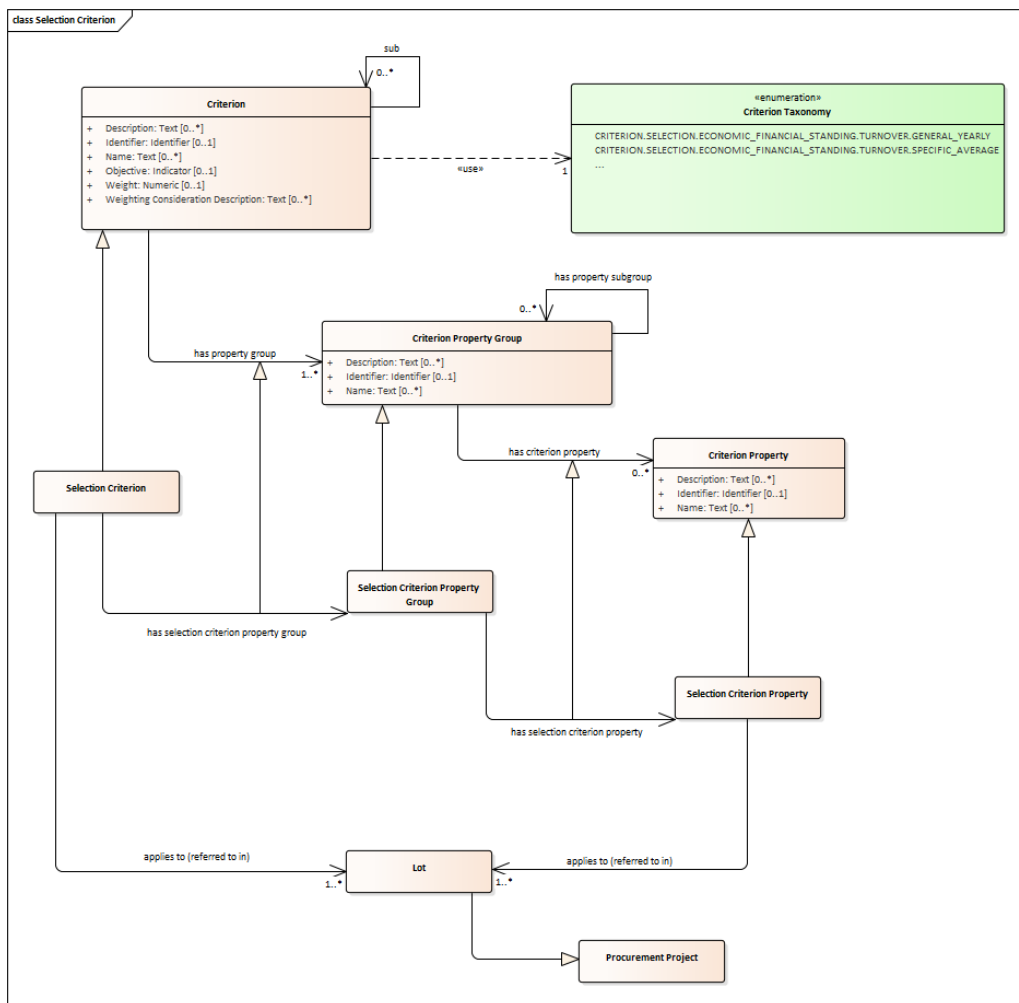


Figure 29. Selection Criterion

1. Selection Criteria aim at ensuring that a candidate or tenderer has the legal and financial capacities and the technical and professional abilities to perform the contract to be awarded (see ePO Glossary for the difference between Candidate and Tenderer).
2. Thus, a Selection Criterion is to be classified using the Criteria Taxonomy (e.g. CRITERION.SELECTION.ECONOMIC_FINANCIAL_STANDING.TURNOVER.GENERAL_YEARLY, CRITERION.SELECTION.ECONOMIC_FINANCIAL_STANDING.TURNOVER.SPECIFIC_AVERAGE, etc.).
3. Selection Criterion is a class that specialises Criterion. The specialisation consists in providing a property to link the Criterion to Lot.
4. Selection Criterion and Selection Criterion Property, both need to link to Lot.

5. This is why the class Selection Criterion needs to provide specialised sub-classes for the Criterion Property Group and Criterion Property, as well as the properties linking them.

Documents

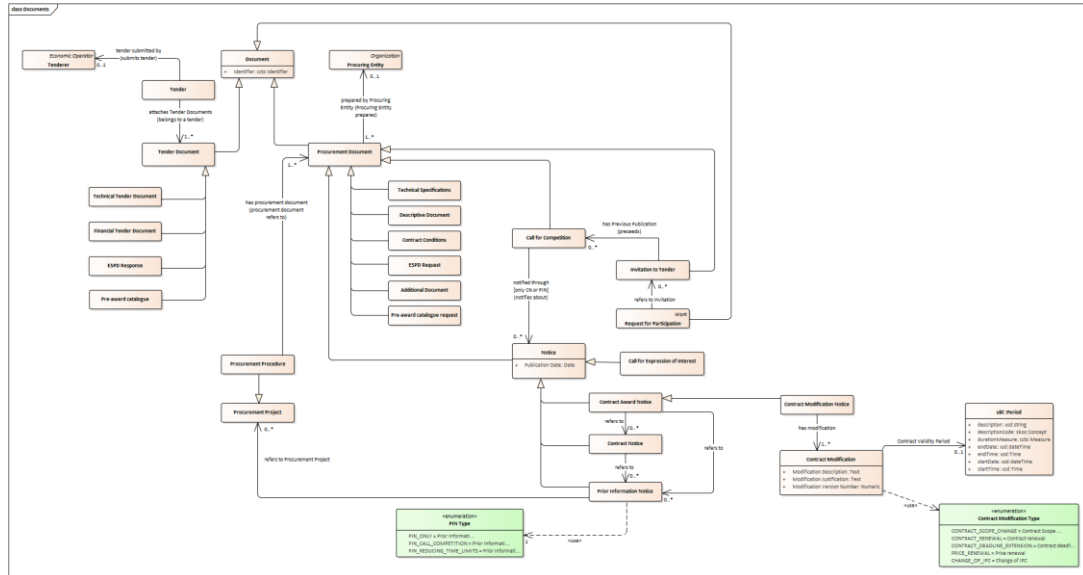


Figure 30. Documents

1. The ePO sees Documents as aggregators of the business domain data. In other words, the content of a Document are individuals that exist in the data graphs. A such (aggregators of individuals) they are ideal artifacts for the interoperability.
2. In the scope of the e-Notification and e-Access time, we can identify "Procurement Documents", whilst during the e-Submission, the Tenderer prepares and sends "Tender Documents".
3. Procurement Documents are prepared by the Procuring Entity and are always particular to a Procurement Procedure.
4. Several groups of Notices can be distinguished: Prior Information Notice, Contract Notice, Contract Award Notice and Call for Expression of Interest.
5. Prior Information Notices are often drafted prior to the existence of the Procurement Procedure and in some cases may refer to more than one Procurement Procedure.
6. Prior Information Notices (PIN) announce Procurement Projects.
7. Contract Notices (CN) announce the initiation of Procurement Procedures as do certain PINs. If the CN follows a PIN previously published, the CN should refer to that PIN.
8. Contract Award Notices (CAN) in turn announce the award of a Contract(s). In the case that a CN has been published prior to the CAN the CN should be referenced in

the CAN. In the case where neither a PIN or CAN have been published prior to the CAN then a justification should be provided.

9. In restricted procedures the need of limiting the number of candidates to a short list may appear and for these cases Invitations to Tender are forward to each one of the candidates. Candidates interested in participating may submit a Request for Participation. The Invitation to Tender may refer to the Notices previously published in the context of the Procurement Procedure.
10. At tendering time, the Tenderer submits its own Tender Documents, which normally encompass a Financial Tender and a Technical Tender among other possible annexes and additional documents.
11. Contracts can experience minor modifications (Contract Modification), otherwise they may carry out new Procurement Procedures. Each modification has to be duly identified (see Contract Modification Type) and justified. These Modifications are to be published via Contract Modification Notices.

Contract Award Notice

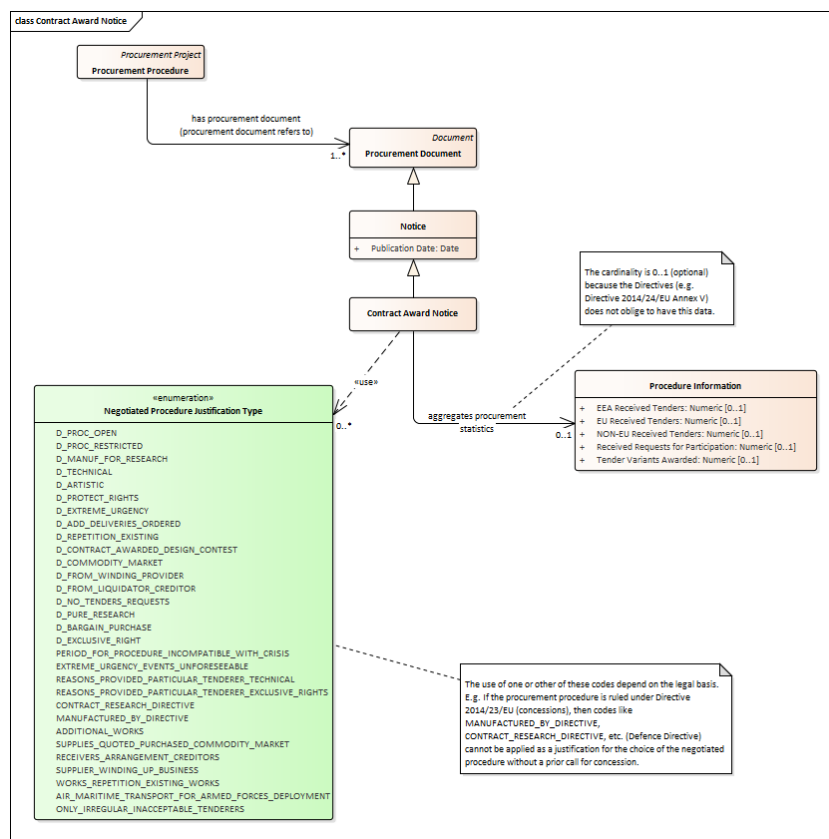


Figure 31. Contract Award Notice

1. Procuring Entities shall publish the award of a contract by means of Contract Award Notices.

- In the case of negotiated procedures without prior publication of a call for competition or for concession, a justification must be provided (Negotiated Procedure Justification Type)

Data Types

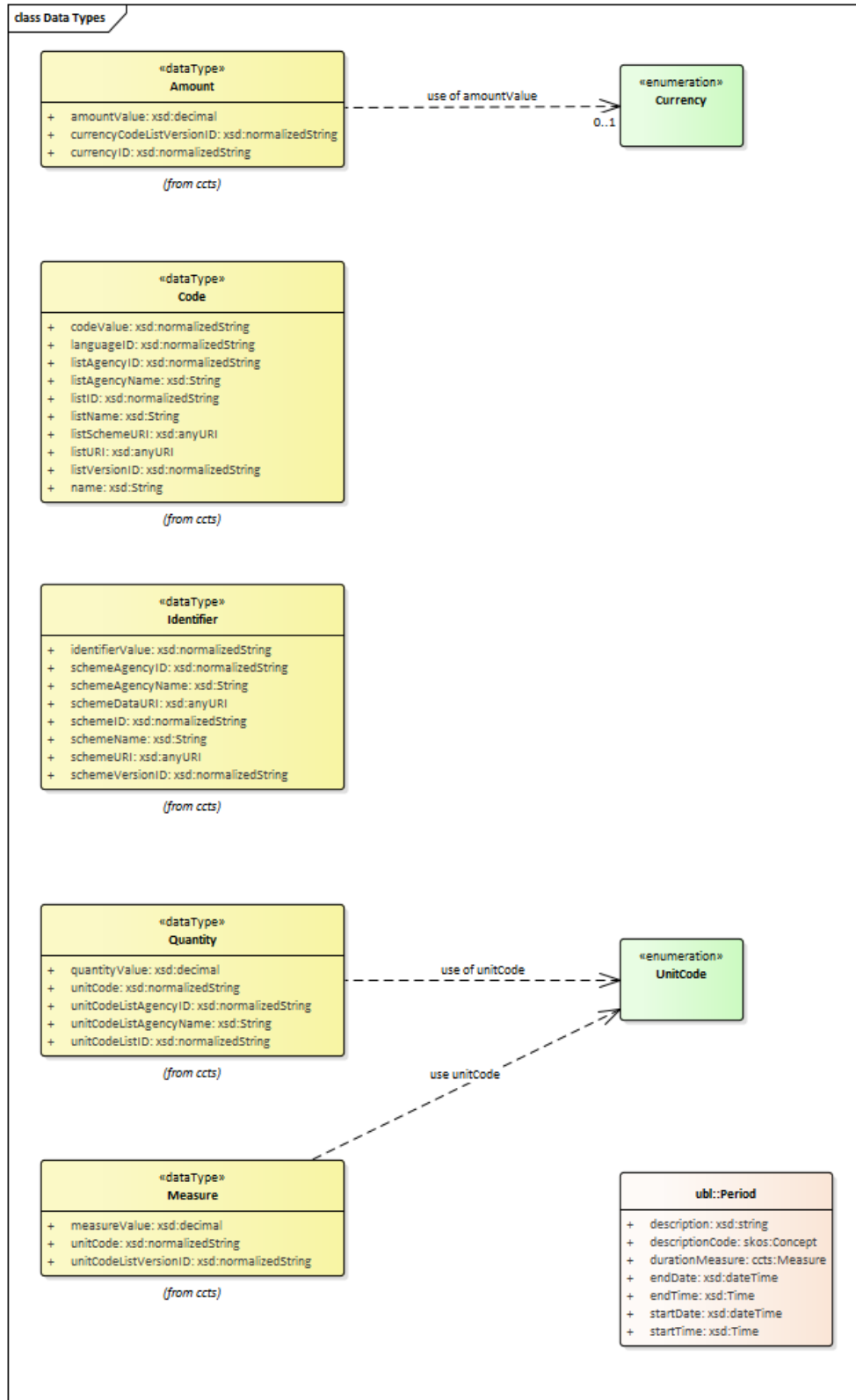


Figure 32. Data Types

The Conceptual Data Model (CM) represents "data properties" (as understood from the ontology perspective) as "class attributes" (as normally represented in UML diagrams). For the representation of literals and other attributes, the CM uses the CCT notation (Text, Numeric, Indicator, Amount, etc.).

Beware, however that this ontology works with two types of data properties, those that can be considered truly "primitive" (like Text, Numeric, Indicator, Date) and those that have additional dimensions (attributes) like Identifier, Amount, Quantity, Measure and Code).

This ePO implementation "primitive" ones as xsd types, string for Text, dateTime for Date and Time, boolean for Indicator, decimal for Numeric, and so on. The rest of complex data types are implemented as classes with their own data properties, including a placeholder for the value (the actual datum). See section [IV. Design and Implementation](#) for details on the Turtle (TTL) implementation.

IV. Design and Implementation

All deliverables produced in the previous tasks, e.g. Glossary and DED, but namely the Conceptual Data Model, were taken into account to produce an OWL ontology.

The outcome of this task are mainly the expression of the ontology [T-Box](#) as an OWL-DL Turtle syntax and a comparison of the tools used ([Protégé 5.2](#) and [VocBench 3.0](#)) for the development of the T-Box.

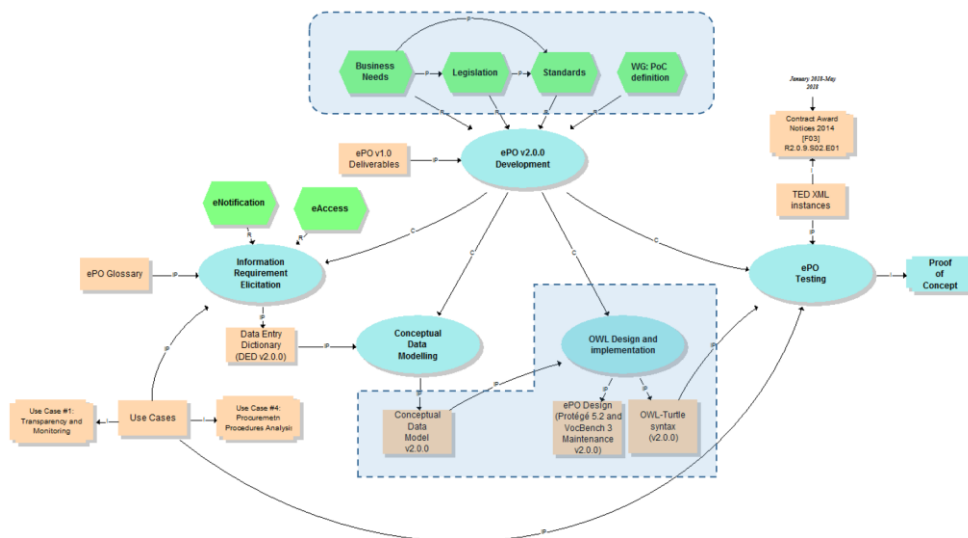


Figure 33. ePO Design and Implementation

ePo 1.0 RDF-XML vocabulary

In ePO v1.0 the Working Group decided that the ontology was to be expressed as an RDF vocabulary. This vocabulary (without axioms defined therein) was expressed as an OWL-XML syntax: see file [eproc_v0.6.owl](#) located in the [root folder](#) of the GitHub code repository.

ePO v2.0.0 OWL-DL syntax

For this new version ePO v.2.0.0 the ePO development team proposed to the WG to approach the development of the ontology with the more expressive language [OWL DL](#) (Description Logic), which allows for advanced reasoning and logic inference, and the [Turtle](#) (TTL) syntax, as it is more human-readable than the OWL-XML equivalent.

Automated generation of the ePO-TTL T-Box

One way of automatising the generation of the OWL-TTL T-Box is to use the DED jointly with a transformation process and artefacts (e.g. XSL-T stylesheets for the conversion of the spreadsheet into TTL). This can be used for the generation of a first draft version that needs to be improved manually, e.g. using Protégé, VocBench or a simple txt editor. The production of such transformation and artefacts were out of the scope of this phase.

As commented above, for the drafting of the TTL syntax the ePO development team used the Stanford's [Protégé 5.2](#) editor. The resulting OWL-TTL file can be accessed from the GitHub [Wiki](#) page (menu [OWL-TTL](#)) or from the [link:https://github.com/eprocurementontology/eprocurementontology/tree/master/v2.0.0/05_Implementation/ttl](https://github.com/eprocurementontology/eprocurementontology/tree/master/v2.0.0/05_Implementation/ttl) repository.

While developing and evolving the OWL-TTL each new version was also loaded and reviewed using the latest version of VocBench 3 (VB3). The objective of this exercise was to check the feasibility of using VB to maintain the ePO.

Class hierarchy: 'Contract Award Notice'



- owl:Thing
 - AcceleratedProcedureJustificationType
 - AcceleratedProcedureJustificationType
 - 'Access Tool'
 - AccountFormatType
 - AccountType
 - Amount
 - 'Award result'
 - 'Award Terms'
 - BidType
 - 'Buyer Profile'
 - Code
 - ConfidentialityLevelType
 - Contract
 - 'Contract modification'
 - 'Contract Terms'
 - ContractModificationType
 - ContractNatureType
 - Country
 - CPVType
 - Criterion
 - 'Criterion property'
 - 'Criterion property datum'
 - 'Criterion property group'
 - CriterionGroupPropertyType
 - CriterionTaxonomy
 - Currency
 - Document
 - 'Procurement Document'
 - 'Additional Document'
 - 'Call For Competition'
 - 'Contract Conditions'
 - 'Descriptive Document'
 - 'ESPD Request'
 - 'Invitation to Tender'
 - Notice
 - 'Call for expression of interest'
 - 'Contract Award Notice'
 - 'Contract Modification Notice'
 - 'Contract Notice'
 - 'Prior Information Notice'
 - 'Pre-Award Catalogue Request'
 - 'Technical Specifications'
 - 'Request for participation'
 - 'Tender Document'
 - 'ESPD Response'
 - 'Financial Tender Document'
 - 'Pre-Award Catalogue'
 - 'Technical Tender Document'
- EOGroupType
- EOIndustryClassificationType
- EORoleType

Figure 34. Protégé 5.2 for the edition of ePO

Codes and Identifiers

A code is a shortened way (a number or a short abbreviated text), leading to the definition of a 'concept'. The code represents the concept and is used by software applications to retrieve the definition of the concept or make automatic decisions.

An Identifier, in turn, can be defined as “a value (represented as a short text, a number or a combination of both) used to establish the identity of, and distinguish uniquely, one occurrence of an object following a pattern” .

The essential distinctive features between identifiers and codes are:

1. Identifiers point at specific occurrences of objects (instances); codes replace concepts;
2. Identifiers are virtually limitless while codes are finite. In other words, identifier lists are “open” (the lists may grow) and code list are “closed” (practically never updated, once consolidated). Hence codes are maintained in 'Codelists' whilst identifiers are usually kept in databases.
3. Identifiers are in principle maintained in the business domain, e.g. buyer identifiers, economic operator identifiers, product identifiers, etc.

Alignment to CCTS

The ePO tries to reuse as much as possible standards, specifications and practices commonly applied in the eProcurement domain. Hence one design decision, coordinated with the WG members, was to use the [ISO 15000-5:2014](#) (Core Components Specification) ^[6] to implement the data types Identifier, Amount, Quantity and Measure.

The package containing the OWL-TTL definition of these data type can be download from the ePO GitHub repository folder [05 Implementation/ttl/](#).

Codes, in ePO, are implemented as [SKOS](#) concepts, but they are also aligned to the requirements commonly needed in eProcurement (see sections below, on codes and code lists).

Codes and code lists

The ePO tries to reuse as much as possible the codes that are already used for e-Procurement. Many of these codes are already published in different formats by the Publications Office and are freely available in the section [link:https://publications.europa.eu/en/web/eu-vocabularies/authority-tables](https://publications.europa.eu/en/web/eu-vocabularies/authority-tables) [Authority Tables] of the OP's [MDR] site. These codes are described in "code lists", in all the EU official languages.

We distinguish at least three layers of codes:

1. **Cross-sector**, common, codes, like the ones defined and maintained by the ISO for currencies, languages, countries, etc.; or the ones defined by the European Commission that can be used in multiple business domains, e.g. the NUTS defined by EUROSTAT;
2. **Business domain-related**, maintained by international or European authorities, e.g. the ones defined by UNECE (as unit codes), or by the OP, e.g. types of procurement procedures (based on the EU Directives);
3. **Project-specific** (or application-specific), i.e. those codes that are particular of the project, e.g. codes used by the OP's applications (eSenders' tools);

Codes are normally maintained in "code lists". All code lists available on the MDR site can be download as HTML, XML, XSD, SKOS or SKOS-EU-AP (a profile of SKOS-XL).

The ePO development team, jointly with the WG members, convened to use the SKOS-EU-AP syntax available on the MDR site. The fragment of code below shows how an instance of a code is referred to in a SPARQL insert query:

Use of SKOS-EU-AP code lists in ePO

```

PREFIX : <http://data.europa.eu/ePO/ontology#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX org: <http://www.w3.org/ns/org#>
PREFIX vcard: <http://www.w3.org/2006/vcard/ns#>
PREFIX rov: <http://www.w3.org/ns/regorg#>
PREFIX adms: <http://www.w3.org/ns/adms#>
PREFIX ccts: <http://www.unece.org/cefact#>
PREFIX euvoc:
<http://publications.europa.eu/ontology/euvoc#>
PREFIX epo-rd: <http://data.europa.eu/ePO/referencedata#>

INSERT
{
    Graph <http://data.europa.eu/ePO/ontology> {
        :PE143899-2018 rdf:type :ProcuringEntity ;
        :usesProcuringEntityType epo-rd:CA ;
        skos:prefLabel "Etat de Fribourg,
Direction des finances, Service de l'informatique et des
télécommunications (SITel)" ;
        org:hasSite :PESite_143899-2018 ;
        :usesJurisdictionalCompetenceLevelType
epo-rd:AUTHORITY_LOCAL .
        :PESite_143899-2018 rdf:type vcard:Individual
;
        vcard:hasAddress :PEAddress143899-2018
.
        :PEAddress143899-2018 rdf:type vcard:Address ;
        vcard:region epo-rd:CH0 ; (1)
    }
}

```

```

Piller 50" ;
                                vcard:street-address "Route André
                                vcard:postal-code "1762" ;
                                vcard:country-name euvoc:CH ; (2)
                                vcard:hasEmail "AOP_SITel@fr.ch" ;
                                vcard:locality "Givisiez"
    }

```

1 NUTS 2016 code, defined by EUROSTAT

2 Country Code, available on the MDR

This other code shows a fragment of the SKOS-EU-AP code list for countries. Click [here](#) to download the file.

SKOS-EU-AP Country Code code list (Luxembourg code)

```

<skos:Concept
rdf:about="http://publications.europa.eu/resource/authority
/country/LUX"
                                at:deprecated="false"
                                at:protocol.order="EU-16">
    <rdf:type
rdf:resource="http://publications.europa.eu/ontology/euvoc#
Country"/>
    <dc:identifier>LUX</dc:identifier>
    <at:protocol-order>EU-16</at:protocol-order>
    <at:authority-code>LUX</at:authority-code>
    <at:op-code>LUX</at:op-code>
    <atold:op-code>LUX</atold:op-code>
    ...
    <skos:topConceptOf
rdf:resource="http://publications.europa.eu/resource/author
ity/country"/>
    <skos:inScheme
rdf:resource="http://publications.europa.eu/resource/author
ity/country"/>
    <owl:versionInfo>20180620-0</owl:versionInfo>
    <dct:dateAccepted
rdf:datatype="http://www.w3.org/2001/XMLSchema#date">2012-
06-27</dct:dateAccepted>
    <dct:created
rdf:datatype="http://www.w3.org/2001/XMLSchema#date">2010-
01-01</dct:created>
    <dct:dateSubmitted
rdf:datatype="http://www.w3.org/2001/XMLSchema#date">2011-
10-06</dct:dateSubmitted>

```

```

    <euvoc:startDate
rdf:datatype="http://www.w3.org/2001/XMLSchema#date">1950-
05-09</euvoc:startDate>
    <euvoc:status
rdf:resource="http://publications.europa.eu/resource/author
ity/concept-status/CURRENT"/>
    <euvoc:order>EU-16</euvoc:order>
... etc.

```

One interesting aspect of the OP's SKOS EU Application Profile (SKOS-EU-AP) is that all the metadata specified as attributes of the ISO 15000 CodeType Core Component Specification are also expressed in the equivalent OP's SKOS-EU-AP code list. As a matter of fact, the features of the SKOS-XL specification which the SKOS-EU-AP is built upon permits the specification of any metadata that can be necessary on both the code list (the "concept scheme") and the individuals of the list (each "concept" of the list).

The figure below shows the set of attributes that can be used for a CCTS CodeType element:

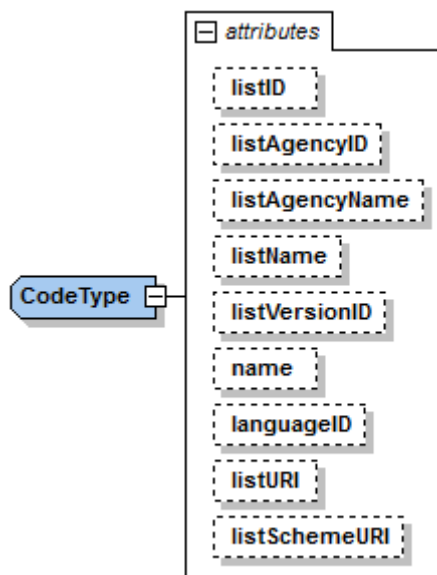


Figure 35. Attributes of the CCTS CodeType element

The table below contains the definitions of each attribute (as defined in [OASIS UBL, ISO/IEC 19845:2015](#)):

Table 1. UBL attributes for codes

| Attribute | Definition |
|---------------|--|
| listID | The identification of a list of codes. |

| | |
|-----------------------|--|
| listAgencyID | An agency that maintains one or more lists of codes. |
| listAgencyName | The name of the agency that maintains the list of codes. |
| listName | The name of a list of codes. |
| listVersionID | The version of the list of codes. |
| name | The textual equivalent of the code content component. |
| languageID | The identifier of the language used in the code name. |
| listURI | The Uniform Resource Identifier that identifies where the code list is located. |
| listSchemeURI | The Uniform Resource Identifier that identifies where the code list scheme is located. |

Identifiers

As commented above, ePO defines a class Identifier in alignment to the ISO 15000 Core Component Specification. This link:[class](#) looks like this:

Definition in ePO of the CCTS-based Identifier class

```

@prefix : <http://www.unece.org/cefact#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
.
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix ccts: <http://www.unece.org/cefact#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix schema: <http://schema.org/> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@base <http://www.unece.org/cefact> .

<http://www.unece.org/cefact> rdf:type owl:Ontology ;
                                owl:versionIRI
<http://www.unece.org/cefact/2> ;

```



```
        rdfs:range xsd:normalizedString ;
        rdfs:comment "The literal identifying
an entity, like a person or an object."@en ;
        rdfs:isDefinedBy
<http://www.everis.com> .
```

```
### http://www.unece.org/cefact#schemeAgencyName
ccts:schemeAgencyName rdf:type owl:DatatypeProperty ,
                        owl:FunctionalProperty ;
                        rdfs:domain ccts:Identifier ;
                        rdfs:range xsd:string ;
                        rdfs:comment "The name of the agency
that maintains the identification scheme."@en .
```

```
### http://www.unece.org/cefact#schemeDataURI
ccts:schemeDataURI rdf:type owl:DatatypeProperty ,
                    owl:FunctionalProperty ;
                    rdfs:domain ccts:Identifier ;
                    rdfs:range xsd:anyURI ;
                    rdfs:comment "The Uniform Resource
Identifier that identifies where the identification scheme
data is located."@en .
```

```
### http://www.unece.org/cefact#schemeID
ccts:schemeID rdf:type owl:DatatypeProperty ,
               owl:FunctionalProperty ;
               rdfs:domain ccts:Identifier ;
               rdfs:range xsd:normalizedString ;
               rdfs:comment "The identification of the
identification scheme."@en .
```

```
### http://www.unece.org/cefact#schemeName
ccts:schemeName rdf:type owl:DatatypeProperty ,
                  owl:FunctionalProperty ;
                  rdfs:domain ccts:Identifier ;
                  rdfs:range xsd:string ;
                  rdfs:comment "The name of the
identification scheme."@en .
```

```
### http://www.unece.org/cefact#schemeURI
ccts:schemeURI rdf:type owl:DatatypeProperty ,
                 owl:FunctionalProperty ;
                 rdfs:domain ccts:Identifier ;
                 rdfs:range xsd:anyURI ;
                 rdfs:comment "The Uniform Resource
Identifier that identifies where the identification scheme
is located."@en .
```

```
### http://www.unece.org/cefact#schemeVersionID
ccts:schemeVersionID rdf:type owl:DatatypeProperty ,
                       owl:FunctionalProperty ;
```

```

        rdfs:domain ccts:Identifier ;
        rdfs:range xsd:normalizedString ;
        rdfs:comment "The version of the
identification scheme."@en .

```

```

#####
#####

```

```

#      Classes

```

```

#####
#####

```

```

###  http://www.unece.org/cefact#Identifier
ccts:Identifier rdf:type owl:Class ;
        rdfs:comment "A character string to
identify and distinguish uniquely, one instance of an
object in an identification scheme from all other objects
in the same scheme together with relevant supplementary
information. This class is based on the UN/CEFACT
Identifier complex type defined in See Section 5.8 of Core
Components Data Type Catalogue Version 3.1
(http://www.unece.org/fileadmin/DAM/cefact/codesfortrade/CC
TS/CCTS-DTCatalogueVersion3p1.pdf). In RDF this is
expressed using the following properties: - the content
string should be provided using skos:notation, datatype
with the identifier scheme (including the version number if
appropriate); - use dcterms:creator to link to a class
describing the agency that manages the identifier scheme or
adms:schemaAgency to provide the name as a literal.
Although not part of the ADMS conceptual model, it may be
useful to provide further properties to the Identifier
class such as dcterms:created to provide the date on which
the identifier was issued."@en ;
        rdfs:isDefinedBy <http://www.ebxml.org/> ,

<http://www.unece.org/cefact> ;
        rdfs:label "Identifier"@en .

```

```

... etc.

```

This code matches the specification and definitions maintained by UN/CEFACT:

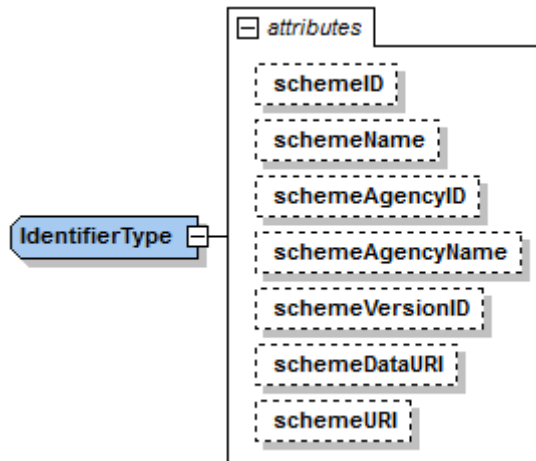


Figure 36. Attributes of the CCTS IdentifierType element

The definition of these definitions, as provided by OASIS UBL (ISO/IEC 19845), follows:

Table 2. CCTS IdentifierType attributes

| Attribute | Definition |
|-------------------------|--|
| schemeID | The identification of the identification scheme. |
| schemeName | The name of the identification scheme. |
| schemeAgencyID | The identification of the agency that maintains the identification scheme. |
| schemeAgencyName | The name of the agency that maintains the identification scheme. |
| schemeVersionID | The version of the identification scheme. |
| schemeDataURI | The Uniform Resource Identifier that identifies where the identification scheme data is located. |
| schemeURI | The Uniform Resource Identifier that identifies where the identification scheme is located. |

V. Proof of Concept

The ePO development team agreed with the WG members to provide a means to test the deliverables produced, especially the Conceptual Data Model and the OWL-TTL implementation. With this purpose in mind a Proof of Concept was planned and executed.

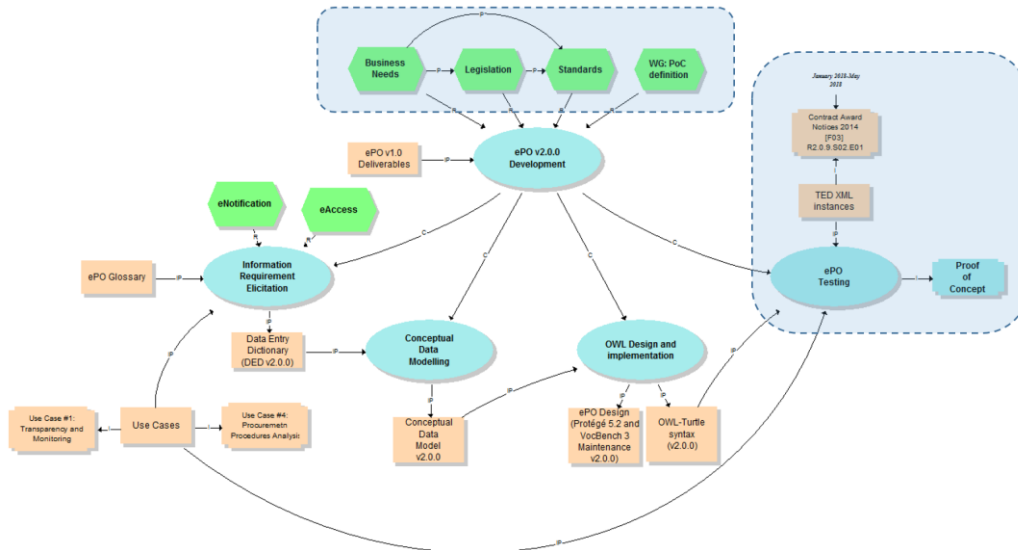


Figure 37. ePO Testing

The main objectives of the Proof of Concept were:

1. Test the coherence of the Conceptual Data Model (of the T-Box);
2. Test the consistency of the data once loaded (in the A-Box);
3. Test the effectiveness of the OWL implementation of the eProcurement Ontology (ePO); and
4. Test the feasibility of the ePO to support the Use Cases defined in ePO v.1.0.

Hence a varied set of activities were planned with these objectives in mind. The diagram below shows the activities that were planned and executed to develop the Proof of Concept:

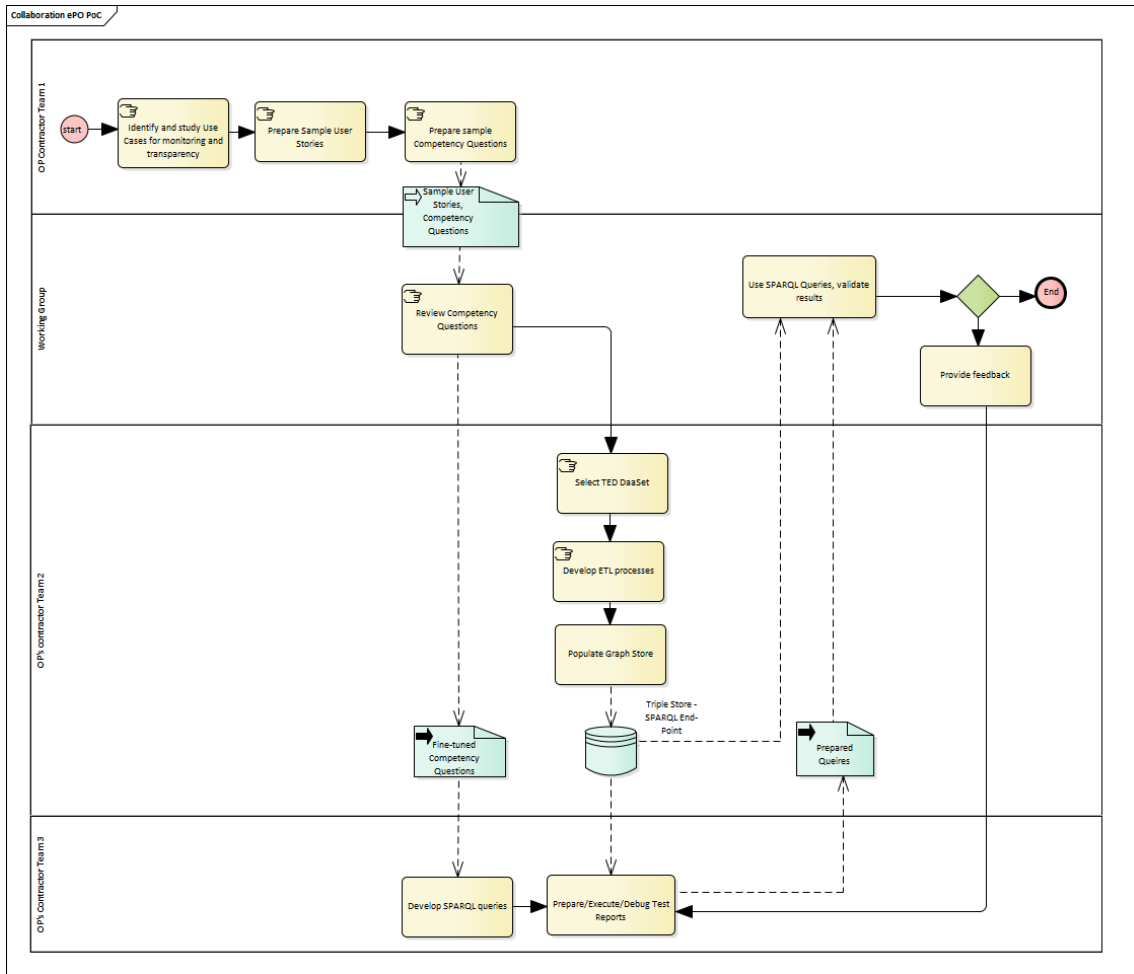


Figure 38. ePO Project v2.0.0 - Proof-Of-Concept

The following subsections explain how each of the activities mentioned in the diagram above has been developed and where to check the inputs, processes and results.

Activity 1: Use Cases

Table 3. Competency Questions, activity summary

| | |
|--------------------------|--|
| Activity name: | Identify and study the Use Cases related to monitoring and transparency. |
| Responsible team: | OP's contractor team. |
| Inputs: | ePO v1.0 Use Case 1 and Issue #11 . |

| | |
|-----------------|---|
| Outputs: | Study of the Use Cases (slightly renaming). |
|-----------------|---|

The ePO v1.0 focused on three different Use Cases:

- [Use Case 1](#): Data Journalism
- [Use Case 2](#): Automated matchmaking of procured services and products with businesses, and
- [Use Case 3](#): Verifying VAT payments on intra-community service provision.

During its development a fourth Use Case was identified as relevant related to Transparency and Monitoring. This use case was proposed through an "[Issue](#)", in the GitHub repository. This Use Case was accepted as as a relevant case for transparency and monitoring.

Hence the ePO v2.0.0, which is focused only on transparency and monitoring, was developed taken into account two Use Cases (slightly renamed):

- [Use Case 1](#): Transparency and Monitoring; and
- [Use Case 4](#): Analyzing eProcurement procedures.

Activity 2: User Stories

Table 4. User Stories, activity summary

| | |
|--------------------------|--|
| Activity name: | Prepare sample (example) User Stories. |
| Responsible team: | OP's contractor team. |
| Inputs: | Use Cases 1 and 4. |
| Outputs: | Examples of User Stories . |

User Stories are a method of helping identify information requirements. The method consists in drafting very simple sentence structured around three main questions:

1. Who is the beneficiary of an action (who benefits from it)?
2. What is the need?

3. What is the benefit?

The structure of the sentence is always like this: “As a <role of the user>, I need <something>in order to <benefit>.”

Example:

As a **contracting authority** (ROLE), I need to know **the number of tenderers** (WHAT DO I NEED?) that have submitted a tender in order **to add it to the award notice** (BENEFIT).

Some examples of User Stories were prepared. The table below shows these sample User Stories for different roles and related to the Use Cases 1 and 4.

| Role | User Stories | Use case |
|-----------------------|--|----------|
| Citizen | As a citizen I want to know who the winner is, what was the total value amount awarded of the contract, what were the selection criteria, who was the procuring entity by CPV, so I can know how much public money is spent. | 1 |
| Citizen - Journalist | As a journalist I want to know the contract value for tram maintenance at country level by selecting awards that refer to some CPV codes in order to compare it with other countries. | |
| Economic Operator | As an Economic Operator, I want to know what award criteria the contracting authorities use for contracts in my business, so that I can know if I have to lower the price of the tender. | 4 |
| Contracting Authority | As a Contracting Authority, I want to know the number of contracts modifications and amount in percentage of these modifications in relation with the awarded/evaluated and modified budget, so I can improve future procurement procedures. | |
| | As a Contracting Authority, I want to know the percentage of contracts awarded segmented by contract nature, so I can prepare the annual report. | |
| | As a Contracting Authority, I want to know the percentage of contracts awarded segmented by procurement procedure, so I can prepare the annual report. | |

Figure 39. Examples of User Stories

Activity 3: Competency Questions

Table 5. Competency Questions, activity summary

| | |
|--------------------------|--|
| Activity name: | Prepare sample (example) Competency Questions. |
| Responsible team: | OP’s contractor team. |
| Inputs: | Use Cases 1 and 4, and the User Stories. |
| Outputs: | Examples of Competency Questions . |

User Stories help also draft very specific questions that need to be answered in order to meet the story. These questions will later on taken into account to draft concrete SPARQL queries.

Some examples of Competency Questions were prepared. The two tables below illustrate how these Competency Questions, linked to their respective User Stories, may look like. The link [Competency Questions](#), in the GitHub [Wiki](#) page, supplies a longer list of concrete examples of CQ for the WG members to get inspiration.

| User Stories | Role | Competency Questions | Use case |
|--|----------------------|---|----------|
| As a citizen I want to know who the winner is, what was the total value amount awarded of the contract, what were the selection criteria, who was the procuring entity by CPV, so I can know how much public money is spent. | Citizen | - For a given period of time, get the procurement procedures awarded, the procuring entity, the winners, the total value amount, and the selection criteria by CPV. | 1 |
| As a journalist I want to know the contract value for tram maintenance at country level by selecting awards that refer to some CPV codes in order to compare it with other countries. | Citizen - Journalist | -For a given CPV, the number of contracts awarded for that CPV and the total value amount awarded. - For a given CPV, the number of contracts awarded for that CPV and and a maximum number of descendant levels (e.g. 3), get the total amount awarded. The result must include all the descendant CPVs and total amount awarded for the CPV and descendants indicated. | |

Figure 40. Example (1/2) of Competency Questions

| User Stories | Role | Competency Questions | Use case |
|--|-----------------------|---|----------|
| As an Economic Operator, I want to know what award criteria the contracting authorities use for contracts in my business, so that I can know if I have to lower the price of the tender. | Economic Operator | For a given period of time, get the procurement procedures awarded and the award criteria segmented by CPV. | 4 |
| As a Contracting Authority, I want to know the number of contracts modifications and amount in percentage of these modifications in relation with the awarded/evaluated and modified budget, so I can improve future procurement procedures. | Contracting Authority | For a given period of time, number of contract modifications , total value amount in percentage of these modifications in relation with the total value amount and modified value amount. | |
| As a Contracting Authority, I want to know the percentage of contracts awarded segmented by contract nature, so I can prepare the annual report. | Contracting Authority | For a given period of time, percentage of contracts awarded segmented by contract nature (percentage of works, percentage of services, percentage of supplies, etc.). | |
| As a Contracting Authority, I want to know the percentage of contracts awarded segmented by procurement procedure, so I can prepare the annual report. | Contracting Authority | For a given period of time, percentage of contracts awarded segmented by procurement procedure (open, restricted, etc). | |

Figure 41. Examples (2/2) of User Stories

Activity 4: Review CQs

Table 6. Revision of Competency Questions, activity summary

| | |
|--------------------------|--|
| Activity name: | Review Competency Questions. |
| Responsible team: | Working Group members. |
| Inputs: | Competency Questions and related User Stories. |
| Outputs: | Comments by the WG members. |

The examples were made available to the Working Group members through the GitHub Wiki page. A special [Add a new competency question](#) to add comments or create new issues related to the CQs was also made available in the [GitHub Wiki page](#).

Activity 5: Input Data Set

Table 7. Select Data Set, activity summary

| | |
|--------------------------|--|
| Activity name: | Select Data Set. |
| Responsible team: | OP's team. |
| Inputs: | Use Cases, User Stories, Competency Questions, agreement with the members of the WG. |
| Outputs: | Documents published on TED, accessed via the OP's FTP server. |

For the extraction of data, the decision was made that the source of data should be the Notices published on the [TED portal](#). This decision was made based on, namely, the following reasons:

- The Contract Award Notice (CAN) contains the data most relevant for Transparency, Monitoring and Procedure control (jointly with the Contract Notice (CN));
- The CAN is the most published document, therefore the sample is richer;
- The structure and elements of the standard form for the CAN are very similar or identical to many of other Notices. This allows to reuse a relevant part of the

extraction and transformation artefacts (XSL-T) to process many other types of Forms.

However the User Interface of the TED Portal does not allow downloading large amount of documents. For this we used the [FTP](#) server supplied by the OP at: <ftp://ted.europa.eu/> (user: **guest**, password: **guest**).

The TED-XML specification has been evolving for the past years. Different versions of XSD Schemas have been maintained in parallel for those years. The result is that, as per today, different schemas are being used to express the data in alignment to the 2014 Directives. For this PoC we decided to use CAN based only on the TED-XML XSD Schema [R2.0.9.S01.E01 TED_EXPORT.xsd](#) and the Contract Award Notice (CAN) form for Directive 2014 supporting the [F03_2014.xsd](#) standard form (all schemas are published on the Publications Office (OP) [MDR site](#)).

For this PoC we downloaded the [*.tar.gz](#) files corresponding to January to May 2018. Bear in mind that, in the context of this PoC, we only extract data and import into the graph store the CANs for Directive 2014. However the TED_EXPORT.xsd includes all the forms (F01 to F25) and the extraction process is able to extract data from many of these forms, as they share a large part of the elements (see "Activity 6: ETL process", just below). If you want a go with these other forms just uncomment the line "#DOCUMENT_TYPE_ID=1,2,3,22,23,24,25 " and comment the line "DOCUMENT_TYPE_ID=3" in the epo.properties file.

Activity 6: ETL process

Table 8. ETL process development, activity summary

| | |
|--------------------------|--|
| Activity name: | Develop ETL process. |
| Responsible team: | OP's team. |
| Inputs: | TED-XML schemas (on MDR) and TED notices published on the TED FTP server. |
| Outputs: | TED to ePO Mapping (Wiring), Java code, XSL-T architecture, other resources (available on the GitHub repository and accessible via the GitHub Wiki page link Data Loading development (ETL)). |

[TED to ePO Mapping](#)

ETL stands for Extraction, Transformation and Loading. The first step (Extraction) requires to identify well where the data of origin are and how they are expressed. For this, the ePO analysts produced a [map](#) putting side by side (wiring) each element of the TED-XML Schema ([R2.0.9.S02.E01](#)) and the corresponding element in ePO.

Technical approach

The ETL process was developed based on two technologies:

1. **Java:** version JDK 1.8 was used to build a Maven project (see [pom.xml](#) configuration file). The output of the build process is a "*.war" file. The [source code](#) is available on the GitHub code repository. This java code is responsible for (i) organising the TED-XML files; (ii) launching the extraction + transformation and/or the loading the data into the graph store, and (iii) log all the events and generate logs for monitoring the process;
2. **XSL-T:** version XSL-T 3.0 was used to draft a set of [stylesheets](#) the mission of which is to read the TED-XML files (Extraction) and transform that information into SPARQL INSERT patterns. Per each TED-XML a new TXT document is created with the mapped SPARQL INSERT patterns. The name of the resulting TXT takes the name of the TED XML file and appends the suffix "_output.txt". The piece of code below illustrates one of those examples (if you use the identifier of the document you should be able to find the TED-XML source in the TED Portal).

Result of transforming the TED-XML instance "091271-2018" into ePO-v2.00
SPARQL INSERT queries

```
PREFIX : <http://data.europa.eu/ePO/ontology#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX org: <http://www.w3.org/ns/org#>
PREFIX vcard: <http://www.w3.org/2006/vcard/ns#>
PREFIX rov: <http://www.w3.org/ns/regorg#>
PREFIX ccts: <http://www.unece.org/cefact#>
PREFIX euvoc:
<http://publications.europa.eu/ontology/euvoc#>
PREFIX ubl: <http://docs.oasis-open.org/ubl#>
PREFIX epo-rd: <http://data.europa.eu/ePO/referencedata#>

INSERT DATA
{
    Graph <http://data.europa.eu/ePO/ontology#>{
        :CAN_091271-2018 rdf:type :ContractAwardNotice
    ;
        :hasPublicationDate "2018-03-
01T00:00:00"^^xsd:dateTime ;
        :hasDocumentIdentifier :CAN_ID_091271-
2018
    }
};
INSERT DATA
```

```

{
    Graph <http://data.europa.eu/ePO/ontology>{
        :CAN_ID_091271-2018 rdf:type ccts:Identifier ;
        ccts:identifierValue "091271-2018" ;
        ccts:schemeAgencyID
"eu.europa.publicationsoffice.epo"
    }
}
...
(1)

```

1 See [GitHub code repository](#) or execute the code for complete examples.

A note about the performance

The Java code developed and the XSL-T approach are extremely fast:

- **Transformation speed:** *1 notice x 2 ms*. One Contract Award Notice transformed into a SPARQL file with multiple INSERT operations in about 2 milliseconds. See the use of XMLStreamReader APIs (e.g. STAX) to capture the metadata about the TED-XML instances in [XSLTTransformer.java](#).
- **Insertion speed:** *1 notice x 0,5 s*. One Contract Award Notice containing hundreds of INSERT operations inserted in the GraphDB as one single transaction in about 0,5 seconds. When the file is greater than 1MB the INSERT operations are split into individual transactions, in which case the operations can consume up to around 1 second. See java code in [KBManagement.java](#).

Code Execution

You can execute the code at least in two ways:

1. Either you clone the project onto your machine, import the Maven project in your preferred Java editor tool and execute the main class [MainETLProcess](#).
2. Alternatively you may unzip the *.war file and execute the compiled code from a console window. The piece of code below provides a very simple script illustrating how this can be done:

Launching the code, a simple bash shell script

```
#!/bin/bash
```

```
arg="$1"
```

```
exec java -classpath "lib/*:classes/." epo.MainETLProcess $arg
```

Beware that the MainETLProcess takes one argument:

Acceptable arguments

Usage: epo.MainETLProcess [-t][[-i]][[-a]]

Valid arguments are:

-t transforms XML into .txt files containing the SPARQL queries, but does not execute the queries.

-i executes the SPARQL queries only.

-a does everything.

Options are mutually exclusive. Only one option is accepted.

Example:

```
java -classpath "lib/*:classes/" epo.MainETLProcess -t
```

```
java -classpath "lib/*:classes/" epo.MainETLProcess -i
```

```
java -classpath "lib/*:classes/" epo.MainETLProcess -a
```

ETL execution configuration

The java code uses a file named **epo.properties**. This file is to be located under the /home/user directory of the computer from where the code is executed. See below an example of how this configuration file looks like. Notice the two lines about the proxy configuration.

The *epo.properties* file, example

```
#Thu Jun 28 10:49:40 CEST 2018
```

```
### Graph db access
```

```
#####
```

```
#####
```

```
#GRAPH_STORE_URL=http://34.249.1.15:7200
```

```
GRAPH_STORE_URL=http://localhost:7200
```

```
GRAPH_STORE_USER=paulakeen
```

```
GRAPH_STORE_PASSWORD=shootingNicely2018Times
```

```
GRAPH_STORE_REPOSITORY=ePO_test
```

```
### Proxy configuration
```

```
#####
```

```
#####
```

```
#PROXY_URL=10.110.8.42
```

```
#PROXY_PORT=8080
```

```
### Directories configuration
```

```
#####
```

```
#####
```

```
## The directory where the TED-XML files are located
```

```
INPUT_DATA_DIR=/TED-Resources
```

```

## The directory where the SPARQL INSERT TXT files,
resulting from the XSL-T transformation, are written.
## This directory is the input directory from where the TXT
files are taken to populate the Graph Store.
OUTPUT_DATA_DIR=/TED-OUTPUT
## The directory where the java application logs the
operations executed and exceptions.
LOG_DATA_DIR=/TED-LOG
## Where the XSL-T architecture files are located. Relative
or absolute paths can be specified.
## Relative paths are relative to the path from where the
etl-process is launched.
TED_TO_EPO_XSL=./src/main/resources/xslt/TEDXSD_to_ePOTTL.x
sl
## Where the TED XSD Schemas are located. Relative or
absolute paths can be specified.
## Relative paths are relative to the path from where the
etl-process is launched.
## @DEPRECATED comment="the latest version uses STAX
XMLStreamReader and works on multiple TED_XSD_VERSIONS
TED_EXPORT_XSD=./src/main/resources/TED_publication_R2.0.9.
S02.E01_003-20170123/TED_EXPORT.xsd
## Subsystem IDs, XSD root element local name of the
Subsystems that produced the XML instances that are
## requested to be processed. A comma separated list of
names is expected.
TED_SUBSYSTEMS=TED_EXPORT
## Version IDs of the TED-XSD schemas upon which the XML
that are requested to be
## processed are instantiated. A comma separated list of
names is expected.
#TED_XSD_VERSIONS=R2.0.9.S02.E01, R2.0.9.S01.E01
TED_XSD_VERSIONS=R2.0.9.S02.E01
## Form types requested to be processed.
#TED_XSD_FORM_TYPES=F01, F02, F03
TED_XSD_FORM_TYPES=F03

```

Notice that each execution of the ETL process generates a log file in the specified directory (property "LOG_DATA_DIR"). The log files append the total number of files transformed and inserted at the end of the file. These figures can be used to study the amount and types of documents that have been published by the OP. For an example see the section following [Activity 7: Populate Graph store](#). The data were extracted from the logs about the transformation of each month of 2018, separately, from January to May.

Activity 7: Populate Graph store

Table 9. Populate the Graph store, activity summary

| | |
|--------------------------|---|
| Activity name: | Populate the Graph store. |
| Responsible team: | OP's team. |
| Inputs: | The result of the XSL-T-based transformation (SPARQL INSERT queries). |
| Outputs: | The Graph store is populated with triples. |

A large amount of TXT files containing the SPARQL INSERT queries was automatically obtained - out of the transformation- for the five first months of 2018. The table and bar graphic below show the exact number of files processed and the number of Contract Award Notices imported into the Graph Store.

The Graph Store chosen for this PoC was the Community version of GraphDB (version 8.5)m which can be freely downloaded from the [Ontotext](#) website.

| Form Type (R2.0.9.S02.E01) | JANUARY 2018 | FEBRUARY 2018 | MARCH 2018 | APRIL 2018 | MAY 2018 | Totals 5 months | Averages |
|-------------------------------|---------------|---------------|---------------|---------------|---------------|-----------------|---------------|
| F01 | 923 | 1.151 | 1.238 | 890 | 1.010 | 5.212 | 1.042 |
| F02 | 11.333 | 14.311 | 17.739 | 15.000 | 14.420 | 72.803 | 14.561 |
| F03 | 16.556 | 13.929 | 14.280 | 13.520 | 13.910 | 72.195 | 14.439 |
| F04 | 51 | 54 | 79 | 52 | 38 | 274 | 55 |
| F05 | 1.172 | 1.418 | 1.706 | 1.551 | 1.494 | 7.341 | 1.468 |
| F06 | 1.811 | 1.473 | 1.510 | 1.451 | 1.426 | 7.671 | 1.534 |
| F07 | 72 | 73 | 52 | 49 | 48 | 294 | 59 |
| F08 | 41 | 43 | 103 | 73 | 80 | 340 | 68 |
| F09 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| F10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| F11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| F12 | 102 | 103 | 139 | 125 | 132 | 601 | 120 |
| F13 | 88 | 72 | 99 | 64 | 100 | 423 | 85 |
| F14 | 4.329 | 4.395 | 5.632 | 4.816 | 5.404 | 24.576 | 4.915 |
| F15 | 351 | 450 | 519 | 381 | 459 | 2.160 | 432 |
| F16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| F17 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| F18 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| F19 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| F20 | 743 | 738 | 853 | 748 | 747 | 3.829 | 766 |
| F21 | 668 | 653 | 933 | 720 | 675 | 3.649 | 730 |
| F22 | 17 | 17 | 11 | 13 | 15 | 73 | 15 |
| F23 | 12 | 8 | 10 | 8 | 16 | 54 | 11 |
| F24 | 124 | 137 | 143 | 115 | 122 | 641 | 128 |
| F25 | 87 | 58 | 66 | 78 | 63 | 352 | 70 |
| OTHER RELEASES | 6.798 | 6.755 | 7.736 | 5.512 | 5.284 | 32.085 | 6.417 |
| Total per month | 45.278 | 45.838 | 52.848 | 45.838 | 45.838 | 235.640 | 47.128 |

Figure 42. Total of Notices and number of Contract Award Notices used to populate the Graph store

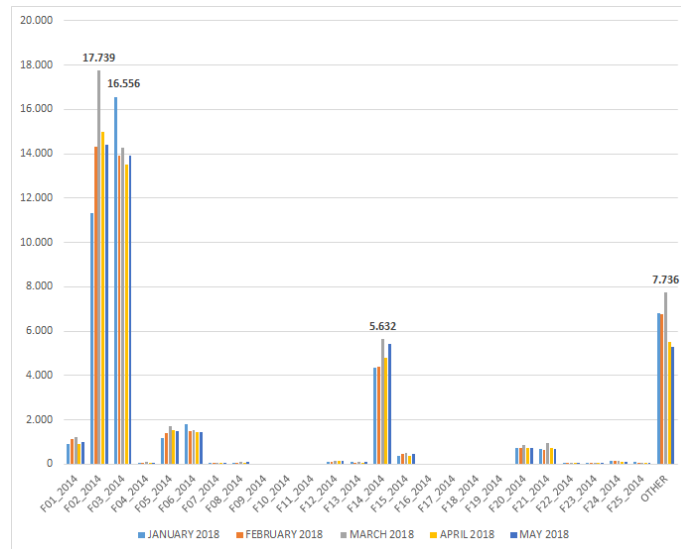


Figure 43. Frequency of Notices

Activity 8: SPARQL Queries

Table 10. Develop SPARQL Queries, activity summary

| | |
|--------------------------|--|
| Activity name: | Develop SPARQL Queries. |
| Responsible team: | OP's team. |
| Inputs: | Competency Questions (QCs examples). |
| Outputs: | The Graph store is populated with triples. |

The document [SPARQL Query examples](#) provides a few examples that were provided for the Working Group (WG) members to have a glimpse at how efficiently the ePO is responding.

Query examples

Example 1: One very first exercise would consist in checking the amount of Contract Award Notice and to compare it to the number of transformations executed and compiled in the log files. For this open a browser, introduce the URL or IP of the GraphDB server (e.g. 34.249.1.15:7200) and copy this SPARQL Query in the textfield of the SPARQL Endpoint:

Counting the number of Contract Award Notices

```
PREFIX : <http://data.europa.eu/ePO/ontology#>
```

```

select ?s (count(?did) as ?c) where {
  ?s a :ContractAwardNotice;
  :hasDocumentIdentifier ?did;
} group by ?s

```

SPARQL Query & Update ⓘ

Editor only Editor and results Results only ⏏

```

1 PREFIX : <http://data.europa.eu/ePO/ontology#>
2 select ?s (count(?did) as ?c) where {
3   ?s a :ContractAwardNotice;
4   :hasDocumentIdentifier ?did;
5 } group by ?s
6

```

Table Raw Response Pivot Table Google Chart

Download as 1 2 3 4 5

Figure 44. Number of Contract Award Notices between the 1s. January and the 30th May 2018

Example 2: List all the winners, the size of the company and the date of award.

Winners, size of the company, date of the awarding

```

PREFIX : <http://data.europa.eu/ePO/ontology#>
PREFIX rov: <http://www.w3.org/ns/regorg#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
select distinct ?Winner_Name ?WinnerSize ?Awarded_Date
where {
  ?Award_Result :hasWinner ?Winner ;
  :hasAwardResultDateOfConclusion
?Awarded_Date .
  ?Winner
:usesEOIndustryClassificationType ?WinnerSize ;
  rov:legalName ?Winner_Name
}

```

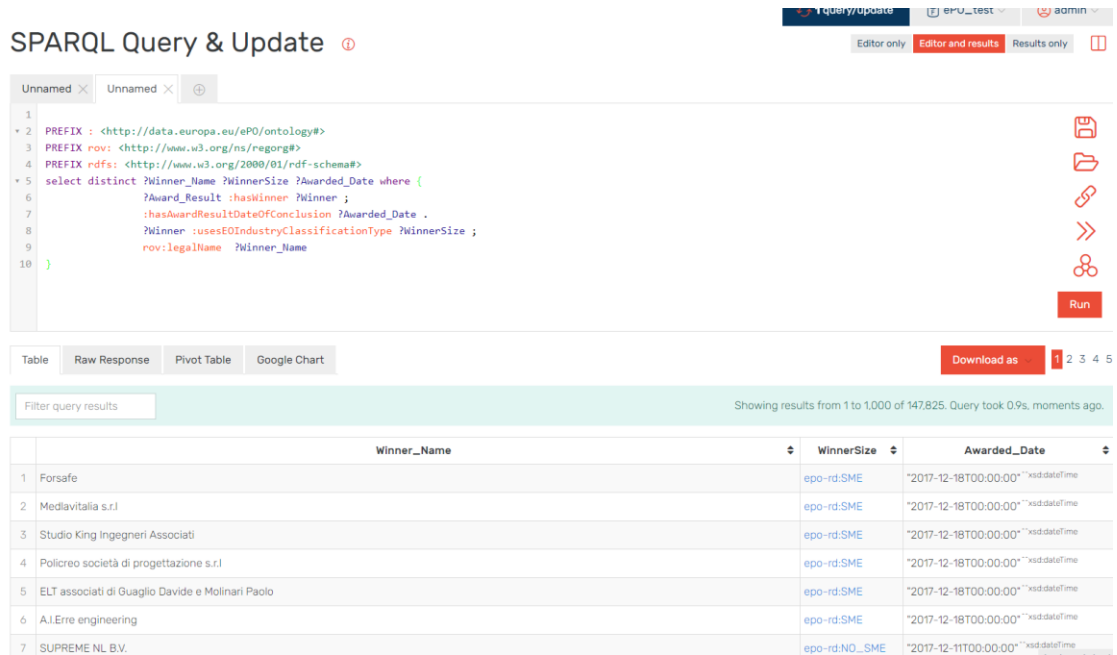


Figure 45. Winners

Example 3: Number of contracts awarded for each CPV (beware that one Contract Award Notice may refer to multiple contracts).

Number of contracts per CPV

```

PREFIX : <http://data.europa.eu/ePO/ontology#>
SELECT ?cpv (COUNT(DISTINCT(?contract)) AS
?number_contracts) where {
    ?contract a :Contract;
    :hasContractPurpose ?purpose.
    ?purpose :hasCPVType ?cpv.
} group by ?cpv order by desc(?number_contracts)

```

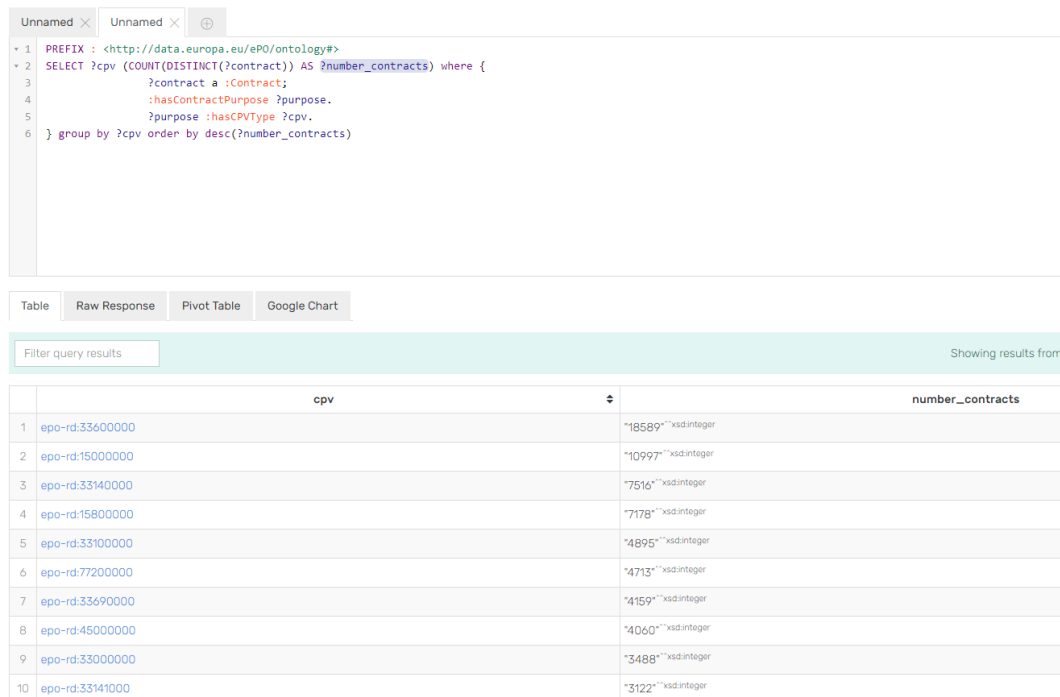



Figure 46. Number of contracts per CPV

See the document [SPARQL Query examples](#) for more contextualisation and examples.

Notes

1. by Natalya F. Noy and Deborah L. McGuinness.
2. by ISA2 Programme
3. See "[MOTPlus, Modélisation par Object Typés](#)", by LICEF, Télug
4. [Directive 2014/24/EU](#), [Directive 2014/23/EU](#), [Directive 2014/25/EU](#), [Directive 2009/81/EC](#) and [Directive 2014/55/EU](#)
5. **A note about codes:** although the CCTS library defines a data type for code (CodeType), we decided to use SKOS to represent code list and taxonomy concepts. Therefore this CCTS element is not used in ePO (see the ePO Conceptual Data Map "[ccts package](#)" and the [ccts.ttl](#) file for these definitions). There are some good reasons on which this decisions was based: (1) The OP's MDR site does already maintain and supply the largest part of the code lists needed by ePO as a SKOS-XL syntax (named SKOS-AP-EU); and (2) SKOS, and specially SKOS-XL, cater for a rich expressivity and metadata extensibility
6. currently maintained by UN/CEFACT, see UNECE's [CCTS](#)