Core Assessment Vocabulary – Anticorruzione Application Profile



CAV Pilot Documentation

**Table of Contents**

[1. Introduction 3](#_Toc58911210)

[2. Objective and Scope of this document 3](#_Toc58911211)

[3. Methodological approach 3](#_Toc58911212)

[4. Use Case 4](#_Toc58911213)

[4.1. Roadmap 4](#_Toc58911214)

[4.2. Core Assessment Vocabulary Application Profile 5](#_Toc58911215)

[4.2.1. Data Model for the CAV Pilot 5](#_Toc58911216)

[4.2.2. T-Box and A-Box of the CAV Pilot 7](#_Toc58911217)

[SPARQL queries 7](#_Toc58911218)

[4.2.3. 7](#_Toc58911219)

[4.2.4. MEAT selection and data visualization 8](#_Toc58911220)

[5. Conclusions 9](#_Toc58911221)

[Annex I – Documentation Provided by Italy 9](#_Toc58911222)

**¡Error! Referencia de hipervínculo no válida.¡Error! Referencia de hipervínculo no válida.¡Error! Referencia de hipervínculo no válida.¡Error! Referencia de hipervínculo no válida.¡Error! Referencia de hipervínculo no válida.¡Error! Referencia de hipervínculo no válida.¡Error! Referencia de hipervínculo no válida.¡Error! Referencia de hipervínculo no válida.¡Error! Referencia de hipervínculo no válida.¡Error! Referencia de hipervínculo no válida.¡Error! Referencia de hipervínculo no válida.**

**Table of Figures**

[Figure 1 CAV Pilot Roadmap 5](#_Toc58911256)

[Figure 2 CAV-Application Profile Data Model 6](#_Toc58911257)

[Figure 3: SPARQL Query 8](#_Toc58911258)

[Figure 4: Pipeline and visualisation 9](#_Toc58911259)

**¡Error! Referencia de hipervínculo no válida.¡Error! Referencia de hipervínculo no válida.**

# Introduction

This document reports the work related to one pilot produced over the Core Assessment Vocabulary (CAV) to test the quality of the vocabulary and improve it if needed. The pilot has been performed together with an Italian Member State institution (Anticorruzione[[1]](#footnote-1)), which provided the use case, sample data and support.

The Core Assessment Vocabulary represents, expresses and defines what an “Assessment” of “Assets” is and how to perform the assessment based on “Criteria”. It is a domain-agnostic vocabulary, meaning that it can be used to assess any asset.

The CAV was developed to be the “core vocabulary” for the CAMSS Ontology. The CAMSS Ontology reuses and customises the CAV. Therefore, CAMSS is an ‘Application Profile’ of CAV.

Previously to CAMSS, no other extension of the CAV had been implemented. Hence the need for new pilots, such as the one presented in this document.

# Objective and Scope of this document

The objective of this document is to present the results of the pilot executed over the CAV, including:

* Conclusions and results on the quality of the vocabulary and its reusability in the domain of eProcurement;
* The improvements implemented in the CAV for the accomplishment of the pilot.

The following artefacts and tasks were developed within the compass of the pilot development, and are also summarised in this document:

* Conceptual data models used for the CAV Pilot;
* Constraints and rules specific to the Pilot;
* A reference implementation of the CAV Pilot as an OWL Turtle syntax.

# Methodological approach

The approach followed for the development of the CAV Pilot sticks to three fundamental principles:

1. Reuse and share (i.e., do not reinvent the wheel);
2. Do not betray the knowledge and experience of the domain, nor the terminology and interpretation of the concepts (i.e. do not invent new terms when they already exist in the communities of practice or generic domains);
3. Isolate technical and business constraints and rules as much as possible; (i.e. externalise them in separate artefacts, e.g. graph and data shapes for the control and validation of the data).

Thus, one way of facilitating the semantic interoperability consists of reusing existing generic ontologies and vocabularies. This way, the semantics of common concepts and properties are agreed without having to re-discuss them. When concepts or properties have not been identified nor defined for the purposes pursued, they have to be proposed either as extensions or from scratch.

The methodological approach followed for the development of the CAV Pilot reuses the following ontologies and vocabularies:

* Friend of a Friend (FOAF);
* eProcurement Ontology (ePO);
* Core Public Service Vocabulary (CPSV);
* .

# Use Case

For the identification of a real use case from a member state, the CAMSS team realized a research through CAMSS stakeholders, and Italy expressed its interest to produce an application profile of the CAV providing a use case which consisted in **the MEAT Award Criteria Assessment “*Most Economic and Advantageous Tender award criteria assessment”*, based on the Electre Method in the eProcurement eEvaluation phase.** This method consists on a set of criteria that allows to assess the advantages and disadvantages between alternatives, in this use case Tenders, in each criterion and prioritize in preference order, the best to the worst.

The objective of this use case was to analyze whether the CAV could be used for this type of assessment or not, and if not, what would be the evolution necessary to meet this objective.

Italy provided a set of documentation needed for the production of the pilot and real data set which was used for the analysis and creation of the data model, the t-box and the a-box. *See the Annex I for the documentation provided by Italy.*

## Roadmap

The roadmap below shows the activities that were planned and executed to develop the CAV Pilot. The different actors involved were the Member State (Italy), CAMSS Team (CT) and the CAMSS Working group (CWG - external experts and the CAMSS Team). The externals were also reviewing and contributing to the new release of the CAV:

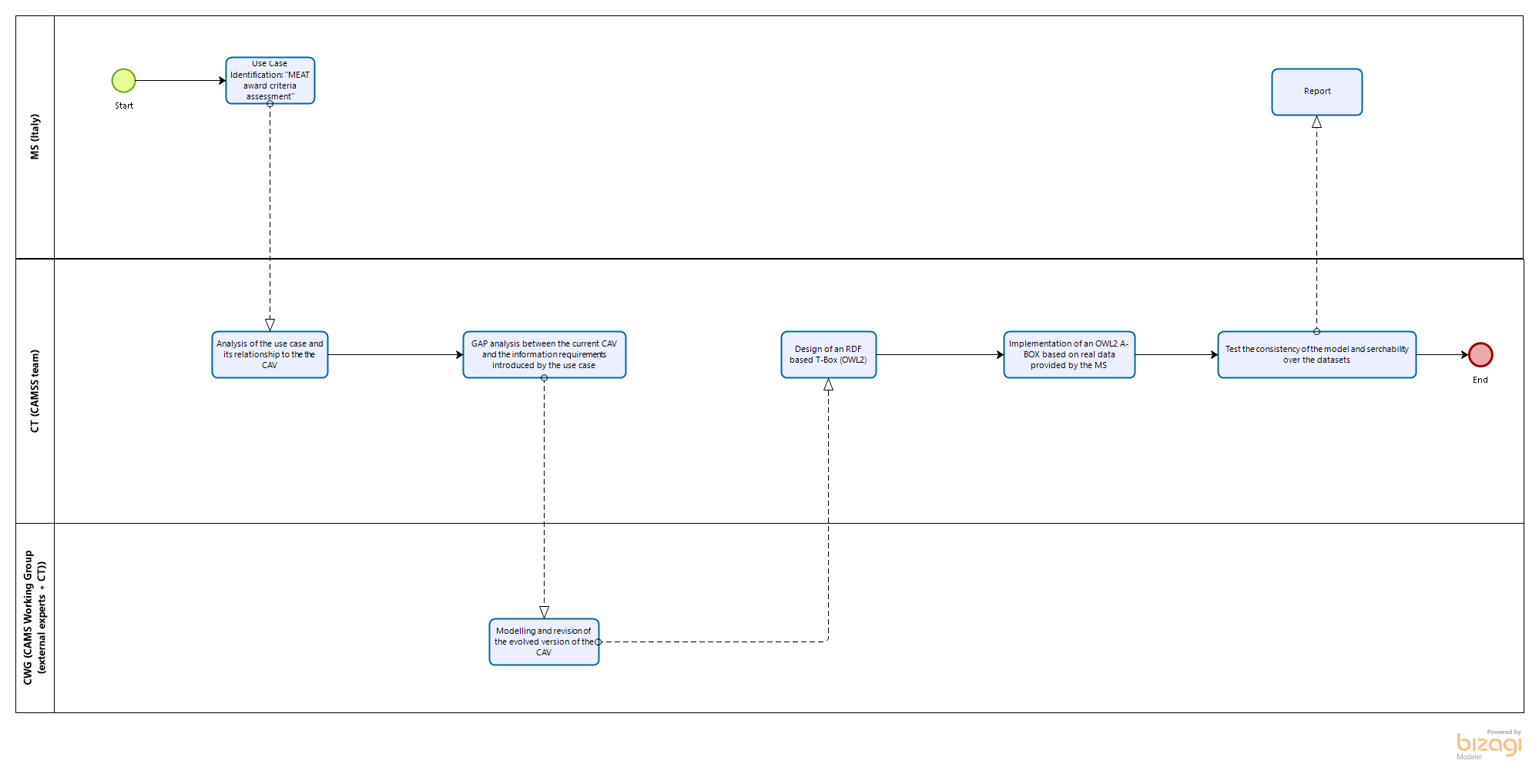


Figure 1 CAV Pilot Roadmap

## Core Assessment Vocabulary Application Profile

Taking as reference the dataset provided by Italy (an evaluation tool[[2]](#footnote-2) based on the ELECTRE method), the following conceptual data model was produced to represent a CAV-AP design:

### Data Model for the CAV Pilot

The figure below captures the extensions, customisations, codelists and restrictions (the Application Profile) used for the implementation of this pilot, on top of the CAV[[3]](#footnote-3).

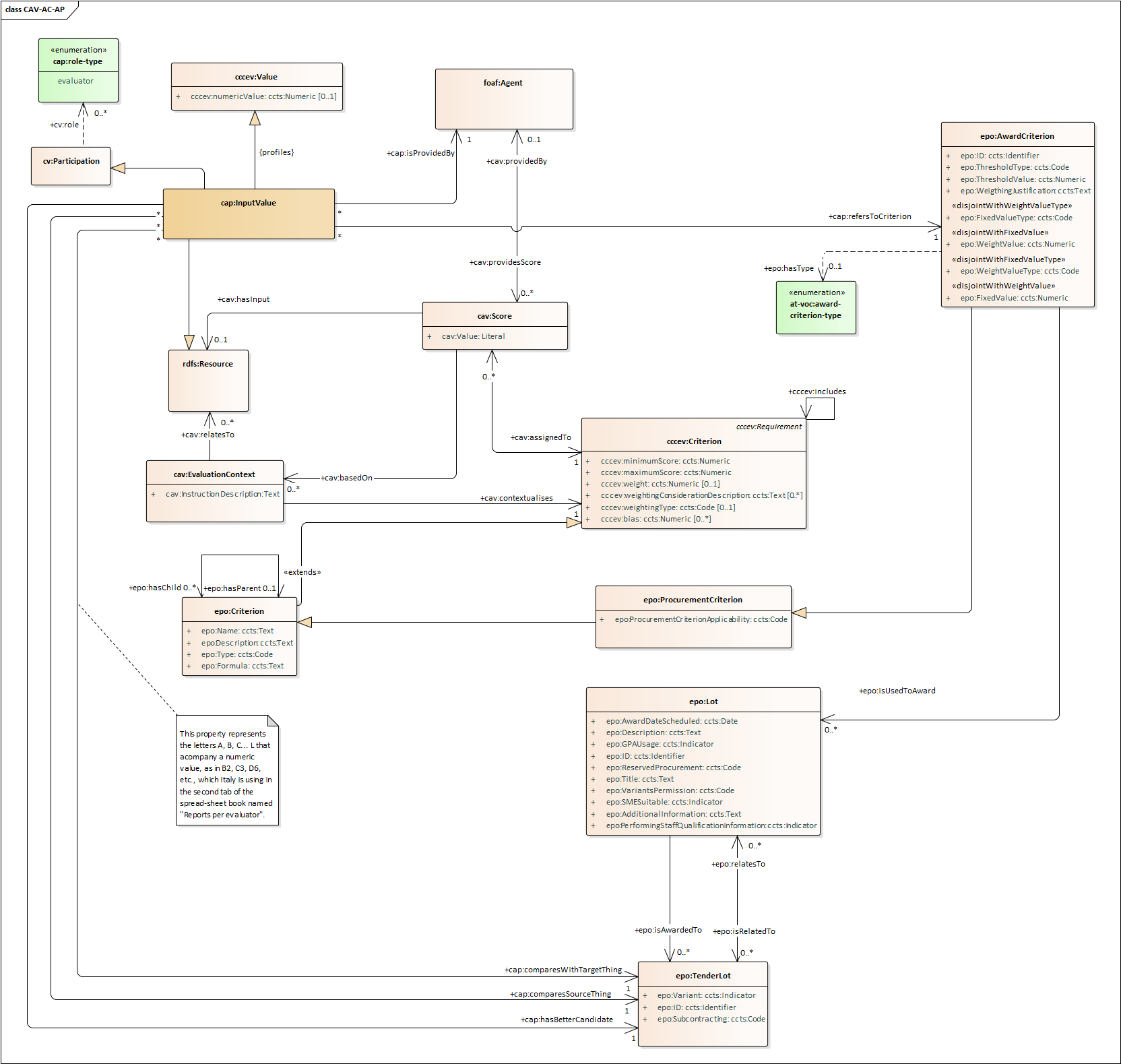


Figure 2 CAV-Application Profile Data Model

#### Interpretation

A *foaf:Agent* is participating in the evaluation process and is playing the role of an evaluator. The evaluator compares the different ***TenderLots*** being evaluated within a procurement procedure.

A ***TenderLot*** is the tender (the bid) submitted by a tenderer for one of the Lots specified in a procurement Procedure.

From this comparison, each evaluator selects which is a better ***TenderLot***candidate assigning a preference value, *cap:InputValue,* to the different ***TenderLots***compared[[4]](#footnote-4)*.* For example, evaluator prefers *a* against *a’* with a preference value of *p*, where:

* *a* and *a’* represent two different ***TenderLots*** amongst all the ones submitted, and for each criterion defined in the procurement documents; and
* *z* is the subjective value assigned by the evaluator to the TenderLot that best fits the criterion. The value ranges between a minimum a maximum threshold established in the call for tenders (e.g. 1 to 6 for criterion 1, 1 to 3 for criterion 2, etc.), and it is used to calculate the relative weight of the preferred TenderLot in relation to each criterion.

The ***AwardCriteria***used for the assessment of the MEAT are grouped in two ‘dimensions’ (codelist *epo:award-criterion-type*), quality and price. These award criteria are used to award the winner *epo:TenderLot*.

### T-Box and A-Box of the CAV Pilot

The UML diagram, representing the pilot CAP-AP design, has been designed as an OWL Turtle file[[5]](#footnote-5) ‘*[TBox](https://github.com/isa-camss/CAV/blob/master/pilot/cav-ac_tbox-empty.ttl)*’.

The dataset provided by the Italian MS was loaded from a slightly modified version of the [spread-sheet](https://github.com/isa-camss/CAV/blob/master/pilot/py/evaluation-tool.xlsm) tool[[6]](#footnote-6) provided by Anticorruzione, and transformed into OWL Turtle (the ‘ABox’) via a python [script](https://github.com/isa-camss/CAV/blob/master/pilot/py/main.py)[[7]](#footnote-7).

Both were imported in a locally installed GraphStore and tested, via simple SPARQL queries.

### SPARQL queries

Based on the use case **MEAT Award Criteria Assessment, a** basic user story and competency question were defined and implemented as a SPARQL queries to test the coherence and consistency of the model, as well as the searchability over the dataset:

* **User Story**: I, as the algorithm calculating and ranking which is the winner of Lot1 of the procurement procedure X, want to get all input values introduced by all evaluators for this procedure's lot and concerning exclusively 'Qualitative Subjective Criteria', so assessment and decision-making can be based on the results.
* **Competency Question**: Retrieve all the input values assigned by all evaluators to 'Qualitative Subjective Criteria', as well as the information related to the input value.
* **SPARQL Query**: see [file](https://raw.githubusercontent.com/isa-camss/CAV/master/pilot/py/select-input-values.sparql) in repository[[8]](#footnote-8):

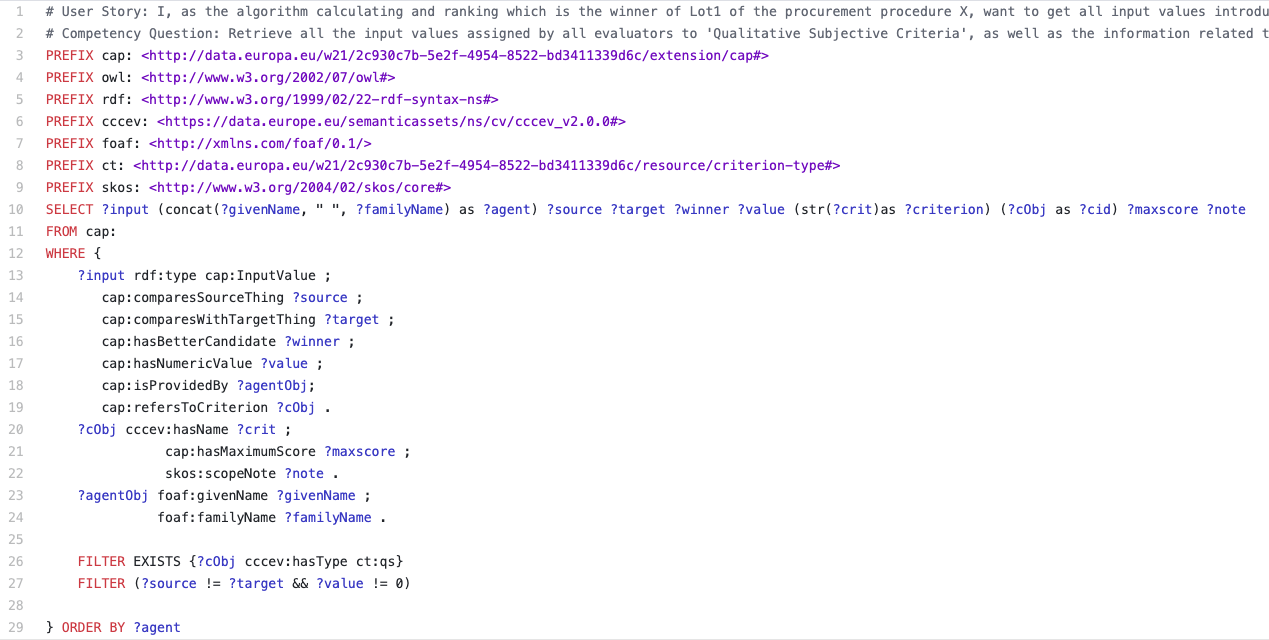


Figure 3: SPARQL Query

### MEAT selection and data visualization

Upon execution of the SPARQL Query, any software development could use the resultset to apply the algorithms described in the ELECTRE methods.

To show this possibility, this pilot has implemented an *ad hoc* version of just the ‘Phase I’ of the method using a python [script](https://github.com/isa-camss/CAV/blob/master/pilot/py/electre.py) [[9]](#footnote-9)(which is sufficient for the first round of comparisons, to outstand the better ***TenderLots*** and grasp obvious differences between them). The objective of this exercice is simply to demonstrate that the CAV-AP modelled and implemented in this pilot is fully functional and usable.

The whole process of data extraction, transformation, loading, querying, TenderLot comparison and results visualization has been documented in a Jupyter Notebook [pipeline](https://github.com/isa-camss/CAV/blob/master/pilot/py/ELECTRE-PHASE-I-DEMO.ipynb)[[10]](#footnote-10), which can be executed and read step by step:

|  |  |
| --- | --- |
|  |  |

Figure 4: Pipeline and visualisation

# Conclusions

* The objective of the current pilot has always been put the CAV in a context of reality and check whether it was complete or if it could be evolved. The dataset, formulae and documentation provided by Italy has shown that the CAV was quite complete but for a couple of attributes: score thresholds and the linking to external resources such as algorithms and formulae (see UML for these two details).
* The application of the CAV to the eProcurement eEvaluation phase, which is what Italy is trying to do, requires taking the CAV and extending it with the ePO specialized classes of criterion. It also requires the use of a reification class (e.g. “InputValue”) to relate all the data about evaluation decision. But this is what application profiles are expected to do in respect of eGovernment Core Vocabularies.
* Modelling and implementing the algorithms and formulae that intervene in the calculation of the results and the issuing of a report goes beyond the objectives of the current pilot. The model, however, sets a good foundation for approaching the development of a ‘Reference Implementation’ of the complete ELECTRE Methods.

# Annex I – Documentation Provided by Italy

* Italy dataset:



* Methods to calculate the MEAT (ELECTRE method page 26):



1. AVCP, ‘Autorità per la Vigilanza sui Contratti Pubblici di Lavore, Servizi e Forniture’, Italy. [↑](#footnote-ref-1)
2. See spreadsheet tool used in: <https://github.com/isa-camss/CAV/blob/master/pilot/doc/Italy-provided/evaluation%20tool.xlsm> [↑](#footnote-ref-2)
3. See UML files in: <https://github.com/isa-camss/CAV/tree/master/pilot/uml>. [↑](#footnote-ref-3)
4. For an exhaustive description of the ELECTRE Methods please see the documentation provided by IT in this repository folder: <https://github.com/isa-camss/CAV/tree/master/pilot/doc/Italy-provided> (‘Quad.07.12.11.\*). [↑](#footnote-ref-4)
5. TBox: <https://github.com/isa-camss/CAV/blob/master/pilot/cav-ac_tbox-empty.ttl> [↑](#footnote-ref-5)
6. Slightly modified IT evaluation tool: <https://github.com/isa-camss/CAV/blob/master/pilot/py/evaluation-tool.xlsm> [↑](#footnote-ref-6)
7. Main python script: <https://github.com/isa-camss/CAV/blob/master/pilot/py/main.py> [↑](#footnote-ref-7)
8. SPARQL query: <https://raw.githubusercontent.com/isa-camss/CAV/master/pilot/py/select-input-values.sparql> [↑](#footnote-ref-8)
9. Ad hoc Phase I ELECTRE calculation: <https://github.com/isa-camss/CAV/blob/master/pilot/py/electre.py> [↑](#footnote-ref-9)
10. Jupyter Notebook pipeline: <https://github.com/isa-camss/CAV/blob/master/pilot/py/ELECTRE-PHASE-I-DEMO.ipynb> [↑](#footnote-ref-10)