

PPROCI Ontology

Miguel Sancho Peña, Jai Grover

February 2, 2019

Abstract

In this work we extend the Pulic Procument ontology (PPROC) that models the tendering and contractual part of a public procurement process. In particular, we incorporate the terminal, invoice, part of the execution phase into the model. This allows for an end-to-end picture of the procurement and a more complete matching of initial bid to final deliverable; this is relevant for transparency, as the latter is a common pit-fall of public processes which frequently run over-time and over-cost.

Contents

1	Introduction	2
2	Ontology Development Process	2
2.1	NeOn Methodology	2
2.2	Ontology Specification	3
2.3	Ontology Schedule	4
2.4	Ontology Re-use and Patterns	5
3	Ontology Structure	5
3.1	Invoice Structure	6
3.2	PPROC Structure and patterns	8
4	Ontology Implementation	11
4.1	OOPS Ontology Evaluation	12
5	Linked Data	13
6	Conclusion	14
7	Appendix:	
	Ontology Requirements Specification Document (ORSD)	16
7.1	Purpose	16
7.2	Scope	16
7.3	Implementation Language	16
7.4	Intended End-Users	16
7.5	Intended Uses	16
7.6	Ontology Non-Functional Requirements	16
7.7	Ontology Functional Requirements	17

7.7.1	Competency Questions	17
7.7.2	Key Words	17
7.8	Non-Ontological Resources	18
7.9	Ontological Resources	18
8	Appendix: Ontology Class Structure	20

1 Introduction

Public procurement refers to the process by which a public agency obtains goods or services from other sectors. The use of public funds for procurement naturally creates a need for accountability in the process; for example, through the timely and transparent publishing of procurement data. While this has always been an important general principle, technological, and social, developments related to the internet and open data have brought the issue to the fore.

Driven by these considerations the Public Procurement (PPROC) [7] ontology, was developed to provide a common framework for the formal specification and analysis of public procurement processes. The PPROC ontology is focuses on the interaction between public and other (often private) institutions towards the resolution of a *public contract*. This concept of a contract forms the core of the PPROC ontology with peripheral concepts then being organised into contributing factors (eg parties involved, initial negotiation, final fulfilment).

However the PPROC ontology does not extend fully to the fulfilment phase of the procurement process; in particular it does not incorporate billing information, reflected in invoices. The fulfilment phase is interesting in the context of transparency as irregularities in the duration, expenses, and activities performed by the contracted organisation could show up as mismatches between the obligations recorded in the public contract and the fulfilment data recorded in the invoice. In this ontology, which we refer to as PPROC*i*, we seek to address this issue by extending the PPROC schema to include *invoice related concepts*.

We develop the report as follows. First, in Section 2 we shall elaborate on the development process we have followed for the PPROC*i* ontology. We then describe in Section 3, the structure we have settled on for the PPROC*i* ontology. In Section 4 we discuss the implementation of the ontology in protégé and test the ontology using the OOPS toolset [6]. In Section 5, we describe the transformation of a public invoice dataset from JSON into RDF. We then conclude in Section 6 summarising the work we have done and suggesting some possible future lines of investigation. In Appendix 7 we provide a standard ontology requirement specification document (ORS*D*) that summarises the report. And in Appendix 8 we provide a more detailed look at the classes we have used for this ontology and their relations.

2 Ontology Development Process

2.1 NeOn Methodology

To develop the ontology we shall follow the NeOn methodology [9, 8], depicted in Fig. 1. The NeOn methodology provides a guideline for the ontology development process emphasising the use and re-use of related ontological, and

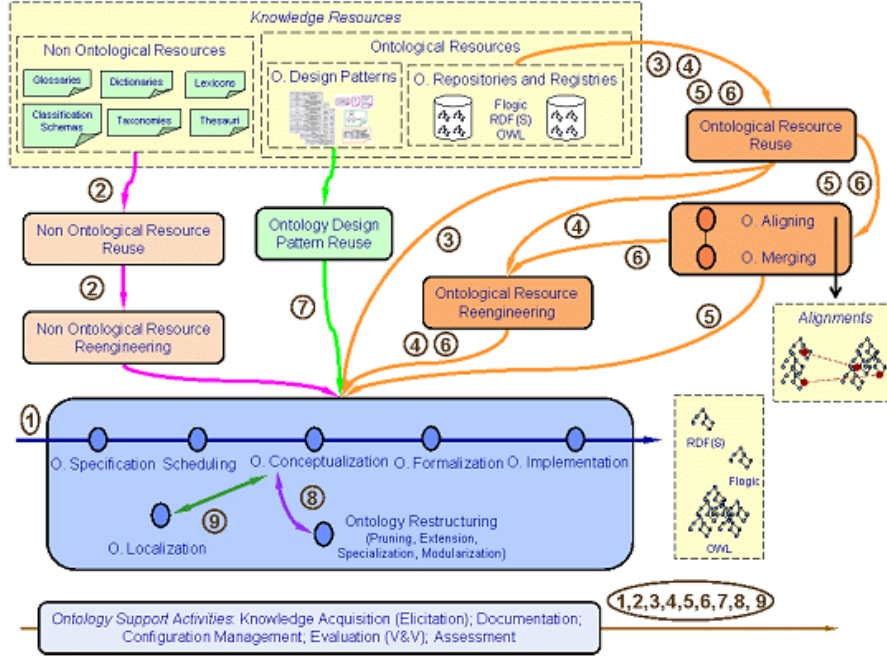


Figure 1: A diagram illustrating the stages of the ontology development process following the NeOn methodology. Each route through the NeOn process corresponds to a different scenario 1-9. We follow a combination of

non-ontological, resources. The extent, and manner, in which these resources can be re-used for a particular ontology problem is delineated across nine NeOn scenarios.

For our particular case we shall be following a combination of: Scenario 9, adopting a design pattern from the PPROC ontology that will prove useful to model invoices; and Scenario 5, where we will re-use the PPROC ontology and merge it with our pattern-based invoice ontology. We describe in some more detail the particular NeOn activities we will need in the preamble to Section 3.

2.2 Ontology Specification

The first step in the NEON methodology is to generate a specification document (ORS); we provide this in an appendix, but here progress through the salient points.

The focus of this work is on providing a schema within which to publish information on the fulfilment of public procurement processes; in particular to conceptualise the invoices that pertain to completed public contracts. The latter are taken to be formulated already within the existing PPROC ontology [7].

The territorial scope is taken to be the city of Zaragoza [?]. Zaragoza has implemented a comprehensive campaign to provide transparent access to the data on the city's public procurement as part of a 'responsible social policy' [?]. Of particular interest for us is the data on invoices (registro de facturas) [?]

supplied by the city in XML/CSV/JSON formats.

Given these two references (PPROC and Zaragoza) we shall look to them to determine the complexity and scope of our ontology. In particular, we shall first take the approach of using PPROC as a reference ontology and building PPROC_i as a local ontology that reuses many of the PPROC concepts and extends the *contract* class structure through a corresponding *invoice* class. We shall prune parts of the resulting ontology that are not relevant to conceptualise the Zaragoza data we are using, or to a list of competency questions that we imagine our end-users could be interested in.

We envisage these end-users as ranging from: public authorities and private organisations, looking for detailed information on the statistics of public contract properties and potential irregularities; and also the general public looking perhaps for information on projects that affect them and basic transparency. The users will be taken to be residing in Spain (due to the Zaragoza dataset) and as such we expect a bilingual framework to be necessary with Spanish and English as the functional languages. Eventually the territorial scope could be extended to include the European Union (EU). A motivation for this is the work on an EU wide public procurement schema, *eprocurement*, that is currently underway [?]. The changes to the language and data set information would then possibly necessitate modifications to the ontology which we try to anticipate but not exhaustively.

When considering competency questions we consider two broad classes: Questions that are essentially of a statistical nature, probing separately basic facts about the invoices, or potentially also the contracts they relate to; and questions that are investigating the coherence of the invoices in relation to the contracts. The latter is targeted at issues of inappropriate use of public funds which may show up as discrepancies between contracted and billed activities.

Some statistical questions are:

- List of invoices by type, date, amount paid
- List of invoices by contracted organisation

Some investigative questions are:

- The difference between contracted and actual amount fees
- The difference between contracted and actual duration

2.3 Ontology Schedule

We include a Gantt chart here Fig. 2 showing the time frame we allotted and largely followed for the development of this ontology. It is not more detailed in part because this would be misleading given our complicated schedules in January; as such a coarse structure allowed for more flexible time management while still having approximate goals to target.

As for the overall life-cycle model of the ontology, it would ideally follow a 5-step waterfall type of approach. It is unclear whether it would ever need a fourth maintenance phase though. We essentially only pass through the other three phases: initial, design, implementation.

pproci Ontology Development Schedule: Weeks in January				
Stage	Week 1	Week 2	Week 3	Week 4
Ontology Specification Document	X			
Ontology Conceptualisation	X			
Ontological Resource Search	X	X		
Ontology Alignment		X		
Ontology Assessment		X		
Ontology Pattern Search			X	
Ontology Merger			X	
Ontology Implementation in Protege			X	X
Ontology Testing in OOPS				X
Linked data conversion from Zaragoza JSON to RDF				X
Import data into Ontology				X

Figure 2:

In this table we show the plan we have followed in developing the PPROCi ontology. It evolved over approximately 4 weeks each with its own set of tasks. These were fairly coarse-grained with overlap to ensure that goals could be met inspite of potential complications or changes of plan.

2.4 Ontology Re-use and Patterns

Our ontology re-use will be principally the PPROC ontology, in order to obtain the relevant domain specific concepts. Where general concepts are necessary they have already been incorporated into PPROC so we will re-use these by proxy as well. Some re-used ontologies that we will need are: Good Relations, w3c Organization, Public Contracts Ontology, Schema.org, and Dublin Core Metadata Terms (dcterms). A list of the re-used ontologies together with their relevance can be found in the ORSD in Section 7.

3 Ontology Structure

Our strategy for constructing the ontology involved working incrementally through several stages, roughly inspired by the NeOn methodology. First we use a bottom-up approach to map the invoice data we wish to model to corresponding classes, thereby *conceptualising* the ontology.

We then consider ontology re-use of the PPROC ontology. Here we need to decide which classes within PPROC, and its own re-used ontologies, we will need for our problem; in so doing, we *align* the ontologies. Where relevant we modify the classes and similarly modify our ontology to remove classes made

```

{
  "id": "1",
  "title": "SUMINISTRO DE AGUA BRUTA DEL CANAL IMPERIAL DE ARAGON DEL 23 AL 30 DE NOV. 2011",
  "entidad": {
    "id": "23780",
    "title": "SERVICIO DE EXPLOTACIÓN DEL AGUA POTABLE",
    "uri": "http://www.zaragoza.es/api/recurso/sector-publico/organismo/23780"
  },
  "tercero": {
    "id": "1657",
    "title": "CDAD-GRAL-REG-CANAL-IMP-ARAGON ",
    "cif": "G50153600"
  },
  "facturaElectronica": "N",
  "amount": 34243.42,
  "ejercicio": 2012,
  "issued": "2012-01-05T00:00:00Z",
  "payment": "2012-05-08T00:00:00Z",
  "status": "Pagada"
}

```

Figure 3: This image shows a representative element of the Zaragoza open invoice data set related to public contracts [5].

redundant by the re-use of PPROC; this corresponds to *assessing* the ontology.

What follows is a stage of ontology pattern re-use, where we modify our remaining class structure to match a pattern shown in the PPROC ontology. This makes the structure more likely to be stable and the overall ontology more transparent. Finally we *merge* the ontologies.

We describe each of these stages in turn with respect to the ontology structure, noting the principal classes for the ontology in each case. The detailed class structure is given in Appendix 8. The final ontology structure is shown in Fig. 5.

3.1 Invoice Structure

We will link this ontology to current public contract and invoice data sets sourced from the Zaragoza open data platform. Given our emphasis on modelling invoices, the restriction of scope to Zaragoza, and the prior existence of PPROC for modelling the contracts, our first task was to correctly mirror the Zaragoza invoice data in our ontology classes.

The Zaragoza invoice data, in its original JSON format, is structured as in Fig. 3. The concepts we need to model the invoice are therefore:

- Invoice
- Contract
- Public organisation
- Contracted organisation
- Amount
- Billing period
- Date of payment
- Date of issue
- Activity

The structure we initially adopt is shown in Fig. 4. The core class is that of the invoice, which forms an n-ary relation with the contract class and several other relevant classes representing concepts from the invoice. The invoice class

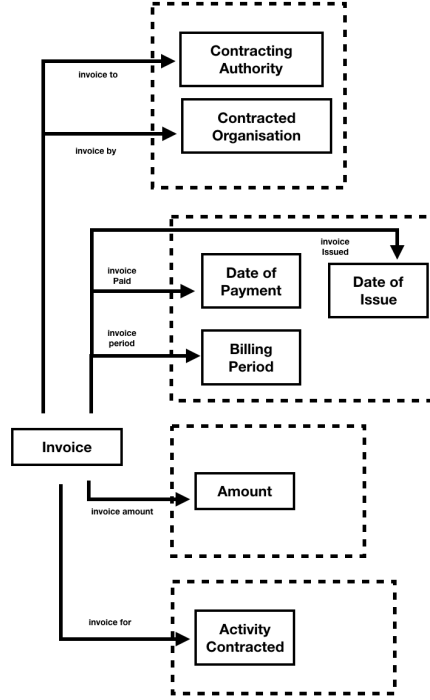


Figure 4: We show the initial data driven ontology conceptualisation. The classes here capture the data fields that were found in the Zaragoza invoice data set [5] and Fig. 3.

is in the domain of several data properties: in particular: `invoice title`, `invoice status`, `invoice uri`, `electronic invoice`. The `invoice id` is used to label each invoice instance.

The `contract` class is reused from the PPROC ontology. Here we expect that the relevant contract for the invoice will be determined via the `invoice title` data property of the invoice.

The `contracting authority` ('entidad'), and `contracted organization` ('tercero'), classes are linked to the invoice class. The data properties they lie in the domain of are `organization id`, `organization title`, and `organization uri`.

The `amount` class is also linked to the invoice, referring to the billing 'amount' in the invoice. As is the `billing period` class, referring to the fiscal period in which the contract was drawn - ie matching the 'ejercicio' field of the invoice.

The `date of payment` and `date of issue` are classes linked to the invoice. They are related to the 'payment' and 'issued' data fields of the Zaragoza invoice data sets.

Finally we introduce an `activity contracted` class to represent the activities that the invoice is payment for. This, possibly, is reflected also in the `invoiceTitle` data property so might be redundant.

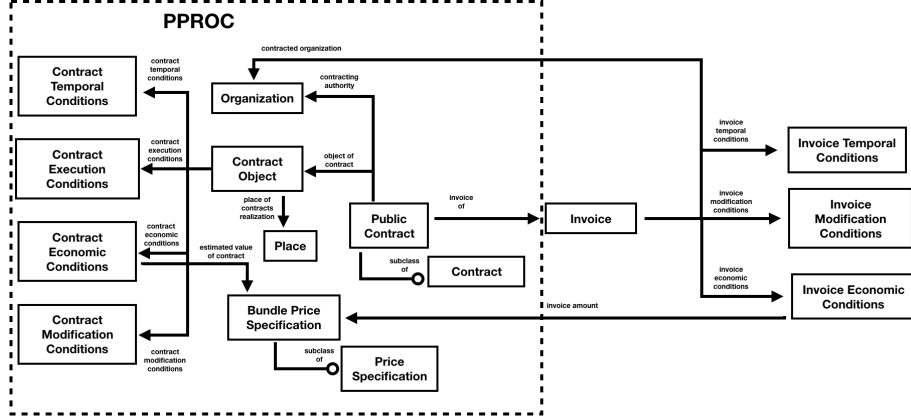


Figure 5: We show the final ontology for PPROC_i. Here we only include the classes from PPROC that are most relevant to our invoice data and competency questions. These classes and their data properties are described in more detail in [7], and in Section 8. Note that only the classes and not their data properties are shown here. The data property relations can be found in Figs. 6, 7, and 8.

3.2 PPROC Structure and patterns

Since we will be re-using parts of the PPROC ontology we briefly review the relevant (to this ontology) parts of its structure; we refer the reader to [7, 3] for more details. We describe some modifications to our ontology that are motivated by the re-use of PPROC, as well as the modifications we make to PPROC in merging the ontologies. The result of this stage is shown in Fig. 5. In addition a summary of the relevant classes and their structure can be found Section 8

A coarse, high-level, organisation of the PPROC classes is according to the various facets of the contract process: the *procedure* involved in negotiating the contract; the *contract* itself and the way it is specified; the institutional *parts* of the process; and the *fulfilment* of the contract. We can ignore parts of the ontology related to the procedure as this does not (at least for this exercise) pertain to the invoice we are modelling.

First consider the classes related to the contract itself. The core of the PPROC structure is the **contract** class re-used from the Public Contracts Ontology (PCO). It contains some basic information about the contract itself and is a n-ary relation to a number of other sub-classes and classes. In particular, the **public contract** sub-class is the only one we will need.

The **public contract** class already has information about the contracting authority, which is described through the **organization** class that PPROC has reused from the w3c organization ontology. As a result we do not need to introduce this class ourselves. Moreover we can make **contracted organization** a link to the **organization** class instead of a class in itself. Where necessary we also add several data types to the **organization** class.

The details and conditions of the contract can be found in the **contract object**. Here we find a pattern which will prove useful to re-use when structuring the invoice data. Namely the partition of the relevant contract con-

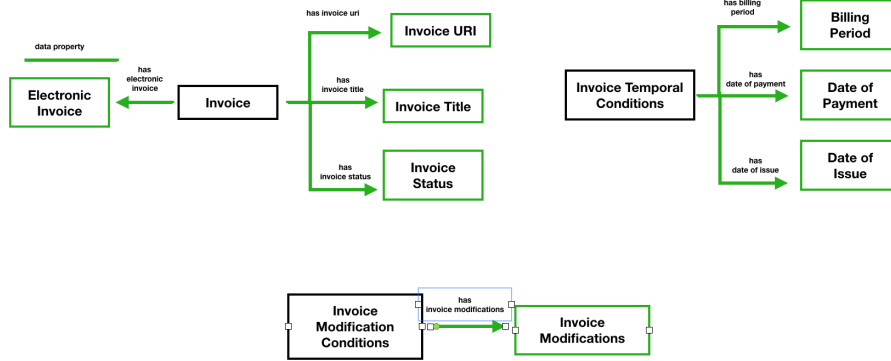


Figure 6: Here we show the data properties associated to the classes that we have included to describe invoices.

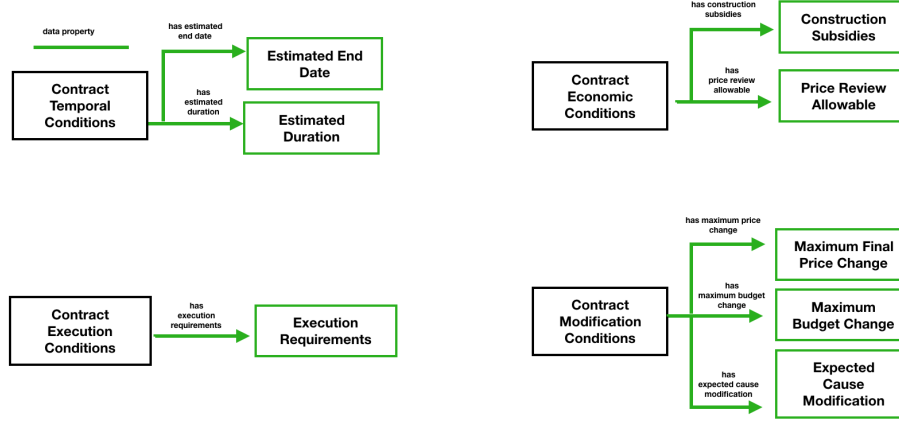


Figure 7: We provide here the data properties linked to the re-used PPROC domain classes that are most relevant to this ontology.

ditions into those that relate to **economic** aspects of, **temporal** aspects of, **modifications** to, and the **execution** of the contract. These fit into contract classes according to the pattern **contract XXX Condition** where **XXX** is one of the above conditions.

We simply replace 'contract' with 'invoice', and introduce the corresponding classes into our ontology. Within these broad condition-type classes it is straightforward to fit all our previous classes as appropriate data properties. This is depicted in Fig. 5 and Section 8.

We also find it useful to keep several general classes from PPROC. In particular price-value/fee information via the class **bundle price specification**. Note that the **bundle price specification** class is a sub-class of the re-used Good Relations ontology **price specification** class. The need for a separate class is based on the need to account for potentially multiple services 'bundled' into the contract price.

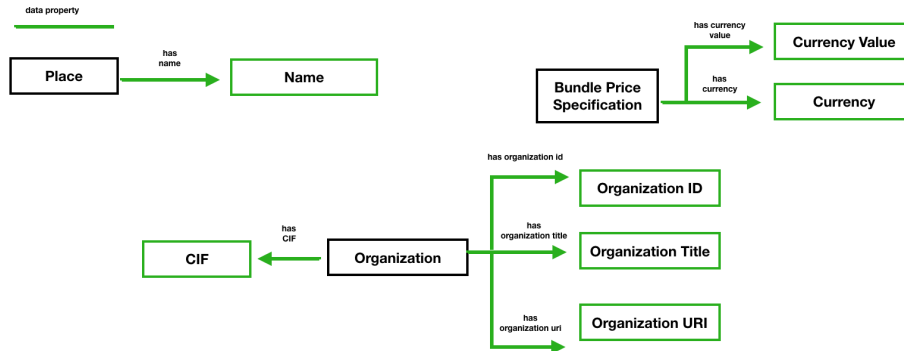


Figure 8: We provide here the data properties linked to the PPROC general classes that are themselves re-used and are most relevant to this ontology.

In addition, location information - place where the contract and invoice is realised - will be described by the re-used schema.org **place** class. In our case this will be almost redundant since we restrict territorially to Zaragoza.

Note here in passing that within the contract condition classes, especially **modification** and **execution**, there are interesting data properties that allow for comparisons between expected activities, prices, durations and those final results reflected in the invoices. This is useful for the investigative purpose of this ontology.

For example we have time related data properties used in the **contract temporal conditions** such as **estimated End Date** and **estimated Duration** that can be compared against the **invoice payment/issue date**. Within PPROC the former two data properties are introduced as sub-properties of DC terms **date** and the w3c XML schema **duration** respectively.

4 Ontology Implementation

We have implemented our ontology in Protege [4]. We did this in two stages. First we defined an ontology based on the structure of the invoices found in the Zaragoza Open Data portal, and the structure of the PPROC ontology which we have re-used. This, strictly invoice related, ontology was implemented in Protege.

We then imported PPROC, Fig. 9. After suitably modifying PPROC, as described in the previous section, we arrived at a merged ontology, which we call PPROC*i* whose structure in Protege is shown in Fig. 10.



Figure 9: PPROC imported into Protege.

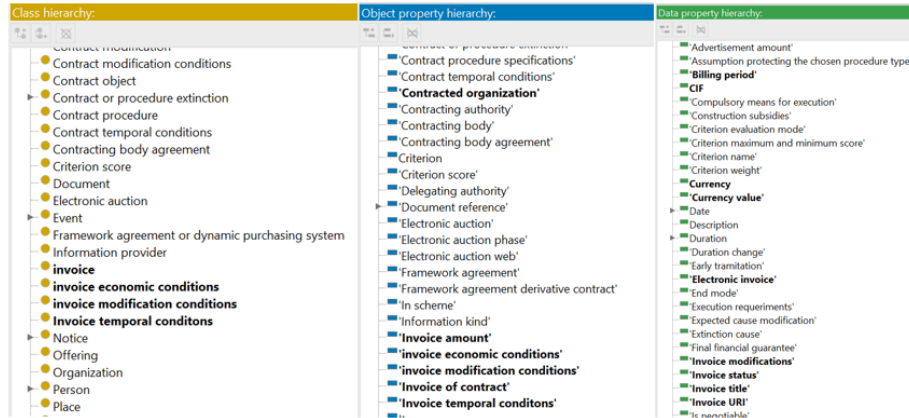


Figure 10: Classes, Object and Data properties of PPROC*i*.

Given that the, initial, territorial scope is the city of Zaragoza we added translations into Spanish for every class, object property and data property, as in Fig. 11; these were kept consistent with the style used in PPROC.

To test our ontology and create the RDF skeleton to produce Linked Data (see Section 5), we inserted a single invoice into our ontology; we then generalised this to introduce all the other invoice instances. We can see in Fig. 12 the data properties for each of the parts of the invoice and its data types.

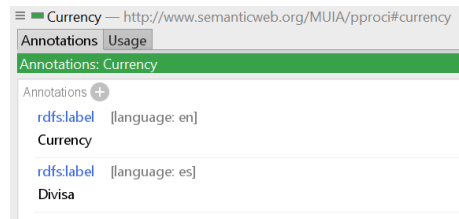


Figure 11: Translations for a data property.

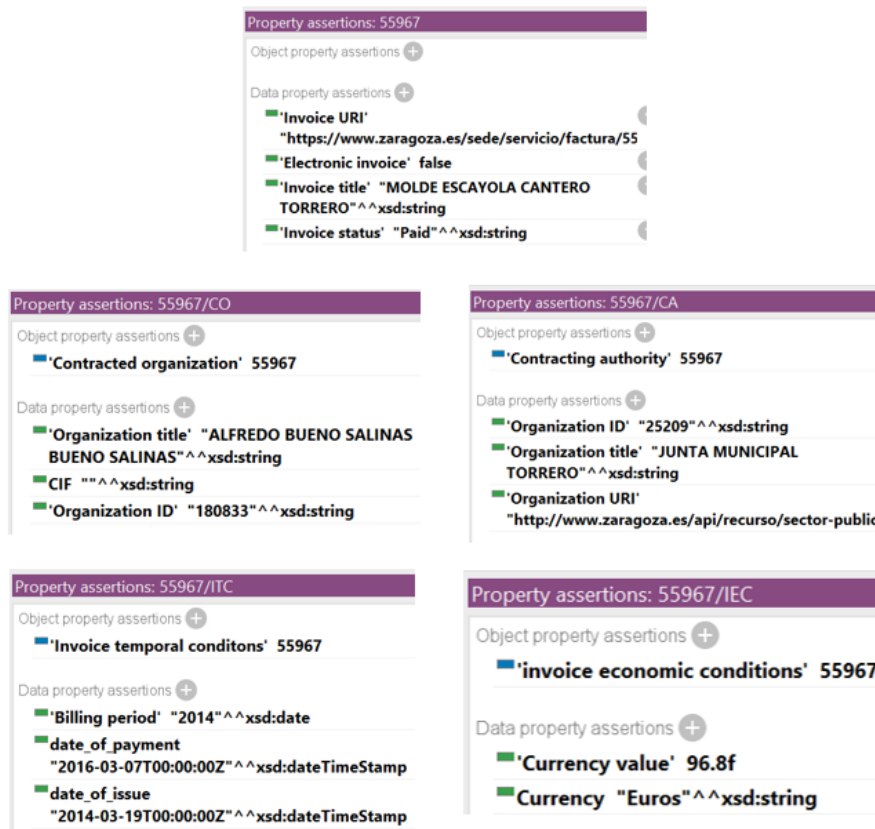


Figure 12: Data for an individual invoice in Protege.

4.1 OOPS Ontology Evaluation

We tested our ontology with OOPS [6]. We did not get any critical errors for our ontology, but we did get 3 errors coming from PPROC as we can see in Fig. 13. We did not want to make changes to PPROC so we have left these as they are.

[Expand All] | [Collapse All]

Results for P04: Creating unconnected ontology elements.	5 cases Minor 🟡
Results for P07: Merging different concepts in the same class.	1 case Minor 🟡
Results for P08: Missing annotations.	189 cases Minor 🟡
Results for P11: Missing domain or range in properties.	60 cases Important 🟠
Results for P12: Equivalent properties not explicitly declared.	1 case Important 🟠
Results for P13: Inverse relationships not explicitly declared.	84 cases Minor 🟡
Results for P19: Defining multiple domains or ranges in properties.	3 cases Critical 🔴
<p>The domain or range (or both) of a property (relationships and attributes) is defined by stating more than one <code>rdfs:domain</code> or <code>rdfs:range</code> statements. In OWL multiple <code>rdfs:domain</code> or <code>rdfs:range</code> axioms are allowed, but they are interpreted as conjunction, being, therefore, equivalent to the construct <code>owl:intersectionOf</code>. This pitfall is related to the common error that appears when defining domains and ranges described in [7].</p> <ul style="list-style-type: none"> • This pitfall appears in the following elements: <ul style="list-style-type: none"> > http://contsem.unizar.es/def/sector-publico/pproc#urgencyType > http://contsem.unizar.es/def/sector-publico/pproc#informationKind > http://contsem.unizar.es/def/sector-publico/pproc#procedureType 	
Results for P22: Using different naming conventions in the ontology.	ontology* Minor 🟡
Results for P24: Using recursive definitions.	3 cases Important 🟠
Results for P32: Several classes with the same label.	1 case Minor 🟡
Results for P41: No license declared.	ontology* Important 🟠
SUGGESTION: symmetric or transitive object properties.	1 case

Figure 13: Critical errors detected with OOPS.

5 Linked Data

We have based our ontology design on the structure found in the invoices from the Zaragoza Open Data Portal [5]. The invoice data is served in several different formats, however, at the moment, the RDF data is not working properly; the file returned by the endpoint does not contain any data. As a result, we chose to work with the JSON format. The motive for this decision is that, despite being a less common format for Open Data portals, which normally serve CSVs, JSON is very widely used in the rest of the industry and is a de facto standard in REST APIs.

Out of the wide range of available tools to generate RDF, we decided to use OpenRefine [2] as it was showcased in class. This tool was created by Google, and can load many relevant data types such as CSV, and JSON while also offering tools that assist users in curating the data. In our case, the Zaragoza invoice data set [5] we were working with was very consistent and did not require extensive curating; for example, it did not have any duplicates, nor did it contain typos or missing data.

OpenRefine does not natively support generating RDF. To work with RDF, an extension is needed. In our case, we chose to use the `grefine-rdf-extension` which is open source and can be downloaded from Github [1]. It is worth noting that this software will not work with the latest version of OpenRefine, but rather with the earlier version 2.7.

Although the extension worked correctly, Fig. 14, we could not map our data to the format required for our ontology. To solve this, we inserted some data into our ontology and extracted the RDF structure for an individual invoice by looking at the generated OWL file. We then implemented a Python script to

map our JSON data set to our specific RDF skeleton. During the process, we saw that although the URIs for the invoices were not included in the data set, we could build them ourselves by combining the URI for the Zaragoza Open Data portal and the invoice ID found in the data set.

	id	result	title	an	iss	sit	pa	eje	fac	en	result
1	58712	GAS MEXICO CENTROS DEPORTIVOS	18551.46	2014-06-12T00:00:00Z	Pagada	2015-01-19T00:00:00Z	2014	N	23960		http://www.zan-publico.org/en
2	57162	Contratación de 7 Camiones para el transporte de Aluminio Público Zona II periodo 01/01/2014 al 31/01/2014	147681.41	2014-06-28T00:00:00Z	Pagada	2015-01-19T00:00:00Z	2014	N	61297		http://www.zan-publico.org/en
3	61177	SERVICIOS DE MANTENIMIENTO EN EL CDM JOSÉ GARCÉS - JUNIO 2014	62342.64	2014-06-30T00:00:00Z	Pagada	2015-04-22T00:00:00Z	2014	N	20231		http://www.zan-publico.org/en
4	62979	ACTIVIDAD FINANCIACIÓN PROYECTOS EN ZARAGOZA ACTIVA	100	2014-06-30T00:00:00Z	Pagada	2015-01-14T00:00:00Z	2014	N	61290		http://www.zan-publico.org/en
5	63811	ELECTRICIDAD AGOSTO FUENTES ORNAMENTALES	20788.49	2014-06-12T00:00:00Z	Pagada	2015-01-28T00:00:00Z	2014	N	23960		http://www.zan-publico.org/en
6	60594	Otros reparos cubiertas, limpieza desmontado en Motos en 2.º piso Exp. 43135009	26135.25	2014-07-17T00:00:00Z	Pagada	2015-01-23T00:00:00Z	2014	N	23960		http://www.zan-publico.org/en
7	59967	MARQUE ESCUELA CANTERO TORREIRO	96.8	2014-03-19T00:00:00Z	Pagada	2015-05-07T00:00:00Z	2014	N	25209		http://www.zan-publico.org/en
8	61736	ACOMPAÑAMIENTO MUSICAL EN PLAZA SANTO DOMINGO 4-2014	300	2014-04-04T00:00:00Z	Pagada	2015-01-19T00:00:00Z	2014	N	61278		http://www.zan-publico.org/en
9	59735	REPARACIÓN FRIGORÍFICO	47.8	2014-06-23T00:00:00Z	Pagada	2015-12-03T00:00:00Z	2014	N	20247		http://www.zan-publico.org/en

Figure 14: JSON data set loaded in OpenRefine.

6 Conclusion

In this report we have provided a description of the PPROC_i extension to the PPROC ontology for describing public contracts. The focus of the extension is the modeling of invoices related to PPROC contracts, and, in particular, we have worked with invoices obtained from an open data set in the city of Zaragoza. We implemented the ontology in the Protege software [4] and tested it with the OOPS tool. Finally, we have converted the invoice data set from Zaragoza from JSON to RDF and linked it to our ontology.

To follow up on this work there are several natural lines. One would be to ensure consistency with European standards, and initiatives such as the eproc ontology in development. Given the re-use of the PPROC ontology and the patterns there-in, and that PPROC maintains consistency with broader European standards, we do not expect it to be difficult to modify PPROC_i to these ends. Another would be to run queries against PPROC_i to make sure that we can actually address all the questions we established at the beginning of this work. This should not be difficult and may suggest changes (necessary or helpful) to the structure of PPROC_i.

References

- [1] Grefine rdf extension. <https://github.com/fadmaa/grefine-rdf-extension>.
- [2] Openrefine. <http://openrefine.org/>.
- [3] Pproc web description. <http://contsem.unizar.es/def/sector-publico/pproc1.0.0>.

- [4] Protege. <https://protege.stanford.edu/>.
- [5] Registro de facturas. <https://www.zaragoza.es/sede/servicio/catalogo/1440>.
- [6] M. Poveda-Villalon, A. Gómez-Pérez, and M. C. Suárez-Figueroa. "oops!(ontology pitfall scanner!): An on-line tool for ontology evaluation". *International Journal on Semantic Web and Information Systems (IJSWIS)*, 2014.
- [7] Jose Felix Munoz Soro, Guillermo Esteban, Oscar Corcho, and Francisco Serón. "pproc, an ontology for transparency in public procurement". *Semantic Web*, abs/1604.00289, 2015.
- [8] Mari Carmen Suárez-Figueroa, Asunción Gómez-Pérez, and Mariano Fernández-López. *The NeOn Methodology for Ontology Engineering*, pages 9–34. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
- [9] M.C. Suárez-Figueroa, A Gómez-Pérez, E Motta, and A Gangemia. "the neon methodology for ontology engineering". 2012.

7 Appendix: Ontology Requirements Specification Document (ORSB)

7.1 Purpose

This ontology will provide a model for investigating the execution phase of a public procurement process, focusing on modelling invoices.

7.2 Scope

We extend an existing public procurement ontology (PPROC) by incorporating aspects of the execution phase into the model; in particular adding invoice-related concepts to the ontology. We restrict the territorial scope to the domain of the city of Zaragoza, making use of their open data platform.

7.3 Implementation Language

This ontology will be implemented in the OWL language.

7.4 Intended End-Users

- Citizens
- Journalists
- Contract Authorities

7.5 Intended Uses

- Provide basic statistical information on invoices and hence end-to-end information on corresponding public contracts in Zaragoza.
- Analyse contract compliance by comparing contractual requirements with delivered actions reflected in invoice.
- Provide for transparency by offering information and tools for studying the properties of public contracts.

7.6 Ontology Non-Functional Requirements

This ontology will be focused on the Zaragoza territorial domain, perhaps with a later extension to a more general EU setting. As such it must:

- **NFR1:** Support multiple European languages, at least Spanish and English.
- **NFR2:** Be consistent with the EU norms where applicable; eg with existing public procurement standards or ongoing related projects. For us PPROC and if possible eproc.

7.7 Ontology Functional Requirements

7.7.1 Competency Questions

We have two broad categories of competency questions that this ontology should be able to address.

Statistically themed questions that allow for basic facts about the invoices and their related contracts to be extracted from the open invoice/contract data sets:

1. The number of invoices per unit time
2. The number of invoices per contracted organization
3. Contract(s) related to an invoice
4. List of companies by number of contracts
5. Details of the invoice: classified by the type, amount, beneficiaries, duration, modifications
6. Details of the contract: expected amount, possibility of modifications, expected duration, expected activities
7. List of contracts given to a particular public sector authority

We also have transparency and compliance related questions. These are typically going to be based on comparisons between contract and invoice information.

1. The difference between contracted and actual amount paid for the contract activities
2. The difference between contracted and actual duration of the contract activities
3. List of contracted, and contracting (public), organizations by time and money differences (expected versus actual)
4. Modifications made to contracted executables upon completion of the contract
5. Ratio of the number of contracts connected to a particular public contracting authority versus the average number of contracts per public authority
6. Ratio of the number of contracts connected to a particular private contracted organization versus the average number of contracts per private organization

7.7.2 Key Words

- | | |
|----------------|------------|
| • Contract | • Duration |
| • Organization | • Amount |
| • Activity | • Money |

- Invoice
- Modification

7.8 Non-Ontological Resources

We restrict, for this work, to the Zaragoza data set, in which case adopting international (or at least European) standards is not entirely relevant. Where it will be relevant is if we later seek to work with E.U. data sets. In which case we would follow PPROC in making use of several standards for presenting currency and time data.

Currency Classification

The International Organization for Standardization has a standard format (ISO-4217) for representing currency values and units. We expect to eventually represent currency data (budget price, fees, expected amount) in this format.

Date Representation

The International Organization for Standardization has a standard format (ISO-8601) for representing dates and times using numbers (YYYY-MM-DD). For a later extension of this work we would represent dates according to this standard.

Zaragoza Invoice Data Set

The Zaragoza open data platform has a data set available for the invoices related to public contracts. We shall be primarily grounding our ontology in this data set.

URL: <https://www.zaragoza.es/sede/servicio/catalogo/1440>

Zaragoza Public Contract Data Set

The Zaragoza open data platform has a data set available for the public contracts that authorities in the city have negotiated.

URL: <https://www.zaragoza.es/sede/servicio/catalogo/147>

7.9 Ontological Resources

The principal resource we will re-use is the PPROC ontology, or at least particular parts of it. This ontology in turn has already incorporated several other general and domain ontologies relating to concepts such as places, time, prices, and organizations.

We list below these ontologies including also the general ones already re-used in PPROC for completeness.

PPROC Public Procurements Ontology

A domain specific ontology that will be the primary ontology on which we base this work. We shall reuse and extend PPROC.

URI: <http://contsem.unizar.es/def/sector-publico/pproc.html>

Public Contracts Ontology

Ontology about public procurement developed by the Czech Open Data initiative. Allows for the core **contract** class to be defined.

URI: <http://purl.org/procurement/public-contracts>

Organization Ontology

Ontology recommendation developed by W3C to describe organizational structures. This ontology is used to define contracting authorities, as well as contracted organizations.

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

Good Relations Ontology

Ontology that allows for price specification.

URI: <http://purl.org/goodrelations/v1>

Schema.org

This is a general ontology used to describe places (and other basic concepts that we have not tracked within PPROC).

URI: <http://schema.org/>

Dublin Core Metadata Terms

Another general ontology used to describe time related concepts (among others that PPROC re-uses it for that we haven't tracked).

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

8 Appendix: Ontology Class Structure

In this section we provide information on the different classes, and their relations, that will be most relevant for analysing the invoice data. For each we indicate via a superscript whether it is a class (c), object property (op), or data property (dp).

Contract^c	
Superclass of:	public contract ^c
In domain of:	object of contract ^{op} , invoice of ^{op} contracting authority ^{op}

Table 1: Contract class from Public Contracts Ontology pc:contract.

Public contract^c	
Subclass of:	contract ^c
In domain of:	object of contract ^{op} , invoice of ^{op} contracting authority ^{op}

Table 2: PPROC Contract class.

Contract Object^c	
In domain of:	contract execution conditions ^{op} , contract temporal conditions ^{op} contract modification conditions ^{op} , contract economic conditions ^{op} place of contracts realization ^{op}
In range of:	object of the contract ^{op}

Table 3: PPROC Contract object.

Invoice^c	
In domain of:	invoice modification conditions ^{op} , invoice temporal conditions ^{op} invoice economic conditions ^{op} , contracted organization ^{op} invoice uri ^{dp} , invoice status ^{dp} , invoice title ^{dp} , electronic invoice ^{dp}
In range of:	invoice of contract ^{op}

Table 4: Invoice class.

Invoice modification conditions^c	
In domain of:	invoice modifications ^{dp}
In range of:	invoice modification conditions ^{op}

Table 5: Invoice modification conditions class.

Invoice economic conditions^c	
In domain of:	invoice amount ^{op}
In range of:	invoice economic conditions ^{op}

Table 6: Invoice economic conditions class.

Invoice temporal conditions^c	
In domain of:	billing period ^{dp} , date of payment ^{dp} , date of issue ^{dp}
In range of:	invoice temporal conditions ^{op}

Table 7: Invoice temporal conditions class.

Contract temporal conditions^c	
In domain of:	estimated end date ^{dp} , estimated duration ^{dp}
In range of:	contract temporal conditions ^{op}

Table 8: PPROC Contract temporal conditions class.

Contract economic conditions^c	
In domain of:	construction subsidies ^{dp} , price review allowable ^{dp}
In range of:	estimated value of contract ^{op} contract economic conditions ^{op}

Table 9: PPROC Contract economic conditions class.

Contract execution conditions^c	
In domain of:	execution requirements ^{dp}
In range of:	contract execution conditions ^{op}

Table 10: PPROC Contract execution conditions class.

Contract modification conditions^c	
In domain of:	maximum budget change ^{dp} , maximum price change ^{dp}
In range of:	expected cause modifications ^{dp} contract modification conditions ^{op}

Table 11: PPROC Contract modification conditions class.

Bundle price specification^c	
Subclass of:	price specification ^c
In domain of:	currency value ^{dp} , currency ^{dp}
In range of:	contract economic conditions ^{op} , invoice economic conditions ^{op}

Table 12: PPROC Bundle price specification class. Modified to include the data properties listed in the table.

Price specification^c	
Superclass of:	bundle price specification ^c

Table 13: Price specification class from the good relations ontology. gr:priceSpecification.

Place^c	
In domain of:	name ^{dp}
In range of:	place of contracts realization ^{op}

Table 14: Place class from schema.org, s:place.

Organization^c	
In domain of:	organization id ^{dp} , organization title ^{dp} , organization uri ^{dp} CIF ^{dp}
In range of:	contracting authority ^{op} , contracted organization ^{op}

Table 15: Organization class from w3c org:organization. Modified with the data properties listed above.