**IRIS ONTOLOGY & NEW TERRITORY CONCEPTS WITHIN THE CITY OS UMBRELLA ONTOLOGY MODEL**

**SUMMARY**

[1. Acronyms and definitions 3](#_Toc498590493)

[1.1 Acronyms 3](#_Toc498590494)

[1.2 Definitions 3](#_Toc498590495)

[2. Introduction 5](#_Toc498590496)

[3. THE IRIS ONTOLOGY 6](#_Toc498590497)

[3.1 Ontology Design Methodology 6](#_Toc498590498)

[3.1.1 Scope 6](#_Toc498590499)

[3.1.2 Modelling 6](#_Toc498590500)

[3.2 Design Principles 11](#_Toc498590501)

[4. NEW TERRITORY CONCEPTS WITHIN THE UMBRELLA ONTOLOGY 12](#_Toc498590502)

[5. CONNECTION WITH THE UMBRELLA ONTOLOGY AND POSSIBLE EXTENSIONS 13](#_Toc498590503)

[6. VALIDATION OF THE INTEGRATED IRIS and TERRITORY ontologies within the cityos UMBRELLA ONTOLOGy 14](#_Toc498590504)

[7. ontology metrics 15](#_Toc498590505)

[8. Related documents 17](#_Toc498590506)

[9. Versioning 18](#_Toc498590507)

# Acronyms and definitions

## Acronyms

Below, the list of acronyms sorted alphabetically which are used in this document.

| **Term** | **Definition** |
| --- | --- |
| OWL-DL | An OWL entailment profile whose vocabulary allows specifying enumeration classes, property values, disjoint classes, boolean class operations, cardinality constraints, and complex classes. Based on SHOIN(D) description logic. More powerful and less more complex than OWL-QL. Decidability of quary answering open for the general case. |
| OWL-QL | An OWL2 entailment profile that contains the intersection between OWL-DL and RDFS. Query answering in LOGSPACE. |
| REST API | A RESTful API is an application program interface that uses HTTP requests to GET, PUT, POST and DELETE data. |
| UML | The **Unified Modeling Language** (**UML**) is a general-purpose modeling language in the field of software engineering, that is intended to provide a standard way to visualize the design of a system. Standard adopted by the Object Management Group (OMG). |
| MIB-CRM | Modelo de información de base. Application developed by IMI.  Customer Relationship Management. |
| RDF | **Resource Description Framework**  Graph template designed to describe web resources and their metadata, so as to enable the automatic processing of such descriptions. |
| OWL | **Web Ontology Language**  Language of constructed knowledge on the RDF data model. It provides means to define web ontologies structured. Its second version became a recommendation from the W3C at the end of 2012. |

## Definitions

Below, the list of terms sorted alphabetically which are used in this document.

| **Term** | **Definition** |
| --- | --- |
| Protege | Open source ontology editor and knowledge management system. |
| Hermit / Reasoner | Reasoner for ontologies written using OWL. A reasoner is a piece of software able to infer logical consequences from a set of asserted facts or axioms. |
| Entity / Class | Abstract objects that are defined by values of aspects that are constraints for being member of the class. |
| Relationship | Relationships between objects in an ontology specify how objects are related to other objects. |
| Restriction/Constraint | Formally stated descriptions of what must be true in order for some assertion to be accepted as input. |
| Semantic Web | The Semantic Web is an extension of the Web through standards by the World Wide Web Consortium (W3C). The standards promote common data formats and exchange protocols on the Web, most fundamentally the Resource Description Framework (RDF).  According to the W3C, "The Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries. |
| Ontology | An ontology is defined as “a formal, explicit specification of a shared  conceptualization” |
| SPARQL | SPARQL is a query language and a protocol that allows you to search, add, modify or delete RDF data available through the Internet. His name is acronym which means SPARQL protocol and RDF Query Language. |

# Introduction

This document includes :

1. The IRIS ontology for the CityOS umbrella ontology;
2. The completion of new territory concepts for the CityOS umbrella ontology, starting from the territory concepts already defined;
3. The connection between the IRIS ontology and the CityOS umbrella ontology.

# IRIS ONTOLOGY

## Ontology Design Methodology

## Scope

The scope of this ontology was well defined by the existing IRIS tool. IRIS tool is a tool provided by the Ajuntament de Barcelona, whose main purpose is to serve as a complaint, suggestion and consultation tool for the citizens, groups, or any other legal entity of the city.

In the IRIS tool, user can choose between:

(a) filing a complaint or suggestion, or

(b) requiring information about the functioning of the city and municipal services.

To do this, one must first choose a category among all the responsibility areas managed by the city. Areas of responsibility manage area components, each of them consisting of many possible “details”. The detail chosen by the user defines the **user request** and can be processed via several possible sequences of **tasks** (which form a **workflow**).

Each task corresponds to a profile and has a status associated with it. User request also has a **status** that depends on the status of the tasks. The functionality of the tool is well defined, as are the possible type of requests that a user can fill in.

Concretely, we first studied the documents received from IMI and tried to understand what the data sets and the variable names meant. This was the first step to figure out what are the concepts handled by the IMI in the implementation of the IRIS system. The files are the following:

* MIB-CRM: noves dades IRIS - Visió.
* arees\_tematiques.xlsx. This is the document that lists the thematic areas for IRIS.
* Servicios REST IRIS: this file was not relevant to the ontology development because it explains the implementation of IRIS´ REST API.
* A list of the tables with IRIS data in CityDB (IMI), together with their column names. Many of the names were not clear and we discussed them with IMI to clarify them.

The next step was to play with the IRIS website and understand its functionality, and how the concepts identified in the first step may map to what is visible in the tool. This raised many additional questions, in particular about possible relationships between concepts.

## Modelling

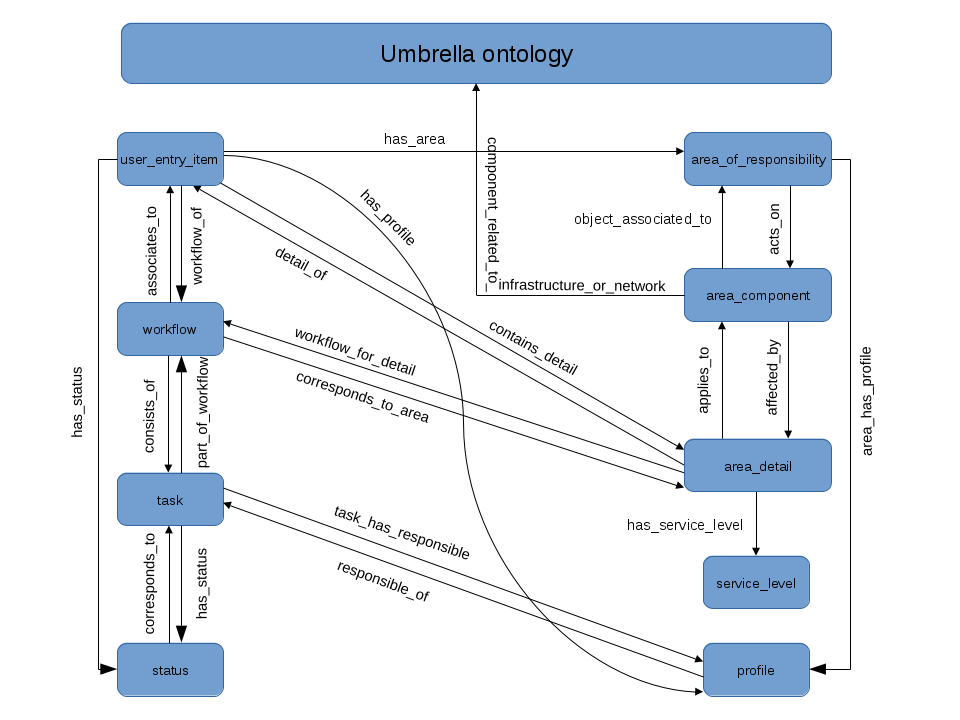
The main conceptual step consists in understanding and extracting the ontology concepts - entities, relationships – and their interactions and properties. We felt that we lacked a more global understanding of the system and its internal workings and we met with Everis – the designers of the IRIS tool, who were very helpful in explaining all the processes underneath the visual user interface. This step was important because it helped to understand that there were relationships not evident from the documents, but also execution flows that could not be represented as meta-data - and therefore not best captured by the ontology – for instance the derivation process.

The IRIS system handles the following types of **entities.** The words in bold are the names of the entities as defined in the ontology.

* **user**: individuals (**citizen**), **group**, and **legal\_entity** (e.g. companies). These are the types of actors that can fill in a user request.
* User request (**user\_entry\_item**): of one of two types - **complain\_and\_suggest\_form** or **inform\_and\_consult\_form**, each of them further divided in several subtypes.
* **area\_of\_responsibility**: the thematic area that corresponds to the user request. Different sets of areas exist for complain\_and\_suggest\_form and inform\_and\_consult\_form.These are areas such as **culture, education, environment, public\_health**, etc**.**
* **area\_component:** are subtypes of areas of responsibility, and have one or several details associated with them.
* **area\_detail:** corresponds to the most specific type of requests that a user can fill in.
* A **workflow** is associated to a detail. Each detail may have different valid workflows; it is the responsible for the user request who chooses one of these and assigns it to the concrete user request.
* A **task** is a step in a workflow.
* **input\_channel** and **response\_channel**: capture how the request was filed (by letter, web, or phone call) and how the user will receive the reply (letter, email, …).
* The **profile** of the person responsible for an area of responsibility, a user request, or a task in the workflow associated to an area detail. From most to least responsible, profiles are coordinators, area responsibles, or area operators.
* Each area detail has a required **service\_level**.
* Both user requests and tasks have a **status**.

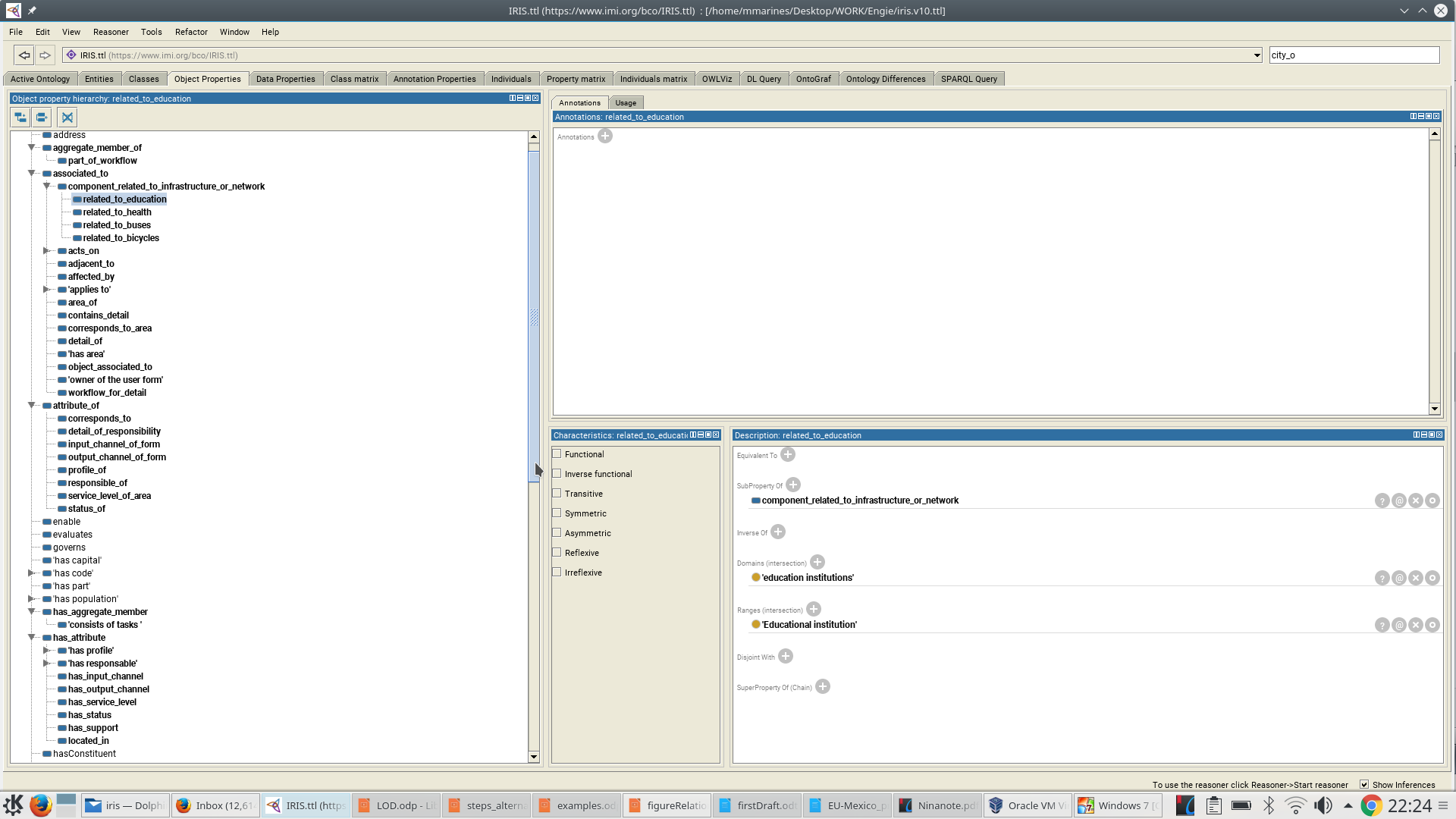
|  |  |
| --- | --- |
| **User request status** | **Task** |
| Pending validation |  |
| Planning | Planning |
| Decision | Decision |
| Pending reply | Reply |
| External processing | External processing |
| Closed |  |
| Canceled |  |
| Not processed |  |

The figure below captures the main **relationships** between these entities.

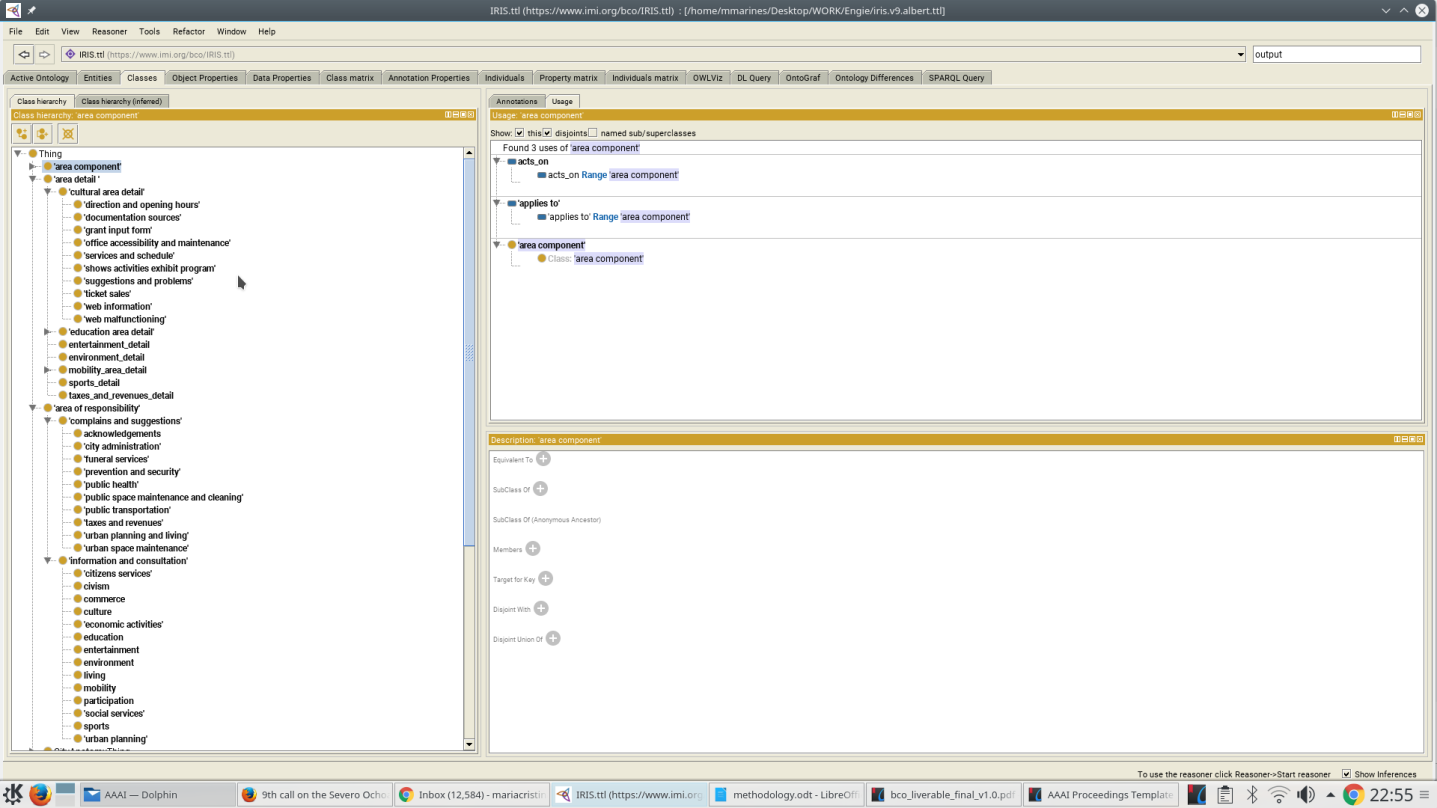


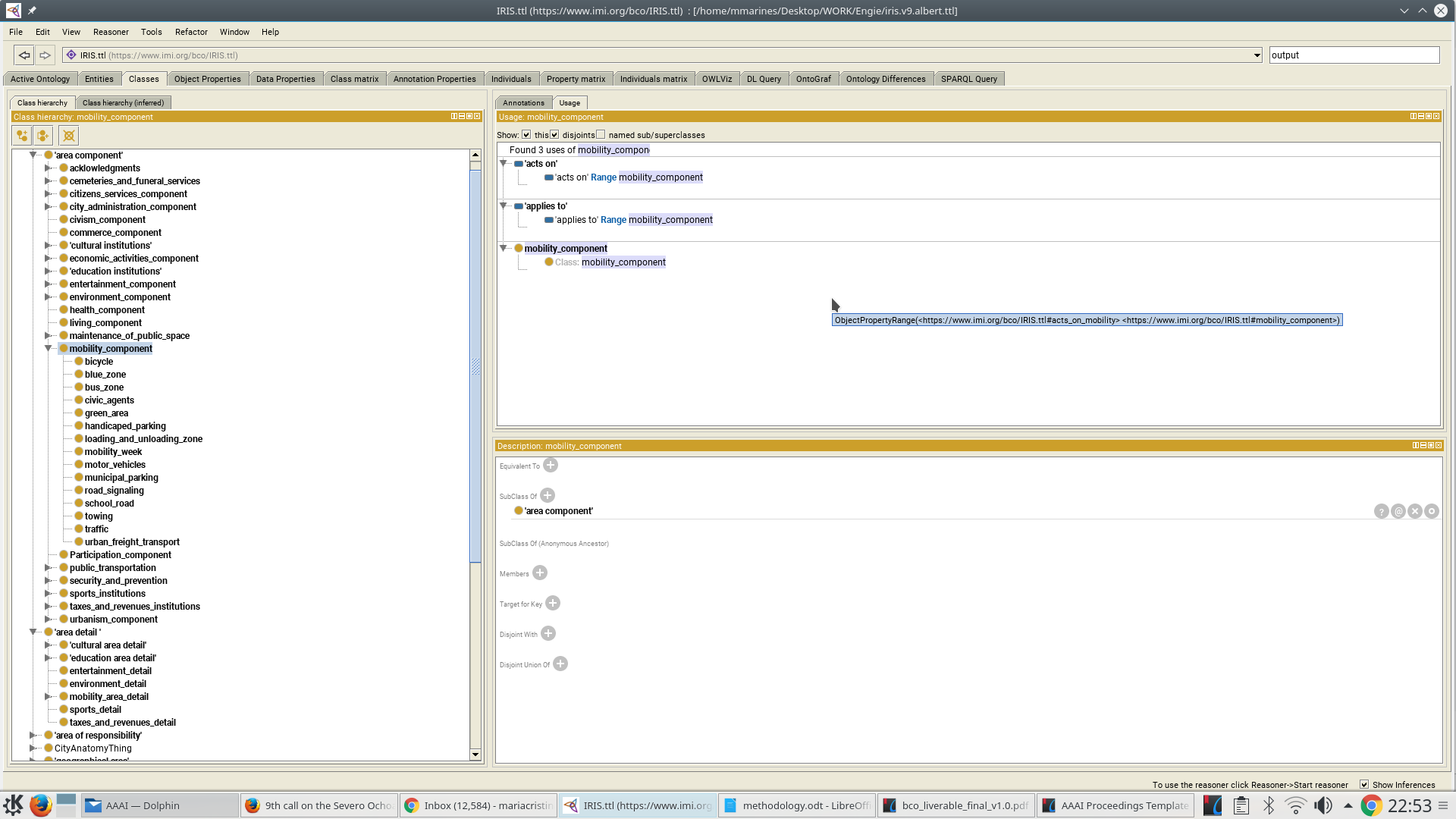
A user request has an area of responsibility (**has\_area**) associated with it, has a status (**has\_status**), an associated profile (**has\_profile**), and refers to a specific detail (**contains\_detail**). The area of responsibility **acts\_on** an area component and has a profile (**area\_has\_profile**). The area component is **affected\_by** an area detail and is currently the connection point with the umbrella ontology via the **component\_related\_to\_infrastructure\_or\_network** relation. The area detail **has\_service\_level** and has a workflow associated of the several possible ones (**workflow\_for\_detail**). A task has a responsible profile (**task\_has\_responsible**), it is part of the workflow – workflow **consists\_of** task – and has a status.

The figure below shows the relationships (object properties) as defined using Protégé (4.3).



**Note:** There are hundreds of area components and details defined in the IRIS tool. The two figures below show the area components, details, and responsibilities that are currently reflected in the ontology. These cover all responsibilities, all components in “complaints and suggestions”, a small subset of “information and consultation” components, and a small subset of area details. They also show a view of the hierarchy of classes for these entities.





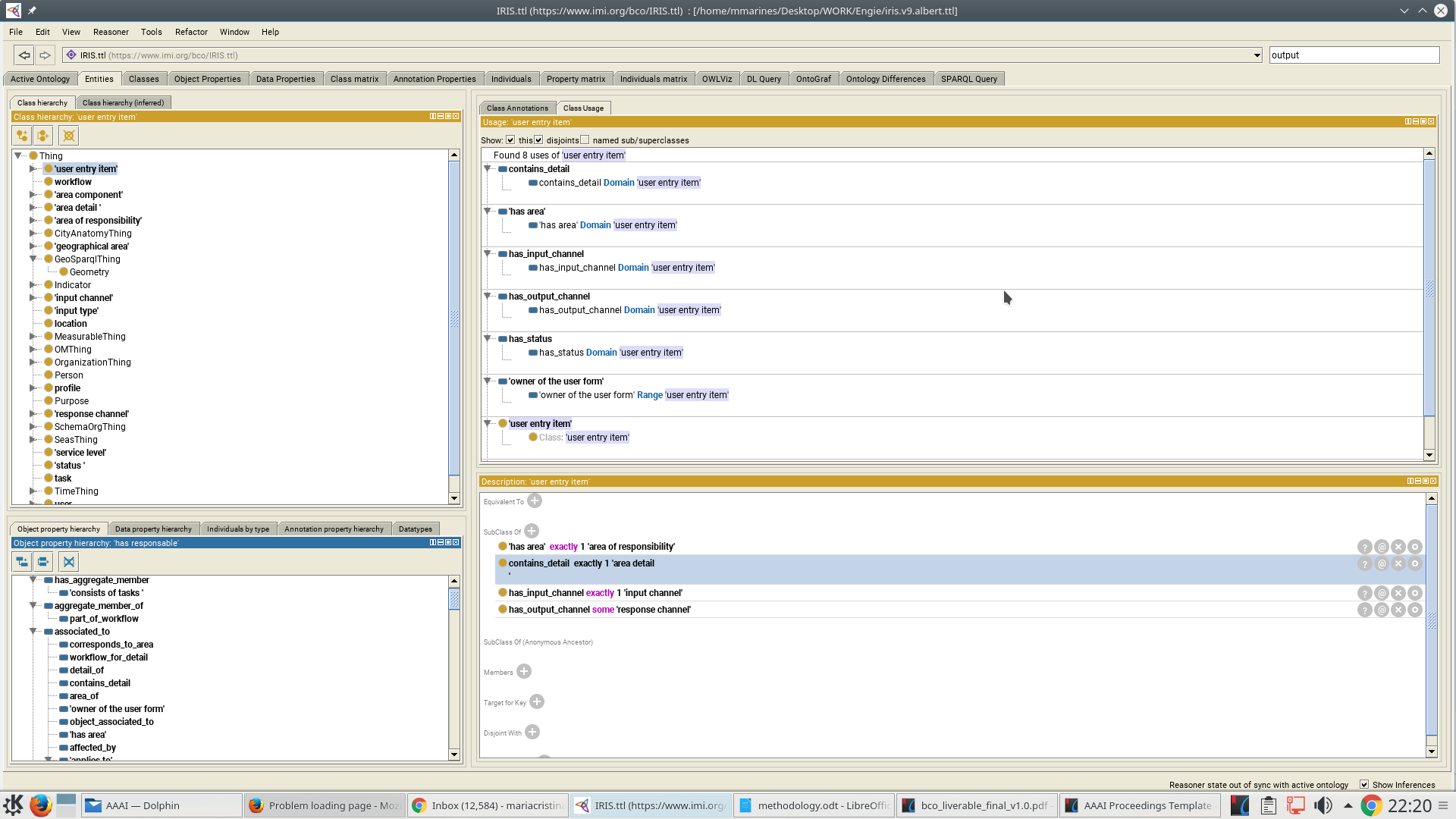
## Design Principles

One of the ontology design principles we are using is to **categorize object properties as one of the five UML relation types**: has attribute, attribute of, has aggregate member, aggregate member of, and associated to. The idea is to be able to perform certain inferences at a more abstract level.

Another principle we are following is to **use a simple dialect of OWL without constraints** (OWL-QL). This is generally more desirable in a context where city data comes from many heterogeneous – and sometimes noisy – sources, in which case defining constraints may prevent actual data from being integrated. This is, in our view, the most plausible scenario for CityOS.

For IRIS it may be the case that defining constraints does not arise any issues, neither of integration nor of scalability. As a result, some of the classes are defined via existential quantifiers or cardinality constraints – see the example in the figure below. For instance, **user\_entry\_item SubClassOf (has\_area exactly 1 area\_of\_responsibility)** defines a user request to correspond to exactly 1 area of responsibility via the relation has\_area. This constraint could also be defined by saying that the relation has\_area is functional.

The idea is to figure out in the evaluation phase with IMI whether these constraint types should be maintained or not. This issue of using OWL-DL vs OWL-QL is a decision that can be discussed when more insight is available.

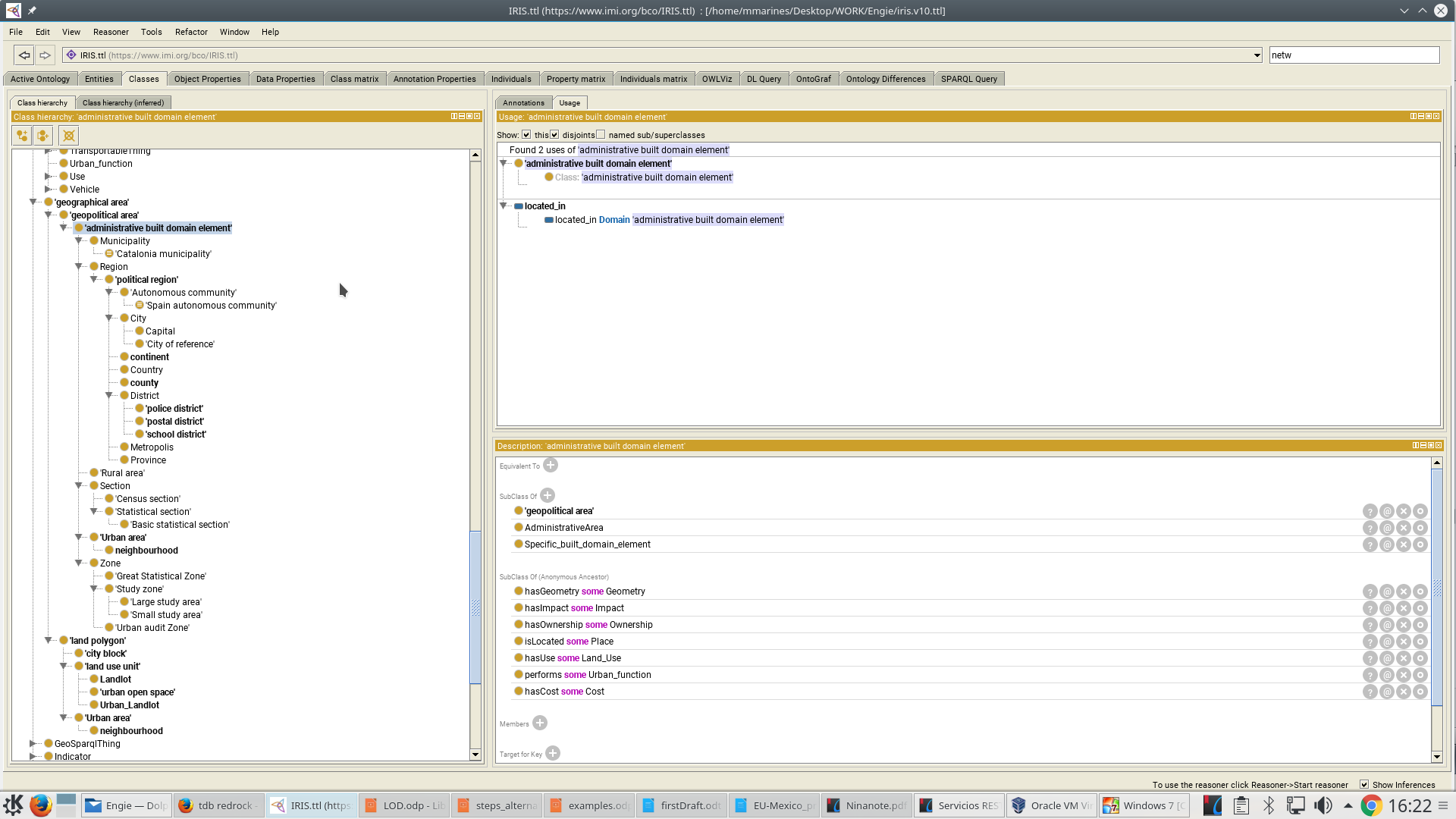


# NEW TERRITORY CONCEPTS WITHIN THE UMBRELLA ONTOLOGY

We extended and slightly reorganized the existent territory concepts in the CityOS umbrella ontology as shown below. In bold you can see the new concepts.

We studied the umbrella ontology and checked the notions it reflects about territory. In addition to the IRIS documentation, we could analyze the document on "Geocodificación y divisiones territoriales" (from IMI) to see which concepts about territory were missing and where to incorporate them in the high-level CityOS umbrella ontology ). In total a number of 18 entities have been added to the existing ontology.

Geographical areas now contain **geopolitical\_area**s and physical areas (**land\_polygon**). Physical areas are new and contain a hierarchy of **city\_block**s, **land\_use\_unit**s, and **urban\_area**s. Land\_use\_unit contain **landlot**s (i.e. parcela), **urban\_open\_space**s, and **urban\_landlot**s (i.e. solar). Geopolitical areas consist mainly of the old City OS concepts, to which we added new ones such as type of **political\_region**s: **continent**, **county**, types of districts (**police\_district**, **school\_district**, and **postal\_district**), and **urban\_area**s- for now **neighbourhood**s.



# CONNECTION WITH THE UMBRELLA ONTOLOGY AND POSSIBLE EXTENSIONS

The IRIS model imports directly the CityOS umbrella ontology developed and extends it with the IRIS and territory domains.

Currently there is little clear overlap between notions in these ontologies, which are limited to some entities in the mobility, health, and education domains. Some other domains as city administration and public space could also be connected to IRIS and the CityOS umbrella ontology. More generally speaking, the type of relationships that we started to formalize are those between area components in IRIS and infrastructure or network components in the high-level CityOS umbrella ontology.

Some other remaining components could also be connected to entities of type **interactions\_layer\_component** or **society\_layer\_component**, but these are very shallow hierarchies.

To summarize, the ontology model could be extended and completed in the future to create new relationships and concepts that would be useful for the final users.

# VALIDATION OF THE INTEGRATED IRIS and TERRITORY ontologies within the cityos UMBRELLA ONTOLOGy

We validated the ontology by running the Hermit 1.3.8 reasoner to check consistency, satisfiability, and class subsumption for the integrated ontology. That means we used the reasoner for classification of the ontology rather than query answering.

Inconsistencies appear, for example, if an instance of a class is defined to be of type of two disjoint classes. An example of unsatisfiability is to define a class that cannot have instances. The subsumption test generate the inferred class hierarchy. Failures appear in red in Protégé and are fixed by clicking on the explanation that the reasoner used (generated) for the inference, identifying the undesired step, and changing the ontology in a way that invalidates it.

For instance assume **culture** is defined to be a subclass of both **complaints\_and\_suggestions** and **information\_and\_consultation**, and these are defined to be disjunct. This is invalid as there connaot exist instances of such a class! We need to either define different “culture” classes or not enforce disjunctivity.

It is important that the client approves the results of this work in order to ensure that it matches expectations and specificities of the city of Barcelona.

Although we did not specify competency questions at the beginning of the process design, however, the ontology designed could answer several inter-esting questions. After populating the ontology, as shown below, we can either address some very generic queries or a very specific ones.

More scenarios could be defined by the client to check if the ontology can answer the identifed needs.

Some possible questions that may be asked over the current ontology are the following (assuming the MyCix tool implements the necessary filtering, query, and aggregation functionality):

1. Which are the days with most complaints?
2. What is the average number of complaints per day of the week?
3. What is the number of complaints per thematic area, per neighbourhood?
4. Give me the list of all the suggestions from individuals (NOT groups or other legal entities) about libraries in area X.
5. How many complaints are there files in a week, per type of input channel (phone, site, letter)?
6. What are the most commonly used workflows?
7. What are the most/least common profiles?
8. What is the ratio of complaints that get assigned to multiple profiles (within the workflow) rather than being solved by a single profile?
9. What is the number of complaints pending validation / canceled / closed in area X in the last week?

# ontology metrics

After having created the IRIS ontology model and having added some new territory concepts in the CityOS umbrella ontology, we currenly have 1360 axiomes, 637 logical axioms, 390 classes, 85 object properties, 21 data properties, and 6 individuals. Theses metric changes when the ontology is modifed.

The figure below shows the aggregated metrics for the CityOS umbrella ontology, after integrating the IRIS and territory ontology.

