**CITY Ontologies & normalization DESIGN**

**SUMMARY**

[1. Acronyms and definitions 3](#_Toc533005539)

[1.1 Acronyms 3](#_Toc533005540)

[1.2 Definitions 3](#_Toc533005541)

[2. Introduction 4](#_Toc533005542)

[3. Ontological model 5](#_Toc533005543)

[3.1 Works from City Protocol 5](#_Toc533005544)

[3.1.1 Introduction 5](#_Toc533005545)

[3.1.2 City Protocol details 5](#_Toc533005546)

[3.2 Deep dive in ontologies 6](#_Toc533005547)

[3.2.1 Fields knowledge model 6](#_Toc533005548)

[3.2.2 Services 7](#_Toc533005549)

[3.3 Overview of a methodology to create ontologies 7](#_Toc533005550)

[4. Normalization of data and processes, tools 9](#_Toc533005551)

[4.1 Web Semantic Language: XML, RDF and OWL 9](#_Toc533005552)

[4.1.1 XML 9](#_Toc533005553)

[4.1.2 RDF 9](#_Toc533005554)

[4.1.3 OWL 10](#_Toc533005555)

[4.2 Protégé: a common tool to build ontologies 10](#_Toc533005556)

[5. Architecture 12](#_Toc533005557)

[5.1 Functional architecture 12](#_Toc533005558)

[5.1.1 Introduction 12](#_Toc533005559)

[5.1.2 Web semantic data repository 13](#_Toc533005560)

[5.1.2.1 Protégé 13](#_Toc533005561)

[5.1.2.2 Data sources 13](#_Toc533005562)

[5.1.2.3 Data ingestion 13](#_Toc533005563)

[5.1.2.4 Preprocessor 13](#_Toc533005564)

[5.1.2.5 Data repository 13](#_Toc533005565)

[5.1.2.6 Query engine interface 13](#_Toc533005566)

[5.1.2.7 Query processor 13](#_Toc533005567)

[6. Related documents 14](#_Toc533005568)

# Acronyms and definitions

## Acronyms

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

| **Term** | **Definition** |
| --- | --- |
| 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** |
| --- | --- |
| 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 deals with the approach of ontology which applies to the city domain. The scope of this document includes:

* A first definition of an ontological model of the city, taking into account the work realized with City Protocol. This part also includes an overview of a methodology for the emergence af a model.
* A Normalization of data and processes, tools.
* The architecture design of the system related to ontology.

With the advent of web 3.0 (or web semantics), data sharing is at the center of concerns of public and private actors. The current digitization of the environment shows that crossing of data is at the origin of many creations of value: sharing of resources, savings, optimization of operation, etc.

Cross-referencing involves the use of a common language, known as the RDF (Resource Description Framework). It is the basic language of the semantic web, developed by the World Wide Web Consortium (W3C). The RDF relies on a simple and decorrelated storage scheme of the subject. The schema is based on a three columns table (subject, predicate, object). Below is the description of the structure:

* subject: resource to describe,
* predicate: property of the resource,
* object: value of the property (data or other resource).

The traditional data model, commonly used in SQL databases, is replaced by an ontology that describes the relations between triplets within the same table.

Writing ontologies is done using the OWL (Web Ontology Language) model, as recommended by W3C. The OWL is a knowledge representation language built on the RDF data model. It provides the means to define structured and standardized (and therefore shareable) web ontologies.

# Ontological model

## Works from City Protocol

## Introduction

Previous work about ontologies has been carried out by the City Protocol intitiative. The aim of the City Protocol is to propose a common foundation and understanding of the complexity and diversity of a city through the ontolgy.

City Protocol is a good start to define a city ontology, and so, it can be reused as a meta-ontology of the city. Each domain revealed by the City Protocol ontology must be deepened and turned towards the uses.

## City Protocol details

According to City Protocol, a city and all of its constituents could be described through three elements:

* The physical structure of the city (Structure),
* The living entities that it contains (Society),
* The flow of interactions and information (Interactions).

The **Structure** is composed of:

* **Environment**, which is composed of:
  + **Nature** (plants and animals),
  + **Air,**
  + **Soil**,
  + **Water**,
* **Infrastructures** (connective structures that enable people to get the resources they need), which is composed of:
  + **Communications network**,
  + **Water cycle** (clean and waste water),
  + **Energy**:
    - **Functional nodes** (nuclear and power plants, wind farms, biomass/bioenergy power plants, hydroelectric plants, solar fields),
    - **Energy networks** (electricity, natural gas, pipelines, ships),
  + **Matter cycle**, it includes:
    - **Matter extraction** from nature and their **transportation**:
      * **Goods,**
      * **Food** from vegetables and animals**,**
    - **Distribution**,
    - **Deliveries**,
    - **Consumption**,
    - **Waste generation**,
    - **Transporting waste to dumps**,
    - **Recycling or producing energy**,
  + **Mobility network** (human/goods transportation):
    - **Railways**,
    - **Airports**,
    - **Highways**,
    - **Roads**,
    - **Streets** (type depending of the type of mobility/transportation: pedestrian, **bicycles**, cars, public transportation, etc.),
  + **Nature** (green infrastructure).
* **Built Domain** (public and private):
  + **Object,**
  + **Dwellings,**
  + **Building,**
  + **Block,**
  + **Neighborhood,**
  + **District**,
  + **City**,
  + **Metropolis**,
  + **Country**,
  + **Continent**,
  + **Planet**.

**Interactions** (relations between the **Structure** and the **Society** reflecting the activities in the city) is composed of:

* **Functions** (living, working, education, shopping, transport, caring for health, etc.),
* **Economy** (commerce, trade, production, distribution, etc.),
* **Culture** (language, tradictions, beliefs, etc.),
* **Information** (Information System : city ontology, city operating system, city kpi, tools and applications, information portal, etc.).

**Society** is composed of:

* **Citizens** (person, family, organizations, businesses),
* **Government**.

## Deep dive in ontologies

As it has been explained in the previous chapter, City Protocol could be assimilated to a meta-ontology, which describes at high level the city anatomy. In order to have an operational ontology, it is necessary to go deeper in each ítems of the City Protocol ontology thanks to usage studying.

The following chapters presents the work that are the result of a preliminary study.

## Fields knowledge model

Fields contains all the classes needed to describe urban equipments of the city.

**Equipments.**

Equipments are real objects. They are devices that communicate, that means, transmit, receive or transform information. For example, camera, lightning, road lightning, variable message panel, etc. are urban equipments located in spaces of the city.

*Anchor with City Protocol – Structure – Built domain – Object.*

**Urban furnitures**.

Urban furniture is used as a support for equipment. For example, a pole is an urban furniture, and could be the support for equipments such as cameras or lightning.

*Anchor with City Protocol – Structure – Built domain – Object.*

**Spaces**:

Spaces are physical areas of a city or territory. For example, a building, a crossroad or a district are considered as spaces.

*Anchor with City Protocol – Structure – Built domain – \* (except object).*

*Anchor with City Protocol – Structure – Infrastructures – Mobility network.*

Equipments can constitute a network in the city. Most of the time, equipments deliver a service, and produce or receive information. These information could be exploited by the Information System of the city, in order to create analytics or indicators, for citizens, elects or services.

## Services

A Service is a prestation consisting in the provisioning of a technical or intelectual capacity. In the urban context, a service is a set of functions performed by equipments, in a specific space, during a period of time, for people or organizations.

A service could be caracterized by two classes:

* Function (or activity): illuminate, count vehicles, provide electricity, etc.
* Mission (objective to achieve): public lightning, safety, etc.

## Overview of a methodology to create ontologies

This global methodology aims to give an overview of the main steps that are needed to represent the territory of a city in a knowledge model. This model should reflect all the specificities and characteristics (multi dimensional) of the territory. Moreover, it should be normalized to facilitate its understanding and sharing.

The methodology is composed of the following main steps:

* Analysis and definition of the scope,
* Emergence, design and implementation of concepts,
* Maintenance and extension.

**Step 1: analysis and definition of the scope.**

This step is dedicated to the good comprehension of the perimeter to study. This step makes it possible to identify the domain and subdomain to deal with. It helps to organize data by domains and to define data governance. To success in the art of ontology, works should be realized with an open mindset: city data architects should be aware of existing models/ontologies on specific domains or subdomains, but also tangent and adjacent models.

In this step, use cases will be described in order to confirm the necessity of creating the ontology and data related to use cases provided.

**Step 2: emergence, design and implementation of concepts**

In this step, data model architects will design and implement concepts and relations between them:

* Definition of concepts/classes and their properties,
* Creation of hierarchy between concepts,
* Creation of categories,
* Creation of relations between concepts,
* Creation of ontologies in a dedicated tool as Protégé,
* Etc.

The tasks presented above are incremental and done with agility.

After the creation of the model, the ontology will be populated with the data to create the instances. That will allow to test, evaluate and validate the ontology.

**Step 3: maintenance and extension.**

After their publication in the real world, ontologies will be maintained, in order to face the potential bugs on the model. In addition, a particular attention will be given to the extensions of the model, which will have to reflect the changes of the real world. The construction phase for maintenance and extension, will be similar to the step 2.

The diagram below represents the methodology to elaborate the ontology of the city:



The methodology to create ontologies is described in more detail in the document “*Focus on the Ontologies Engineering Methodology.pptx”*, which link is inclued in Chapter 6 of Related Documents.

# Normalization of data and processes, tools

The methodology described in the previous chapter will help to build the ontology of the city, from the highest to the deepest concepts. As explained before, the most known and used language to write ontologies is OWL (Web Ontology Language). This language provides a set of primitives to construct ontologies, based on the RDF data model and the XML syntax.

The standards XML, RDF and OWL to build the ontology of the city will help to guarantee interoperability. The tool to edit ontologies is Protégé.

## Web Semantic Language: XML, RDF and OWL

XML, RDF and OWL are the basement of a semantic web standards. They provide a framework for managing resources, integrating, sharing and reuse data. Consequently, thanks to XML, RDF and OWL, data related to domains could be shared by many users, even if they do not share the same software, promoting interoperability.

## XML

XML, or Extensible Markup Language, is a generic markup metalanguage, which brings syntax, rules and structured documents technology to build data on the Web. It enables to define different languages with each their own vocabulary and grammar. XML schemas are used in order to define and validate XML documents.

## RDF

RDF is a specification from W3C in the Semantic Web domain, which purpose is to provide a metadata data model. This metadata data model is used to describe web resources through various artefacts, such as syntax notations (or vocabulary) and data serialization formats.

Vocabulary contains all the term needed to describe resources:

* Classes:
  + rdf:
    - rdf:XMLLiteral
    - rdf:Property
    - rdf:Statement
    - rdf:Alt, rdf:Bag, rdf:Seq
    - rdf:List
    - rdf:Nil
  + rdfs:
    - rdfs:Resource
    - rdfs:Literal
    - rdfs:Class
    - rdfs:Datatype
    - rdfs:Container
    - rdfs:ContainerMembershipProperty
* Properties:
  + rdf:
    - rdf:type
    - rdf:first
    - rdf:rest
    - rdf:value
    - rdf:subject
    - rdf:predicate
    - rdf:object
  + rdfs:
    - rdfs:subClassOf
    - rdfs:subPropertyOf
    - rdfs:domain
    - rdfs:range
    - rdfs:label
    - rdfs:comment
    - rdfs:member
    - rdfs:seeAlso
    - rdfs:isDefinedBy

Serialization formats define the way to represents RDF. It could be:

* Turtle
* N-Triples
* N-Quads
* JSON-LD
* N3
* RDF/XML

The serialization format used to store objects of a project in database can be N-Triples. This output must be compliant with the tools choosen for a platform as Hadoop framework, Apache Jena, etc.

## OWL

OWL, which means Web Ontology Language, is a knowledge-representation system based on a description-logic approach. It is considered as an extension of RDFS (RDF Schema), and facilitate the construction of web ontologies to describe formal domains. Where RDFS is well construct to describe objects and their relation, OWL brings vocabulary to give more precision about these objects.

OWL will help the work of inference and querying engine, such as SPARQL, which can be probably used in a platform.

## Protégé: a common tool to build ontologies

Protégé is a free, open source ontology editor and a knowledge management system. Protégé provides a graphic user interface to define ontologies.

It will be used to modelize the ontologies of the city, using the OWL knowledge-representation language.

Protégé is compliant with the OWL specification. It means that it is able to manipulate all the concepts defined by OWL. However, there are some differences of vocabulary:

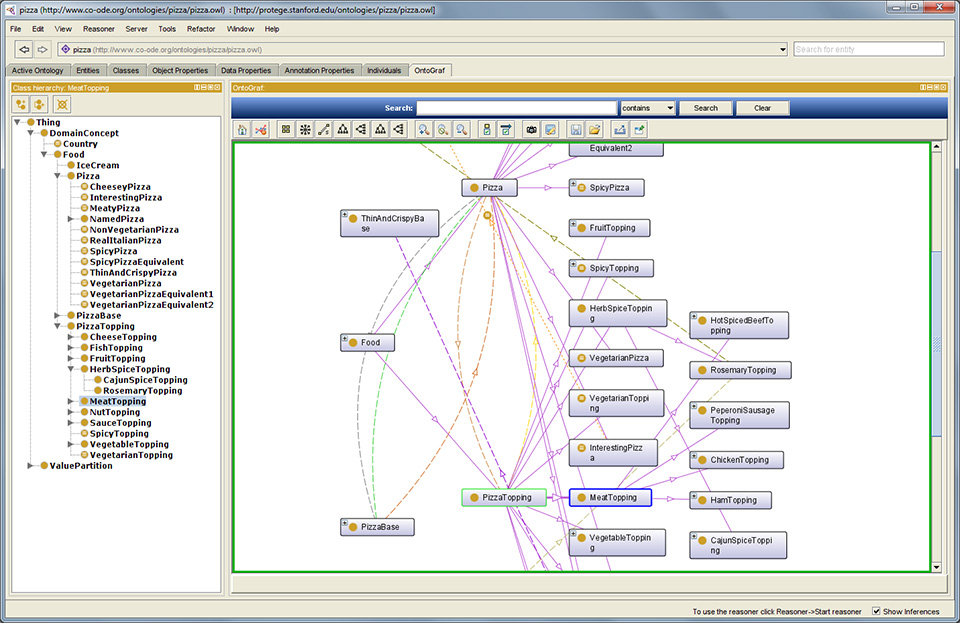
* OWL “Individuals” 🡪 Protégé “Instances”.
* OWL “Properties” 🡪 Protégé “Slots”
* OWL “Classes” 🡪 Protégé “Classes” or “Concepts”

Individuals (or Instances) represents objects in a specific domain. It is considered that Individuals are instances of Classes.

Slots (or Properties) are used to describe the relation or link between two Individuals.

Classes (or Concepts). Classes are a concrete representation of concepts: a class will contain items which describe the concept. Classes could implements hierarchical relationships.

Example of Protégé GUI:



# Architecture

## Functional architecture

## Introduction

A platform will integrate a Web Semantic data repository, which will contain a lot of data from the city systems. Data contained inside must be structured according to the ontology data model built for the city.

The diagram below represents the functional architecture of the ontology ecosystem, from data ingestion to user querying:



## Web semantic data repository

The following chapters describe the schema introduced in the previous chapter.

## Protégé

Protégé is the tool which will be used to describe the ontology of the city. Protégé is a stand-alone application installed on workstation. It provides all the features needed to create ontology, based on the XML, RDF and OWL specifications. The result of the work created with Protégé will be loaded in the database dedicated to to store ontologies data model.

## Data sources

Data sources represent all the systems which generate data that will be ingested in the web semantic data repository. These data could be internal or from third party applications, and provide data in various ways: files, database, web services, etc.

These data will be ingested through the data ingestion component.

## Data ingestion

This component will be in charge of extract, transform and load data from data sources to data repository. In order to achieve this work, many connectors (depending of data sources) will be implemented to extract the data. The process of transformation will process the data in order to transform data in an RDF representation, and to compare it with the data model to be sure that the data is known by the system and well-contructed. If the comparaison fails, the data is rejected and a specific workflow could be triggered by the directive module.

## Preprocessor

Before being stored in the repository, and after the validation of the data, a set of preprocessor will do operations on the RDF data, to transform them in N-Tripples. The N-Tripples will be stored on the repository, which could be considered as a web semantic data repository.

## Data repository

It represents the Big Data repository where the data are stored after their identification, transformation and preprocessing.

## Query engine interface

This interface provide tools to applications/users to querythe data repository. The query is sent to the query processor which will execute the query.

## Query processor

Query processor will execute a lot of operation in order to transform the query to a comprehensive format for the data repository (ontology language).

# Related documents

City Protocol – City Anatomy:

[*http://www.cptf.cityprotocol.org/CPAI/CPA-I\_001-v2\_Anatomy.pdf*](http://www.cptf.cityprotocol.org/CPAI/CPA-I_001-v2_Anatomy.pdf)

Focus on ontologies engineering methodology:

[*Focus on the Ontologies Engineering Methodology.pptx*](file:///D:\Users\CR5189.D60\Desktop\City%20OS%20BCN\Entregables%20i%20Moduls%20Engie%20-%20Livrables%20et%20Modules%20ENGIE\Ontology%20-%20CRIGEN\CityOS_Focus%20on%20the%20Ontologies%20Engineering%20Methodology_Nov2016.pptx)