# An Ontology for Open 311 Data

*Soroosh Nalchigar[a], Mark S. Fox[ab]*

*[a]Computer Science Department, University of Toronto*
*[b]Mechanical and Industrial Engineering, University of Toronto*
*msf@eil.utoronto.ca*

## Abstract
Last decade has seen a rapidly increasing interest in the publishing of city data. Applying data analytics to these data could result in discovery of city knowledge, insights and thereafter data-driven decision making and action. A major challenge in this context is to integrate data coming from different sources for later analyzes. This paper proposes a formal foundation ontology, called Open 311 Ontology that provides a unified terminology and a reference model for representing the 311 data of cities. It is illustrated that the this ontology could be used for reasoning and answering competency questions as well as mapping and integrating data coming from various sources.

Keywords: City, 311 Data, Ontology, Semantic Web

## 1. Introduction
Being a data-driven city (i.e., being able to intelligently use data to better deliver core city services) has quickly become a goal for modern cities. Towards this end, the application of data analytics has been chosen as a means to reach this goal [1]. Today, cities are publishing a broad range of data using Open Data standards, linking disparate data sources, and allowing citizens to retrieve and post data with smart phone apps. Canadian cities such as Ottawa, Toronto, and Vancouver as well as American cities such as New York, San Francisco, and Chicago all have major efforts underway to make city data, such as 911 and 311 calls, publicly available [3]. 311 is the name and the telephone number of city agencies that provide non-emergency municipal services to the public. The main goal of 311 systems is to enhance accessibility of city services, increase cities effectiveness in responding to public inquiries, and hence to improve city life. Open311[1] contributes to this trend by providing a standardized protocol and collaborative model for civic issue tracking to cities. It publishes a set of free and publicly available APIs that provide access to an existing 311 service.

A fundamental aspect of a smart city is to integrate and combine the data coming from various sources and places. Data integration is a challenging task, partially due to differences in the schema and content of data sources. Consider the existing 311 data of cities as an example. In order to combine data from multiple cities in order to perform comparative analysis, there are some issues that have to be tackled in advance. The first one is to find and map equivalent attributes between existing data sets. Is the attribute "Responsible Agency" in San Francisco's dataset equivalent to Toronto's "Division", "Section-unit", or both? Is

[1] http://www.open311.org/

Toronto's "Service Request Name" equivalent to "Request Type" in San Francisco's dataset? The second issue is to define a mapping between values of equivalent attributes. Provided that Toronto's "Service Request Name" is equivalent to San Francisco's "Request Type", how should the values of these attributes be integrated? Is San Francisco's "Sign Repair" equivalent to Toronto's "Sign Maintenance" or "Missing/Damaged Signs" or both? Without identifying and mapping equivalent attributes and values across 311 datasets, it is not possible to integrate, merge, and analyze the data [3]. To make these issues more clear, we note that each city's 311 dataset has a different number of attributes. Even worse, Toronto's 311 is using 371 different names for describing the service request types, while New York, San Francisco, and Chicago are using 120 and 25, and 12 different names (unique within each dataset) for representing service requests, respectively.

In this paper, we focus on ontologies that enable the longitudinal analysis of cities' 311 data (i.e., changes over time for a single city), and transversal analysis (i.e., comparison of two or more cities). We develop an ontology for 311 data, referred to as Open 311 Ontology[2], with which specific city's 311 data models and instances can be defined. The ontology is aimed to provide a unified and complete terminology and definitions that could be utilized for the integration of existing open data sets, enabling city data analytics, and hence facilitating the current movement towards smart and data-driven cites.

The rest of this paper is organized on the basis of the ontology development methodology proposed in [5]. Section 2 provides an overview of the existing datasets of four cities and describe their data schemas. Section 3.1 presents two motivating scenarios to highlight the need for the ontology and results in a set of competency questions in Section 3.2. The main classes and properties of the open 311 ontology are presented in Section 3.3. Thereafter, Section 3.4 describes other existing ontologies that are related and used in design of Open 311 Ontology. In Section 3.5 the main classes of the ontology are formally defined using Description Logic. Section 4 presents the evaluation of the ontology.

## 2. Analysis of Published 311 Data

We analysed the 311 data sets of four cities: Toronto, New York, San Francisco, and Chicago. To choose these cities, we considered factors such as availability of 311 data as well as existence of enough instances of service requests for understanding the domain and creating the unified terminology. This section describes the datasets of each of the cities.

### 2.1. Toronto

Toronto's open 311 dataset[3] includes 6 attributes. **Service Request Name** is the unique title of an individual service request. **Problem Code** is a unique identifier of the service request names. **Creation Date** indicates the date and time in which the corresponding request instance is submitted to 311. The attributes **Division** and **Section-Unit** represent the responsible City division and the 311's section or unit under which the service request is listed. Finally, **Internet Self-Serve** shows if the service request is reported via the web. Table 1 shows a service request record in this dataset.

---

[2] http://ontology.eil.utoronto.ca/o311o.owl.

[3] `http://www.toronto.ca/311`

Table 1: A service request record in Toronto's 311 dataset

| Data attributes | Example values |
|---|---|
| Creation Date | 04-26-2010 16:02:36 |
| Service Request Name | Residential: Garbage Bin: Exchange to Medium |
| Division | Solid Waste Management Services |
| Section – Unit | Collections |
| Problem Code | SWBNMTC-26 |
| Internet Self-Serve | Yes |

## 2.2. San Francisco

The San Francisco 311 dataset[4] includes 15 attributes. In this dataset, fields such as **Category** and **Responsible Agency** are equivalent to the attributes Service Request Name and Section–Unit, respectively, in Toronto's data set. Clearly this data set has more attributes than Toronto, such as **Status, Address,** and **Point** (latitude and longitude coordinates). Table 2 shows an example of a service request record in this dataset.

Table 2: A service request record in San Francisco's 311 dataset

| Data attributes | Example values |
|---|---|
| **Case ID** | 2441829 |
| **Opened** | 06-03-2013 |
| **Closed** | 06-03-2013 |
| **Status** | Closed |
| **Work Status** | New |
| **Responsible Agency** | 311 Supervisor Queue |
| **Address** | 2329 Castro St, San Francisco, CA, 94131 |
| **Category** | Street and Sidewalk Cleaning |
| **Request Type** | Sidewalk-Cleaning |
| **Request Details** | Furniture |
| **Source** | Voice In |
| **Supervisor District** | 9 |
| **Neighborhood** | Inner Mission |
| **Updated** | 06-03-2013 6:49 |
| **Point** | (37.750540724, -122.419933447) |

4 http://data.sfgov.org

### 2.3. New York

New York's open 311 dataset[5] includes 52 attributes. **Created Date**, **Closed Date,** and **Agency** are equivalent to Opened, Closed, and Responsible agency attributes form San Francisco's dataset, respectively. Other attributes such as **Complaint Type**, **Latitude,** and **Longitude** have obvious equivalents, but with a different name than in the San Francisco dataset. Some of the attributes that are appearing only in this dataset are **Due Date**, **Facility Type**, **Cross Street**. It should be noted this dataset has some attributes whose values are unspecified, NA, or missing value for the whole dataset. Those attributes are not considered in design of the ontology.

### 2.4. Chicago

Chicago's open 311 dataset[6] has 15 attributes and are provided in separate files, where each file includes requests of a specific type (e.g., tree debris, garbage carts, etc.). All the attributes in this dataset have an equivalent attribute in either or both San Francisco and New York datasets. However, they have different names. For example the **Completion Date** here is equivalent to the Closed attribute in San Francisco. Interested readers are referred to the URL of the New York and Chicago datasets given in the footnotes.

### 2.5. Observations

It is clear that there is little commonality across cities in the structure and content of their open 311 data sets. They vary in the number of attributes, the naming of the attributes and the naming of values. The records can also be incomplete. Finally, it is clear that some cities are less open than other cities. Toronto's open 311 data does not contain information on where the problem occurred nor status of the problem.

The Open311 standard does not attempt to address these problems. It provides a standard API for requesting a city's attribute names and values and then using that information to enable to reporting of attribute values to a city via a smart phone app. It does not attempt to introduce a standard vocabulary for 311 attributes and values.

## 3. Open 311 Ontology

### 3.1. Usage Scenarios

In order to illustrate and motivate the need for Open 311 Ontology, this section provides two hypothetical use case scenarios. These scenarios are later used to develop a set of competency questions and to indicate how the ontology would be helpful in these cases.

**Customer inquiries.** The contact center of the city 311 department receives numerous calls from customers who have inquiries about their previously reported service requests. Usually, the customers call to check the status of their request and to get updates on that, having the unique reference number of their submitted service request. To answer those inquiries, the contact center needs to access the stored data of service requests. To this end, the city 311 needs to keep records of the date and time in which the request was submitted as well as its latest status (open, closed, etc.).

[5] http://nycopendata.socrata.com

[6] http://data.cityofchicago.org

**Performance management**. Every day, the 311 call center receives thousands of service requests from the crowd, through various channels such as email, smart phone apps, and phone calls. The mayor's office understands that in the current rapidly changing business environment, deriving insights from raw data and making data-driven decisions is important. Towards this end, the 311 department has developed a standard reporting system that addresses the information needs of the mayor's office. Among others, the mayor's office wants to know what the busiest agencies are, i.e., which agencies are receiving highest number of service request. This information would help them to assign more employees to busy agencies, balance the workload, and hence reduce the time it takes to address the requests. Also, each service request is about a different subject, e.g., garbage bins, graffiti, roads, etc. The mayor's office wants to know what the most reported service topics are. These will help them in aggregating messages arising from the crowd and use it to gain insights about the city problems. Beside these reports, the mayor's office is interested in comparisons and cross-city analyses. They like to know how other cities are different from them in term of environmental pollutions and crime. In particular, they like to know which cities are having more reports about dead animals as well as reports about  by law contravention. In order to generate these reports, the 311 department needs accurate, relevant and timely data.

### 3.2. Competency Questions

Competency questions are essential for evaluating ontologies [5]. Based on the above scenarios, we have identified three categories of competency questions. The *first* category focuses simple retrieval of attribute values:

- **QC-1:** What is the submission date of a given service request with the unique code "XYZ"?
- **QC-2:** What is the status of a given service request with the unique code "XYZ"?

The *second* category of competency questions focus on the aggregation of information:

- **QC-3:** What are top five busiest 311 agencies in terms of number of submitted service requests?
- **QC-4:** How many service requests about "Subject1" are reported since the beginning of the year?

The *third* category of competency questions focuses on cross city comparisons:

- **QC-5:** Which cities have more than 1000 reports categorized as "illegal issues"?
- **QC-6:** What are top three cities with most number of reports of the subject "dead animals"?

### 3.3. Classes and Properties

In this section we illustrate the construction of Open 311 Ontology and explain the primitive classes as well as object and data properties of the ontology[7]. This ontology is expressed in OWL-DL and is implemented using the Protégé ontology editor.

At the core of the Open 311 Ontology is the class ServiceRequest, which is the class of all service requests submitted to the 311 department. A ServiceRequest contains following "standard" data properties that can be provided by most cities:

- **AddressType:** Type of the address of the service request (e.g., Blockface).

---

[7] Available at:
`http://ontology.eil.utoronto.ca/o311o.owl`

- **Borough:** Borough of the service request (e.g., Manhattan).
- **CloseDate:** The unique ID for each instance of the service request.
- **CommunityBoard:** The community board of the service request (e.g., 04 Manhattan).
- **CrossStreet:** The two cross streets nearest to the location of event.
- **Details:** Further information about the service request.
- **DueDate:** The due date and time of the service request.
- **EventID:** The unique ID for each instance of the service request.
- **EventZip:** The zip code of the service request.
- **Intersection:** The intersection streets close to the location of service request.
- **LocationType:** The type of the location of the service request (e.g., Residential building).
- **Neighborhood:** The neighborhood of the service request.
- **OpenDate:** The open date and time of the service request.
- **Source:** Represents how the service request was made (e.g., voice in).
- **Status:** Represents the status of the service request.
- **UpdateDate:** Date and time of the last update of the service request.
- **Ward:** shows the ward number of the service request.

Along with these data properties, the class ServiceRequest has following object properties:

- **hasType:** whose range is the class of 311Type and identifies the category of the service request.
- **isHandledBy:** whose range is the class Agency, represent the 311 agency that handles the service request.
- **hasSPS:** whose ranges is the class SpsPoint, identifying the exact location of the service requests.
- **isSubmittedTo:** whose range is the class org:Division, showing the 311 responsible division to which the service request submitted.

Another important class in our ontology is 311Type which, as the name suggests, represents the type to which each instance of ServiceRequest belong. One of the main issues in creation of Open 311 Ontology was that each city has its own vocabulary for describing the service request types. For example a service request about a damaged street signs is recorded as "Sign Maintenance" in Toronto dataset, while it appears as "Sign Repair" and "Street Sign - Damaged" in San Francisco and New York datasets respectively. To solve this problem we define the 311Type class as having the following properties:

- **hasSubject:** whose range is the class Subject and defines what the corresponding Type is about. The subclasses of the class Subject include but are not limited to TransportationRoutes, RoadSymbol, GarbageContainer, etc. Each of these has its own subclasses. See the OWL file of the ontology for a full list of subclasses of Subjects as well as their subclasses.
- **needAction:** whose range is the class Action and represent the action that the Agency needs to undertake in response to the ServiceRequest. The class Action has subclasses such as Replace, Repair, Remove, Reinstall, Install, Inspect, etc.
- **hasCategory:** whose range is the class MessageCategory and includes subclasses such as Complaint, Report, Inquiry.

These definitions enable each 311 department to relate its own service request types with their specific naming values to our ontology and thereafter populate it with real instances of service requests. In this way, our ontology facilitates integration of data across various cities and supports reasoning about service requests as well as querying, and analysis of the integrated data.

### 3.4. Related Ontologies

The Open 311 Ontology is built on a foundation of existing ontologies such as Organization Ontology, Time Ontology, and GeoNames Ontology. In this section we briefly introduce these ontologies and indicate how they are related to Open 311 Ontology.

**Organization Ontology.** Organization ontology[8], defined by Fox et al. [4], focuses on organization structure, roles, authority and empowerment. It is developed as part of the TOVE Enterprise Modelling Project [2]. One of the core classes in this ontology is the class Organization, defined as a set of constraints on the activities performed by agents. This class contains following data and object properties:

- **hasName:** a text showing the name of the organization.
- **hasLegalName:** represents the legal official name of the organization.
- **hasGoal:** whose range is the class Goal and shows the goal of organization.
- **consistsOf:** whose range is the class Division and represents the subdivisions of the organization.

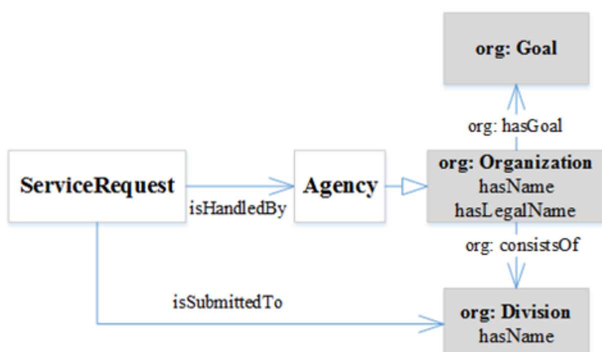


**Figure 1:** Open 311 Ontology in relation to Organization Ontology. In all the figures in this paper, arrows with open arrow head represent the `rdfs:subClassOf` properties. Regular arrows symbolize the object property of the given label.

Figure 1 depicts how the Open 311 Ontology is related to the Organization ontology. In this figure we specialize the class Organization to the classes Agency. This will allows the class Agency to inherit the properties of the Organization as defined in Organization ontology, e.g., hasName.

**Placename Ontology.** The service requests submitted to 311 are associated with a geographic area, which would be a borough, park, cemetery, building, etc. Therefore, a requirement for the Open 311 Ontology is the ability to identify the geographic area to which the service request is related. The GeoName geographical database includes over 8.3 million

---

[8]This ontology is available at `http://ontology.eil.utoronto .ca/organization.owl`. In this paper, the prefix "`org:`" is used to show the classes as well as data and object properties of this ontology.

placenames. Beyond names of places in various languages, this database integrates geographical data such as latitude, longitude, elevation, population and postal codes from various sources. All the placenames are instantiations of the GeoNames Ontology[9] that integrates a number of ontologies including Schema.org. The most fundamental class in GeoNames Ontology is the class gn:feature which includes the following properties:

- **name:** text, representing the main international name of a feature., e.g., “New York”.
- **altenativeName:** a number of alternative names for the feature.
- **countryCode:** a two letters country code in the ISO 3166 list.
- **population:** population of the feature.
- **wikiPediaArticle:** a Wikipedia article of which subject is the resource.

Figure 2 shows how the Open 311 Ontology is related to GeoNames and Schema.org ontologies. This figure indicates that the object property hasCity connects the class ServiceRequest to the class sc:City which inherits the properties of the class gn:Feature.


Figure 2: Open 311 Ontology in relation to GeoNames and Schema.org ontologies.

**International Contacts Ontology.** The current data of service requests includes the address for which the request is made. The address text usually includes number, street name, as well as the postal code. Hence, the Open 311 Ontology requires representing the address. This will allow the ontology to be better refined and represent the location of the service request. International Contact (iContact) Ontology[10] provides basic classes and properties for the representation of street addresses, phone numbers and emails. One of the important classes in this ontology is ic:Address that includes following properties:

- **hasStreet:** text, showing the name of the street.
- **hasUnitNumber:** a non-negative integer representing the unit number where the request is located.
- **hasPostalCode:** text, representing the postal code of the location.
- **hasStreetDirection:** shows the direction of the street (e.g., north, east).
- **hasStreetType:** whose range is the class ic:StreetType and shows the type of the street (e.g., avenue, road, boulevard).

---

[9] This ontology is available at `http://www.geonames.org /ontology/ontology_v3.1.rdf`. In this paper, the prefix “`gn:`” is used to show the classes as well as data and object properties of this ontology.

[10] This ontology is available at `http://ontology.eil.utoronto .ca/icontact.owl`. In this paper, the prefix “`ic:`” is used to show the classes as well as data and object properties of this ontology.

Figure 3 shows how the Open 311 Ontology is related to iContact Ontology.

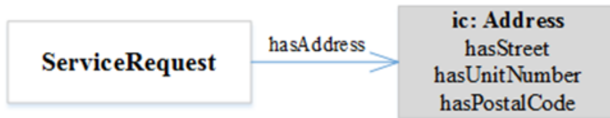


Figure 1: Open 311 Ontology in relation to iContact Ontology

**Time Ontology**. The service requests are associated with temporal information such as the submission date, the closing date, etc. It is important for Open 311 Ontology to capture and represent the time attributes of service requests. To do that, we use the Time Ontology[11] for representing temporal properties of service requests. Time Ontology provides a standard set of classes and relations for representing facts about topological relations among instants and intervals, as well as information about durations and datetime information. One of the main classes in this ontology is DateTimeInterval that is connected to the class DateTimeDescription through the object property hasDatetimeDescription. The class DateTimeDescription includes various data properties such as second, minute, hour, day, month, year, etc. Figure 4 illustrates that the class ServiceRequest form Open 311 Ontology is connected to Time Ontology through four different object properties, namely hasOpenDate, hasClsoeDate, hasUpdateDate, and hasDueDate.

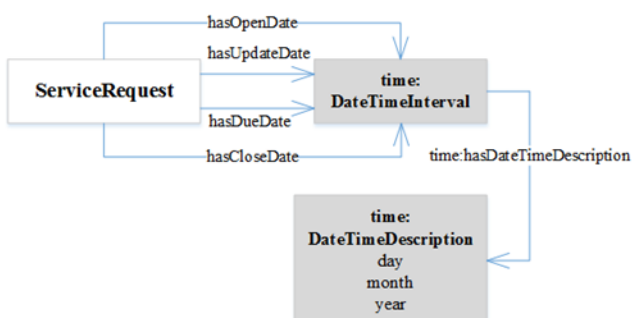


Figure 4: Open 311 Ontology in relation to Time Ontology

**Other related ontologies.** In the Open 311 Ontology, the class ServiceRequest is connected to the class 311Type via the object property hasType. The class 311Type is connected to the class 311Subject through the object property has311Subject. One of the subclasses is TransportationRoutes, meaning that a service request could be about a transportation route such as an expressway. In order to provide enough expressivity and have sufficiency, we needed to define subclasses of the TransportationRoutes. Several urban ontologies, e.g., Towntology ontology [7], CityGML ontology [6], contain transportation-related classes. Although these have been created with a specific task in mind, they could be used to identify some of the subclasses of TransportationRoutes. Figure 5 shows how our ontology is connected to the Towntology, CityGML, and DBpedia ontologies.

[11] This ontology is available at `http://www.w3.org/2006/time`. In this paper, the prefix "`time:`" is used to show the classes as well as data and object properties of this ontology.

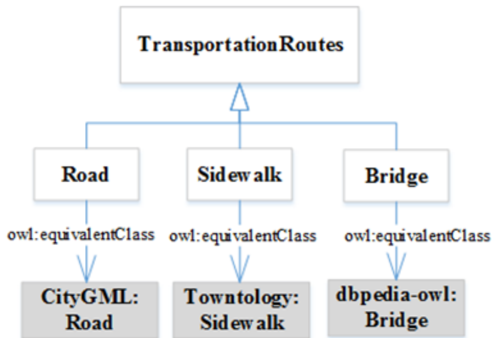


Figure 5: Equivalent Classes of Open 311 Ontology to Other Ontologies

It should be noted that within Open 311 Ontology, the class TransportationRoutes has other subclasses that were identified by careful review of 311 city datasets, e.g., Expresseway, Boulevard. Moreover, there are other classes in our ontology, such as Plants, Animal&Insects that are related to other ontologies. Interested readers are referred the OWL-DL file of our ontology for further details.

### 3.5. Axioms

In this section, using Description Logic (DL), we present the axioms that define the ServiceRequest and 311Type classes.[12]:

```
ServiceRequest ≡
311Thing ⊓
=1 has311Type.311Type ⊓
=1 isHandledBy.Agency ⊓
=1 hasAddress.Address ⊓
=1 hasCity.City ⊓
=1 hasOpenDate.DateTimeInterval ⊓
≤1 hasCloseDate.DateTimeInterval ⊓
≤1 hasDueDate.DateTimeInterval ⊓
≥0 hasUpdateDate.DateTimeInterval ⊓
=1 EventID.string ⊓=1 Source.string ⊓
=1 Status.string ⊓≤1 AddressType.string ⊓
≤1 Borough.string ⊓≤1 CommunityBoard.string ⊓
≤2 CrossStreet.string ⊓ ≤1 Deatils.string ⊓
≤1 Intersection.string ⊓
≤1 LocationType.string ⊓
≤1 Neighborhood.string ⊓
≤1 Ward.string
```

Moreover, the class 311Type is defined in terms of following formulation:

[12] To represent the “exactly one” cardinality in these formulations, we contract the ≥1 and ≤1 constructors to =1, due to space limitations.

```
311Type ≡
311Thing ⊓
≥1 has311Subject.311Subject ⊓
≥1 need311Action.311Action ⊓
=1 has311MessageCategory.311MessageCategory ⊓
=1 311TypeCode.string ⊓
=1 311TypeName.string
```

## 4. Evaluation

We evaluate the Open 311 ontology in two parts. The first part evaluates the ability of ontology to represent the data that is needed to answer the competency questions of Section 3.2. The second part evaluates the ontology by illustrating how data of each city is represented in out ontology.

### 4.1. Answering the Competency Questions

This Section presents the competency questions and shows how the SPARQL query language [8] could be used to retrieve the relevant data from the ontology and to answer the questions[13].

**QC-1:** What is the submission date of a given service request with the unique code "XYZ"? In order to answer the first competency question, we need to retrieve the date in which the given service request was submitted to the city 311. Following query finds the answer:

```
SELECT ?day ?month ?year
WHERE{
  ?ServiceRequest O311O:EventID "XYZ".
  ?ServiceRequest O311O:hasOpenDate ?DTInterval.
  ?DTInterval time:hasDateTimeDescription ?DTD.
  ?DTD time:day ?day.
  ?DTD time:month ?month.
  ?DTD time:year ?year
 }
```

In our ontology, the ServiceRequest class is connected to the class DateTimeInterval (imported from Time Ontology) via the object property hasOpenDate. In Time Ontology, the DateTimeInterval class is connected to the class DateTimeDescription through the object property hasDateTimeDescription. This data that is required to answer the first competency question are represented as data properties of the class DateTimeDescription.

**QC-2:** What is the status of a given service request with the unique code "XYZ"?
Following query answers the question:

```
SELECT ?status
```

[13] All the proposed queries assume that the namespace prefix `o311o` refers to the IRI `http://ontology.eil.utoronto.ca/o311o.owl`. Moreover, it is assumed that the prefix `time` refers to the IRI `http://www.w3.org/2006/time`.

```
WHERE {
 ?ServiceRequest O311O:EventID "XYZ".
 ?ServiceRequest O311O:Status ?status
}
```

In the ontology, the class ServiceRequest has the data property of Status whose value is a string. The query finds the answer to the second competency question could be obtained from this data property.

**QC-3:** What are top five busiest 311 agencies in terms of number of received service requests?
The answer to the third competency question is obtained by following SPARQL query:

```
SELECT ?Name (COUNT (?ServiceRequest) AS ?Total)
WHERE{
  ?ServiceReuqest O311O:isHandledBy ?Agency.
  ?Agency org:hasName ?Name
}
GROUP BY ?Name
ORDER BY ?Total
LIMIT 5
```

In our ontology, the object property isHandledBy connects the class ServiceRequest to the class Agency. The class Agency has the data property of hasName which is a unique string representing the name of agency that handles the service request. In order to compute the answer to the forth competency question, this query counts total number of service requests that are submitted to the city agencies. Then, by ordering and finding the top 5 instances of the class Agency, the answer to the third competency question is found.

**QC-4:** How many service requests about "Subject1" are reported since the beginning of the year?
Regarding the forth competency question, we need to retrieve and count service requests of the given subject that are reported in the current year. To do that, following SPARQL query is used:

```
SELECT (COUNT(?ServiceRequest) AS ?Total)
WHERE{
 ?ServiceRequest O311O:has311Type ?311Type.
 ?311Type O311O:has311Subject ?Subject.
 ?Subject a O311O:"Subject1".
  ?ServiceRequest   O311O:hasOpenDate  ?DateTimeInterval.
 ?DateTimeInterval time:hasDateTimeDescription ?DTD.
 ?DTD time:year ?Year.
 FILTER (?Year == 2014)
}
```

In our ontology, each instance of the class ServiceRequest is associated with its 311Type through the object property has311Type. Moreover the class ServiceRequest is associated with the class DateTiemInterval from Time Ontology, to keep the time information in which a request is reported. Within Time Ontology, the class DateTimeInterval is connected to the class DateTimeDescription through the object property hasDateTimeDescription. The first step in answering this competency question is to retrieve the set of all instances of the class ServiceRequest whose 311Type instance has the subject that is the given in the competency questions. Having this set, the next step is to exclude those instances which are not submitted in the current year and count total number of service requests that are remained.

**QC-5:** Which cities 311 has received more than 1000 reports categorized as illegal issues? Following query computes the answer to this competency question:

```
SELECT ?City (COUNT (?ServiceReuqest) AS ?Total)
WHERE{
      ?ServiceReuqest O311O:hasCity ?City.
      ?ServiceReuqest O311O:has311Type ?311Type.
      ?311Type O311O:has311MessageCategory ?Category.
      ?Category a O311O:IllegalIssue
}
GROUP BY ?City
HAVING COUNT (?ServiceReuqest) > 1000
```

In our ontology, the class 311Type is connected to the class 311MessageCategory via the object property has311MessageCategory. The class 311MessageCategory has various subclasses one of which is Complaint which has IllegalIssue as a subclass. To compute the answer of this competency question, for each city, all the instances of ServiceRequest whose category is illegal issue are retrieved and counted. This will result in a list of cities along with their corresponding number of illegal issue reports. The last step is to exclude those cities which have less than 1000 service requests of the specified category.

**QC-6:** What are top three cities with most number of reports of the subject “dead animals”? In this competency question, cities should be compared with regarding to the number of submitted service requests about dead animals. The answer to this question results from following query:

```
SELECT ?City (COUNT (?ServiceRequest) AS ?Total)
WHERE{
      ?ServiceReuqest O311O:hasCity ?City.
      ?ServiceReuqest O311O:has311Type ?311Type.
      ?311Type O311O:has311Subject ?Subject.
      ?Subject a O311O:DeadAnimal
}
GROUP BY ?City
ORDER BY ?Total
LIMIT 3
```

In the ontology, the class 311Type is connected to the class 311Subject via the object property has311Subject. The class 311Subject has various subclasses one of which is Pests. The class Pests has two subclasses, namely Animal and Insects. The DeadAniumal class is a subclass of the class Animal. Similar procedure to previous question is used here to answer this competency question. The only difference is that in this question we look for ServiceRequest instances that are connected to the DeadAnimal class via the object property has311Subject.

## 4.2. Mapping Datasets to the Open 311 Ontology

In this section, we illustrate the possibility of mapping/representing existing datasets to/in the ontology. As previously was explained in Section 2, Tables 1 and 2 presents an examples of a service request records in Toronto and San Francisco datasets. Figures 6 and 7 show how these examples are represented in the Open 311 Ontology. It should be mentioned that we also have mapped data sets of the cities New York and Chicago to our ontology. However, due to lack of space the examples are not presented here.

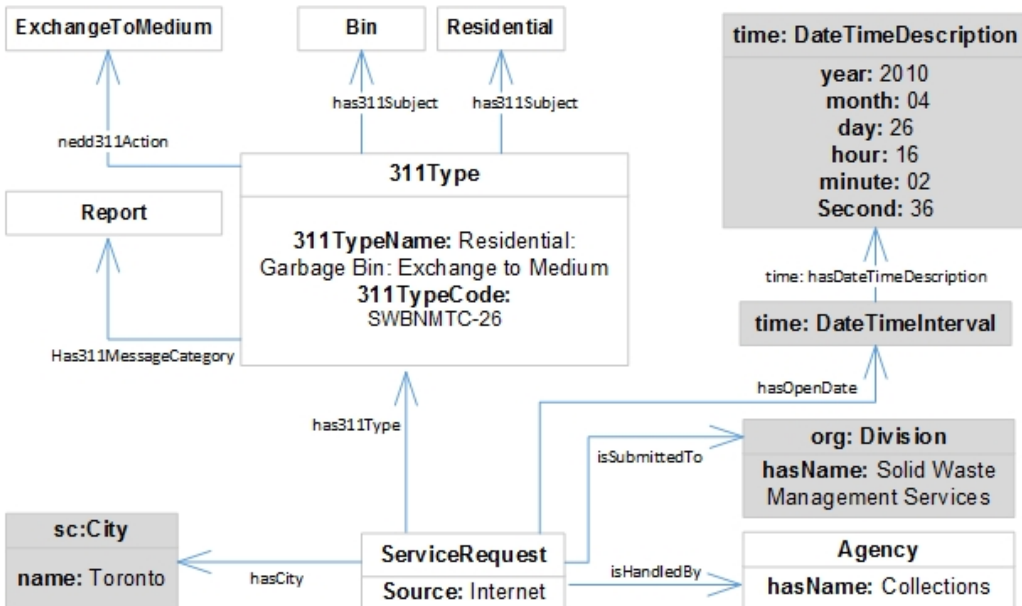


Figure 5: Mapping Toronto's Dataset to Open 311 Ontology

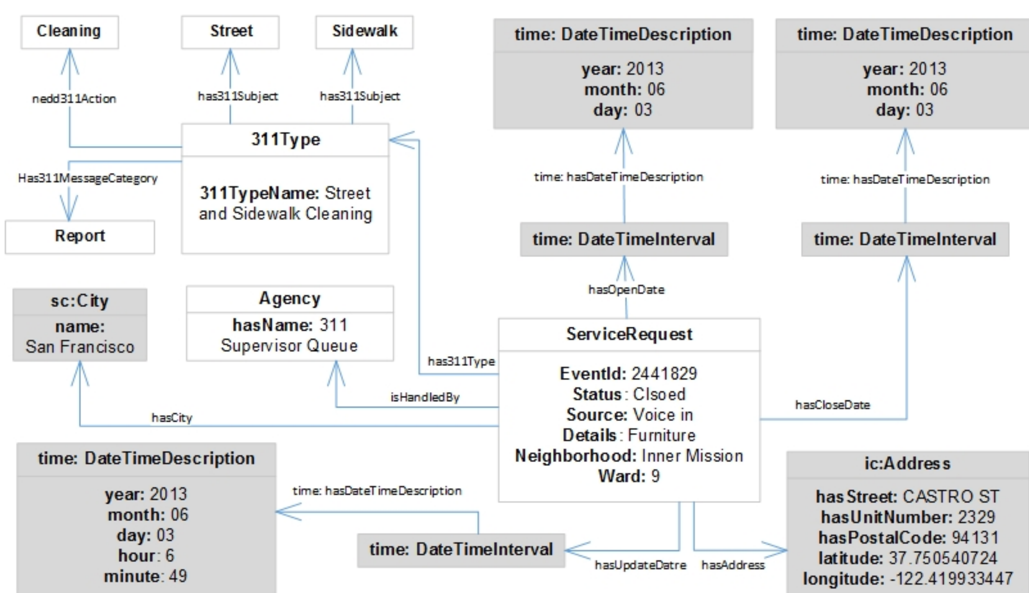


Figure 6: Mapping San Francisco's Dataset to Open 311 Ontology

## 5. Conclusion

This paper describes the Open 311 Ontology for representing and reasoning about 311 data. To evaluate the ontology, we illustrated that it satisfies the competency questions based on the scenarios. Moreover, we illustrated how a city's 311 data is mapped onto it, thereby making possible to perform aggregate and comparative analyses of multi-city 311 data.

The process of creating of this ontology clearly illustrates the lack of and need for common vocabularies and ontologies for 311 and other city data. The challenge now is to persuade cities to adopt this standard.

## 6. References


[1] Flowers, M. 2013. Beyond Open Data: The Data-Driven City. In Brett Goldstein and Lauren Dyson, editors, *Beyond Transparency: Open Data and the Future of Civic Innovation,* Chapter 15, 185-198.

[2] Fox, M. S. 1992. The TOVE project towards a common-sense model of the enterprise. *Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*. Springer Berlin Heidelberg, 25-34.

[3] Fox, M. S. 2013. City Data: Big, Open and Linked. Working Paper, Enterprise Integration Laboratory, University of Toronto.

[4] Fox, M. S., Barbuceanu, M., and Gruninger, M. 1996. An organisation ontology for enterprise modeling: preliminary concepts for linking structure and behaviour. *Computers in industry* 29.1: 123-134.

[5] Gruninger, M., Fox, M. S. 1995. Methodology for the design and evaluation of ontologies, In *Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing at IJCAI*.

[6] Gröger, G., Kolbe, T., and Czerwinski, A. 2007. Candidate OpenGIS® CityGML Implementation Specification (City Geography Markup Language). *Open Geospatial Consortium Inc, OGC*.

[7] Keita, A., Laurini, R., Roussey, C., and Zimmerman, M. 2004. Towards an ontology for urban planning: The towntology project. In *CD-ROM Proceedings of the 24th UDMS Symposium, Chioggia* (Vol. 12, No. 1).

[8] Prud'Hommeaux, E., and Seaborne, A. 2008. SPARQL query language for RDF.*W3C recommendation*, *15*.