Modèle d'enrichissement et d'intégration des données

Antoine Zimmermann

Modèle d'enrichissement et d'intégration des données

ANR OpenSensingCity – ANR-14-CE24-0029
Livrable 3.3

Auteur:
Antoine Zimmermann (ARMINES-FAYOL)
Contributeurs:
Éric Noulard (Antidot), Olivier Boissier (ARMINES-FAYOL)

29 septembre 2017

Table des matières

Résumé

Ce livrable décrit le modèle de données utilisé pour décrire à la fois les données, les métadonnées, les jeux de données, les catalogues de données ouvertes, les flux de données et les séries temporelles dans le cadre des villes intelligentes.

1 Introduction

Comme l'ont révélé les entretiens conduits dans le cadre du projet OpenSensingCity, l'un des problèmes majeurs dans la réutilisation de données ouvertes est l'hétérogénéité des formats (CSV, XML, JSON, Excel, PDF, HTML, etc.). À cela s'ajoute le manque d'information complémentaire permettant d'interpréter le contenu des données. Par exemple, un nombre peut parfois indiquer une quantité (p.ex., le nombre de places dans un parking), une mesure (p.ex., la largeur d'une place, en mètre) ou bien un simple numéro de série (auquel cas la valeur numérique du nombre n'a aucune signification).

Dans le projet OpenSensingCity, nous envisageons de présenter toutes les données de façon homogène et enrichie. Ce livrable décrit le modèle de données choisi et les éléments d'enrichissement permettant de donner plus de sémantique aux données. Nous présentons également les standards utilisées pour décrire les métadonnées de jeux de données, de catalogues, de séries temporelles et flux. Nous discutons de quelques alternatives.

Pour arriver à notre modèle, nous décrivons d'abord les exigences (section 2) qui nous amènent à privilégier le modèle RDF, décrit en section 3. Nous détaillons ensuite les structures RDF utilisées pour la description de flux, de séries temporelles et de jeux de données.

2 Éxigences

Nous souhaitons préconiser un modèle de données unique qui devrait être privilégié lorsqu'un portail open data expose des données ouvertes. Ce modèle doit :

- être suffisamment souple pour pouvoir être employé dans tout domaine d'application, donc éviter les standards propre à un domaine, tel que l'énergie (p.ex., CIM), le bâtiment (p.ex., BIM), le transport (p.ex., GTFS). On se tournera donc vers des modèles génériques.
- à partir d'un document suivant ce modèle, il doit être possible de comprendre la structure et le sens des données. En particulier, il doit y avoir un lien explicite entre un jeu de données et la description de son schéma et/ou de son vocabulaire.
- lorsque les données existent dans plusieurs formats, l'accès à un des formats doit suffire à découvrir tous les autres formats disponibles. Si les données conformes au modèle que nous préconisons ont été générées à partir de sources de données autres, la provenance de la donnée exposée doit être accessible.

— les données et les métadonnées doivent être accessibles de façon uniforme. On doit donc éviter les situations où les données peuvent être téléchargées dans leur format natif, tandis que les métadonnées ne sont visibles que sur la page Web du jeu de données.
— il doit être possible de créer des liens explicites entre jeux de données.
— le modèle doit également couvrir le cas des flux de données et des séries temporelles.

Ces contraintes ne sont, à l'heure actuelle, pas entièrement couvertes par les modèles de données existants, mais le modèle RDF en satisfait la majorité. Nous choisissons donc RDF comme base de notre modèle d'enrichissement de données.

TODO (from Olivier) : Donner quelques références de modèles de données existants ? et arguments (références ?) permettant de justifier une telle critique ?

3 Le modèle de données RDF

Dans cette section, nous présentons de façon un peu simplifiée le modèle de données RDF (*Resource Description Framework* [6]). Nous ajoutons également des notions concernant les données liées (ou *Linked Data*) dont les principes (énoncés initialement dans [2]) ont une importance pour le choix du modèle d'enrichissement de données. RDF repose essentiellement sur :

— une identification uniforme de toute chose d'intérêt à l'échelle du Web, en utilisant des URL ou *Uniform Resource Locators* [3], c'est-à-dire, des adresses Web ; [1]
— une description des choses d'intérêt par des déclarations sous la forme sujet-verbe-complément, c'est-à-dire, sous forme de triplets.

Ce qu'on appelle ici "chose d'intérêt" est toute chose que l'on souhaite décrire, soit en lui associant des données (p.ex., une place de parking dont on veut indiquer la taille en mètre) soit en la reliant à d'autres choses d'intérêt (p.ex., une place de parking se trouve dans un parking). Dans la terminologie RDF, une chose d'intérêt s'appelle une ressource. Une ressource peut être concrète ou abstraite, réelle ou fictive. De même, la terminologie RDF dénome les composants d'un triplet RDF le sujet, le prédicat (l'équivalent du verbe) et l'objet (l'équivalent du complément). Le prédicat représente une propriété du sujet et toute propriété est potentiellement une chose d'intérêt, donc une

1. En réalité, le standard utilise des IRI, ou *Internationalized Resource Identifiers* [7] qui forme un surensemble des URL. Mais dans le contexte d'OpenSensingCity, nous n'utiliserons que des URL.

ressource, que l'on peut identifier par des URL.

Le fait d'utiliser des URL pour identifier les ressources permet de réaliser deux choses en une seule fois : l'identification de la ressource et la localisation d'information concernant cette ressource.

Exemple 1. *Un exemple d'URL pouvant être utilisé pour identifier le parking Bellecour dans Lyon est* `http://resource.grandlyon.com/parking/C77`*. Puisqu'il s'agit également d'une adresse en plus d'être un identificateur, et que cette adresse se réfère au nom de domaine du Grand Lyon, celui-ci peut fournir des données relatives au parking lorsqu'on se rend à cette URL. Idéalement, ces données sont fournies dans un format compatible avec le modèle de données RDF.*

Exemple 2. *La propriété d'avoir un nombre de places pour un parking existe pour tous les parkings du monde. En identifiant cette propriété avec une URL, tout jeu de données relatif aux parkings pourrait réutiliser la même URL pour décrire le nombre d'emplacements parkings. Par exemple* `http://www.fnms.fr/parking/nb_places` *où* `fnms.fr` *correspond au nom de domaine du site de la fédération nationale des métiers du stationnement.*

Un groupe d'URL pointant vers le même site Web ou document en ligne, et servant à identifier les termes d'un même thème ou domaine d'application est appelé un vocabulaire. Par exemple, on peut parler du vocabulaire des parkings de la FNMS. Lorsqu'un document définissant un vocabulaire est complété par des propriétés formelles sur les termes, ou des relations formelles explicites liant ses termes, on parle d'ontologie. Par exemple, une ontologie des parkings fournit une classification des éléments et concepts du parking, ainsi que des contraintes formelles sur ce qui peut exister dans un parking ou une place de parking.

À l'aide de ces termes et des URL identifiant les objets d'intérêt, on peut former des déclarations sous forme de triplets.

Exemple 3. *La déclaration indiquant que "la place de parking numéro 32 dans la rue Mortier à Lyon se trouve dans le parking Bellecour-Campanile" peut se représenter en RDF par le triplet :*

sujet	`http://resource.grandlyon.com/parking/C77/place/32`
prédicat	`http://www.fnms.fr/se_trouve_dans`
object	`http://resource.grandlyon.com/parking/C77`

Pour simplifier les exemples, nous utiliserons une notation très courante des URL où le début de l'URL est remplacé par un préfixe court suivi de deux points. Ainsi, nous écrirons le triplet précédent :

```
pklyon:C77/place/32  fnms:se_trouve_dans  pklyon:C77
```

Dans certains, nous souhaitons utiliser une valeur concrète pour une propriété d'une ressource. Par exemple, le nombre de place dans un parking. Dans ce cas, nous utilisons la notion de littéral.

```
pklyon:C77  fnms:nb_places  "613"^^xsd:integer
```

Le littéral `"613"^^xsd:integer` est composé de deux éléments : sa forme lexicale `613` qui est une transcription en chaine de caractère de la valeur que l'on veut indiquer, et son type de données `xsd:integer` qui permet de savoir comment interpréter la forme lexicale. Notons encore que le type de données est identifié par une URL (ici, il s'agit de `http://www.w3.org/2001/XMLSchema#` qui dénote les entiers relatifs).

On appelle *graphe RDF* un ensemble de triplets RDF car chaque triplet peut être représenté comme un arc orienté, étiqueté par le prédicat, reliant le nœud-sujet et le nœud-objet, comme dans la figure 1.

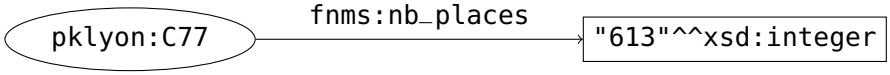

FIGURE 1 – Un exemple de graphe RDF.

Si plusieurs triplets ont le même sujet, alors plusieurs arcs partent du même nœud. Enfin, il est possible de décrire des entités anonymes, c'est-à-dire, n'ayant pas d'identificateur. Si le sujet ou l'objet d'un triplet n'est pas identifié, on dit que c'est un nœud vide et on l'illustre dans un graphe par un cercle sans étiquette, comme dans la figure 2.

Dans la suite du document, pour avoir des exemples compacts, nous utiliserons un format de représentation des graphes RDF appelé Turtle [17]. Le graphe de la figure 2 s'écrit comme suit :

```
@prefix ex: <http://data.example.com/> .
pklyon:C77 a fnms:Parking;
  fnms:nb_place "613"^^xsd:integer;
  fnms:gestionnaire [
    a ont:Entreprise;
    ont:pdg ex:jacques-dupont .
  ] .
```

Exemple RDF 1 – Le graph RDF de la figure 2 en notation Turtle.

La lettre `a` est un raccourci pour l'URL `http://www.w3.org/1999/02/22-rdf-syntax-ns#type` (en abrégé `rdf:type`)qui désigne la relation qui lie une entité à une classe d'entités à laquelle elle appartient. Les crochets indiquent que l'on décrit un nœud vide, c'est-à-dire, que le contenu du

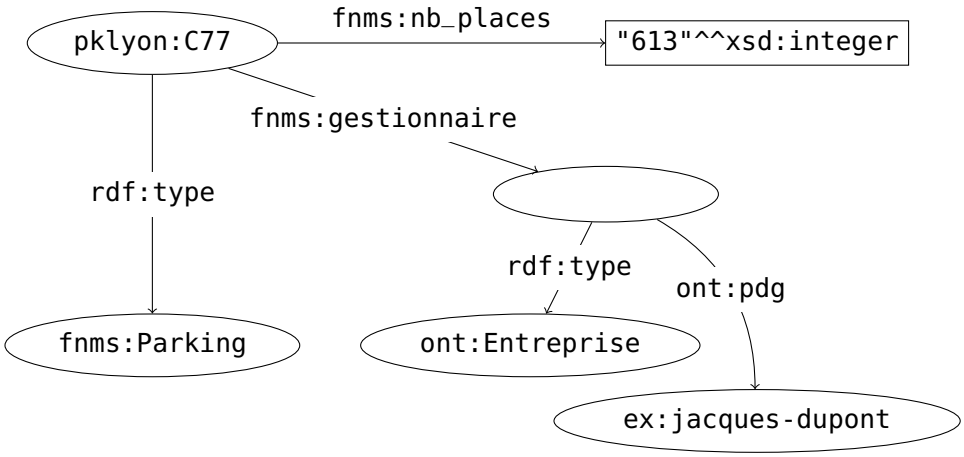

FIGURE 2 – Un exemple de graphe RDF.

crochet représente des paires propriété-objet. Le point-virgule sépare des paires propriété-objet pour un même sujet, et la virgule sépare des objets pour une même paire sujet-propriété. Ainsi, cet exemple contient 8 triplets.

4 Modèles pour les flux de données et séries temporelles

Nous souhaitons utiliser le modèle de données RDF pour décrire des flux de données, permettant ainsi de fournir des données de capteurs dans un format compatible avec les données du Web de données et du Web sémantique.

Deux approches sont communément employées pour encoder des flux de données en RDF :

— par annotation,
— par réification,
— par liens hypermédia.

4.1 RDF et annotations

Dans le premier cas, on suppose que les données émises par le capteur (ou le flux) décrit une situation actuelle. Par exemple, à tel endroit, il fait 27°C. Cette situation se décrit facilement sous forme d'un triplet RDF :

```
ex:lieu  meteo:a_pour_temperature  "27 Cel"^^cdt:ucum
```

Ici, nous utilisons une notation standardisée (UCUM ou *Unified Code for Units of Measures*) pour les unités de mesures, identifiée par le type de données `cdt:ucum`. Pour prendre en compte l'évolution temporelle de ces données, on associe aux triplets une étiquette temporelle. L'information transmise par un flux est une séquence de paires $\langle G, t\rangle$ où G est un graphe RDF et t un indicateur temporelle (soit un instant précis, soit un intervalle indiquant la plage de validité des déclarations contenues dans le graphe G).

```
2017-07-05T09:42:42+02:00
ex:lieu  meteo:a_pour_temperature  "27 Cel"^^cdt:ucum
```

Ce modèle n'est cependant pas suffisant pour décrire des séries temporelles. En effet, on souhaite également décrire une série temporelle avec sa provenance, sa fréquence d'échantillonnage, le début et la fin de la série, etc. Pour cela, on pourrait ajouter un identifiant à un ensemble de paires, mais une alternative consiste à utiliser un autre format standard : les graphes nommés et les jeux de données RDF (en anglais *named graphs* et *RDF datasets*). Un *graphe nommé* est une paire $\langle u, G\rangle$ où u est une URL [2] appelé le *nom du graphe* et G est un graphe RDF. Un *jeu de données RDF* est composé d'un graphe RDF distinct, qu'on appelle le *graphe par défaut* du jeu de données, et d'un ensemble de graphes nommés tels qu'une URL ne peut apparaitre qu'une seule fois comme nom de graphe.

Ce modèle permet de représenter une série temporelle comme le montre l'exemple 2. Cet exemple utilise la syntaxe TriG [5] qui étend la syntaxe Turtle avec la possibilité de nommer des graphes en associant une URL à un bloc Turtle inséré dans des accolades.

```
# Graphe par défaut
ex:mesure1 a ont:MesureDeTemperature;
  ont:date "2017-07-05T09:42:42+02:00"^^xsd:dateTime;
  ont:serieTemp ex:serie1 .
ex:mesure2 a ont:MesureDeTemperature;
  ont:date "2017-07-05T10:42:42+02:00"^^xsd:dateTime;
  ont:serieTemp ex:serie1 .
ex:serie1 a ont:SerieTemporelle;
  ont:debut "2015-03-31T09:42:42+02:00"^^xsd:dateTime;
  ont:fin "2015-07-05T12:42:42+02:00"^^xsd:dateTime;
  ont:frequence "1 mesure par heure"^^xsd:string .
# ...

# Graphes nommés
ex:mesure1 {
  ex:lieu meteo:temperature "27 Cel"^^cdt:ucum
}
```

2. Ici aussi, le standard autorise les IRI, mentionnés dans la note de bas de page précédente, mais nous simplifions le modèle dans le cadre d'OpenSensingCity.

```
ex:mesure2 {
  ex:lieu meteo:temperature "29 Cel"^^cdt:ucum
}
# ...
```

Exemple RDF 2 – Exemple de jeu de données RDF décrivant une série temporelle.

4.2 Réification

La réification en RDF consiste à faire d'une déclaration une entité à part entière décrite dans le même graphe RDF qui décrit les données de la déclaration. Plusieurs modèles permettent de faire cela : la réification standard RDF [4, §5.3], le modèle des relations N-aires [15], les propriétés singletons [14], le modèle NdFluents [9].

4.2.1 La réification standard en RDF

Le standard RDF propose une manière de réifier une déclaration RDF en créant une entité représentant la déclaration et en la reliant aux trois composantes du triplet que l'on veut décrire, comme dans l'exemple 4. On remplace ainsi un triplet par au moins trois triplets et on y ajoute toutes les métadonnées nécessaires.

```
# triplet initial
ex:lieu
   meteo:temperature
      "27 Cel"^^cdt:ucum
            .
```

Exemple RDF 3 – Triplet non réifié.

```
[] a rdf:Statement;
  rdf:subject ex:lieu;
  rdf:predicate meteo:temperature;
  rdf:object "27 Cel"^^cdt:ucum;
  ont:date "2017-07-05T09:42:42+02:00"^^xsd:dateTime .
```

Exemple RDF 4 – Réification standard en RDF.

4.2.2 Les relations N-aires

Les triplets RDF ne permettent de représenter *que des relations binaires*. Or, il existe de nombreuses situations où l'on est confronté à des relations ternaires, voire quaternaires, etc. Afin de proposer aux utilisateurs de RDF des solutions pour exprimer de telles relations à l'aide de triplets RDF, le W3C [3] a publié une note préconisant deux approches pour l'expression des relations N-aires en RDF :

— en transformant la relation définie par le prédicat en une classe (exemple 5),

3. *World Wide Web Consortium*, organisme de normalisation des langages et protocoles du Web tels que HTML, XML, CSS.

— en scindant la relation en deux relations et en faisant appel à un nœud intermédiaire (exemple 6).

```
[] a ont:Mesure;
 ont:objet-de-la-mesure ex:lieu;
 ont:valeur-de-mesure "27 Cel"^^cdt:ucum;
 ont:date "2017-07-05T09:42:42+02:00"^^xsd:dateTime .
```

Exemple RDF 5 – Relation réifiée en classe.

Le modèle définissant une classe à la place d'une propriété est utilisé dans l'ontologie SSN (*Semantic Sensor Network* [10]) élaborée au sein du W3C pour décrire des données de capteurs. Ce modèle oblige à définir au cas par cas les classes et propriétés utilisées. L'autre modèle se construit comme dans l'exemple 6.

```
ex:lieu meteo:temperatureS [
 meteo:temperatureP "27 Cel"^^cdt:ucum;
 ont:date "2017-07-05T09:42:42+02:00"^^xsd:dateTime
] .
```

Exemple RDF 6 – Relation dédoublée.

Le modèle scindant la relation en deux est utilisé par la base de connaissances collaborative Wikidata, dont la structuration est expliquée dans [8].

4.2.3 Les propriétés singletons

Les propriétés singletons consistent à introduire une relation qui n'existe qu'entre le sujet et l'objet d'un triplet à annoter, puis à associer les métadonnées à cette relation spéciale, comme montré dans l'exemple 7. La relation pour cet objet est reliée à la relation initiale par le biais du prédicat spécial `rdf:isSingletonPropertyOf`.

```
ex:lieu ex:temperature#lieu-27cel "27 Cel"^^cdt:ucum .
ex:temperature#lieu-27cel ont:date "2017-07-05T09:42:42+02:00"^^xsd:dateTime;
   rdf:isSingletonPropertyOf meteo:temperature .
```

Exemple RDF 7 – Propriété singleton.

Les propriétés singletons ont une sémantique particulière, c'est-à-dire, que si l'on veut faire des raisonnements sur des données utilisant ces propriétés, on ne peut pas utiliser un moteur de raisonnement automatique standard pour RDF. Il faut implémenter des règles de déductions supplémentaires.

4.2.4 NdFluents

NdFuents est à la fois une manière de représenter des informations contextuelles et une ontologie permettant de raisonner sur cette représentation. Dans

cette approche, on considère que si une entité est en relation avec une autre, il ne s'agit pas de l'entité dans sa globalité et selon toutes ces dimensions, mais l'entité dans le contexte où la relation existe. Par exemple, une personne est en relation avec son âge signifie en fait qu'il y a un âge à un instant pour la "tranche" (en anglais, *slice*) de la personne existante à cet instant. On dit que la personne est un *fluent* car ses caractéristiques fluctuent dans le temps (par opposition à un *perdurant* qui est une entité atemporelle, qui perdure). De même, on peut considérer comme vrai l'affirmation "la Terre est plate" dans le contexte de l'opinion des platistes. Dans l'approche NdFluents, cela revient à dire qu'il existe une "tranche" de la Terre qui n'existe que dans l'opinion des platistes. Cette tranche de Terre est effectivement dans la catégorie des objets plats.

Ainsi, si l'on dit qu'en tel lieu il fait 27°C, il faut comprendre que la "tranche" de ce lieu à cet instant selon l'instrument de mesure indique 27°C.

```
ex:lieu#2017-capt1 meteo:temperature "27 Cel"^^cdt:ucum .
  nd:isContextualPartOf ex:lieu;
  nd:inContext ex:context-2017-07-05-capteur1 .
ex:context-2017-07-05-capteur1 ont:date "2017-07-05T09:42:42+02:00"^^xsd:dateTime;
  ont:provenance ex:capteur1 .
```

Exemple RDF 8 – Approche NdFluents.

Un des avantages de cette méthode est qu'elle préserve les raisonnements, c'est-à-dire, que si on peut effectuer une déduction à partir de la déclaration hors contexte (il fait 27°C), on peut également le déduire avec l'information en contexte. En l'occurence, si nous avons une règle de déduction indiquant qu'un lieu à 27°C est chaud, on peut déduire que `ex:lieu` est chaud avec la déclaration hors contexte, et l'on peut également déduire que `ex:lieu#2017-capt1` est chaud avec la déclaration mise en contexte.

4.3 Modèle de données OpenSensingCity

Pour les besoins du projet OpenSensingCity, nous faisons le choix d'un modèle hybride. Les jeux de données statiques sont mis à disposition en RDF à une adresse Web. Ceci équivaut à construire un graphe nommé dont le nom est l'URL du fichier, et le graphe associé est le contenu du fichier. Aussi, les entités décrites dans les données sont elles-mêmes identifiées par des URL, conformément aux principes des données liées. À ces URL, on doit pouvoir retrouver une description en RDF desdites entités. Ceci permet effectivement de lier les jeux de données entre eux. De cette manière, nous souhaitons faire de la plateforme de données ouvertes d'OpenSensingCity une véritable

plateforme de données liées conforme au standard *Linked Data Platform 1.0* [19].

Concernant les données dynamiques, les données courantes sont accessibles à l'URL du flux de données. Ainsi, un flux correspond à un graphe nommé dont le contenu évolue dans le temps. Quant aux séries temporelles, elles sont identifiées avec leur propre URL et liées aux différentes mesures en utilisant le modèle des relations N-aires passant par une classe de mesures : le modèle des réseaux de capteurs sémantiques SSN évoqué en section 4.2.2.

5 Catalogues, jeux de données, métadonnées

Dans l'approche OpenSensingCity, les métadonnées sur les jeux de données forment elles-mêmes un jeu de données pouvant avoir ses propres métadonnées. Les jeux de données sont regroupés au sein de catalogues de données. Généralement, à un portail de données ouvertes correspond un catalogue : il existe le catalogue des jeux de données du portail de la ville de Paris, celui de Rennes, de Toulouse, etc. Sur certains portail open data de grande ampleur, plusieurs catalogues peuvent coexister. Dans ce cas, un catalogue correspond soit à une thématique (p.ex., transport, énergie), soit à un organisme fournisseur (p.ex., INSEE, ING, ministère).

La description d'un catalogue de jeux de données et des jeux de données eux-mêmes peut se faire en RDF à l'aide d'un standard : DCAT [12]. La figure 3 donne un aperçu des concepts, attributs et relations définis par, ou pouvant être utilisés avec DCAT.

Dans une description DCAT, un catalogue (entité de type `dcat:Catalog`) est rattaché à des jeux de données (entité de type `dcat:Dataset`) et peut être associé à une taxonomie de thèmes (entité de type `skos:ConceptScheme`) pour organiser les jeux de données en catégories (p.ex., transport, services). Un jeu de données est une entité abstraite qui n'est pas liée à un format. Au contraire, un jeu de données peut être lié à plusieurs formats de distribution de la donnée. On associe donc à chaque jeu de données une ou plusieurs distributions (entité de type `dcat:Distribution`) correspondant aux objets numériques concrets pouvant être téléchargés, copiés, traités. Chaque jeu de données est associé à un thème (entité de type `skos:Concept`) correspondant à un concept de la taxonomie utilisée par le catalogue. Enfin, il est possible d'externaliser les métadonnées sur les jeux de données dans un enregistrement de catalogue (entité de type `dcat:CatalogRecord`), lié à un unique jeu de donnée.

Chacun des éléments décrits dans un catalogue DCAT peut avoir des

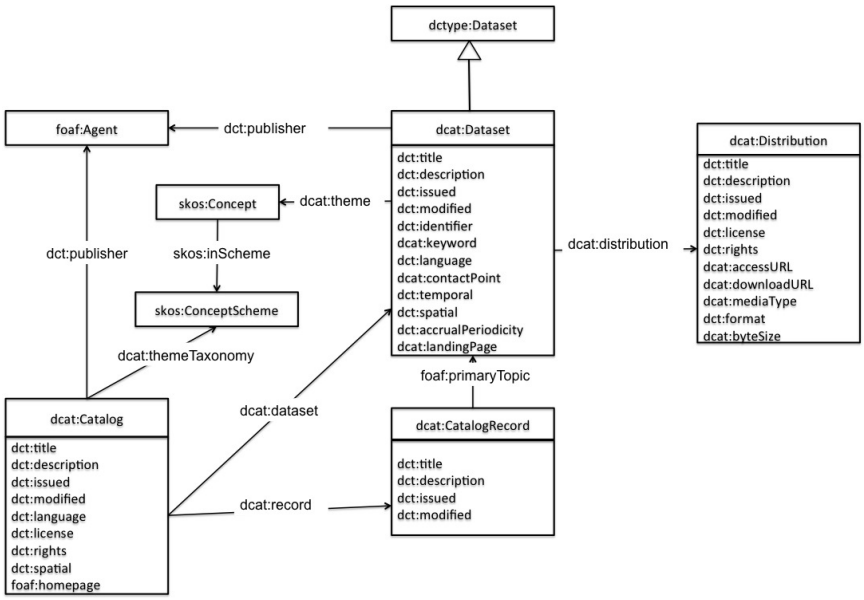

FIGURE 3 – Aperçu des termes du vocabulaire DCAT et de leur lien. Image extraite de [12] disponible à `https://www.w3.org/TR/vocab-dcat/dcat-model.jpg`. Copyright World Wide Web Consortium, (MIT, ERCIM, Keio, Beihang). `http://www.w3.org/Consortium/Legal/2015/doc-license`

métadonnées issues d'autres vocabulaires que DCAT. En particulier, les termes commençant par `dct:` dans la figure 3 sont recommandés par le standard et issus de la norme *Dublin Core Metadata Terms* [20]. De même, DCAT utilise des termes du standard W3C pour les systèmes simples d'organisation de la connaissance (*Simple Knowledge Organization Systems* (SKOS) [13]).

Dans le cadre du projet OpenSensingCity, nous envisageons d'utiliser les vocabulaires RDF suivants pour la description des métadonnées de jeux de données et de catalogues (en indiquant l'organisme de standardisation qui le définit et le préfixe que nous utilisons) :

— Creative Commons Rights Expression Language [4] (Creative Commons, préfixe `cc:`) : description des licenses associées aux données,
— DCAT (W3C, préfixe `dcat:`) : description des catalogues et jeux de données,
— Dublin Core Metadata Terms (DCMI, préfixe `dct:`) : métadonnées

4. Describing Copyright in RDF. `https://creativecommons.org/ns`

génériques de documents physiques ou numériques,
— GeoSPARQL [16] (OGC, préfixe `gsp`) : description de la couverture géographique des jeux de données,
— l'ontologie des organisations Org [18] (W3C, préfixe `org:`) : métadonnées sur les organismes, institutions et entreprises publiant les données,
— l'ontologie de la provenance PROV-O [11] (W3C, préfixe `prov:`) : pour indiquer la source des informations et tracer la chaine de traitement,
— Simple Knowledge Organization Systems (W3C, préfixe `skos:`) : pour expliciter la catégorisation des jeux de données et thématiques.

Tous ces vocabulaires sont spécifiés dans des standards et couvrent différentes parties des métadonnées de l'open data. Les données elles-mêmes utiliseront les ontologies répertoriées dans le cadre de la tâche 4 du projet et indexées dans la plateforme Smart City Artifacts [1] [5].

Références

[1] Noorani Bakerally, Olivier Boissier, and Antoine Zimmermann. Smart City Artifact Web Portal. In Harald Sack, Giuseppe Rizzo, Nadine Steinmetz, Dunja Mladenic, Sören Auer, and Christoph Lange, editors, *The Semantic Web - ESWC 2016 Satellite Events, Heraklion, Crete, Greece, May 29 - June 2, 2016, Revised Selected Papers*, volume 9989 of *Lecture Notes in Computer Science*, pages 172–177. Springer-Verlag, May 2016.

[2] Tim Berners-Lee. Linked data. Published online at `http://www.w3.org/DesignIssues/LinkedData.html`, 2006. W3C Design issue.

[3] Tim Berners-Lee, Larry Masinter, and Mark P. McCahill. Uniform Resource Locators (URL). Technical report, Internet Engineering Task Force, December 1994.

[4] Dan Brickley and Ramanathan V. Guha. RDF Schema 1.1, W3C Recommendation 25 February 2014. W3C Recommendation, World Wide Web Consortium, February 25 2014.

[5] Gavin Carothers and Andy Seaborne. RDF 1.1 TriG - RDF Dataset Language, W3C Recommendation 25 February 2014. W3C Recommendation, World Wide Web Consortium, February 25 2014.

[6] Richard Cyganiak, David Wood, and Markus Lanthaler. RDF 1.1 Concepts and Abstract Syntax, W3C Recommendation 25 February

5. Smart City Artifacts web portal `http://opensensingcity.emse.fr/scans/`

2014. W3C Recommendation, World Wide Web Consortium, February 25 2014.

[7] Martin J. Dürst and Michel Suignard. Internationalized Resource Identifiers (IRIs). Technical report, Internet Engineering Task Force, January 2005.

[8] Fredo Erxleben, Michael Günther, Markus Krötzsch, Julian Mendez, and Denny Vrandecic. Introducing Wikidata to the Linked Data Web. In Peter Mika, Tania Tudorache, Abraham Bernstein, Christopher A. Welty, Craig A. Knoblock, Denny Vrandecic, Paul T. Groth, Natasha Fridman Noy, Krzysztof Janowicz, and Carole A. Goble, editors, *The Semantic Web - ISWC 2014 - 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part I*, volume 8796 of *Lecture Notes in Computer Science*, pages 50–65. Springer-Verlag, October 2014.

[9] José Miguel Giménez-García, Antoine Zimmermann, and Pierre Maret. NdFluents : An Ontology for Annotated Statements with Inference Preservation. In Eva Blomqvist, Diana Maynard, Aldo Gangemi, Rinke Hoekstra, Pascal Hitzler, and Olaf Hartig, editors, *The Semantic Web - 14th International Conference, ESWC 2017, Portorož, Slovenia, May 28 - June 1, 2017, Proceedings, Part I*, volume 10249 of *Lecture Notes in Computer Science*, pages 638–654. Springer-Verlag, May 2017.

[10] Armin Haller, Krzysztof Janowicz, Simon Cox, Danh Le Phuoc, Jamie Taylor, and Maxime Lefrançois. Semantic Sensor Network Ontology, W3C Working Draft 04 May 2017. W3C Working Draft, World Wide Web Consortium, May 4 2017.

[11] Timothy Lebo, Satya Sahoo, and Deborah L. McGuinness. PROV-O : The PROV Ontology, W3C Candidate Recommendation 11 December 2012. W3C Candidate Recommendation, World Wide Web Consortium, December 11 2012.

[12] Fadi Maali and John Erickson. Data Catalog Vocabulary (DCAT), W3C Recommendation 16 January 2014. W3C Recommendation, World Wide Web Consortium, January 16 2014.

[13] Alistair Miles and Sean Bechhofer. SKOS Simple Knowledge Organization System, Reference, W3C Recommendation 18 August 2009. W3C Recommendation, World Wide Web Consortium, August 18 2009.

[14] Vinh Nguyen, Olivier Bodenreider, and Amit Sheth. Don't like RDF reification ? : making statements about statements using singleton property. In Chin-Wan Chung, Andrei Z. Broder, Kyuseok Shim, and Torsten

Suel, editors, *23rd International World Wide Web Conference, WWW '14, Seoul, Republic of Korea, April 7-11, 2014*, pages 759–770. ACM Press, April 2014.

[15] Natasha Fridman Noy and Alan L. Rector. Defining N-ary Relations on the Semantic Web, W3C Working Group Note 12 April 2006. W3C Note, World Wide Web Consortium, April 12 2006.

[16] Matthew Perry and John Herring. OGC GeoSPARQL - A Geographic Query Language for RDF Data. Ogc implementation standard, Open Geospatial Consortium, September 10 2012.

[17] Eric Prud'hommeaux and Gavin Carothers. RDF 1.1 Turtle - Terse RDF Triple Language, W3C Recommendation 25 February 2014. W3C Recommendation, World Wide Web Consortium, February 25 2014.

[18] Dave Reynolds. The Organization Ontology, W3C Recommendation 16 January 2014. W3C Recommendation, World Wide Web Consortium, January 16 2014.

[19] Steve Speicher, John Arwe, and Ashok Malhotra. Linked Data Platform 1.0, W3C Recommendation 26 February 2015. W3C Recommendation, World Wide Web Consortium, February 26 2015.

[20] DCMI usage board. Dcmi metadata terms. Technical report, Dublin Core Metadata Initiative, June 14 2012.

A Portails de données ouvertes français utilisant DCAT

Ceci est une liste, peut-être non exhaustive, de portail open data utilisant DCAT. Cette liste a été extraite en examinant le recensement des portails open data par l'entreprise OpenDataSoft [6].

- `https://tourisme62.opendatasoft.com/`
- `https://data.angers.fr/` Portail urbain d'Angers.
- `https://bistrotdepays.opendatasoft.com/`
- `https://datanova.legroupe.laposte.fr/`
- `https://data.sarthe.fr/`
- `https://data.enedis.fr/`
- `https://opendata.hauts-de-seine.fr/`
- `https://data.grandpoitiers.fr/` Portail urbain de l'agglomération de Poitiers.

6. La liste des Portails Open Data dans le Monde. `https://www.opendatasoft.com/a-comprehensive-list-of-all-open-data-portals-around-the-world`

— https://data.iledefrance.fr/
— https://datainfolocale.opendatasoft.com/
— https://data.haute-garonne.fr/
— https://navitia.opendatasoft.com/
— https://www.data.corsica/
— https://opendata.paris.fr/ Portail urbain de la ville de Paris.
— https://data.ratp.fr/
— https://data.rennesmetropole.fr/ Portail urbain de l'agglomération de Rennes.
— https://ressources.data.sncf.com/
— https://opendata.stif.info/
— https://data.toulouse-metropole.fr/ Portail urbain de l'agglomération de Toulouse.
— https://data.agen.fr/ Portail urbain d'Agen.
— https://data.issy.com/ Portail urbain d'Issy-les-Moulineaux.