

Support uniforme de types de données personnalisés dans RDF et SPARQL

Maxime Lefrançois, Antoine Zimmermann

Univ Lyon, MINES Saint-Étienne, CNRS, Laboratoire Hubert Curien UMR 5516,
F-42023 Saint-Étienne, France
prenom.nom@emse.fr

Résumé. Les littéraux sont les nœuds terminaux du modèle de données RDF, et permettent d’encoder des données telles que des nombres (`"12.5"^^xsd:decimal`), des dates (`"2017-01-26T23:57:15"^^xsd:dateTime`), ou tout autre type d’information (`"vert pomme"^^ex:couleur`). Les moteurs RDF/SPARQL savent tester l’égalité ou comparer les littéraux RDF dont le type de données leur est connu (ce qui est le cas de `xsd:decimal` et `xsd:dateTime`). Mais lorsqu’un type de données est inconnu d’un moteur RDF/SPARQL (comme `ex:couleur`), il n’a à priori aucun moyen d’en « découvrir » la sémantique. Dans cet article, nous attaquons ce problème et étudions comment permettre: (i) aux éditeurs de données de publier la définition de types de données personnalisés sur le Web, et (ii) aux moteurs RDF/SPARQL de découvrir à la volée ces types de données personnalisés, et de les utiliser de manière uniforme. Nous discutons de différentes solutions possibles qui tirent partie des principes du Web des données, et détaillons une solution concrète basée sur le déréférencement et le langage JavaScript, suffisamment générique pour être utilisée pour des types de données personnalisés arbitrairement complexes.

1 Introduction

Un littéral RDF est composé d’une chaîne de caractères UNICODE (la forme lexicale) et d’une IRI de type de données¹ qui identifie un type de données (Cyganiak et al., 2014, §5). Ils forment avec les IRI et les nœuds anonymes les atomes du modèle de données RDF. Les IRI forment les liens qui tissent la toile du Web des données, mais ce sont les littéraux qui *in fine* encodent les données telles que des nombres (`"12.5"^^xsd:decimal`), des dates (`"2017-01-26T23:57:15"^^xsd:dateTime`), ou tout autre type d’information (`"vert pomme"^^ex:couleur`). Les moteurs RDF/SPARQL savent tester l’égalité ou comparer les littéraux RDF dont le type de données leur est connu (ce qui est le cas de `xsd:decimal` et `xsd:dateTime`). On dit alors qu’ils *supporte* le type de données. En pratique, les moteurs sont programmés pour supporter un ensemble fini de types de données.²

1. Et lorsque l’IRI du type de données est `rdf:langString`, d’une étiquette de langue. Cependant ce type de littéral ne nous intéresse pas dans le cadre de cette étude.

2. Il s’agit au moins de l’ensemble des types de données XSD, mais certains moteurs en supportent d’autres comme ceux du standard OGC GeoSPARQL (Perry et Herring, 2012).