The Unified Code for Units of Measure in RDF: cdt:ucum and other UCUM Datatypes

Maxime Lefrançois, Antoine Zimmermann

The Unified Code for Units of Measure in RDF: cdt:ucum and other UCUM Datatypes

Maxime Lefrançois, Antoine Zimmermann

Univ Lyon, MINES Saint-Étienne, CNRS, Laboratoire Hubert Curien UMR 5516,
F-42023 Saint-Étienne, France
{maxime.lefrancois,antoine.zimmermann}@emse.fr

Abstract. Being able to describe quantity values and their units is a requirement that is common to many applications in several industrial sectors such as manufacturing, transport and logistics, personal and public health, smart cities, energy, environment, buildings, agriculture. Different ontologies have been developed to describe units, their relations, and quantities with their values. In this paper we propose an alternative approach that leverages the Unified Code of Units of Measure, a code system intended to include *all* units of measures being contemporarily used in international sciences, engineering, and business. Our approach consists of a main UCUM datatype identified by IRI http://w3id.org/lindt/custom_datatypes#ucum, abbreviated as cdt:ucum. This datatype can be used for lightweight encoding and querying of quantity values, in a wide range of applications where representing and reasoning with quantity kinds and values is more important than reasoning with units. We compare our approach with existing approaches, and demonstrate it with our implementation on top of Apache Jena and an online testing tool.

1 Introduction

Applications in many industry sectors rely on quantity values with units of measures, for sensor observations, actuation, design calculations/simulations, quantitative general knowledge, etc. A typical way to convey the value of a quantity in RDF consists in using a structure with one triple providing a numerical value as a literal in standard datatypes (xsd:float, xsd:double, xsd:decimal), and a triple with an IRI identifying the unit. A dedicated ontology can define the properties that connect the quantity value to the numerical value and the unit. An alternative approach relies on custom datatypes [5].

In this paper, we introduce an RDF datatype, cdt:ucum, that transposes to RDF the full expressive power of the Unified Code for Units of Measure (UCUM [8]), enabling lightweight descriptions and querying of physical quantities using a single datatype. We currently provide 32 more specific datatypes such as cdt:speed and cdt:length to further specify the quantity kind of quantity values, but more datatypes may be introduced in the future.

We first show in Section 2 how quantity values are typically described in RDF with existing ontologies or custom datatypes. Then, in Section 3, we introduce the cdt:ucum datatype, highlighting the conciseness of the representation with many examples. Finally, in Section 4, we describe our implementation as an extension of Apache Jena with support for cdt:ucum in SPARQL queries, with an online testing tool.

2 Related Work

We identify two approaches to represent physical quantities in RDF: using ontologies, or using custom datatypes.

Using ontologies of units of measurements. The classical approach consists in using an ontology to describe units, their relations, and measurements. A recent survey [4] compares and evaluates eight well known ontologies for units of measurements, among which MUO [6], QUDV [1], OM [7], QUDT [3]. This survey also report on the Wikidata corpus[1] that currently contains over 4.4 k measurement units and 4.1 k non-prefixed units. Using such ontologies, quantity values are usually represented as OWL individuals linked to some numeric value and to some individual representing a unit of measure. For example, Listing 1.1 represents the quantity value 29 °C using QUDT 1.1.

Listing 1.1: Description of a quantity value using QUDT 1.1

```
@prefix qudt-1-1: <http://qudt.org/1.1/schema/qudt#> .
@prefix qudt-unit-1-1: <http://qudt.org/1.1/vocab/unit#> .
[ ] a qudt-1-1:QuantityValue ;
     qudt-1-1:unit qudt-unit-1-1:DegreeCelsius ;
     qudt-1-1:numericValue "29"^^xsd:double .
```

Not all possible units of measurement are (or will be) defined in these ontologies, for example QUDT 1.1 defines a unit for kilowatt hour, but not megawatt hour. Application developers in the energy domain can force themselves to use units they are not used to, or they can define missing units using the definition mechanism provided by QUDT. This extension mechanism uses concepts such as base units, conversion offsets and multipliers, numerator and denominator. For example, Listing 1.2 illustrates how the unit megawatt hour may be defined. Even then, two energy operators may define the same unit using different URIs, leading to potential interoperability issues.

Listing 1.2: Description of a new unit using QUDT 1.1

```
ex:Megawatthour a qudt-1-1:EnergyAndWorkUnit ;
    rdfs:label "Megawatthour" ;
    qudt-1-1:symbol "MW-hr" ;
    qudt-1-1:conversionOffset "0.0"^^xsd:double ;
    qudt-1-1:conversionMultiplier "3.6E9"^^xsd:double ;
```

Datasets using quantity values defined with such ontologies require 4 triples every time a quantity needs to be linked to a quantity value, and complex mechanisms are needed to canonicalize quantity values so as to query them uniformly. We are not aware of any existing support of QUDT or OM custom units in any RDF or SPARQL engine.

Using datatypes. DBpedia has many datatypes[2], which are hard-coded in OntologyDatatypes.scala and listed in the DBpedia Mappings Wiki for reference. Dbpedia defines datatypes for physical dimensions (http://dbpedia.org/datatype/Area)

[1] https://www.wikidata.org/
[2] http://mappings.dbpedia.org/index.php/DBpedia_Datatypes

along with datatypes for specific units of measures (http://dbpedia.org/datatype/cubicInch). Yet, these datatypes do not dereference, so one cannot understand if Inch here is in the international customary units, U.S. survey lengths, British Imperial lengths, for example. Again, not all possible units of measurement are (or will be) defined in the Dbpedia ontology, and complex mechanisms are needed to canonicalize quantity values so as to query them uniformly.

We previously proposed an approach for RDF and SPARQL engines to support arbitrarily complex custom datatypes on-the-fly by dereferencing their URIs and retrieving specifications in JavaScript [5]. In this paper we are exclusively interested in datatypes for quantity values, and do not consider on-the-fly support capabilities.

3 Specification of cdt:ucum and other UCUM Datatypes

The Unified Code for Units of Measure (UCUM) [8] is a code system intended to include *all* units of measures being contemporarily used in international science, engineering, and business.

We define a RDF datatype UCUM identified by IRI http://w3id.org/lindt/custom_datatypes#ucum, abbreviated as cdt:ucum. Its lexical space is the concatenation of an xsd:decimal, optionally followed by `e` or `E` and the lexical form of an xsd:integer, at least one space, and a unit chosen in the case sensitive version of the UCUM code system. The value space corresponds to the set of measures, or quantity values as defined by the International Systems of Quantities. The lexical-to-value mapping maps lexical forms with a UCUM unit to their corresponding measures according to the International Systems of Quantities.

We also define a set of additional datatypes such as cdt:length and cdt:speed that further specify the quantity kind of quantity values. Their lexical spaces, value spaces, and lexical-to-value mappings are subsets of those of cdt:ucum. More such datatypes may be defined in the future. Table 1 lists examples of valid cdt:ucum literals, and their equivalent using more specific datatypes.

`"1 mA"^^cdt:ucum`	1 milli ampère. Same value as `"1e-3 C/s"^^cdt:ucum` and `"1 mA"^^cdt:electricCurrent`
`"1 MW.h"^^cdt:ucum`	1 mega watt hour.
`"1 [nmi_i]"^^cdt:ucum`	1 nautical mile, international customary unit. Same value as `"1852.0 m"^^cdt:length`, or `"0.9993618864985154 [nmi_br]"^^cdt:length` (nautical mile, British imperial unit)
`"1 cd/cm2/[pi]"^^cdt:ucum`	1 candela per square centimeter and divided by Pi: Valid UCUM unit definition equivalent to the Lambert brightness unit. Same value as `"1 Lmb"^^cdt:ucum`
`"1.8 [ppm]"^^cdt:ucum`	1.8 parts per million. Literals may be formed with stranger units such as `"1.8 M[ppm]"^^cdt:ucum` (mega parts per million, which is equal to 1). Same value as `"1.8 [ppm]"^^cdt:dimensionless`

Table 1: Some valid UCUM literals.

4 Implementation of UCUM Datatypes on Apache Jena

The UCUM specification has implementations in different languages. We used the latest version of `systems-ucum-java8`[3], an implementation leveraging the recent Java units of measurement API 2.0 (JSR 385), to add support of the 33 datatypes specified above on top of Apache Jena. Our extension, named `jena-ucum`, is open-source and available online[4]. It overloads native SPARQL operators (=, <, etc.) to compare UCUM literals, and arithmetic functions (+, -, *, /) to manipulate quantity value literals: 1. Add two commensurable quantity value literals; 2. Subtract a quantity value literals to a commensurable one; 3. Multiply two quantity value literals, or a quantity value literal and a scalar (xsd:int, xsd:decimal, xsd:float, xsd:double); 4. Divide a quantity value literal by a quantity value literal, a quantity value literal by a scalar, or a scalar by a quantity value literal. We additionally define a custom SPARQL function with IRI: `http://w3id.org/lindt/custom_datatypes#sameDimension` which takes two parameters and returns true if they are commensurable quantity values.

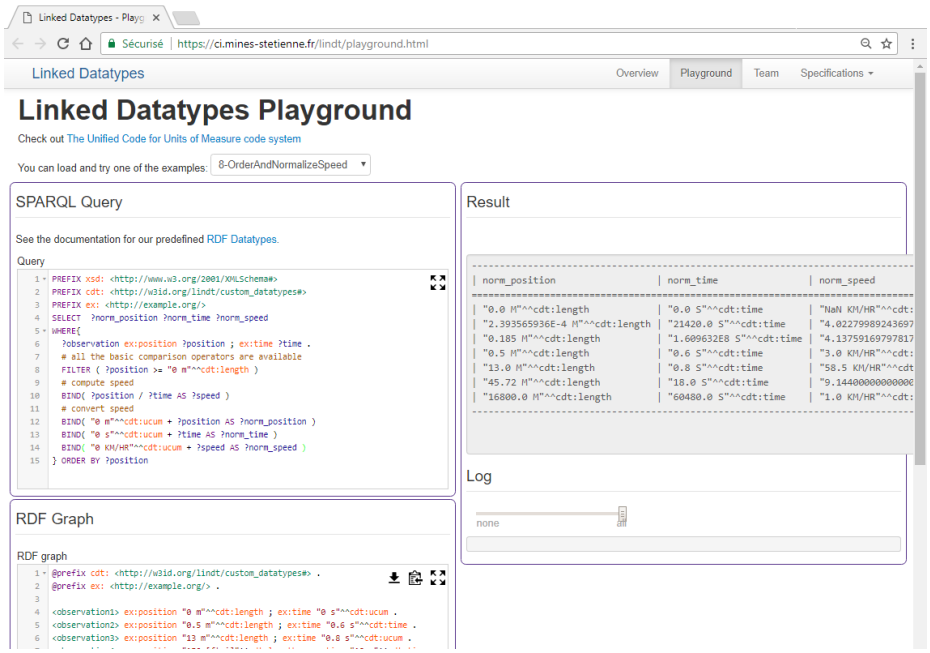

Fig. 1: Screenshot of the UCUM datatypes playground `https://w3id.org/lindt/playground.html`.

[3] Implementation of UCUM we used: `https://github.com/unitsofmeasurement/uom-systems/tree/master/ucum-java8`

[4] Our implementation on Jena: `https://github.com/OpenSensingCity/jena-ucum`

5 Demonstration

We demonstrate the UCUM datatypes using a playground illustrated on Figure 1 and accessible online.[5] The user can enter a SPARQL Construct or Select query and the default graph of the RDF Dataset on which it is evaluated. The result is computed in real-time and returned to the user using the WebSocket protocol.

Queries are predefined to progressively introduce the use of SPARQL comparison operators, arithmetic functions, solution sequence modifiers (`ORDER BY`). Other predefined queries are predefined to illustrate each of the 33 currently defined UCUM datatypes. We will also showcase how the UCUM datatypes may be used in combination with other vocabularies such as SOSA/SSN [2].

6 Conclusion

Using the UCUM datatypes, one only requires 1 triple to link a quantity to a fully qualified value, and one does not require custom mechanisms to canonicalize literals based on external descriptions of units of measurements. Using UCUM Datatypes, datasets are therefore drastically lightened, and queries are also simpler. The UCUM datatype can inherently represent an infinite set of custom units, and is therefore suitable for an open set of application domains.

A similar datatype could be defined to support amounts of money, potentially with any type of currencies and a timestamp for this currency.

Acknowledgments. This work has been partly funded by the ANR 14-CE24-0029 OpenSensingCity project.

References

1. Quantities, Units, Dimensions, Values (QUDV). SysML 1.2 Revision Task Force Working draft, Object Management Group, October 30 2009.
2. Armin Haller, Krzysztof Janowicz, Simon J D Cox, Danh Le Phuoc, Kerry Taylor, and Maxime Lefrançois. Semantic Sensor Network Ontology. W3C Recommendation, W3C, October 19 2017.
3. Ralph Hodgson, Paul J. Keller, Jack Hodges, and Jack Spivak. QUDT - Quantities, Units, Dimensions and Data Types Ontologies . Technical report, NASA, 2014.
4. Jan Martin Keil and Sirko Schindler. Comparison and evaluation of ontologies for units of measurement. *Semantic Web journal*, 2018. To appear, http://www.semantic-web-journal.net/content/comparison-and-evaluation-ontologies-units-measurement-1.
5. Maxime Lefrançois and Antoine Zimermann. Supporting Arbitrary Custom Datatypes in RDF and SPARQL. In *Proceedings of the Extended Semantic Web Conference, ESWC*, May 2016.
6. Luis Polo and Diego Berrueta. MUO - Measurement Units Ontology, Working Draft - DD April 2008. Working draft, Fundación CTIC, April 2008.
7. Hajo Rijgersberg, Mark van Assem, and Jan L. Top. Ontology of units of measure and related concepts. *Semantic Web Journal*, 4(1):3–13, 2013.
8. Gunther Shadow and Clement J. McDonald. The Unified Code for Units of Measure. Technical report, Regenstrief Institute, Inc., October 22 2013.

[5] UCUM Datatype playground https://w3id.org/lindt/playground.html