Introduction to the QUDT Ontology

Quantities, Units, Dimensions and Types

April 28, 2020

QUDT

Why have a units ontology?

- Provides a formal way of specifying units explicitly, thereby avoiding tacit conventions that are prone to misinterpretation.

- QUDT distinguishes between variants of a given unit.
  - unit:BTU_IT
    - British Thermal Unit (BTU or Btu) is a traditional unit of energy equal to about 1.0550558526 kilojoule.
  - unit:BTU_TH
    - British Thermal Unit (thermochemical definition, BTUth) is a traditional unit of energy equal to about 1.0543502645 kilojoule. It is approximately the amount of energy needed to heat 1 pound (0.454 kg) of water from 39°F to 40°F

Why have a units ontology?

- QUDT distinguishes between units of different types that are commonly referred to with the same name.
  - Second - unit:SEC (the unit of time)
  - Second - unit:ARCSEC (the unit of plane angle)
- The ontology provides explicit conversion information
  - Built-in conversion factors support mathematical operations combining different (commensurate) units
  - The data carries its metadata along with it, so unit conversion errors are avoided
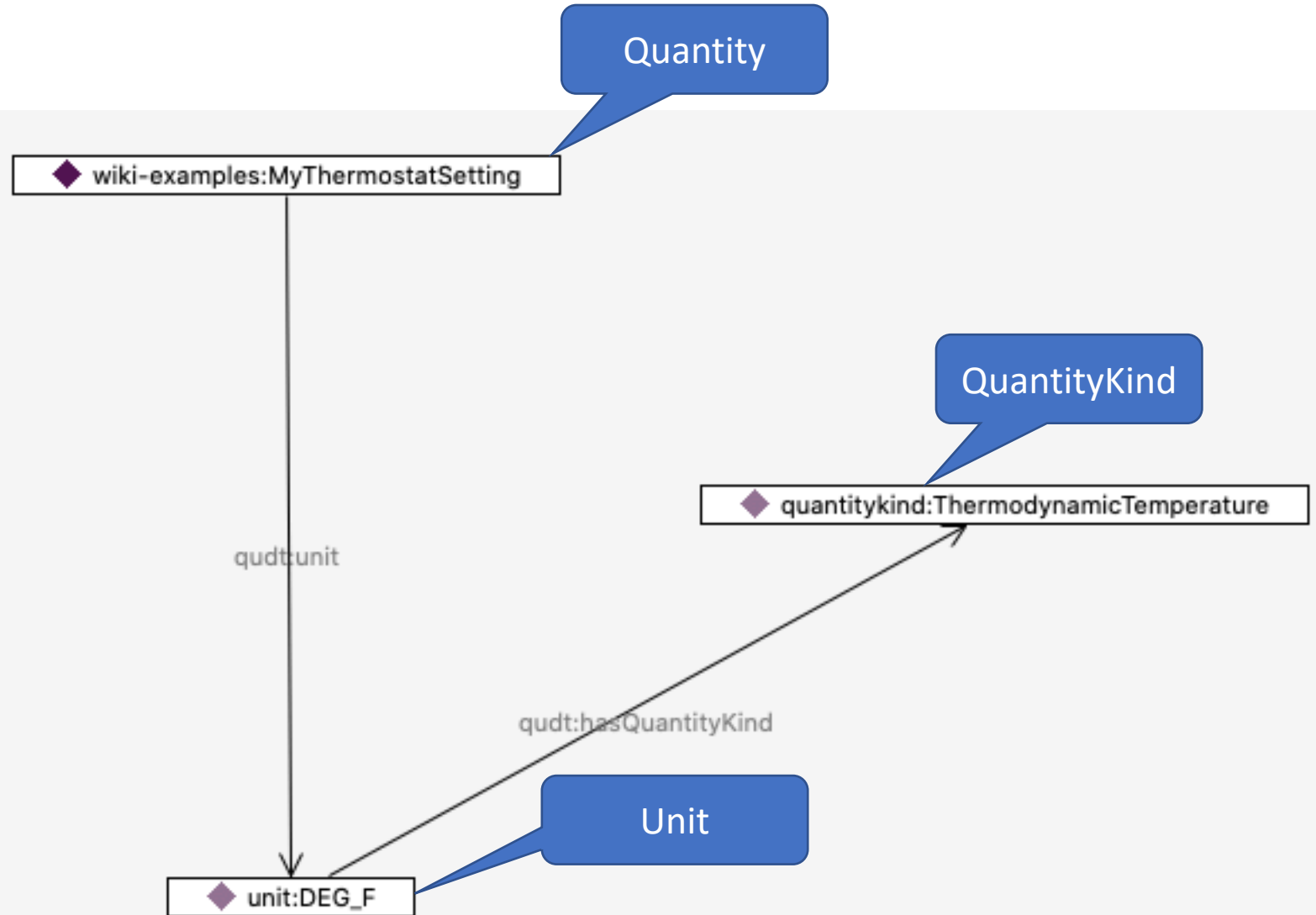
QUDT

The core design pattern

A simple example
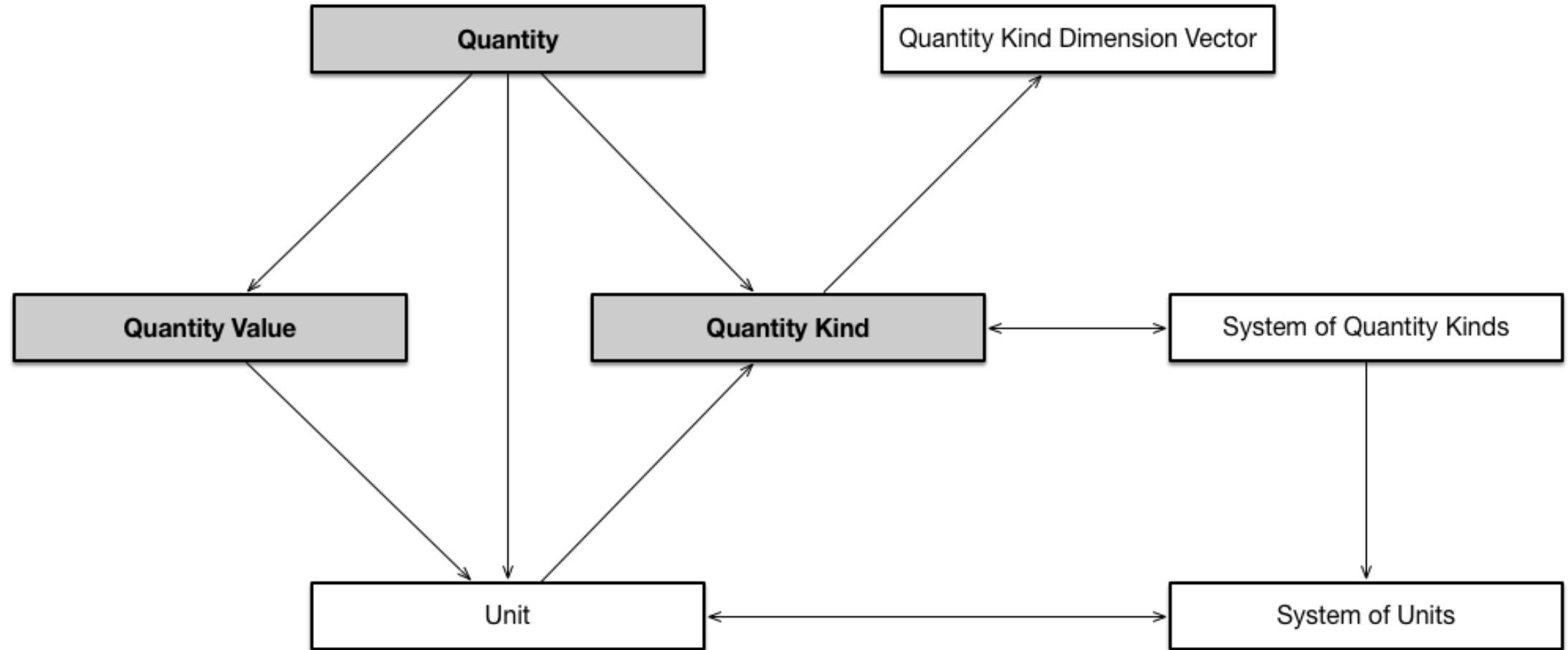
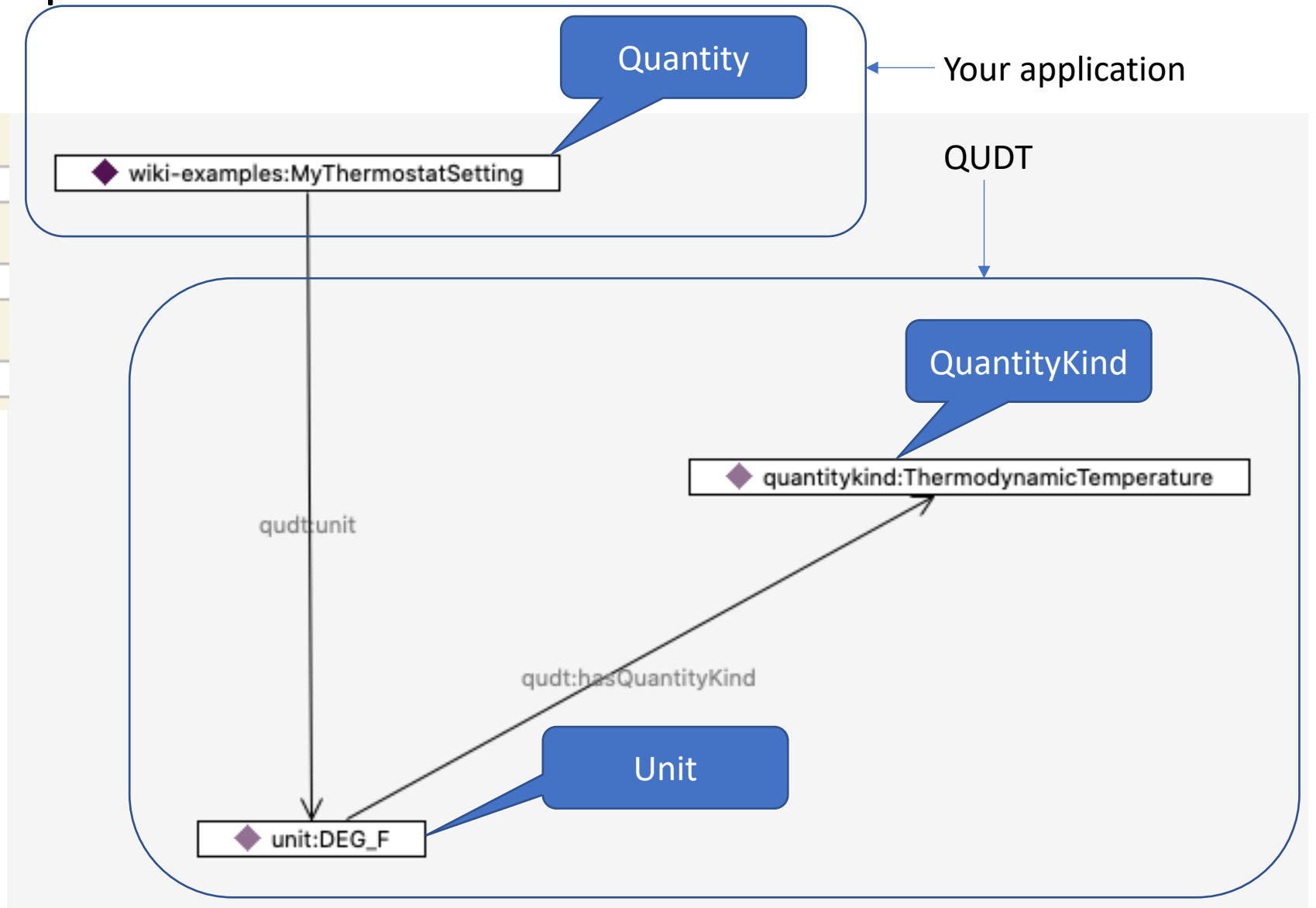

wiki-examples:MyThermostatSetting
  rdf:type qudt:Quantity ;
  qudt:unit unit:DEG_F ;
  qudt:value "72" .

A simple example

Deeper structure in QUDT

- All QuantityKinds are defined in terms of dimension vectors
  - E.g. the 7 base units of S.I. correspond to the 7 base dimension vectors
    - Length - meter (m)
    - Time - second (s)
    - Amount of substance - mole (mole)
    - Electric current - ampere (A)
    - Temperature - kelvin (K)
    - Luminous intensity - candela (cd)
    - Mass - kilogram (kg)
  - This allows automated
    - Unit conversion
    - Dimensional analysis
    - Identification/validation checking for commensurate units

QUDT

Automated generation of unit conversions

```
PREFIX qudt: http://qudt.org/schema/qudt/
PREFIX rdfs: http://www.w3.org/2000/01/rdf-schema#
PREFIX smf: http://topbraid.org/sparqlmotionfunctions#
PREFIX fn: http://www.w3.org/2005/xpath-functions#
SELECT DISTINCT ?toConvert ?label ?into ?otherUnitLabel ?multiplyBy ?multiplier
WHERE {
BIND ("To convert" AS ?toConvert) .
BIND ("into" AS ?into) .
BIND ("multiply by" AS ?multiplyBy) .
?unit a qudt:Unit .
?unit rdfs:label ?label .
FILTER (fn:contains (smf:lowerCase(?label), "milligray")) .
?unit qudt:conversionMultiplier ?cm1 .
?unit qudt:hasQuantityKind/qudt:hasDimensionVector ?qkdv .
?otherUnit qudt:hasQuantityKind/qudt:hasDimensionVector ?qkdv .
?otherUnit a qudt:Unit .
FILTER (?otherUnit != ?unit) .
?otherUnit qudt:conversionMultiplier ?cm2 .
?otherUnit rdfs:label ?otherUnitLabel .
BIND ((?cm1/?cm2) AS ?multiplier) .
}
ORDER BY ?label ?multiplier
```

This SPARQL query finds all units commensurate with the MilliGRAY, and calculates the conversion factor

To convert	MilliGRAY	into	Kilocalorie per Gram	multiply by	2.390057361376673E-10
To convert	MilliGRAY	into	MegaJ PER KiloGM	multiply by	1.0E-9
To convert	MilliGRAY	into	CAL_IT PER GM	multiply by	2.388458966274959E-7
To convert	MilliGRAY	into	calorieIT per gram (calIT/g)	multiply by	2.388458966274959E-7
To convert	MilliGRAY	into	CAL_TH PER GM	multiply by	2.390057361376673E-7
To convert	MilliGRAY	into	calorieTH per gram (calTH/g)	multiply by	2.390057361376673E-7
To convert	MilliGRAY	into	BTU-IT-PER-lb	multiply by	4.299226139294927E-7
To convert	MilliGRAY	into	British Thermal Unit (TH) Per Pound	multiply by	4.30210433032265E-7
To convert	MilliGRAY	into	J PER GM	multiply by	1.0E-6
To convert	MilliGRAY	into	BTU_IT PER LB_F	multiply by	4.216100966554345E-6
To convert	MilliGRAY	into	Gray	multiply by	0.001e0
To convert	MilliGRAY	into	Joule per Kilogram	multiply by	0.001e0
To convert	MilliGRAY	into	Sievert	multiply by	0.001e0
To convert	MilliGRAY	into	Rad	multiply by	0.1e0
To convert	MilliGRAY	into	Rem	multiply by	0.1e0
To convert	MilliGRAY	into	MilliSV	multiply by	1.0e0
To convert	MilliGRAY	into	ERG PER GM	multiply by	10.0e0
To convert	MilliGRAY	into	Erg per Gram	multiply by	10.0e0
To convert	MilliGRAY	into	MilliR equivalent man	multiply by	3875.968992248062e0
To convert	MilliGRAY	into	MilliR	multiply by	3875.968992248062e0

QUDT statistics

- At last count there are:
  - Number of units: 1522
  - There are 746 quantitykinds
  - There are 148 dimension vectors
- QUDT includes entries/cross references to units and codes from:
  - S.I. (BIPM/NIST)
  - ISO/IEC 80000
  - IEC 61360
  - UCUM
  - UN/ECE
  - BACnet (in progress)

QUDT

⋮ | Explore ▾ | Modify ▾ | Edit | ✱ ≡ ⓘ ✎ ▣ ⚠

Erg
URI http://qudt.org/vocab/unit/ERG

Merged View ▾

▾ Properties

symbol:	erg
has quantity kind:	Energy and work ▾

▾ Descriptions

description: An erg is the unit of energy and mechanical work in the centimetre-gram-second (CGS) system of units, symbol 'erg'. Its name is derived from the Greek ergon, meaning 'work'. An erg is the amount of work done by a force of one dyne exerted for a distance of one centimeter. In the CGS base units, it is equal to one gram centimeter-squared per second-squared ($g \cdot cm^2/s^2$). It is thus equal to 10^{-7} joules or 100 nanojoules in SI units. $1 erg = 10^{-7} J = 100 nJ$, $1 erg = 624.15 GeV = 6.2415 \times 10^{11} eV$, $1 erg = 1 dyne \cdot cm = 1g \cdot cm^2/s^2$.

latex definition: $g \cdot cm^2/s^2$

informative reference: http://en.wikipedia.org/wiki/Erg?oldid=490293432

dbpedia match: http://dbpedia.org/resource/Erg

▾ Equivalent Units

om unit:	http://www.ontology-of-units-of-measure.org/resource/om-2/erg ▾
iec-61360 code:	0112/2///62720#UAA429
unece common code:	A57
ucum code:	erg ERG
ucum case-sensitive code:	erg
ucum case-insensitive code:	ERG

▾ Conversion

conversion multiplier:	1.0E-7
conversion offset:	0.0

QUDT

Useful links

- http://qudt.org – Homepage
- https://github.com/qudt/qudt-public-repo – Github repository
- https://github.com/qudt/qudt-public-repo/wiki/User-Guide-for-QUDT – User Guide (in progress)

- There is much more to the model, including support for other systems of units, dimensionless units, complex datatypes, treatment of non-linear units and scales…
- As an open source project, we are always looking for participants to help error-check, add new units, provide domain expertise

QUDT