Semantic Web 0 (0) 1
IOS Press

Ontology of Units of Measure and Related Concepts

Editor(s): Carsten Keßler, University of Münster, Germany; Mathieu d'Aquin, Open University, United Kingdom; Stefan Dietze, Leibniz University Hanover, Germany
Solicited review(s): Werner Kuhn, University of Münster, Germany; Michael Compton, CSIRO, Australia; Simon Scheider, University of Münster, Germany; one anonymous reviewer

Hajo Rijgersberg [a,*], Mark van Assem [b] and Jan Top [a,b]
[a] *Food & Biobased Research, Wageningen University and Research centre, Bornse Weilanden 9, Wageningen, The Netherlands. E-mail: {first.last}@wur.nl*
[b] *Dept. of Computer Science, VU University, De Boelelaan 1081A, 1081 HV, The Netherlands. E-mail: mark@cs.vu.nl*

Abstract. This paper describes the Ontology of units of Measure and related concepts (OM), an OWL ontology of the domain of quantities and units of measure. OM supports making quantitative research data more explicit, so that the data can be integrated, verified and reproduced. The various options for modeling the domain are discussed. For example, physical quantities can be modeled either as classes, instances or properties. The design choices made are based on use cases from our own projects and general experience in the field. The use cases have been implemented as tools and web services. OM is compared with QUDT, another active effort for an OWL model in this domain. We note possibilities for integration of these efforts. We also discuss the role OWL plays in our approach.

Keywords: Ontology, Quantities, Units, OWL, Comparison

1. Introduction

Quantities and units, such as the *length* of a ship measured in *meters*, are vital to the exact sciences and engineering. Large amounts of quantitative data are used and produced in scientific experiments and in designs of artifacts. These data are stored in formal representations so that they can be manipulated by analysis and design tools. The need to integrate data from several sources has increased, e.g. to make new inferences on previously disconnected existing research efforts. In practice researchers often store their results in proprietary formats, such as spreadsheets or mathematical software packages, and only informally annotate the data (e.g. text entered in the head of a table such as "l (m)"). This lack of standardization and formal meaning of data hinders interoperability.

To improve the annotation and interpretation of quantitative research data, an ontology of units of measure is required. A number of different ontologies of units of measure exist, such as EngMath,[1] an ontology for mathematical modeling in engineering, written in Ontolingua by Gruber and Olsen [5]. UCUM,[2] created by Schadow *et al.* [16], is a system of codes of units and quantities to refer to in e.g. electronic data interchange (EDI) protocols. Another ontology is MUO,[3] the Measurement Units Ontology, in RDF [22], which adopts the units and quantities of UCUM and gives

*Corresponding author. E-mail: hajo.rijgersberg@wur.nl

[1] http://www.ksl.stanford.edu/knowledge-sharing/papers/engmath.html
[2] http://www.unitsofmeasure.org
[3] http://forge.morfeo-project.org/wiki_en/index.php/Units_of_measurement_ontology

them URLs. A number of other, older ontologies exist [3,8,13,11], as we have described previously [14].

Inspired by these different proposals, we have developed OM, the Ontology of units of Measure and related concepts. We have distilled a semi-formal description of the domain of units of measure from several paper standards that we have analyzed. An example of such a standard is the Guide for the Use of the International System of Units [19], by the NIST. The semi-formal description for example states that "multiples and submultiples of units combine a prefix and a singular unit". For a full list of statements and the sources we have used, see previous work [14]. OM is meant for use in science and engineering practice. Therefore we have based it on the technical standards used by physicists, chemists, engineers, food scientists, etc., such as the documents published by the NIST [19]. We have made no explicit efforts to link to terminology in measurement theory, as this appears to use a somewhat different terminology. For example, measurement theory doesn't seem to distinguish between what are called measurement scales and units in the technical standards. Moreover, we have not yet grounded OM in foundational ontologies such as DOLCE, the Descriptive Ontology for Linguistic and Cognitive Engineering. DOLCE aims at capturing ontological categories underlying natural language and human common sense [9]. Studying the precise relations between concepts in OM and in DOLCE is definitely an interesting option. However, it is beyond the scope of our present work, and not needed to achieve our goals in operational support for scientific and engineering. In the paper we include a short discussion on this issue.

OM is modeled in OWL 2 [21]. The choice for OWL 2 is motivated by the fact that it allows us to link instances to classes, and classes to instances. We consider this to be a required feature in the design of our ontology. We need it for expressing the relations between application area instances and quantity classes, and between quantity classes and commonly-used unit-of-measure instances. OM is published as Linked Open Data through our vocabulary and ontology portal Wurvoc.[4] OM can be used freely under the Creative Commons 3.0 Netherlands license. It was created by the authors using text editors and versioned using SVN.[5]

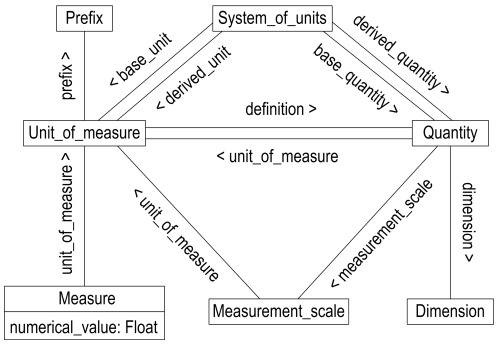

Fig. 1. Simplified class diagram (UML) of OM.

Figure 1 shows a part of the structure of OM. Quantities are related to units of measure and measurement scales that can express them. Units of measure need to be fixed by some observable standard phenomenon, such as the length of the path travelled by light in a vacuum during a time interval of 1/299 792 458 of a second, which is the definition of the meter. Measures, such as "3 kilogram" are used to indicate amounts of quantities. Units of measure can have a prefix. Quantities and units of measure are organized in systems of units such as the International System of Units (SI). Quantities have a dimension. Ontological choices in OM such as subclassing quantities and the distinction between units and scales are discussed in previous work [14].

In this article we discuss the domain of quantities and units and the design of our ontology that models this domain. The contributions of this article are twofold. Firstly, we present the domain and determine which use cases benefit from an ontological representation of this domain. These include mathematical applications such as unit conversion and dimensional analysis. Existing software products already perform these applications but rely on their own proprietary data formats. Secondly, we present OM and discuss the modeling choices we have made. We compare these modeling choices with those underlying the QUDT ontology,[6] which is another active effort to comprehensively model this domain in OWL.[7]

[4] http://www.wurvoc.org/vocabularies/om-1.8/
[5] http://subversion.apache.org/

[6] http://www.qudt.org
[7] The OASIS QUOMOS effort has an OWL version in the planning stage, see http://wiki.oasis-open.org/quomos/.

2. Domain Description

One of the main reasons to specify quantities and units is to use them for recording observations of the physical world. These observations are used for various goals such as creating new models and theories in science and developing new artifacts in engineering. A basic record consists at least of the elements (1) phenomenon (object or event being observed); (2) quantity kind (aspect of phenomenon being measured such as length or weight); (3) unit of measurement (e.g. meter); and (4) numerical value (e.g. "5.0"). In everyday language the term "quantity" is often used to denote just the quantity kind (e.g. "the quantity length"), but also sometimes a value and unit (e.g. "a quantity of 3 meter"). However, in the physical sciences this term may also refer to the combination of the quantity kind and the phenomenon, for example 'the density of water'. The quantity may have been measured, i.e. a numerical value and unit may be known for it. If the value and unit are known, the quantity can also be regarded as a record, e.g. "height (2 m)".

Some quantity kinds are more specific than others (e.g. diameter is a kind of length; work is a specific kind of energy, when a force acts against resistance to produce motion of a body). A unit together with a numerical value expresses the amount of one particular quantity; this is called a measure (e.g. "3 meter"). The amount of a particular quantity can only be expressed with a specific set of units (e.g. meter, yard, light year, etc. for distance).[8] A unit is defined by reference to a standard measurement. For example, 1 kilogram represents the mass of the International Kilogram Prototype, a platinum cylinder stored at the International Bureau of Weights and Measures in France.

Each unit can ultimately be expressed in terms of a set of *base units*. Which units are chosen as the base units depends on the *system of units*. For example the SI uses seven base units including meter and second. The CGS system on the other hand uses centimeter, gram and second as base units, plus different extensions to cover electromagnetism. Base units are considered to be mutually independent units (although e.g. the meter is defined through the second) within a system of units; they cannot be converted into one another. Non-base units are called *derived units*, and are defined by multiplication, division and exponentiation of base units. For example, newton is a derived unit (in the SI) defined as $\mathrm{kg{\cdot}m/s^2}$.

Units can be combined with a prefix such as milli or mega, which represents a multiplication factor (e.g. one micrometer is 10^{-6} meter). Its main use is to allow short numerical values in actual measures (e.g. 1 μm instead of 0.000001 m). The combination of prefix and unit is called a *multiple of a unit* (e.g. megameter) or a *submultiple of a unit* (e.g. millimeter). *Compound units* – units expressed as multiplication, division or power of other units – cannot be prefixed as a whole, only *singular* units such as meter and newton can be prefixed.

Quantities and units have a *dimension*, which is an abstraction of a quantity, ignoring magnitude, sign and direction aspects. The dimension of a quantity or unit can be viewed as a vector in a space spanned by an independent set of base vectors (i.e. *base dimensions*). For example, the quantity speed has a dimension that can be decomposed into base dimension *length* and base dimension *time* (with certain magnitudes as we show below). In principle we could also have expressed time in terms of base dimensions distance and speed. The base dimensions of SI are length (L), mass (M), time (T), electric current (I), thermodynamic temperature (Θ), amount of substance (N) and luminous intensity (J). For example, speed is defined as length divided by time, written as $\mathrm{L^1\ M^0\ T^{-1}\ I^0\ \Theta^0\ N^0\ J^0}$ [1].

Each quantity can have more than one *measurement scale*, which can be nominal, ordinal (e.g. Beaufort), interval or ratio. Of these, the interval and ratio scale type express amount using numerical values in combination with units of measure. Ratio scale types such as the Kelvin scale have an absolute zero point, while interval scale types such as the differential Celsius scale do not.

Different quantity kinds are typically associated with different application areas. For example, the area of space and time concerns quantity kinds such as length and speed. Some areas are more specific than others; e.g. sailing uses the nautical mile to measure speed rather than km/h. This is practical knowledge of how quantities and units are *used*, instead of knowledge concerning the mathematical nature of quantities and units themselves. Standards such as SI provide no information on such matters.

[8] For simplicity we ignore dimensionless units in this article. OM does model them.

3. Use Cases

The use cases below were identified in the context of the Tiffany project at the Dutch food research organization TI Food and Nutrition.[9] In this project a semantic research repository is being created to support collaboration between food researchers and to enable knowledge transfer to food industry. The use cases are also inspired by experiences in other domains, as for example described by Hey *et al.* [6]. The main goals of such efforts are to enable (1) replication and verification of experiments done by others; (2) integration of research data from different sources; and (3) analysis of existing research data. These goals require an explicit semantic description of the data (using an ontology). Data owners not familiar with semantic technologies should be supported in providing descriptions.

UC1: Representing and checking observation records. The ontology must allow us to represent statements about the physical world. It can be used to represent inputs and outputs of experiments to the advantage of scientific research (see e.g. Roure *et al.* [15]). It should for example be possible to state that "the viscosity of ketchup sample 1 is around 70.000 cP". This requires relating a phenomenon to a quantity kind, a numerical value and a unit. It should be possible to check if the unit used is consistent with the quantity kind. Therefore, the ontology should model the relationship between quantity kinds and units.

UC2: Manual annotation assistance. Scientists and engineers should be supported in the process of annotating their data (numerical values) with quantities and units. An example is annotating the header of a table that contains experimental results. In the domain of food science, which we work in, many researchers use Excel to store tabular data. To assist annotation we have developed an Excel add-in that allows them to quickly select the right quantity and unit. Optionally, the user first selects an application area. This allows the tool to limit the drop-down list to relevant quantities only (e.g. astronomy uses different quantities than food research). After a quantity has been selected, the tool limits the set of units in the unit drop-down list. In this way the selection process is simplified.

[9] http://www.tifn.nl

UC3: Unit conversion. In order to integrate data from different sources, and for the purpose of data analysis, it is necessary to convert between units (for example from degrees Celsius to degrees Fahrenheit). This requires a conversion factor between the units (in this case 9/5). In the case of absolute temperature values, also an offset is required (in this case +32), because different temperature scales have different zero points.

UC4: Representing and checking formulas. Research in the exact sciences often uses formulas, either in the process itself or as output when a newly discovered "law" is given a formal notation. Formulas are either expressed as quantities (e.g. Newton's $force = mass{\cdot}acceleration$) or combinations of quantities and units f[N] = m[kg]·a[m/s^2]. To prevent mistakes the formulas can be checked on their dimensional consistency and their unit consistency. For example, the dimensional exponents of force are the same as those of mass multiplied by those of acceleration. A formula can be dimensionally consistent without being unit consistent, e.g. v[km/h] = s[m] / t[s] is not unit consistent. Formulas need to be specified formally, including the units and quantities contained in them, to allow such consistency checks.

UC5: Automated annotation. Disclosing legacy data contained in e.g. spreadsheet files without costly human intervention necessitates automated annotation software. The structure of the ontology should assist in deriving annotations from text, which a flat list of quantities and units cannot. In previous work [20] we describe a system that performs automatic annotation of table headers with quantities and units. Human-made tables contain ambiguous information, e.g. the symbol F can refer to over ten quantities and units. If the cell contains the text "F (Hz)" it is clear to humans that F refers to frequency (because the unit hertz (Hz) expresses frequency and not for example force, to which capital F usually refers). Such ambiguity can only partly be resolved by improving the standards (see a discussion on this issue in the SI by Foster [4]), because humans will probably keep using older, ambiguous notations (a phenomenon inherent in standardization efforts), and self-invented abbreviations.

We discovered that the following knowledge can be used to heuristically disambiguate: (1) relationship between quantities and units (e.g. frequency is expressed in hertz); (2) application areas and their units (e.g. nautical mile belongs to sailing); (3) relationships between units and dimensions; and (4) terms used in everyday

language to indicate specific quantities. The first allows F in "F (Hz)" to be matched to frequency. The second allows "m" to be matched to metre instead of nautical mile (when no additional information is given, the more generic unit is the more likely interpretation). The third heuristic allows "g/l" to be matched to gram per liter instead of other possible unit combinations, such as gauss per liter. Quantities with that dimension are not available. In essence, dimensional analysis allows to distinguish units that are not listed in the ontology into those that are "used in practice" (e.g. gram per liter) from those that are not plausible (e.g. gauss per liter). The fourth heuristic (i.e. recognizing jargon) allows to detect the right quantity although a colloquial, non-standard term was used in the text. For example, people tend to write down "weight (gram)" where it should be "mass (gram)" (weight is a force expressed in e.g. newton). Other confusions are velocity and speed, frequency and rotational speed. This type of knowledge might also be used in a system that teaches this domain to students, a future use case which we will not consider here.

4. Design and Usage of OM

Quantity Kinds, Quantities and Units. There are three basic options to model quantity kinds and quantities. In the first option, "quantity-kinds-as-classes", subclasses of Quantity are used to model the quantity kinds, e.g. Length. This is the approach OM takes. It allows us to incorporate the hierarchical relations between quantity kinds in the class hierarchy; e.g. Diameter is a subclass of Length. Instances of Quantity represent specific occurrences of quantities, such as "the diameter of apple1". In that case, "apple1" is an instance of the class Fruit. The property om:phenomenon links a quantity to the phenomenon, for example the quantity "diameter" has phenomenon "apple1".

In the second option, "quantity-kinds-as-instances", quantity kinds are modeled as instances of class QuantityKind, e.g., "length" and "mass" are instances of QuantityKind. The hierarchy between quantity kinds should then be modeled with a property that relates instances of QuantityKind to each other. In the third option, "quantity-kinds-as-properties", quantity kinds are modeled as properties that connect phenomena to quantities, e.g. hasLength. The hierarchy is modeled with the subproperty mechanism, e.g. hasDiameter is a subproperty of hasLength. These three alternatives represent possible ways to model the same information

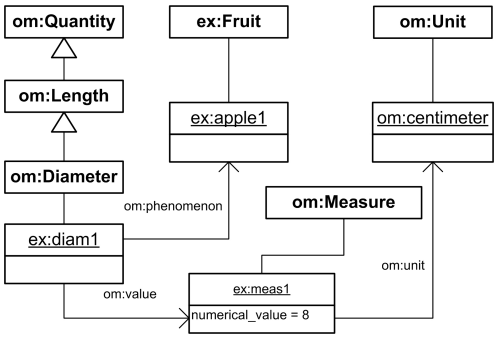

Fig. 2. UML diagram of a measurement of diameter of apple1 in OM.

from slightly different perspectives. They are compatible in that rules may be formulated to automatically translate one in the other. Which perspective should be preferred then depends on practical concerns, e.g. which perspective allows useful reasoning (in the chosen representation language) not easy to realize in another perspective. We return to this issue in the Discussion.

Requirements from use cases. To represent observation records (UC1), the unit and numerical value have to be recorded. OM groups the numerical value and unit of a quantity in an instance of class Measure. Quantity instances are linked to a measure through property om:value (see also Figure 2).

UC1 and UC2 both require a link between quantities and units. However, the set of units is different. In UC1 (checking annotations) the set of units is all units *allowed* in principle. The set of allowed units is potentially large: each unit can also be expressed as (sub)multiple units that combines a binary or SI-specified prefix with the unit (e.g. kilometer, millivolt, etc.). Even more possible combinations occur for compound units (megameter per minute, centimeter per megasecond, etc.).

An intensional description of the allowed units for a quantity can be given using OWL restrictions. It is relatively easy to specify all allowed (sub)multiple units; see the example for electric potential in Figure 3. For compound units the restriction can get quite large and complicated. Instead of specifying them all by hand we investigated a generative approach. The ontology currently contains the intensional description of the (sub)multiples for all quantities.

In UC2 (manual annotation support) the user first selects a quantity and should then be given a list of units to select from. This list should be much smaller

```
om:Electric_potential
  rdfs:subClassOf om:Quantity ;
  rdfs:subClassOf [
    a owl:Restriction ;
    owl:onProperty om:value ;
    owl:allValuesFrom
     [ a owl:Restriction ;
      owl:onProperty om:unit_of_measure_or_scale ;
      owl:allValuesFrom om:Electric_potential_unit ]].

om:Electric_potential_unit a owl:Class ;
  rdfs:subClassOf om:Unit_of_measure ;
  owl:equivalentClass
   [ rdf:type owl:Class ;
    owl:unionOf (om:Volt_multiple_or_submultiple ;
    owl:oneOf( om:volt  om:abvolt
          om:statvolt om:watt_per_ampere ))] .

  om:Volt_multiple_or_submultiple a owl:Class ;
    rdfs:subClassOf om:Unit_multiple_or_submultiple ;
    owl:equivalentClass
    [ a  owl:Class ;
      owl:intersectionOf (
    [ a  owl:Restriction ;
      owl:hasValue om:volt ;
      owl:onProperty om:singular_unit]
    [ a  owl:Restriction ;
      owl:cardinality "1"^^xsd:nonNegativeInteger ;
      owl:onProperty om:singular_unit]
    [ a owl:Restriction ;
      owl:allValuesFrom om:SI_prefix ;
      owl:onProperty om:prefix]
    [ a owl:Restriction ;
      owl:cardinality "1"^^xsd:nonNegativeInteger ;
      owl:onProperty om:prefix ])] .
```

Fig. 3. Example definition of the quantity kind "electric potential" and units that are allowed to appear in any Measures of electric potential (Measures have units and are connected to quantity kinds through property om:value). We list (1) the singular unit (e.g. volt); and (2) all (sub)multiples of that singular unit (Volt_multiple_or_submultiple). All other allowed units (e.g. watt per ampere and its (sub)multiples) are added in the same way.

than the set of allowed units, as many of the theoretically possible units are irrelevant in most cases (e.g. yoctoliter is not used to measure volumes, and people rarely use multiples of time such as megasecond or megaminute). This set of "commonly used units" cannot be specified intentionally, so we list them explicitly (see Figure 4 for an example). The units are linked directly to the class using the property unit_of_measure. This information cannot be expressed as a restriction on the property unit_of_measure_or_scale (as with allowed units), because it is not *forbidden* to use other units than the commonly used ones. Neither should it be modeled as a restriction on the property unit_of_measure itself for the same reason: it is not forbidden that instances of measurements (quantities with values) can specify commonly used units individually.

The set of commonly used units is actually the same as the union of all units of a quantity specified in all

```
om:Electric_potential
  om:unit_of_measure om:millivolt ;
  om:unit_of_measure om:volt ;
  om:unit_of_measure om:kilovolt .
```

Fig. 4. Definition of commonly used units for the quantity electric potential. Actual definition contains more units.

application areas taken together, but this equivalence cannot be expressed in OWL. Through specification of the commonly used units, UC2 can be covered in our annotation tool. Our selection of application areas and commonly used units is not yet completed and the choices are preliminary, to be considered as input for debate on this matter.

Checking datasets. Although we have specified the allowed units of quantities in OM, this alone does not allow for checking of datasets in terms of correct use of units, as OWL DL uses the open world assumption. For example, if a value of electric potential would be expressed using the unit "inch", OWL DL would conclude that inch is a member of the class Electric_potential_unit, instead of declaring the ontology to be inconsistent. We can solve this by adding disjointness axioms between e.g. Electric_potential_unit and Length_unit. This allows reasoners such as Pellet to identify the erroneous specifications.

Compound units. Compound units are defined by classes Unit_Multiplication, Unit_Division and Unit_Exponentiation. Instances of these classes are linked to their constituents with the properties term_1 and term_2 (multiplication; range Unit), numerator and denominator (division; range Unit), and base and exponent (exponentiation; base has range unit, exponent has range integer). Note that all divisions can also be expressed as multiplications (e.g. $\mathrm{m{\cdot}s^{-1}}$ instead of m/s). We have still included division in OM as it is often used to represent these units, accepted in all standards. An advantage of divisions is that the exponents are always positive. The ontology thus contains concepts that are compositionally different, but mathematically equal (i.e. they are not owl:sameAs). This has to be taken into account in applications. For example, when searching for data annotated with a division, the search process should also formulate the query as the equivalent multiplication in order to obtain all relevant results.

Unit conversion. UC3 requires that conversion relationships between units are modeled. This relationship consists of a source and target unit, and a conversion factor (expressing how many of the target unit is the

same as one of the source unit). Notice that the target unit and the conversion factor actually express the amount of a quantity – a "measurement". For this reason OM reuses the class Measure to express the relationship between units. For example, the unit foot is linked to an instance of Measure that groups the unit metre and the numerical value "3.048e-1" (in the scientific e notation), denoting that one foot is equal to 0.3048 meter. When the standards *define* a unit in terms of another unit, the property definition is used to link the units. For example, newton is linked to the compound unit m·kg/s^2. In cases where the conversion factor is 1, we link directly to a unit rather than a measure (i.e. we omit the factor). The link allows conversion of newton to base units, after which further conversion is possible to other units. In OM, conversion factors are given for all singular derived units.

In case of conversion between interval scale types, and interval and ratio scale types also an offset is required, as the zero points of the scale types differ (this requirement is almost exclusive to temperatures; most scales have uniquely-defined zero points).[10] OM represents this by adding to the link between Measurement_scales a factor and an offset value. Note that conversions are only possible if a unit or scale is related directly or indirectly to the target unit or scale. OM provides definitions of units/scales that allow most conversions to take place directly or indirectly.

Labels. UCs 1 and 2 require that names and symbols of quantities and units are provided, so that users can find the appropriate concept to annotate with. UC5 (automated annotation) requires that also unofficial and alternative names/symbols of concepts are provided. In OM, quantities and units have a preferred label and a preferred symbol, derived from the standards. Other labels needed for UC5 such as plural forms of units (e.g. "meters") and contractions of compound unit symbols (e.g. "Pas" instead of "Pa s" for pascal second) are not given but can be generated. Exceptions are for example the hectare (not "hectoare") and kilohm (although "kiloohm" is also allowed), and US/British spelling differences (meter/metre). Symbols for compound units can be generated from their constituent unit symbols (e.g. "s2" and "m/s"). Some quantities such as mass are sometimes referred to as "weight" in everyday language. During automated annotation (UC5), incorrect mentions of weight have to result in annotation with mass. To reach this goal, we add unofficial_labels to mass and other cases in OM (unofficial_label is a subproperty of skos:hiddenLabel). We have also added a number of frequently used abbreviations for quantities and units, including "sec", "temp" and "ul" (instead of "µl" for microliter), stored in unofficial_abbreviation (another skos:hiddenLabel).

Application areas. UC2 and UC5 require that application areas and their quantities and units are modeled. In OM, the class Application_area has instances such as sailing and astronomy. The quantities and units belonging to a specific area are linked to these instances. Two areas may have the same quantities, but may use different units. For example, the parsec is a unit of distance in astronomy, while it is not used in sailing.

An application area is linked to its units and to its quantities by two separate properties. Application areas that form a selection (subset) of quantities and units in another application area are linked to each other with property uses_application_area. For example, sailing is linked to space and time.

Dimensions and systems of units. UC4 (checking formulas) and UC5 (automated annotation) require that the dimensions of quantities and units are modeled. In OM, the class Dimension has instances such as density-dimension. The dimensional exponents are given in properties such as SI_mass_dimension_exponent and SI_length_dimension_exponent (for density-dimension the values of these exponents will be 1 and -3, respectively). Dimensions are linked to quantity classes by the property dimension. Instances of class System_of_units are used to group together the base and derived units of a system such as SI. OM defines a few other systems of units and their base and derived quantities and units. However, OM does not contain a representation of the dimensional properties of other systems. This is not necessary as the SI can support the necessary computations and analyses defined in the UCs for all other systems of units.

Software. All of the use cases mentioned are implemented as freely-accessible SOAP services,[11] an annotation plugin for Excel (released in the near future) and an automated annotation system (freely available). As far as we know we are the first to supply elementary services based on an ontology for units of measure-

[10] Look for example at length scales (such as the meter scale), mass scales (such as the kilogram scale), density scales (such as the kilogram per cubic meter scale), etc.: 0 m, 0 kg, 0 kg/m^3 are all clear zero points of these scales.

[11] `http://www.wurvoc.org/services/oum.jsp`

ment, as opposed to embedding the functionality in a monolithic software infrastructure intended to support a specific program.

Linking. We have written a SILK [7] specification that links OM to DBpedia.[12] Using a strict comparison to ensure high precision (but lower recall), we generated 88 quantity (skos:exactMatch) links and 130 unit links. Note that recall in practice is higher than these figures suggest, because DBpedia does for example not include all (sub)multiple units that OM has.

5. Comparing OM with QUDT

In this section we compare OM with QUDT v1.0.0, focusing on main modeling choices and their consequences for the use cases. QUDT is an OWL ontology under development by NASA and TopQuadrant in the context of the NExIOM project.

Quantity Kinds, Quantities and Units. QUDT does not use the "quantity-kinds-as-classes" approach that OM uses, but "quantity-kinds-as-instances". Quantity kinds are modeled as instances of QuantityKind. Instances of QuantityValue group together a numerical value and a unit (similar to om:Measure). Instances of Quantity link to a QuantityValue and a QuantityKind, but not to a a phenomenon to represent a complete data record. The hierarchy between quantity kinds such as velocity and linearVelocity is indicated with a special-purpose property qudt:generalization (see example in Figure 5). QUDT does not provide an intensional description of "allowed" units, neither does it specify commonly used units of quantities.

Units in QUDT are, like in OM, instances of Unit. Units are linked to their quantity by the property quantityKind. The units allowed for one quantity are grouped together in classes such as LinearVelocityUnit (similar to OM). The property qudt:exactMatch is used to indicate that units are equivalent, e.g. knot and nautical mile per hour. QUDT does not contain disjointness axioms between its unit classes, making it impossible to check observation records using OWL DL as presented for OM.

QUDT does not represent (sub)multiple units and compound units in terms of their constituents. For example, femtometer is not explicitly related to the prefix femto and the unit metre. It is not clear why some (sub)multiples have been included and others not. For example, millihenry is included but millimetre is not.

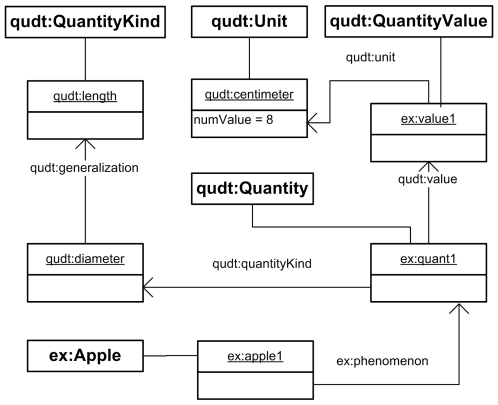

Fig. 5. UML diagram of a measurement of diameter of an apple in QUDT. Note that "diameter" is not currently defined in QUDT.

Labels of units in QUDT are included with rdfs:label, symbol and abbreviation. The label for e.g. millisecond is "Millisecond", symbol "ms" and abbreviation "ms" (abbreviations of units appear to be the same as the symbol; abbreviations such as "msec" would be more useful).

QUDT specifies 239 quantity kinds and 801 units; OM specifies 463 quantity kinds (subclasses of om:Quantity) and 1,033 units (206 singular units, 599 (sub)multiples and 228 compounds). In general, OM specifies more quantity kinds per application area. QUDT covers additional areas: biology, communication and currency; OM covers acoustics and astronomy.[13] QUDT contains some quantities that are not derived from standards, such as EnergyPerElectricCharge and ForcePerArea. They might have been added to group quantities (e.g. quantForcePerArea groups Pressure and Stress) based on their dimensions. Such "organizing quantities" may be more confusing than helpful as no one is familiar with them.

QUDT specifies many physical constants (e.g. the Planck constant or the speed of light in vacuum) whereas OM defines only a few. Note that many of the 641 QUDT constant instances concern the same constant (e.g. Planck) expressed in different units.

[12] http://www.afsg.nl/InformationManagement/images/escience/om_dbpedia_units.nt and http://www.afsg.nl/InformationManagement/images/escience/om_dbpedia_quantities.nt

[13] Surprisingly, NASA's QUDT does not contain parsec, light year and other astronomical units.

Application areas. Application areas are modeled in QUDT as instances of the class QuantityKindCategory, such as SpaceAndTimeQuantityKind and MechanicsQuantityKind. These instances are at the same time also classes; they are subclass of qudt:QuantityKind. These classes group together all quantities that belong to an area. Other classes are used to group the units that belong to a quantity, e.g. qudt:SpaceAndTimeUnit (subclass of Unit). There are two differences with OM's approach. Firstly, in QUDT quantities and units of one area are not grouped together. The units that belong to an area are not directly accessible from the application area instance, but only through the link they have with their quantities. Secondly, OM's approach allows to make more fine-grained groupings. For example, in OM we can express that microbiology typically uses milliliter (rather than e.g. megaliter).

Use cases. We are not aware of software written for QUDT that enables verifying the use cases mentioned. QUDT doesn't support all use cases. The lack of (sub)multiple units and different types of label hinders automated annotation (UC3), manual annotation (UC2) and representation of data records (UC1). Checking records (UC1) will require additional knowledge on which units are allowed for quantities. The ontological definitions that facilitate unit conversion (UC3) are often unclear. For example, for the qudt:abvolt an offset of 0 and a multiplier of 1.0e-8 are given, but the target unit (presumably qudt:volt) is not given. A more problematic example is the qudt:newton, for which correct offsets and factors are provided (namely 0 and 1), but again the target unit is not specified. This case is more problematic because the likely target unit, metre kilogram per second squared, is not specified in QUDT.

6. Discussion and Future Work

Terminology. The terms that are used in OM are based on a number of technical standards [19,2,23,10]. These terms are used differently in standard works about observation and measurement theory [17,18]. E.g. in measurement theory, scales and units as appearing in the technical standards do not seem to be distinguished. However, we concentrate on use in operational science and engineering and therefore use a different reference framework, with different terminology (see also Introduction).

OWL DL compatibility. Both OM and QUDT are not valid OWL 1 DL. In OM this is caused by instances of om:Application_area that link to the quantity classes, and quantity classes that link to instances using om:unit_of_measure. In QUDT this problem has been avoided since it uses the "quantity-kinds-as-instances" approach (i.e. quantities are not classes but instances). However, QUDT has chosen to model application areas as a meta-class qudt:QuantityKindCategory, the instances of which are classes themselves (e.g. qudt:MechanicsQuantityKind. In OWL 1 DL entities are not allowed to have multiple roles (entities are either class, instance or property). OWL 2 DL does support multiple roles through "punning", which only disallows any reasoning that involves the entity in both its roles (to the reasoner they are simply two different entities). This is fine because the application-area part of the model is only needed to lookup which quantities and units belong to it (see UC2), i.e. no reasoning over them is needed. Moreover, the major use case for an OWL-compatible ontology (UC1) can be supported with generally available OWL 2 DL reasoners. In short, it does not appear to be necessary to support OWL 1 DL.

Generating ontologies. OM's and QUDT's unit classes such as om:Electric_potential_unit and qudt:EnergyUnit are predictable in structure. Such classes could be generated automatically (from each singular unit, and from each quantity, respectively). The labels of units are also highly regular (e.g. a compound unit's name can be constructed from the labels of its constituents; a unit's plural form is often created by suffixing -s). Therefore, instead of tedious and error-prone manual curation, it would be beneficial to automatically generate these elements. However, we do not know of an OWL-based ontology editor or manager that would allow us to specify this type of meta-knowledge and generate the ontology from it. Such issues might also play a role in other ontologies, so we suggest that this is a lacking component in the ontology management life cycle.

Future work and open issues. We aim to specify more precisely the application areas, and extend the number of quantities and units with some more specific ones such as half-life and resonance energy (which we found missing through our automatic annotation experiments [20]). The mapping to DBpedia will be improved.

For usage of OM in applications related to quantitative research (representing tables, formulas, inputs and

outputs of computations, hypotheses) we are developing an Ontology of Quantitative Research (OQR).

A number of open issues remain. Firstly, an open problem for both OM and QUDT is how to deal with quantities that are a combination of an existing quantity and a mathematical operator such as total pressure and average speed, other than specifying them as atomic quantities. Secondly, how to represent measures with just a number rather than a number with a unit. An example is the countable quantity (e.g. number of apples). This issue is not straightforward. Thirdly, the proper definition of application areas and commonly used units belonging to quantities. Fourthly, unresolved ambiguities in SI are a source of problems (see Foster [4]). For example, duplicate symbols for units and prefixes (e.g., "d" and "h" for respectively "day" and "deci", and "hour" and "hecto"), which makes compound units such as "hW" (hour watt or hectowatt?) ambiguous. A fifth issue is which perspective on quantity kinds ("quantity-kinds-as-classes" vs. "quantity-kinds-as-instances" vs. "quantity-kinds-as-properties") is most appropriate. When comparing OM and QUDT we found no particular reason to favor one over the other. For example, OM provides subclass reasoning between quantity kinds and under OWL semantics QUDT's transitive generalization property provides similar functionality. One advantage of "quantity-kinds-as-instances" is that it allows an OWL 1 DL compatible ontology (although both QUDT and OM are currently not OWL 1 DL), but it is not clear whether this is really needed in practice.

Integration. An attempt to integrate the two ontologies (i.e. merge them into one new ontology) could simply select one of the perspectives and drop the other. Another option is to allow both models to coexist but harmonize them such that one is automatically translatable into the other. This will allow users to choose based on the use case, without sacrificing interoperability. Services written for one could also handle data from the other. Difficulties in merging will be in the missing unit information in the definitions of derived units in QUDT and deviating names of quantities in QUDT and OM.

A complete ontology of this domain should in any case contain information currently exclusive to both ontologies. OM provides additional label types, compositional units, clear representation of unit conversion characteristics and specification of allowed units which enables automatic consistency checks through OWL; QUDT provides physical constants. Perhaps the OASIS QUOMOS working group is a useful forum for integration, as it aims to integrate several (OWL and non-OWL) standards such as QUDT and UCUM.[14] This would entail merging and selecting among the (partially overlapping) quantities and units defined in the separate approaches.

An open question is whether ontologies such as OM and QUDT, although defined for practical purposes, can be aligned with foundational ontologies such as DOLCE. The class Quality in DOLCE can not be defined as a superclass of OM class om:Quantity just like that; there are clear differences between these two classes. Firstly, DOLCE qualities have specific properties, such as a temporal index. OM can be used to express dynamic and static data; in itself it does not make a choice. Additional concepts are needed to express assertions and functions, for expressing for example time dependence. Secondly, qualities in DOLCE have scales that represent their possible values [12]. In OM, most quantities are related to units. In DOLCE, qualities can be grouped through spaces, which can be related to units of measure. So, relating quantities in OM and qualities in DOLCE is not straightforward and must be investigated. What can be related to DOLCE is a phenomenon such as the class Fruit in Figure 2 by making it a subclass of Endurant or Perduant in this ontology.

Modeling domain practice. Modeling issues do arise when we leave the standardized part of the domain, and look at how quantities and units are used in practice. Firstly, which application areas to include and which units belong to an area is hard to ascertain, as are the commonly used units belonging to a quantity. Secondly, when modeling allowed units we find units that are theoretically possible (e.g. microstatvolt, megasecond, millifoot) but that are not used in practice. An empirical study is needed to decide on which units are more or less common. In many cases the use of these may even not be recommended. A reason to include these as allowed units is that when they do appear it is at least possible to interpret them. Thirdly, in everyday language and textual notes and tables, people use non-standard terms to refer to quantities and units. It is not enough to model the standards in an ontology, or even to reduce ambiguity within standards as proposed by Foster [4]. Unofficial terms, symbols and abbreviations should be linked to official ones in order to enable the use cases.

[14] http://www.unitsofmeasure.org/

We conclude that the practical usability of an ontology (in terms of use cases) should be a central design principle, which has not yet received enough attention in ontologies of this domain. In OM we provide a starting point for further discussion and development in the future.

Acknowledgements

This publication was supported by the Dutch national program COMMIT. The authors thank Jeen Broekstra, Mari Wigham and Rinke Hoekstra for useful discussions, and Jeen and Mari for contributions to the services and ontology.

References

[1] Bridgman, P. (1922). *Dimensional Analysis*. Yale University Press.

[2] Cohen, E. and Giacomo, P. (1987). Symbols, units, nomenclature and fundamental constants in physics. *Physica*, 146A:1–68.

[3] Davenport, J. and Naylor, W. (2003). Units and dimensions in OpenMath. Technical report, OpenMath. Available at `http://www.openmath.org/documents/Units.pdf`.

[4] Foster, M. P. (2010). The next 50 years of the SI: a review of the opportunities for the e-Science age. *Metrologica*, 47(6).

[5] Gruber, T. and Olsen, G. (1994). An ontology for engineering mathematics. In Doyle, J., Torasso, P., and Sandewall, E., editors, *Fourth International Conference on Principles of Knowledge Representation and Reasoning*. Morgan Kaufmann, Bonn, Germany.

[6] Hey, T., Tansley, S., and Tolle, K., editors (2009). *The Fourth Paradigm - Data-Intensive Scientific Discovery*. Microsoft Research. Available at `http://research.microsoft.com/en-us/collaboration/fourthparadigm/contents.aspx`.

[7] Isele, R., Jentzsch, A., and Bizer, C. (2010). Silk Server – Adding missing Links while consuming Linked Data. In Hartig, O., Harth, A., and Sequeda, J., editors, *Proceedings of the First International Workshop on Consuming Linked Data (COLD2010)*, volume 665. CEUR, Shanghai, China.

[8] Leal, D. and Schröder, A. (2002). RDF vocabulary for physical properties, quantities and units. Technical report, ScadaOnWeb. Available at `http://www.s-ten.eu/scadaonweb/NOTE-units/2002-08-05/NOTE-units.html`.

[9] Masolo, C., Borgo, S., Gangemi, A., Guarino, N., and Oltramari, A. (2003). WonderWeb Deliverable D18. Technical report, Laboratory For Applied Ontology, Trento, Italy. Available at `http://wonderweb.semanticweb.org/deliverables/documents/D18.pdf`.

[10] NIST (2004). The NIST Reference on Constants, Units, and Uncertainty, International System of Units (SI), Prefixes for binary multiples. Technical report, National Institute of Standards and Technology (NIST). Available at `http://physics.nist.gov/cuu/Units/binary.html`.

[11] Pinto, H. and Martins, J. (2001). Revising and extending the Units of Measure "subontology". In *Proceedings of the Workshop IEEE Standard Upper Ontology, at IJCAI'2001*. AAAI Press, Seattle, Washington.

[12] Probst, F. (2008). Observations, Measurements and Semantic Reference Spaces. *Journal of Applied Ontology*, 3(1-2).

[13] Raskin, R. and Pen, M. (2005). Knowledge representation in the Semantic Web for Earth and Environmental Terminology (SWEET). *Computers & Geosciences*, 31:1119–1125.

[14] Rijgersberg, H., Wigham, M., and Top, J. L. (2011). How semantics can improve engineering processes: A case of units of measure and quantities. *Advanced Engineering Informatics*, 25(2):276–287.

[15] Roure, D. D., Goble, C., and Stevens, R. (2009). The design and realisation of the Virtual Research Environment for social sharing of workflows. *Future Generation Computer Systems*, 25(5):561 – 567.

[16] Schadow, G., McDonald, C., Suico, J., Fohring, U., and Tolxdorff, T. (1999). Units of measure in clinical information systems. *Journal of the American Medical Informatics Association*, 6(2):151–162.

[17] Suppes, P., Krantz, D., Luce, R., and Tversky, A. (1989). *Foundations of Measurement*. Academic Press, San Diego.

[18] Suppes, P. and Zinnes, J. (1962). Basic measurement theory. Technical report, Stanford University, Stanford, California. Available at `http://suppescorpus.stanford.edu/techreports/IMSSS_45.pdf`.

[19] Taylor, B. N. (2008). Guide for the use of the International System of Units (SI). 2008 edn. Technical report, National Institute of Standards and Technology. Available at `http://physics.nist.gov/cuu/pdf/sp811.pdf`.

[20] van Assem, M., Rijgersberg, H., Wigham, M., and Top, J. (2010). Converting and Annotating Quantitative Data Tables. In Patel-Schneider, P. F., Pan, Y., Hitzler, P., Mika, P., Zhang, L., Pan, J., Horrocks, I., and Glimm, B., editors, *Proceedings of the 9th International Semantic Web Conference*, number 6496 in LNCS. Springer-Verlag.

[21] W3C OWL Working Group (27 October 2009). *OWL 2 Web Ontology Language: Document Overview*. W3C Recommendation. Available at `http://www.w3.org/TR/owl2-overview/`.

[22] W3C RDF Working Group (10 February 2004). *RDF Primer*. W3C Recommendation. Available at `http://www.w3.org/TR/rdf-primer/`.

[23] Weast, R., editor (1976). *Handbook of Chemistry and Physics*. CRC Press, Cleveland, Ohio.