Semantics of Governmental Statistics Data

Denny Vrandečić
KIT
Karlsruhe, Germany
denny.vrandecic@kit.edu

Christoph Lange
Jacobs University
Bremen, Germany
ch.lange@jacobs-university.de

Michael Hausenblas
DERI, NUI Galway
Galway, Ireland
michael.hausenblas@deri.org

Jie Bao
Tetherless World Constellation, Rensselaer Polytechnic Institute
Troy, NY, USA
baojie@cs.rpi.edu

Li Ding
Tetherless World Constellation, Rensselaer Polytechnic Institute
Troy, NY, USA
dingl@cs.rpi.edu

ABSTRACT

A number of governments have started to publish their statistical data online. Some of them are adhering to Linked Data principles, others are using other standards which often can be transformed in order to be published as Linked Data. RDF vocabularies, such as SCOVO (Statistical Core Vocabulary) are being used for publishing the data. Many of the current translations of the statistical data are published using simple transformation approaches. The semantics of the statistical data points, such as the real-world concepts they describe, or how they have been derived, is typically not explicit in these transformations. We propose a path towards a collaborative approach for *a posteriori* explicit sense-making.

Keywords

Semantic Web, Linked Open Data, Government, Statistics

1. INTRODUCTION

A quickly growing amount of statistical government data is being published online as Linked Open Data [14]. The UK government has published their statistical data natively as RDF[1] [19], whereas the institute RPI is in the process of translating the US government's public statistical data to RDF[2] [8]. Currently most effort is going into publishing statistical data in the RDF syntax without careful modeling of the meaning of the data, even though RDF could also provide for that. This has been criticized.[3] In this paper we will discuss how this can be remedied *after* the initial publication of the data, using the fact that Semantic Web standards enable *a posteriori* schema definitions.

RDF is a formal model that enables the integration of diverse data sources [11] and that allows using custom vocabularies in order to describe the data. SCOVO (Statistical Core Vocabulary) is such a lightweight vocabulary, a community effort to represent statistical data on the Web, adopted by the UK government initiative data. SCOVO closely follows the usual tabular structure of statistical data [10]. Widely deployed exchange formats for statistical data like SDMX (Statistical Data and Metadata Exchange)[4] are not using linked open data standards yet, but currently work is ongoing to remedy this and connect SDMX and SCOVO [7]. RDF allows the reuse of other existing data from the Semantic Web, be it on the instance level, or of existing vocabularies on the schema level. SCOVO does not yet represent the semantics of what the statistical data actually describes.

In this paper we present the SCOVOLink ontology that enables us to state the link between the data and the described entities explicit by grounding the statistical data in existing vocabularies and appropriate mathematical functions. SCOVOLink is designed in such a way to be enabled by Semantic MediaWiki (SMW). SMW [12] is an extension to MediaWiki that allows the collaborative editing of metadata. Both the UK government and the translation of the US government data to RDF are supported by an SMW system and thus would allow the methodology described in this paper to be used.

The next section describes an example of data published at `data.gov.uk` which will offer a running example for the rest of the paper. Section 3 describes the SCOVOLink ontology and how it can be used to ground SCOVO data to external vocabularies. Section 4 describes how to ground derived values with OpenMath. Section 5 offers an example of how data integration can be accomplished once the necessary annotations are available, and Section 6 describes how the current wiki infrastructure can be used in order to provide these annotations and thus realize collaborative sense-making. We end with the conclusions, including open issues and suggestions for data publishing.

2. EXAMPLE

Table 2 shows an extract from data published by the UK government.[5] The data was extracted using the SPARQL

[1] `http://data.gov.uk`

[2] `http://data-gov.tw.rpi.edu`

[3] `http://www.mkbergman.com/846/when-linked-data-rules-fail/`

[4] `http://sdmx.org`

[5] For legibility we present simple triples in N3 [3] and more

Web Science Conf. 2010, April 26-27, 2010, Raleigh, NC, USA.
.

Area \ Year	2007	2008
Norfolk	14,888	9,038
Isle of Wight	606	693

Table 1: Number of geese in the given counties.

```
ahs:EG74 scv:dimension env:norfolk ;
  scv:dimension env:year-2007 ;
  scv:dimension env:geese ;
  rdf:value "14888"^^xsd:decimal ;
  scv:dataset ahs2:livestock .
ahs:EH74 scv:dimension env:norfolk ;
  scv:dimension env:year-2008 ;
  scv:dimension env:geese ;
  rdf:value "9038"^^xsd:decimal ;
  scv:dataset ahs2:livestock .
ahs:EG100 scv:dimension env:isle-of-wight ;
  scv:dimension env:year-2007 ;
  scv:dimension env:geese ;
  rdf:value "606"^^xsd:decimal ;
  scv:dataset ahs2:livestock .
ahs:EH100 scv:dimension env:isle-of-wight ;
  scv:dimension env:year-2008 ;
  scv:dimension env:geese ;
  rdf:value "693"^^xsd:decimal ;
  scv:dataset ahs2:livestock .
```

Table 2: RDF describing Table 1.

endpoint of the data.gov.uk website. Table 1 displays the data in a tabular form, as human readers would usually read it.

The RDF describes the table and its content. We see four items (representing the four cells), each with three dimensions (representing the two dimensions of the table and that the table is about geese; the RDF data also includes further animals) and the connection of the items with a dataset (i. e. the table).[6]

As human readers we understand that two of the dimensions point to a region and to a time respectively, and that the item provides the number of geese. A machine can render the data in a range of charts if it understands SCOVO, offering powerful methods to slice and dice along the dimensions of the dataset. But currently there is no way to describe what the datasets and their items actually mean, thus hindering automatic integration of different datasets (e. g. with the US dataset, as will be discussed in Section 5).

The above example uses only scv:dimension to define the dimensions. This has the drawback of possibly requiring an exploding number of instances for the dimension values. For example, imagine a statistic about migration, stating the number of residents moving from one area to another. In this case we have two dimensions that are areas, i. e. we may choose to state that both are of the type env:local-authority in our example. But in this case it would be not possible anymore to discern the direction of the migration flow. SCOVO requires the publisher to introduce a new dimension type for the values, like env:to-local-authority, and completely duplicate all instances within that dimension.

Instead we suggest to introduce subproperties of the property scv:dimension (an approach also suggested by [7]).

For the example in Table 1 we would introduce the following three dimension properties, one for each dimension of the table:

- ahs:region for the region where the count was performed (the y-axis in the table)
- ahs:year for the year when the count was performed (x-axis in the table)
- ahs:animal for the animal that was counted (not displayed in the table)

We can automatically translate the above example into using the new, more specific properties by applying the following SPARQL query and adding the result to the original dataset (we only give one example for brevity, the other properties are analogous). Note that this semantics cannot be expressed by an OWL axiom.

```
CONSTRUCT { ?item ahs:region ?area }
WHERE {
  ?item scv:dimension ?area .
  ?area rdf:type env:local-authority .
}
```

In the end, the following triples will be added (only given for one of the items, the others are analogous).

```
ahs:EG74 ahs:region env:norfolk ;
  ahs:year env:year-2007 ;
  ahs:animal env:geese .
```

We will use this enriched ontology for further annotation.

3. THE SCOVOLINK ONTOLOGY

We propose to extend SCOVO with SCOVOLink (sl), a vocabulary enabling the connection of the SCOVO-described datasets with external vocabularies. This improves discoverability, reusability, and semantic integrability. This allows us, for example, to perform more complex data analysis automatically. The following is an example of the description of the above dataset.

```
ahs2:livestock rdf:type sl:CountDataset .
ahs2:livestock sl:counts dbpedia:Animal .
ahs2:livestock sl:numberOfDimensions "3"^^xsd:int .
ahs2:livestock sl:georegionDimension ahs:region .
ahs2:livestock sl:timepointDimension ahs:time .
ahs2:livestock sl:entitytypeDimension ahs:animal .
```

SCOVOLink additionally offers a small vocabulary for mathematical or statistical functions used to obtain original data points (e. g., defining the items in the above dataset as the count of certain livestock in a geographic area, as seen above) or to compute derived data points (e. g., the change in geese density over the years, see Section 4). SCOVOLink extends the SCOVO representation of statistical data by grounding the description of the datasets in existing, established vocabularies.

The meaning of the property sl:numberOfDimensions can be formalized with the following axiom:

```
EquivalentClasses(
  DataHasValue(sl:numberOfDimensions "3"^^xsd:int)
  ObjectExactCardinality(scv:dimension "3"^^xsd:int)
)
```

This axiom allows to ensure completeness when constructing the class description in the next step (see also the SPARQL

complex OWL axioms in OWL 2 Functional Syntax [17].

[6] All namespaces are given at the end of the paper in Table 8.

query in Table 7). As can be seen in the triples of the above example, we rely on the OWL 2 feature called *punning* that allows us to reuse names of entities of one ontological type (class, property, individual) also for other types, e. g. using the property name `ahs:time` as an instance name.

Whereas SCOVOLink merely contains the terms, offering a light-weight ontology to annotate SCOVO datasets, we also define the formal semantics of the terms here (but not in OWL, since it is not possible to express the complete semantics in OWL). Here we give a rough sketch of the formal semantics. We do not expect these axioms to be reasoned with, but rather to be used as a reference either to answer discussions on the meaning of the ontologies, or for ontology discovery, matching and alignment tasks. We will now take a closer look at the description of `sl:CountDataset`, a subclass of `scv:Dataset`.

First we introduce the property `sl:classCardinality`. Analogously to `owl:cardinality` on properties we define `sl:classCardinality` for classes, stating how many different individuals instantiate the given class. An example:

```
ex:EUCountry sl:classCardinality "27"^^xsd:int .
```

This states that the class `ex:EUCountry` has exactly 27 instances that are different from each other. If we know of less, we know that the knowledge is not complete; if we know of more, we know that some of them have to be referring to the same individual; otherwise we have an inconsistency.[7]

Now we need to construct the actual class description that is being counted. This is a conjunction of atoms being constituted by the counted class (connected to the `scv:Dataset` by the `sl:counts` property) and a description for each `scv:dimension`, determined by the actual `scv:dimension` and its value for each single `scv:Item`. Note that the counted class may be a complex class description itself. This description can be generated automatically out of the SCOVOLink metadata. In our example above, the class description for the upper left value would be:

```
ObjectUnionOf(
  dbpedia:Animal
  ObjectHasValue(ex:livesIn ahs:norfolk)
  ObjectHasValue(ex:inYear ahs:year-2007)
  ObjectHasValue(ex:species ahs:geese)
)
```

Now using the `sl:classCardinality` property we can state that this class description has the class cardinality given by the statistical data (i. e., in this case 14,888). This way the formal semantics of the given statistic are exactly defined and can be grounded in external vocabulary terms.

4. DERIVED VALUES

Having shown how to ground original data points in existing vocabularies using SCOVOLink, we will now study how to ground derived data points. Derived data points are computed from original data points or from other derived data points using mathematical functions. Making this information explicit gives access to computational services, such as deriving new values from existing data points, or verifying existing derivations, as will be shown below.

4.1 Units of measure

The original data points studied so far are counts. For measurements, such as the area of a country or the age of a person (or a goose), we additionally need to represent the unit of measurement. Applications of units often involve computation: a dataset about livestock might contain areas in square meters, but a Canadian goose studying it might be more familiar with acres. An abundance of non-standard unit vocabularies exist.[8] Some of them specialize on making conversions computable, whereas others aim at covering as many units as possible, including customary ones. While these two goals do not exclude each other conceptually, there is currently no one-size-fits-all solution. Therefore, we only demonstrate two extreme cases and suggest how to unify them, and leave the remaining work to an OASIS Technical Committee for developing a standard ontology for "quantities, systems of measurement units, and base dimensions for use across multiple industries" that has recently been formed.[9]

The QUDT ontology (Quantities, Units, Dimensions and Data Types)[10] is oriented towards information retrieval rather than computation. It links hundreds of units to their corresponding DBpedia resources but on the other hand only comes with few conversion rules. Computation-oriented unit ontologies usually cover fewer units and do not link them to other datasets. They usually decompose derived, compound, and prefixed units into basic units; for example, they would define a square kilometer as the square of 1000 meters. By that same mechanism they facilitate the definition of new units as required. Well-designed examples of that family of ontologies comprise the SysML QUDV ontology (Quantities, Units, Dimensions and Values),[11] the SWEET ontologies (Semantic Web for Earth and Environmental Terminology),[12] and the OpenMath unit content dictionaries [20] – listed in decreasing order of the number of terms and, at the same time, ascending order of suitability for computation. We will take a close look on the OpenMath language in the following sections.

4.2 Ontologies for Mathematical Functions

As an example, consider geese density, which is defined as the ratio of geese and area, or the mean number of geese per county. The SWEET ontologies used in Table 3 contain several mathematical concepts, including arithmetic operators and statistical functions. The ontology allows for modeling the derivations of the two example data points mentioned before. Comparing the second to the first example shows the limits of SWEET: The input sequence for a statistical operation cannot be decomposed into *elements*, i. e. original data points. Moreover, SWEET only defines that the mean is a statistical operation, but it cannot define how it is computed. It is safe to assume wide availability of libraries that know how to compute a mean, but two problems remain:

1. Neither SWEET nor any other RDF-based ontology is

[7] Class cardinality can be expressed in OWL indirectly by using the `ObjectOneOf` construct, i. e.
C `classCardinality` n
would be semantically equivalent to
EquivalentClasses(C **ObjectOneOf**($i_1 \dots i_n$)).

[8] See `http://ontolog.cim3.net/cgi-bin/wiki.pl?UoM` for a summary

[9] `http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=quomos`

[10] `http://www.qudt.org/`

[11] `http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-qudv:qudv_owl`

[12] `http://sweet.jpl.nasa.gov/`

```
# read: There is a division operation,
# whose operands are ...
[] a oper:Division ;
  # the original data points:
  # Goose population of Norfolk (see above)
  oper:hasFirstOperand ahs:EH74 ;
  # Area of Norfolk
  oper:hasSecondOperand area:norfolk ;
  # the derived data point
  oper:hasOutput ex:geese-density-norfolk .

[] a stat:Mean ;
  # the original data points: a sequence that
  # cannot be defined more closely in SWEET
  # here: the absolute goose population
  # count per county
  oper:hasInput ahs:geese-per-county ;
  # the derived data point
  oper:hasOutput ahs:geese-average .
```

Table 3: Computational Semantics of Data Points using SWEET

   widely recognized as a standard vocabulary of mathematical operations. Therefore, we would have to hardcode the mapping of `stat:Mean` to the mean function from a mathematical library in our application.

2. There are many less common derived values around. Consider the Human Development Index (HDI) of a country.[13] Assuming that four auxiliary data points have already been computed (LE = life expectancy index, ALI = adult literacy index, GEI = gross enrollment index, and GDP = an index computed from the gross domestic product per capita at purchasing power parity, all normalized to a scale between 0 and 1), the HDI is defined as $\frac{1}{3}(LE+\frac{2}{3}ALI+\frac{1}{3}GEI+GDP)$. With SWEET, one would not be able to express more than that the HDI is some mathematical operation.

4.3 Defining Functions and their Computation in OpenMath

This is where OpenMath comes into play [18].[14] OpenMath is an XML-based language for exchanging mathematical expressions across applications – originally and still primarily between computer algebra systems and automated theorem provers, but also educational software and scientific publishing. In the mathematical domain, it is widely recognized as a standard. OpenMath comes with its own lightweight ontology language and a number of standard ontology modules (Content Dictionaries) containing descriptions of common mathematical symbols: operators, functions, constants, sets, etc. Additional content dictionaries have been developed by the community. The `s_data1` content dictionary is part of the OpenMath standard; it defines the `mean` function as shown in Table 4.

```
<CD xmlns="http://www.openmath.org/OpenMathCD">
  <CDBase>http://www.openmath.org/cd</CDBase>
  <CDName>s_data1</CDName>
  <CDStatus>official</CDStatus>
  ...
  <Description>This CD holds the definitions of the
    basic statistical functions used on sample data.
    ...</Description>
  <CDDefinition xml:id="mean">
    <Name>mean</Name>
    <Role>application</Role>
    <Description>This symbol represents an n-ary
      function denoting the mean of its arguments.
      That is, their sum divided by their number.
    </Description>
    <FMP>
      <!-- This is OpenMath for
           $\mathrm{mean}(\{l_1,\ldots,l_n\}) = \frac{l_1+\cdots+l_n}{n}$,
           where $L = \{l_1,\ldots,l_n\}$ is a list -->
      <OMOBJ xmlns="http://www.openmath.org/OpenMath"
       version="2.0"
       cdbase="http://www.openmath.org/cd">
        <OMA>
        <!-- A = application of an
                 operator to arguments -->
          <!-- the equality relation, here being
               definitional equality -->
          <OMS cd="relation1" name="eq"/>
          <!-- S = symbol -->
          <OMA>
            <OMS cd="fns2" name="apply_to_list"/>
            <OMS cd="s_data1" name="mean"/>
            <OMV name="L"/>
            <!-- V = variable -->
          </OMA>
          <OMA>
            <OMS cd="arith1" name="divide"/>
            <OMA>
              <OMS cd="fns2" name="apply_to_list"/>
              <OMS cd="arith1" name="plus"/>
              <OMV name="L"/>
            </OMA>
            <OMA>
              <OMS cd="set1" name="size"/>
              <OMV name="L"/>
            </OMA>
          </OMA>
        </OMA>
      </OMOBJ>
    </FMP>
    ...
  </CDDefinition>
  ...
</CD>
```

Table 4: Definition of the mean function.

The URI of such an OpenMath symbol is constructed by concatenating the base URI, the name of the content dictionary and the name of the symbol, resulting in `http://www.openmath.org/cd/s_data1#mean` in our example [5]. Thus, given the information that `ahs:geese-average` has been computed from `ahs:geese-per-county` using the `sdata:mean` function, a mathematical application has two choices:

1. If there is a mapping from `sdata:mean` to a built-in

[13] http://en.wikipedia.org/wiki/Human_Development_Index

[14] Whenever we say "OpenMath" in this paper, we could equally say "Content MathML 3". Content MathML is the semantic sublanguage of the W3C MathML language [1]. In the upcoming version 3, the semantics of Content MathML will be aligned with OpenMath. Both languages will continue to exist, as two syntaxes for the same formal semantics. OpenMath, however, has a stronger history in computation, and so far there are few computational applications that support Content MathML.

mean function, it can directly perform the computation. For many common computer algebra systems such mappings – called *phrasebooks* – between their native language and OpenMath have been provided.

2. If there is either no phrasebook mapping for `sdata:mean`, or if the application does not have a built-in *mean* function, but if the application understands the OpenMath content dictionary language, it can reduce the *mean* function to more elementary functions.

It can generally be assumed that, if there are OpenMath phrasebooks for a particular mathematical application, they support a large subset the official content dictionaries that are part of the OpenMath standard.

So far, there is no content dictionary supplying the above-mentioned definition of the HDI. Therefore, we have written our own one, which defines the `ex:hdi` function as well as functions to compute the auxiliary data points. For enabling mathematical applications to communicate with each other, OpenMath specifies a protocol for negotiating the set of content dictionaries and symbols that they support.

A web service infrastructure for OpenMath-aware computation services exists. A simple HTTP interface that accepts OpenMath expressions and evaluates them in Mathematica has been developed by the MathDox team [16, 6]. There have also been various attempts at designing a unified infrastructure for symbolic computation that completely abstracts from the underlying systems and their native languages; SCIEnce (Symbolic Computation Infrastructure for Europe [9]) is the most recent one. SCSCP, the Symbolic Computation Software Composability Protocol, enables mathematical applications to exchange expressions, request calculations to be made, and store and retrieve remote objects. SCSCP libraries have so far been provided for a number of computer algebra systems. Additionally, there are Java libraries, as well as SOAP and REST interfaces giving access to SCSCP-aware applications.

4.4 Connecting OpenMath and RDF

We have said that OpenMath symbols have URIs, but how exactly will we reproduce the listing in Table 3 using OpenMath? From Table 4 it is clear that OpenMath is not immediately compatible to RDF. A straightforward representation of mathematical expressions in RDF has been proposed earlier for the related Content MathML language [15], but this idea has not been taken up in practice. We see mainly two reasons for that:

1. Mathematical expressions are *ordered* n-ary trees. The order of elements in a set and arguments to an operator often matters, and the number of elements that a set can have or arguments that an operator takes is often not limited to a fixed number (which would allow for addressing them by named relations). n-ary constructs and order require auxiliary structures in the triple-oriented RDF, which do not go well along with RDF-based reasoning and querying.

2. RDF-based querying and reasoning is usually confined to decidable subsets of first order logic, such as description logic or Horn rules. Symbolic computation in computer algebra systems and automated theorem provers usually relies on more complex foundations. Therefore, mathematical applications usually do not support RDF-based knowledge representations.

Therefore, we refrain from expressing the full mathematical semantics of statistical data in RDF. We leave part of it in OpenMath and instead show how to connect both worlds by grounding RDF vocabulary terms in OpenMath via the mutual use of URIs.

SCOVOLink has a small set of properties for modeling mathematical operations as references to OpenMath symbols and for passing arguments into them. The design is inspired by SWEET. We assume the mathematical operations to be defined in OpenMath, and computations to be performed by OpenMath-aware mathematical services. That results in the following SCOVOLink/OpenMath reimplementation of the listing in Table 3, plus the HDI example mentioned above:

```
ex:geese-density-norfolk sl:computedFrom [
  sl:function arith:divide ;
  sl:arguments
    [ sl:argPosition "1"^^xsd:int ;
      sl:argName "dividend" ;
      sl:argValue ahs:EH74 ] ,
    [ sl:argPosition "2"^^xsd:int ;
      sl:argName "divisor" ;
      sl:argValue area:norfolk ]
] .

ahs:geese-mean sl:computedFrom [
  sl:function sdata:mean ;
  sl:arguments
    [ sl:argValue ahs:EH01 ] ,
    [ sl:argValue ahs:EH02 ] ,
    [ sl:argValue ahs:EH03 ] ,
    ...
] .

ex:HDI-Germany sl:computedFrom [
  sl:function ex:hdi ;
  sl:arguments
    [ sl:argName "LE" ;
      sl:argValue ex:LE-Germany ] ,
    [ sl:argName "ALI" ;
      sl:argValue ex:ALI-Germany ] ,
    [ sl:argName "GEI" ;
      sl:argValue ex:GEI-Germany ] ,
    [ sl:argName "GDP" ;
      sl:argValue ex:GDP-Germany ]
] .
```

Now we have to specify how a SCOVOLink-aware application can use this information for verifying or recomputing data points. A function call has to be translated from RDF to an OpenMath expression, before it can be sent to a mathematical service for evaluation. OpenMath usually represents operations as the application (`OMA`) of an operator or function symbol (`OMS`) to some number of arguments. In the SCOVOLink setting, the arguments are numbers. For each data point that is referenced as the `sl:argValue` of an argument of the function call, we obtain its `rdf:value`. We map `xsd:int` values to OpenMath integers (`OMI`) and `xsd:decimal` and any other representation of real numbers to OpenMath floating-point numbers (`OMF`).[15]

[15] OpenMath only has one datatype for double precision floating-point numbers built into the language. Additional datatypes can be introduced by defining new constructor symbols in content dictionaries. This has been done, e. g., for complex numbers, but it should also be done for XML

```
curl -o - -H "Content-type: text/plain" ↩
 -d "<OMOBJ xmlns='http://www.openmath.org/OpenMath' version='2.0' cdbase='http://www.openmath.org/cd'>↩
  <OMA>↩
    <OMS cd='arith1' name='divide'/>↩
    <OMI>14888</OMI> <!-- the goose population of Norfolk -->↩
    <OMI>5371</OMI> <!-- the area of Norfolk in km$^2$ -->↩
  </OMA></OMOBJ>" http://mathdox.org/phrasebook/mathematica/eval_openmath_native
```

Table 5: Sending an OpenMath Expression to Mathematica for Evaluation, via an HTTP Frontend

Listing 5 shows the OpenMath translation of the first operation and how it is evaluated using Mathematica and the MathDox HTTP interface[16]. The `arith:divide` operator takes two arguments, first the dividend, second the divisor. We indicate this order by numeric position properties. OpenMath expressions always assume a fixed order of arguments. In this example, the names of the arguments merely improve readability but have no meaning for the translation to OpenMath. In the second example, the mathematical operation works on a set; therefore, the order of arguments does not matter. The OpenMath output has the same structure as in the first example and is omitted for brevity. The third example exclusively relies on named arguments. This is possible when the OpenMath definition has a fixed number of arguments with distinct names. This is the case with our `hdi` symbol. The left hand side of its definition looks as follows:

```
<OMA>
  <OMS cd="relation1" name="eq"/>
  <OMA> <!-- left hand side -->
    <OMS cdbase="http://www.example.org"
     cd="hdi" name="hdi"/>
    <OMV name="LE"/>
    <OMV name="ALI"/>
    <OMV name="GEI"/>
    <OMV name="GDP"/>
  </OMA>
  <!-- right hand side:
        $\frac{1}{3}(LE + \frac{2}{3}ALI + \frac{1}{3}GEI + GDP)$ -->
</OMA>
```

Note the difference to listing 4: The `sdata:mean` function takes an arbitrary number of arguments; therefore, it is defined implicitly, using the auxiliary `fns2#apply_to_list` operator, which applies a function to a list of arguments. Thus, to get the order of the arguments for `hdi` right when mapping from RDF to OpenMath, we have to look into the content dictionary. Assuming that the content dictionary has been published as linked data[17], we can do so by dereferencing the URI `ex:hdi` and requesting content in the `application/openmath+xml` MIME type. This gives us the OpenMath source of the definition of the `hdi` symbol, from which four simple XML queries determine the order of the four named arguments, so that we can finally construct the OpenMath expression.

```
data961:entry1 rdf:type twc:DataEntry ;
  rdf:value "3231"^^xsd:int ;
  data961:data_item "GEESE␣-␣INVENTORY" ;
  data961:state "ALABAMA" ;
  data961:state_fips "1" ;
  data961:year "2007" .
```

Table 6: Number of geese in Alabama, according to dataset 961 of `data.gov`.

5. DATA INTEGRATION

In Section 2 we have shown an example of publishing data about the number of geese in British counties. A dataset about the same topic for the United States also exists.[18] It has been partially translated to RDF. Table 6 gives an extract of the data.[19]

The `data.gov` translation to RDF does not use SCOVO, but is structurally very similar. It also uses an entity per data cell (in this case of class `twc:DataEntry`), and then introduces a number of properties to describe the dimensions. The current translation to RDF uses literals for the dimension values, which can be translated to entities with appropriate axioms like the following:

```
EquivalentClasses(
  DataHasValue(data961:state_fips "1")
  ObjectHasValue(twc:state twc:Alabama)
)
EquivalentClasses(
  DataHasValue(data961:data_item
              "GEESE - INVENTORY")
  ObjectHasValue(twc:type dbpedia:Goose)
)
EquivalentClasses(
  DataHasValue(data961:year "1997")
  ObjectHasValue(twc:year env:year-1997)
)
```

Furthermore we can map the ontology to the SCOVO ontology:

```
scv:Item owl:equivalentClass twc:DataEntry .
twc:year rdfs:subPropertyOf scv:dimension .
twc:state rdfs:subPropertyOf scv:dimension .
twc:type rdfs:subPropertyOf scv:dimension .
```

Note that the mapping can be done externally. Furthermore, we can use the same approach as in Section 2 to

Schema's numeric datatypes.

[16] Here, we demonstrate the access from the command line for easy reproduction by the reader.

[17] The standard content dictionaries of OpenMath have not yet been published as linked data but will soon be – not just as OpenMath, but also as RDF. This RDF excludes the detailed structures of mathematical expressions but makes the metadata of content dictionaries and symbols available to RDF-based clients. We have already generated RDF descriptions of all OpenMath content dictionaries, but so far they are only internally used in a semantic wiki for maintaining the content dictionaries [13].

[18] http://www.data.gov/tools/961/

[19] The complete dataset can be found at http://data-gov.tw.rpi.edu/raw/961/data-961-00001.rdf

```
SELECT ?area ?goosenumber
WHERE {
  ?item scv:dataset ?dataset .
  ?dataset sl:counts dbpedia:Animals .
  ?dataset sl:timepointDimension ?timeprop .
  ?dataset sl:georegionDimension ?geoprop .
  ?dataset sl:entitytypeDimension ?typeprop .
  ?dataset sl:numberOfDimensions "3"^^xsd:int .
  ?item ?geoprop ?area .
  ?item ?timeprop env:year-2007 .
  ?item ?typeprop dbpedia:Goose .
  ?item rdf:value ?goosenumber .
}
```

Table 7: SPARQL query over the merged dataset displaying the number of geese in 2007

annotate the dataset and thus ground the meaning of the properties in external ontologies:

```
data961:dataset sl:numberOfDimensions "3"^^xsd:int .
data961:dataset sl:counts dbpedia:Animal .
data961:dataset sl:georegionDimension twc:state .
data961:dataset sl:timepointDimension twc:year .
data961:dataset sl:entitytypeDimension twc:type .
```

Assuming that both datasets – the UK dataset presented in Section 2 and the US dataset presented here – are annotated as described, we can now merge the datasets and issue a single SPARQL query to access the mashed-up data from both datasets in order to display the number of geese in 2007 in a uniform result set (see Table 7). There are two points of notice about the SPARQL query:

- we first need to select the properties representing the correct dimensions. Since both datasets use their own properties to represent dimensions (which will often be the case in automatically translated datasets), we can use the annotations in the dataset to find the matching properties

- we need to state the number of dimensions of the dataset, since otherwise we could get numbers meant to be more fine-grained items. For example, dataset 1425 of the US datasets has data about the number of geese per state *per ethnicity and gender of the farm owner.* If we did not explicitly state the number of dimensions, we would get all these numbers back as well, but without the additional dimensions being specified (since we did not anticipate them).

The next section will offer a possibility on how to gather the required annotations in order to enable the automatic integration of different statistical data sources published in RDF as described here.

6. COLLABORATIVE SENSE-MAKING

Both RPI's translation to RDF and the UK's publishing of government data are accompanied by a Semantic MediaWiki (SMW) [12] to foster collaboration amongst the user community. The wikis can be used to describe the published datasets, and are meant for the community to offer mutual help and user-contributed documentation. But since they are semantic wikis, they can actually be used not only to describe the datasets informally, but also to provide the required formal, machine readable definitions of the used terms in order to enable the automatic integration described in the previous section.

As we have seen in the count dataset example above, the metadata describing the dataset using the SCOVOLink ontology is comprised of simple property and class assertions with the dataset being in the subject position. This structure was chosen as it is highly suitable to be represented in an SMW instance. We can use the special property `equivalent URI` in SMW to map the internal entities to external URIs, and thus state their mapping to SCOVOLink and other external ontologies. This allows to add exactly the kind of metadata needed above.

The SMW contains a page for every dataset. For example, `http://data-gov.tw.rpi.edu/wiki/Dataset_961` is the wiki page for the dataset about the livestock census in the United States. So we can define the mapping here, and an external service can now reuse the mapping and provide the integration. This allows to collaboratively figure out the semantics of the datasets *after* publishing them on the Web. We expect this to become an interesting case study in collaborative sense-making, since the community is asked to join in annotating the meaning of the datasets.

7. CONCLUSIONS

Currently a lot of energy is being put in publishing government data on the Semantic Web. These published datasets are huge, but often these translations are simple and merely syntactic.

A major claim of the Semantic Web standards is that it is not required to define the schemas and models beforehand, but that it can also be provided *a posteriori*, after the act of publishing. In this paper we examined this claim in greater detail, showing a case study with a concrete dataset. We propose SCOVOLink, an ontology that allows to ground datasets in either external ontologies or mathematical functions. We also demonstrated its use and applicability.

We identify and suggest the following steps:

- Instead of `scv:dimension` more specific subproperties of it should be used. This will very practically help with aligning and reusing the datasets.

- An ontology for Units of Measurements needs to become prevalent. This is a basic and yet unresolved issue.

- OpenMath needs to connect tighter to the RDF world, and provide clean interfaces. We have outlined some possibilities for their interaction in Section 4, but regard this as merely preliminary thoughts.

- Semantic MediaWiki's current use of external vocabularies is not designed to be used collaboratively. This needs to improve in order to enable the use cases in Section 6.

- The SCOVOLink ontology needs to be further developed.

- It is unclear if collaborative *a posteriori* sense-making really is feasible, and what kind of community management it requires. We expect this to be an interesting case study for Web Science.

We regard all these steps as feasible and helpful for integrating government data into the growing Semantic Web.

Prefix	Namespace
sl	`http://vocab.deri.ie/scovolink#`
scv	`http://purl.org/NET/scovo#`
ahs	`http://environment.data.gov.uk/statistics/agriculture-horticulture-survey/june-2008#livestock-`
ahs2	`http://environment.data.gov.uk/statistics/agriculture-horticulture-survey/june-2008#`
env	`http://environment.data.gov.uk/statistics/dimension#`
twc	`http://data-gov.tw.rpi.edu/2009/data-gov-twc.rdf#`
data961	`http://data-gov.tw.rpi.edu/raw/961/data-961-00001.rdf#`
oper	`http://sweet.jpl.nasa.gov/2.0/mathOperation.owl`
stat	`http://sweet.jpl.nasa.gov/2.0/mathStatistics.owl`
sdata	`http://www.openmath.org/cd/s_data1#`
arith	`http://www.openmath.org/cd/arith1#`
dbpedia	`http://dbpedia.org/resource/`
ex	`http://www.example.org/` (not existent, for example use)

Table 8: RDF prefixes used in this paper. `rdf`, `rdfs`, `owl`, and `xsd` are defined as in the W3C standard documents and not repeated here for brevity.

Acknowledgments

The work presented in this paper has been partially funded by the EU FP7 project ACTIVE[20] and a grant from DARPA's Transformational Convergence Technology Office. We would like to thank Dan Roozemond and Peter Horn from the MathDox and SCIEnce projects for their help with OpenMath-based computation services.

8. REFERENCES

[1] R. Ausbrooks, S. B. D. Carlisle, G. Chavchanidze, S. Dalmas, S. Devitt, A. Diaz, S. Dooley, R. Hunter, P. Ion, M. Kohlhase, A. Lazrek, P. Libbrecht, B. Miller, R. Miner, M. Sargent, B. Smith, N. Soiffer, R. Sutor, and S. Watt. Mathematical Markup Language (MathML) version 3.0. W3C Candidate Recommendation, World Wide Web Consortium (W3C), 12 2009.

[2] S. Autexier, J. Campbell, J. Rubio, V. Sorge, M. Suzuki, and F. Wiedijk, editors. *Intelligent Computer Mathematics, 9th International Conference, AISC 2008 15th Symposium, Calculemus 2008 7th International Conference, MKM 2008 Birmingham, UK, Proceedings*, number 5144 in LNAI. Springer Verlag, 2008.

[3] T. Berners-Lee. Notation 3 - a readable language for data on the web, 2006. available at `http://www.w3.org/DesignIssues/Notation3`.

[4] C. Bizer, T. Heath, T. Berners-Lee, and M. Hausenblas, editors. *Linked Data on the Web (LDOW 2010)*, CEUR Workshop Proceedings, Apr. 2010.

[5] S. Buswell, O. Caprotti, D. P. Carlisle, M. C. Dewar, M. Gaetano, and M. Kohlhase. The Open Math standard, version 2.0. Technical report, The Open Math Society, 2004.

[6] H. Cuypers, A. M. Cohen, J. W. Knopper, R. Verrijzer, and M. Spanbroek. Mathdox, a system for interactive mathematics. In *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2008*, pages 5177–5182, Vienna, Austria, June 2008. AACE.

[7] R. Cyganiak, S. Field, A. Gregory, W. Halb, and J. Tennison. Semantic statistics: Bringing together SDMX and SCOVO. In Bizer et al. [4].

[8] L. Ding, D. DiFranzo, A. Graves, J. R. Michaelis, X. Li, D. L. McGuinness, and J. Hendler. Data-gov wiki: Towards linking government data. In *AAAI Spring Symposium on Linked Data Meets AI*, 2010.

[9] S. Freundt, P. Horn, A. Konovalov, S. Linton, and D. Roozemond. Symbolic computation software composability. In Autexier et al. [2], pages 285–295.

[10] M. Hausenblas, W. Halb, Y. Raimond, L. Feigenbaum, and D. Ayers. Scovo: Using statistics on the web of data. In *European Semantic Web Conference 2009*, 2009.

[11] G. Klyne and J. Carroll. Resource Description Framework (RDF): Concepts and abstract syntax. W3C Recommendation 10 February 2004, 2004.

[12] M. Krötzsch, D. Vrandečić, M. Völkel, H. Haller, and R. Studer. Semantic Wikipedia. *Journal of Web Semantics*, 5:251–261, September 2007.

[13] C. Lange. `wiki.openmath.org` – how it works, how you can participate. In J. H. Davenport, editor, *22nd OpenMath Workshop*, July 2009.

[14] Linked data – connect distributed data across the web. `http://linkeddata.org`.

[15] M. Marchiori. The mathematical semantic web. In A. Asperti, B. Buchberger, and J. H. Davenport, editors, *Mathematical Knowledge Management, MKM'03*, number 2594 in LNCS. Springer Verlag, 2003. Keynote.

[16] MathDox – OpenMath translation servlet. `http://mathdox.org/phrasebook/`.

[17] B. Motik, P. F. Patel-Schneider, and B. Parsia. OWL 2 web ontology language: Structural specification and functional-style syntax. W3C recommendation, World Wide Web Consortium (W3C), 10 2009.

[18] OpenMath. `http://www.openmath.org`.

[19] J. Sheridan and J. Tennison. Linking UK government data. In Bizer et al. [4].

[20] J. Stratford and J. H. Davenport. Unit knowledge management. In Autexier et al. [2], pages 382–397.

[20] `http://www.active-project.eu`