Semantically Enriched Spreadsheet Tables in Science and Engineering

Jan Top[1,2], Mari Wigham[1] and Hajo Rijgersberg[1]
[1] Wageningen UR, Food and Biobased Research
Wageningen, The Netherlands
{firstname.secondname@wur.nl}

[2] VU University Amsterdam
Amsterdam, The Netherlands

***Abstract*—Tabular data are common in science and engineering. Datasets found in practice are often not very well specified, and are therefore hard to understand and use. Semantic standards are available to express the meaning and context of the data. However, present standards have their limitations in expressing heterogeneous datasets with several types of measurements. Such datasets are abundant in science and engineering. We propose the RDF Record Table vocabulary for semantically modelling tabular data. It complements the existing RDF Data Cube standard. RDF Record Table has a nested structure of records that contain self-describing observations. A first implementation of the model shows that it facilitates finding and integrating data from multiple spreadsheets. This support helps scientists to get the most out of available quantitative data with a minimum of effort.**

Keywords-semantics; table; spreadsheet; e-science.

I. INTRODUCTION

In science and engineering, datasets can be very complex, in particular, if they combine different experiments and observations. We propose a format that has observations and records, rather than traditional tables, as its basic building blocks.

Tabular data are common in science and engineering. Tools to handle such data, such as spreadsheets, are extremely popular because of their flexibility and ease of use. However, this flexibility often leads to data being ambiguous or even incomprehensible, and their provenance being unknown [1][2]. The possibility to immediately proceed to the analysis and visualization of the data, often has a negative effect on the quality of the actual registration in terms of complete and systematic recording. This makes finding, understanding and reusing the data very difficult [3]. As the amount of available data is exploding, it is essential to be able to efficiently locate and reuse existing datasets.

The traditional way to present tabular data is in tables on paper or on a screen. Rows and columns of cells make up their structure. In such a table, an individual recording shows up as a single value in one of the table cells. The associated header cell along the same column or row explains the meaning of this value, for example ‘m (kg)’ for mass measured in kilograms. In datasets found in practice, this header information is often ambiguous and incomplete. In fact, much of the information about the actual observation is frequently left out. This may even be done on purpose, in order to clean the data for presentation or processing. Tables also become more compact, if all records contain the same quantities, the same unit of measure and have the same interpretation. In this way, the ‘bare’ numerical or string value in the table cells is separated from the metadata, and directly visible for comparison and available for numerical computation. Researchers are trained in reading such tables and can interpret them immediately.

However, to further exploit datasets in science and engineering, we are not bound to the traditional two-dimensional table format. We can use richer representations to express more contextual information. Many methods have been developed over the last decades to express tabular datasets in a more flexible and rich manner. The W3C RDF (Resource Description Framework) standard provides a more general, graph-based language to do so [4]. RDF Data Cube is a prominent example of such an RDF-based standard [5].

Representing datasets semantically has major advantages. Firstly, the meaning of the measurements is independent of for example the precise text in a spreadsheet, so that data can be found and understood regardless of typos, abbreviations, local terminology and even different languages. Secondly, the use of semantic concepts makes tables machine readable, meaning that they can be (semi-) automatically processed, from simple unit conversion up to complex computations. Finally, allowable numerical values and units can be defined, making it possible to check or clean the data. Moreover, semantic tables can be used as templates for future observations and experiments.

Which requirements should a semantic standard meet to facilitate and stimulate structured annotation of tabular data? First, it should be able to annotate the individual data elements. For example, it should be possible to state that ‘the mass of this sample is 2.95 grams’, ‘the city considered is Amsterdam’, or ‘this event has occurred 5 minutes and 6.3 seconds later’. Good scientific recordings contain extensive information about each observation, for example on which object it has been measured, by which method and by whom. The annotation (metadata) of the individual data elements explains them and describes their provenance and relations. A standard has to build on existing (domain)

 ISBN: 978-1-61208-355-1

ontologies in order to facilitate shared understanding of the individual observations.

Secondly, a semantic standard for tabular data should make explicit the grouping together of scientific observations that collectively form a 'snapshot' of the world. The observations are combined since they are generated in one experiment, using the same experimental protocol or by a single apparatus. A collection of snapshots, or *records*, is used to detect patterns, similarities or correlations. This grouping is essential for correct interpretation of the data. Within one experiment, the structure of the records is often quite similar. However, when comprehensive recording of all possibly relevant effects is required, datasets can be less homogeneous and well-formed. This, in particular, holds for datasets that combine observations from different origins. Moreover, exact science typically deals with quantities having diverse scales, units and other specifications; values may be missing or occasionally additional measurements are available. Consider for example research that combines input from a number of labs around the world. Some of them have recorded the environmental temperature in degrees Fahrenheit and others in degrees Celsius. One lab has not measured temperature at all. Semantic standards should allow these variations and at the same time provide enough structure.

In this paper, we intend to find a format that is sufficiently rich and flexible to handle complex datasets in science and engineering. In Section II, we first briefly describe existing approaches, in particular the RDF Data Cube vocabulary. This is a recommended W3C standard for multidimensional tables. To be able to handle more heterogeneous datasets, we propose RDF Record Table in Section III, as a supplement to RDF Data Cube. RDF Record Table uses self-contained observations and recursive records. In Section IV, we describe which steps can be taken to cope with the verbosity that is a consequence of the very explicit character of RDF Record Table datasets. This is followed by a description of a first implementation in Microsoft Excel in Section V. Finally, we conclude in Section VI, also listing a number of open issues.

II. RELATED WORK

Many methods take the relational database approach when they convert tables or databases into an RDF-based representation [6]-[8]. They assume that a table consists of a header row defining variables and other rows that contain strings or numbers representing the value of the variable in the same column. In general, they do not support more complex structures. All columns are translated into RDF properties of a single object. At this point, no other metadata is available than what is given in the header and data cells.

A richer format is defined by the RDF Data Cube vocabulary [5], a recommended W3C standard. This vocabulary has been developed in the context of statistical data in social sciences and policy studies, but is also being applied in other areas. Information about the meaning of the data and its provenance is expressed by linking to concepts from other ontologies, most typically the SDMX vocabulary [9]. Data Cube organizes observations as multidimensional datasets. Each observation is a point in n-dimensional space, defined by the associated values of the *dimensions*. Typical dimensions in RDF Data Cube are 'time', 'area' and 'gender'. Each observation contains one or more *measures*, for example 'life expectancy = 83.5 years'. Observations can have *attributes* that provide additional information about them, for example the unit of measure used. A separate section of an RDF Data Cube defines its *structure*; this section can be used as a template for future observations. Another section gives information for external reference to the entire dataset.

In its normalized form, each observation in a data cube contains all its dimensional values. One way to reduce redundancy is by moving shared attributes to the structure definition section. Further reduction can be obtained by introducing 'slices'. A slice is a lower-dimensional representation, which also serves as a proposed interpretation of the dataset. Moreover, one can refer to a slice as an independent entity. Table I shows the example table that RDF Data Cube definition uses to explain the vocabulary [5]. This reference shows the full model of Table I.

TABLE I. LIFE-EXPECTANCY DATA IN DIFFERENT REGIONS OVER TIME

	2004-2006		2005-2007		2006-2008	
	Male	Female	Male	Female	Male	Female
Newport	76.7	80.7	77.1	80.9	77.0	81.5
Cardiff	78.7	83.3	78.6	83.7	78.7	83.4
Monmouthshire	76.6	81.3	76.5	81.5	76.6	81.7
Merthyr Tydfil	75.5	79.1	75.5	79.4	74.9	79.6

The RDF Data Cube vocabulary is very well suited for modelling well-formed, complete datasets such as are produced by statistics offices. Software tools are available to provide useful views of the data. However, these advantages are the result of some restrictions on the data. We submit that these restrictions make the RDF Data Cube less suitable for heterogeneous, multi-scale data such as exist in science and engineering. The requirement to choose *a-priori* between dimensions and measures is problematic in those fields. Rather than assuming some causal order between quantities, we can only state that they have been observed together. For example, for Table I, RDF Data Cube assumes 'sex' (male or female) to be a dimension and 'life expectancy' (values in the table) to be a measure. This assumption is not needed and limits data analysis; it is sufficient to say that 'sex' and 'life expectancy' have been measured simultaneously.

One striking consequence of the hypercube approach is that multiple measures in a single observation are difficult to handle. This is, however, a common experimental setting in science and engineering. For example, imagine that in the above example in addition to 'life expectancy', also the quantities 'weight', 'waste size' and 'length' have been observed. RDF Data Cube has two alternative ways to handle such a dataset, which cannot be used simultaneously. In the *multiple measures* approach one observation can contain more than one measured quantity. However, all quantities must have the same attributes, for example, the same type and unit of measure. This rules out this approach for most exact science applications. The second approach restricts observations to having a single measured value. It allows a dataset to carry multiple measures by adding an extra dimension, a measure dimension. This turns a measured value into a kind of semi-dimension. We submit that this construction complicates the model unnecessarily and may influence the interpretation of the data.

Another characteristic of RDF DataCube is that it makes extensive use of properties (rather than classes) as its main organizing mechanism. The design introduces many different types of properties. It is questionable whether these different properties are needed to express the meaning of the data. They make the design of a model rather complex.

RDF Data Cube is intended for describing ‘well-formed’ datasets. As a result, several constraints are placed on the data, for example that each observation must have a value for every measure. For example, if for one measurement in the example it is not known whether this person is a man or a woman, this data point cannot be included in the model. Another assumption is that the multidimensional structure is a regular (hyper)cube, not permitting rows with varying length for a single dimension. If we know the standard deviation of the life expectancy value for Cardiff and a few other regions, we cannot add this to the above in Table I. Another complication would arise if some life expectancy values were expressed in years-with-decimal (as in the table), and others in years-and-months.

Whereas RDF Data Cube and other standards define the structure and context of tabular data, they are not intended for expressing provenance of data on the web. For that purpose, additional vocabularies have been developed. The W3C-standard PROV is becoming increasingly popular for this purpose [10]. It describes the origins of any type of data, helping the user to evaluate how appropriate and trustworthy the data is for a particular use. PROV basically says that a `prov:Agent` performs a `prov:Activity`, in which he uses or generates a `prov:Entity`. Tables, records, slices and individual measurements can all be seen as subclasses of `prov:Entity`. The previously defined Dublin Core Terms [13] vocabulary complements the PROV model with detailed concepts about publications and authorship.

III. RDF RECORD TABLE

Experience with researchers over the past ten years has confronted us with many different datasets. Many of them are contained in spreadsheets and data analysis tools such as Matlab [11] and SPSS [12]. Our work on introducing electronic lab notebooks in the multidisciplinary domain of food science has revealed many issues in data recording in the lab. Annotation of the data is often scarce and ambiguous due to the focus of researchers on the research itself rather than its bookkeeping. In addition, large amounts of data are produced by automated measurement equipment in the lab. These devices tend to produce more systematic metadata, but linking data from different sources is as yet difficult and labor intensive. Initially, we proposed templates to stimulate systematic annotation of research data, but experience has shown that this restricts the creative and essentially unstructured character of scientific research. Moreover, researchers are typically reluctant to spend a lot of time on data bookkeeping. Inspired by other initiatives to annotate datasets using RDF, we have devised an approach that can work in the tools commonly used by researchers and at the same time support rich annotation. This approach has developed into a model for tabular data called RDF Record Table.

The RDF Record Table vocabulary is intended for recording original and processed data in science and engineering. It models datasets in terms of observations and records (see Fig. 1, using `rec:` as a prefix for the RDF Record Table namespace). An *observation* is a statement about an entity or the property of an entity, such as ‘the temperature of this object measured by a pt-sensor is 36.5C’ or ‘this milk sample is from batch 20140612YTU’. A *record* combines observations to form a snapshot, thus conveying the assumption that in some way the observations are related - in time, location, subject, conditions, or in another way.

To express composite structures, in RDF Record Table any record can recursively contain sub-records, which again are of the type RDF Record Table. For example, an experiment may observe multiple samples at one fixed temperature. For each sample its viscosity, composition and mass are measured over time. This means that the entire dataset consists of a RecordTable that at its highest level contains (i) the observed temperature and (ii) a sub-record for each sample. Each sub-record in turn contains the sample identifier and sub-records that describe viscosity, composition and mass for that sample measured at a point in time. In the most explicit form, all sub-records are expanded into non-nested records. In this example, the top level RecordTable only contains sub-records, each of them stating the observed temperature, time point, sample id and the other measured properties.

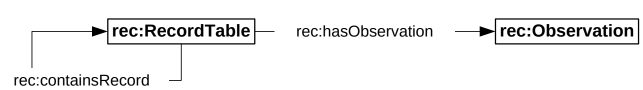

Figure 1. Basic RDF Record Table schema

In Turtle format RDF Record Table is defined as follows.

```
rec:RecordTable
      a  rdfs:Class ;
      rdfs:subClassOf prov:Entity .

rec:hasObserved
      a   owl:ObjectProperty ;
      rdfs:domain rec:RecordTable ;
      rdfs:range rec:Observation .
```

```
rec:containsRecord
      a  owl:ObjectProperty ;
      rdfs:domain rec:RecordTable ;
      rdfs:range rec:RecordTable .

rec:Observation
      a  owl:Class ;
      rdfs:subClassOf prov:Entity .
```

In practice, we see that two types of observations frequently occur, i.e., *identified entities* and *properties measured on a scale*. Examples of *identified entities* are ‘sample XY876b’, ‘Newport’ and ‘Peter’. Quantities such as ‘length’, ‘mass’, and ‘temperature’ are examples of properties measured on a scale. These two types extend the basic schema by subclassing `rec:Observation`, as shown in Fig. 2.

In traditional tables, identified entities are typically represented by a unique, human readable identifier as a value, and a type indication in the associated header cell. RDF Record Table uses externally available domain ontologies to express all that is needed to know about such an entity by pointing to the relevant instance. In Table I, besides ‘life expectancy’ also ‘periods’, such as 2004-2006, can be considered as identified entities since they are not supposed to be read as numerical values.

For the other type of observation, a *property measured on a scale*, RDF Record Table uses ontologies that define quantitative or qualitative values defined on a scale, possibly with units of measure. In Table I, ‘sex’ and ‘life expectancy’ are typical measured properties, one on a nominal scale and the other on a rational scale, with unit ‘Year’. In our work we use OM (Ontology of units of Measure and related concepts) [14] for expressing quantitative measurements. OM contains a large number of quantities and units of measure suited to scientific and engineering datasets. It also provides the necessary properties for linking the quantities, domain concepts and units. However, other ontologies such as QUDT [15] and SDMX [9] can be used equally well. The measured quantities can be properties of the observed entities, but do not need to be related to anything specific. For example, in Table I, the life expectancy measured is that of the associated geographical region. On the other hand, ‘time’ is usually not connected to a specific entity (except for example to a ‘time zone’).

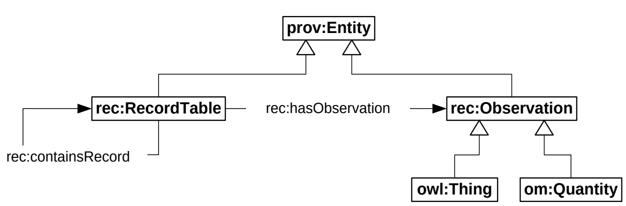

Figure 2. RDF Record Table expressing domain and provenance information

Finally, by making `rec:RecordTable` and `rec:Observation` subclasses of `prov:Entity` we ensure that all provenance information can be expressed for individual measurements and for records.

To illustrate the use of the RDF Record Table format, we show how the cells with values 76.7 and 83.3 in Table I are modelled. We see that the first level of nesting defines four records (:o1, :o2, :o3, :o4), one for each region. We use the ontology for geographic areas (as *identified entities*) that was also used in the RDF Data Cube example [5]. The next level specifies the three time periods, again using instances that were also used in the data cube example. At the third level of sub-records, we register two properties measured on a scale, viz. ‘sex’ and ‘life expectancy’. For indicating the variable ‘sex’, we use an sdmx-code, as in that data cube; to illustrate the use of OM [14], we use the concept `om:Duration` from that ontology to describe ‘life expectancy’. The value of a quantity in OM is of the type `om:Measure`, which is a combination of a numerical value and a unit.

```
:dataset1  a  rec:RecordTable ;
  rec:containsRecord :o1 , :o2 , :o3 , :o4 .

:o1  a   rec:RecordTable ;
   rec:hasObserved ex-geo:newport_00pr ;
   rec:containsRecord :o11 , :o12 , :o13 .

:o11  a   rec:RecordTable ;
   rec:hasObserved
<http://reference.data.gov.uk/id/gregorian-
interval/2004-01-01T00:00:00/P3Y> ;
   rec:containsRecord :o111 , :o112 .

:o111  a  rec:RecordTable ;
   rec:hasObserved sdmx-code:sex-M ,
        :lifeExpectancy_76_7YR .

:lifeExpectancy_76_7YR a om:Duration ;
   om:value :_76_7YR .

:_76_7YR  a  om:Measure ;
   om:numerical_value "76.7"^^xsd:string ;
   om:unit_of_measure_or_measurement_scale om:year
.

...

:o2  a   rec:RecordTable ;
   rec:hasObserved ex-geo:cardiff_00pt  ;
   rec:containsRecord :o21 , :o22 , :o23 .

:o21  a   rec:RecordTable ;
   rec:hasObserved
<http://reference.data.gov.uk/id/gregorian-
interval/2004-01-01T00:00:00/P3Y> ;
   rec:containsRecord :o211 , :o212 .

...

:o212  a  rec:RecordTable ;
   rec:hasObserved sdmx-code:sex-F ,
   :lifeExpectancy_83_3YR .
```

```
:lifeExpectancy_83_3YR a om:Duration ;
   om:value :_ 83_3YR .

:_83_3YR  a  om:Measure ;
   om:numerical_value "83.3"^^xsd:string ;
   om:unit_of_measure_or_measurement_scale  om:year
.
```

We now discuss a number of differences between RDF Record Table and RDF Data Cube. The most salient difference between RDF Data Cube and OQR Record Table is the fact that RDF Data Cube sees complex datasets as n-dimensional hypercubes, whereas RDF Record Tables are defined recursively via nesting. The second major distinction between the two approaches is that RDF Data Cube distinguishes between dimensions and measures, whereas OQR Record Table does not make a priori assumptions about the roles of individual observations. We consider making such decisions to be the task of the data analyst. Moreover, RDF Record Table has no centralized section describing the structure of the table. If it is necessary to prescribe an observation protocol or template, it suffices to list the identified entities and properties measured as the items to register in each record. Finally, where the RDF Data Cube definition makes intensive use of properties, RDF Record Table only has a few simple properties and further builds on concepts from dedicated, external ontologies.

RDF Data Cube does not allow missing variable-values or an occasional extra measurement. In contrast, in RDF Record Table any record can contain an arbitrary set of measurements, with different types and sub-records. Missing values or varying units of measure or other attributes within a single dataset are no problem. We do not demand completeness or regularity of the data, in the sense that a record can contain any set of entities and properties. This better reflects the reality of datasets in science and engineering, in particular, when datasets from different sources are combined. It can be argued that such datasets can be modelled in RDF DataCube simply by violating the integrity constraints. This is, however, a bad approach to using a standard, and can lead to interoperability problems between tools developed for the standard.

For example, in Table I we can add 'the measured average weight of the inhabitants of this region' to an existing observation using the OM quantity `om:Mass`. We can also switch to 'life expectancy' measured in months rather than years for this single observation. This is shown here:

```
:o431  a  rec:RecordTable ;
   rec:hasObserved sdmx-code:sex-M ,
   :lifeExpectancy_74_9MONTH ,
   averageWeight_71kg ;

:lifeExpectancy_74_9MONTH a om:Duration ;
   om:value :_ 74_9MONTH .

:_74_9MONTH  a  om:Measure ;
   om:numerical_value "74.9"^^xsd:string ;
   om:unit_of_measure_or_measurement_scale
om:month .

:averageWeight_71kg a om:Mass ;
   om:value :_71kg.

:_71kg  a  om:Measure ;
   om:numerical_value "71"^^xsd:string ;
   om:unit_of_measure_or_measurement_scale
   om:kilogram .
```

We conclude that RDF Record Table can be viewed as a generalized RDF Data Cube, making fewer assumptions about the regularity and completeness of the data. It can act as a precursor in the data cleaning, analysis and integration process. If a dataset that was originally drafted as an RDF Record Table meets certain requirements, it is in principle possible to automatically transform it into an RDF Data Cube. Any dataset expressed in RDF Data Cube, on the other hand, can be modeled as RDF Record Table.

IV. REDUCING REDUNDANCY IN RDF RECORD TABLE

In RDF Record Tables, the individual observations are in principle self-contained, allowing an extremely flexible approach. However, making all metadata available for each observation in practice leads to very large data files. In a single experiment, records are often very similar and much information is redundant. This means that many details can be referred to rather than repeated. In the traditional table, metadata is typically condensed in the header row, assuming that the reader knows that it holds for all rows. In an RDF-based graph model, we can be more flexible. We can use any completely specified value as a template for other observations. It is then possible, using for example SPARQL [16], to generate the full, extensive description from the reduced version when needed. This is in particular effective if the expansion to the fully explicit (normalized) form can be done locally, i.e., only for the interesting parts of a table.

Fig. 3 shows how RDF Record Table supports compression of datasets by giving metadata information by referring to a similar measurement. Each `rec:Observation` can hold a literal value (the string or number ending up in a table cell) and emulate another observation, which has identical attributes other than the value. These referencing observations are collected in records, just like normal observations.

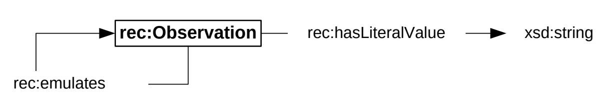

Figure 3. Describing an observation by reference.

In Turtle format, the definition is as follows.

```
rec:emulates
      a   owl:ObjectProperty ;
      rdfs:domain rec:Observation ;
      rdfs:range rec:Observation .

rec:hasLiteralValue
      a    owl:DatatypeProperty ;
```

```
      rdfs:domain rec:Observation ;
      rdfs:range xsd:string .
```

For example, the observation from Table I that in Monmouthshire the life expectancy of women in the period 2006-2008 was 81.7 years, is originally expressed in

```
:o332  a  rec:RecordTable ;
   rec:hasObserved sdmx-code:sex-F ,
   :lifeExpectancy_81_7YR .
```

as

```
:lifeExpectancy_81_7YR a om:Duration ;
   om:value:_81_7YR .

:_81_7YR  a  om:Measure ;
   om:numerical_value "81.7"^^xsd:string ;
   om:unit_of_measure_or_measurement_scale  om:year
.
```

Using the fact that all details for `:lifeExpectancy_81_7YR` are the same as for `:lifeExpectancy_76_7YR` from observation `:o111`, except for the actual value, we can summarize this as

```
:lifeExpectancy_81_7YR a rec:Observation ;
   rec:emulates :lifeExpectancy_76_7YR ;
   rec:hasLiteralValue "81.7"^^xsd:string.
```

For this example, this may not seem an impressive compression. However, if more metadata is included, such as descriptions, the devices used, methods applied and other background information, the reduction of the size will be substantial. This, in particular, holds for datasets with large numbers of similar measurements. Finally, further reduction of datasets is possible by applying general compression algorithms [17].

V. IMPLEMENTATION EXAMPLE

A good model of tabular data is useless if the data can't easily be input. Given the popularity of the classic table format in tools such as spreadsheets, it should be possible to use these for data entry and then construct semantic datasets from there. In order to make this process as easy as possible, it should fit into existing work procedures and tools and minimize additional effort by the user. Since Microsoft Excel is extremely popular, we have implemented the RDF Record Table model as an add-in for this tool, called Rosanne [14]. Rosanne supports engineers and scientists in creating semantic tables (as yet simple tables, i.e., rectangular with one header row or column). Similar functionality for the RDF Data Cube has been implemented in TabLinker [18]; however this is a standalone tool which cannot be accessed from within Excel. Rosanne allows users to enter their data in a simple table format. Rosanne then uses OM (Ontology of units of Measure and related concepts) [14] to assist users in adding relevant quantities and units of measure to the table. In addition, other domain-specific ontologies are available for annotating identified entities in the table, such as samples, objects, locations, etc.

The user is not confronted with the Record Table model nor do they have to have any knowledge of ontologies. The user selects the concepts they want from dropdown lists showing the user-friendly labels from the ontologies. The URIs (Uniform Resource Identifiers) for the ontology concepts are stored in the Record Table model by the add-in. The add-in can also automatically annotate existing data with units and quantities from OM, based on heuristics [19]. This does not always produce accurate results, but saves time for the user by creating an initial annotation which can be corrected where necessary. Finally, Rosanne allows users to search for annotated tables and integrate them.

Fig. 4 shows an example from food science. In this experiment, the researcher wishes to combine rheological measurements on protein samples with sample composition data. Without semantic support, this task would require her to find the relevant files somehow, then to copy and paste different data by hand, with plenty of scope for error. With Rosanne, she can find the files easily via the search function. The table has been annotated using OM and a domain ontology. She then selects 'Protein' as the identifier, and 'Storage Modulus' and 'Composition' as the variables of interest. Rosanne creates a query to find the relevant data, and generates the integrated table.

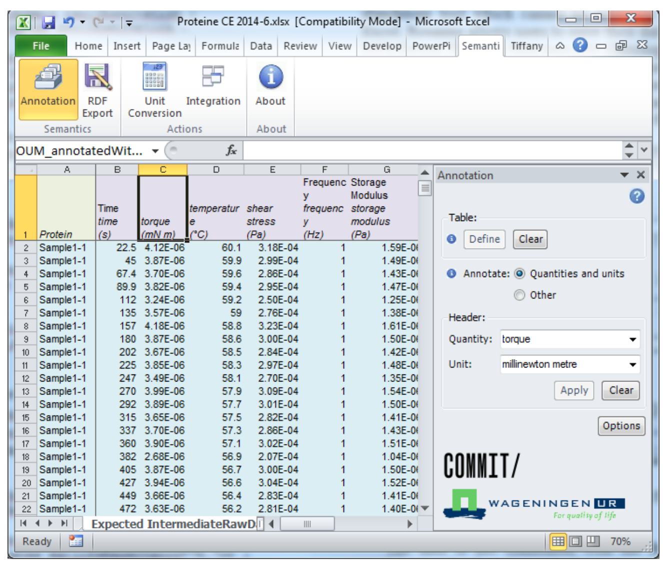

Figure 4. Rosanne using RDF Record Table.

VI. CONCLUSION AND FUTURE WORK

Looking to the future, semantic datasets are a step towards advanced quantitative e-science. The data can be documented and linked to the scientific process, assisting the researcher and ultimately leading to full transparency and reusability of quantitative scientific knowledge.

In practice, this means that data entry tools can be developed which use ontologies to support the user in adding contextual information. Describing the content and structure of tabular data semantically makes it possible to easily find data even in disparate sources, to understand and clean the data and to combine it semi-automatically. This way, much richer datasets will be published in the future, so that others can fully understand them and build further on them.

We have proposed RDF Record Table as a way to organize observational data semantically. The model complements the RDF Data Cube vocabulary. RDF Data Cube offers the benefits of semantic modelling to domains such as statistics, with regular, standardized datasets. RDF Record Table offers more flexibility in storing heterogeneous data, and therefore extends those same benefits to the more complex world of science and engineering. A first implementation of the RDF Record Table model in Microsoft Excel, called Rosanne, demonstrates the benefits of semantic tables. This includes semi-automatic integration of datasets. This functionality is presently being evaluated by a number of R&D organizations of multinationals in food production, cooperating in TI Food and Nutrition [20]. In another area, we are using RDF Record Table for statistical analysis with the popular language R.

For full implementation of this model, several issues must still be solved. We mentioned the automatic (local) expansion and compression of datasets, mapping to and from RDF Data Cube, and the translation to and from two-dimensional representations. In addition to these, the recovery of legacy data needs attention. There is a wealth of data stored in existing spreadsheets, which have, in general, an informal structure and no annotations. Current results for fully automatic annotation are still of insufficient quality [19], so more research is needed to find how to unlock this legacy data. We plan to submit RDF Record Table to the CSV on the Web Working Group [21] for consideration and inspiration in their work to provide better interoperability for tabular data.

ACKNOWLEDGMENT

This publication was supported by the Dutch national program COMMIT.

REFERENCES

[1] Y. L. Simmhan, B. Plale, and D. Gannon. ‘A survey of data provenance in e-science.’ ACM SIGMOD Record, 2005. doi:10.1145/1084805.1084812

[2] A. Garcia, O. Giraldo, and J. Garcia, ‘Annotating Experimental Records Using Ontologies.’ Int. Conference on Biomedical Ontology, Buffalo, NY, USA, 2011. Available from: http://ceur-ws.org/Vol-833/paper12.pdf. Retrieved June, 2014.

[3] J. Gray, ‘Jim Gray on eScience: a transformed scientific method.’ in T. Hey, S. Tansley, K. Tolle (Eds.), The Fourth Paradigm: Data-Intensive Scientific Discovery, Microsoft Research, 2009, pp. xvii–xxxi.

[4] Semantic Web, W3C. Available from: http://www.w3.org/standards/semanticweb/. Retrieved: June, 2014.

[5] R. Cyganiak, D. Reynolds, (eds). RDF Data Cube Vocabulary, W3C, 2012. Available from: http://www.w3.org/TR/vocab-data-cube/. Retrieved June, 2014.

[6] L. Han, T. Finin, C. Parr, J. Sachs, and A. Joshi, ‘RDF123: From spreadsheets to RDF.’ Lecture Notes in Computer Science, Vol. 5318 LNCS, 2008, pp. 451–466. doi:10.1007/978-3-540-88564-1-29

[7] J. Cunha, J. Saraiva, and J. Visser, ‘From spreadsheets to relational databases and back.’ In Proceedings of the 2009 ACM SIGPLAN workshop on Partial evaluation and program manipulation - PEPM ’09 (p.179), 2009.

[8] C. Bizer, and R. Cyganiak, ‘D2R Server – Publishing Relational Databases on the Semantic Web.’, World, p. 26, 2006.

[9] S. Capadisli, S. Auer and A.-C. Ngonga Ngomo, ‘Linked SDMX Data’. Semantic Web, 2013. doi:10.3233/SW-130123

[10] P. Groth, L. Moreau. (eds), PROV Overview, W3C, 2013. Available from: http://www.w3.org/TR/2013/NOTE-prov-overview-20130430/. Retrieved June, 2014.

[11] Matlab, The Language of Technical Computing. Available from: http://www.mathworks.nl/products/matlab/. Retrieved July, 2014.

[12] SPSS Statistics. Available from: http://en.wikipedia.org/wiki/SPSS. Retrieved July, 2014.

[13] M. Nilsson, A. Powell, P. Johnston, and A. Naeve. ‘Expressing Dublin Core metadata using the Resource Description Framework (RDF).’, 2008. Available from: http://dublincore.org/documents/dc-rdf . Retrieved June, 2014.

[14] H. Rijgersberg, M. Wigham, and J. L. Top, ‘How semantics can improve engineering processes: A case of units of measure and quantities.’ Advanced Engineering Informatics, *25*(2), 2010, pp.276–287. doi:http://dx.doi.org/10.1016/j.aei.2010.07.008

[15] R. Hodgson, P. J. Keller, J. Hodges, and J. Spivak, ‘QUDT - Quantities, Units, Dimensions and Data Types Ontologies’. Available from: http://qudt.org/. Retrieved June, 2014.

[16] W3C, SPARQL Query Language for RDF. Available from: http://www.w3.org/TR/rdf-sparql-query/. Retrieved July, 2014.

[17] J. Urbani, J. Maassen, N. Drost, F. Seinstra, F., and H. Bal, ‘Scalable RDF data compression with MapReduce.’ Concurrency Computation Practice and Experience. Vol. 25, pp. 24–39, 2013. doi:10.1002/cpe.2840

[18] TabLinker, 2012. Available from: http://www.data2semantics.org/2012/02/19/tablinker/. Retrieved June, 2014.

[19] M. van Assem, H. Rijgersberg, M. Wigham, and J.L Top, ‘Converting and annotating quantitative data tables’. The Semantic Web - ISWC 2010, vol. 6496/2010, 2010, pp. 16–31. doi:10.1007/978-3-642-17746-0_2.

[20] TI Food and Nutrition. Available from: http://www.tifn.nl. Retrieved July, 2014.

[21] CSV on the Web Working Group Charter, 2013. Available from: http://www.w3.org/2013/05/lcsv-charter.html. Retrieved June, 2014.

 ISBN: 978-1-61208-355-1