Challenges on Developing Tools for Exploiting Linked Open Data Cubes

Evangelos Kalampokis[1,2], Bill Roberts[3], Areti Karamanou[1,2], Efthimios Tambouris[1,2], Konstantinos Tarabanis[1,2]

[1] Centre for Research & Technology - Hellas, 6th km Xarilaou-Thermi, 57001, Greece
[2] University of Macedonia, Egnatia 156, 54006 Thessaloniki, Greece
{ekal, akarm, tambouris, kat}@uom.gr
[3] Swirrl IT Limited, 20 Dale Street, Manchester, M1 1EZ, United Kingdom
{bill}@swirrl.com

Abstract. A major part of open data provided by international and governmental organizations include facts and figures that are described in a multi-dimensional manner (aka data cubes). The real value, however, of these open data cubes will unveil from combining and exploiting them in analytics across the Web. Linked data paradigm promises to facilitate the realization of this vision. The RDF data cube (QB) vocabulary, which enables modeling multidimensional data as RDF graphs, is a major step towards this direction. Based on the QB vocabulary a number of data cubes are provided as linked data either by the owners of the data or by third parties. However, existing linked open data cubes do not facilitate the development of generically applicable tools that could use data from different sources. The aim of this paper is to present challenges related to the development of software tools that combine and exploit linked open data cubes in analytics and visualizations. These challenges have been emerged during the development of the OpenCube suite of tools that support the whole linked data cube lifecycle. We anticipate that the identified challenges will enable publishing linked data cubes of high quality.

Keywords: Linked data, open data, statistics, data cube, OLAP, data analytics.

1 Introduction

A major part of open data provided by international and governmental organizations include facts and figures that are described in a multidimensional manner [1]. For example, based on a sample of 100 datasets from data.gov.uk that ranked highest for 'popularity', 45% of the datasets were multidimensional data. Multidimensionality means that a measured fact is described based on a number of dimensions, e.g. unemployment rate on different countries, years, and age groups. This type of data is compared to a cube, where the location of a cell is specified by the values of the dimensions, while the value of a cell specifies the measured fact [2]. Hence we onwards refer to this type of datasets as data cubes or just cubes.

Linked data has been introduced as a promising paradigm for opening up data because it facilitates data integration on the Web [3]. In the case of data cubes, linked data has the potential to create added value by combining figures and facts from various sources and performing analytics [4] [5]. A fundamental step towards this vision is the RDF data cube (QB) vocabulary, which enables modeling data cubes as RDF graphs [6].

The process that enables publishing raw data cubes as linked data, combining cubes from multiple sources, and exploiting them in data analytics and visualizations has been recently described in the literature [7]. Moreover, software tools that support this process have been developed with a focus on linked data cubes creation and exploitation. In the first case the tools aim at transforming data from legacy technical formats ranging from CSV, JSON-stat and SDMX-ML to relational and OLAP databases into RDF data adhering to the RDF Data Cube (QB) vocabulary (e.g. [8], [9], [10], [11], [12]). In the case of exploitation existing tools enable exploring cubes in two-dimensional tables and on maps, and creating charts (e.g. [9], [13], [14]).

At the moment, a number of linked data cubes are also available on the Web. Some of them are official endeavors launched by the organizations that own the data. For example, the European Commission's Digital Agenda[1] provides its Scoreboard as linked data cubes. Census data of 2011 from Ireland has been also published as linked data cubes[2]. The Department for Communities and Local Government (DCLG) in the UK[3] and the Flemish Government[4] also provides statistics as linked data. At the same time cubes from Eurostat, European Central Bank, World Bank, UNESCO and other international organizations have been also transformed to linked data in third party activities [15].

Although this activity proves that both academia and businesses gained much experience in the area during the last years, we are still far from having tools that can be applied successfully to a wide range of datasets and data from different datasets and publishers that can be compared and combined. The aim of this paper is to present challenges and opportunities that are related to combining and exploiting linked data cubes in analytics and visualizations. These challenged have emerged during the development of a number of software tools aiming at dealing with linked data cubes as well as the use of these tools in exploiting linked data cubes mainly from DCLG, Flemish Government, Irish CSO, and the Digital Agenda. The development and evaluation of the tools has been performed in the OpenCube project.

The rest of the paper is structured as follows: Section 2 sets the background of our work and describes the linked data cubes lifecycle along with the tools that have been developed in the OpenCube project. Section 3 presents the identified challenges while section 4 briefly discusses these results. Finally, section 4 draws conclusions.

---

[1] http://digital-agenda-data.eu/data
[2] http://data.cso.ie
[3] http://opendatacommunities.org/data
[4] http://data.opendataforum.info

2 Software tools for exploiting linked data cubes

In this section we present the tools that we have developed to deal with linked data cubes. The tools support the whole linked data cube lifecycle. Sub-section 2.1 briefly presents the three phases of the lifecycle, while sub-section 2.2 briefly present the tools.

2.1 Linked data cube lifecycle

Linked data cubes go through three phases in order to create value [7]. The first phase deals with transforming raw data into linked data cubes and addresses the following activities:

- Discover & pre-process raw data in various data formats such as CSV files, XLS files, RDBMS.
- Create RDF data adhering to the Data Cube vocabulary
- Manage and re-use controlled vocabularies (concept schemes, code lists etc.)
- Publish cubes through different interfaces i.e. Linked Data, SPARQL endpoint etc.
- Manage metadata

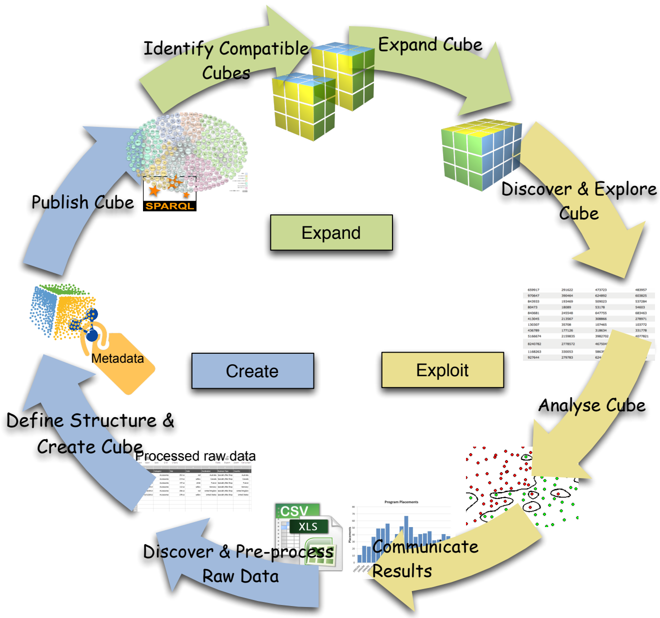

Fig. 1. The Linked Data Cubes lifecycle

The second phase deals with expanding linked data cubes by joining them with other cubes on the Web and addresses the following tasks:

- Discover compatible to join linked data cubes in existing collections of cubes.
- Establish typed links between compatible to join cubes.

- Create a set of compatible cubes from an initial linked data cube by computing aggregations across a dimension or a hierarchy.
- Create expanded cubes by increasing the size of one of the sets that define a cube i.e. measures, objects of a dimension's level, levels of a dimension, or dimensions.

The final phase deals with exploiting linked data cubes in data analytics and visualizations and considers the following tasks:

- Discover and explore linked data cubes.
- Perform OLAP operations on linked data cubes.
- Perform statistical analyses on linked data cubes e.g. compute descriptive statistics, calculate statistics such as correlation coefficient, and create learning models.
- Communicate results through visualizations.

In this paper, we identify challenges related to the two latter phases in order to provide feedback at the first one. Towards this end, we develop software tools that expand and exploit existing linked data cubes.

2.2 Tools

The following tools per phase of the lifecycle have been developed during the OpenCube project. Two linked data management platforms serve as a backbone for the tools, namely Information Workbench and PublishMyData.

- Creating linked data cubes
    - *Grafter* is an ETL framework designed specifically to create RDF for linked data publishing purposes.
    - The *JSON-stat2qb* tool facilitates the automatic transformation of cubes in JSON-stat format to linked data cubes.
    - The *R2RML extension* for data cubes enables the transformation of cubes structured in tabular sources to linked data cubes.
- Expanding linked data cubes
    - The role of the *Aggregator* is twofold. First, given an initial cube with n dimensions the aggregator creates 2^n-1 new cubes taking into account all the possible combinations of the n dimensions. Second, given an initial cube and a hierarchy of a dimension, the aggregator creates new observations for all the attributes of the hierarchy.
    - Given an initial cube in the local RDF store of the infrastructure, the main role of the *Compatibility Explorer* is to (a) search into the Linked Data Web and identify cubes that are compatible to expand the initial cube, and (b) establish typed links between the local cube and the compatible ones.
    - The *Expander* creates a new expanded cube by merging two compatible cubes.
- Exploiting linked data cubes

- The *OLAP browser* enables performing OLAP operations (e.g. pivot, drill-down, and roll-up) on top of linked data cubes.
- The *MapView* enables the visualization of RDF data cubes on a map based on their geospatial dimension using choropleth and markers maps.
- The *Spreadsheet Builder* provides a form of wizard to assist a user to build up a table of data, with geographical areas as rows and slices of data cube dataset as columns. It has been designed to help analysts with no programming or SPARQL skills to select specific data from multiple cubes for direct comparison or easy download for analysis. The table can be viewed online or downloaded as a CSV file for analysis or visualisation with other tools.
- The *R statistical analysis* tool enables processing linked data cubes with R and present the results using charts and visualizations.

3 Challenges

In this section the challenges that we faced during the development of the tools that support the expanding and exploiting phases of the linked data cubes lifecycle are described. We have categorized the challenges as follows:

- Challenges related to the different practices that can be followed in applying the RDF data cube (QB) vocabulary.
- Challenges related to the misuse of the QB vocabulary.
- Challenges related to the re-use of controlled vocabularies and code lists.
- Challenges related to the lack of data.
- Challenges related to use of proposed extensions of the QB vocabulary.
- Challenges related to conceptual issues.

3.1 Different practices in applying QB vocabulary

In many cases, the flexibility of the QB vocabulary enables publishers to follow different practices for publishing linked data cubes. These different practices hampers (a) the development of generic tools that can be used across different linked data cubes as well as (b) the combination of cubes across multiple sources.

Here the most important challenges are related to the understanding of the semantics of a measure. A widely adopted practice when referring to a qb:MeasureProperty is to use sdmx-measure:obsValue. For example Digital Agenda uses sdmx-measure:obsValue but it also defines an indicator dimension for which values are measured. The indicator takes values from a code list. In addition, DCLG, the Flemish government and the Irish CSO define measures as rdfs:subPropertyOf sdmx-measure:obsValue. DCLG data generally uses measure properties with quite specific semantics. In other data collections, Swirrl has used a small number of more generic measure properties, such as ‘count’ and ‘ratio’, defined as subproperties of sdmx-measure:obsValue. These work in conjunction with a sdmx-attribute:unitMeasure which defines what the observation is a ‘count’ of: for example people or households. These unitMeasure values are re-used across datasets wherever possible to maximise the opportunities for combining and comparing data.

Often multiple measures need to be included in a datacube. The QB vocabulary proposes two approaches to include multiple measures in data cubes: (a) multimeasure observation or (b) qb:measureType.In the first approach the multiple measures can be declared as qb:MeasureProperty components in the structure of the cube. Each observation can be then attached with multiple observed values. One problem with this approach is that it allows the attachment of only a single attribute to each observation that will describe only one of the measured values. This could be fixed using the qb:componentAttachment property so as to attach one attribute to each qb:MeasureProperty but this attachment will regard the whole data set and can't vary between observations. The qb:measureType approach overcomes the previous problems. More precisely the second approach suggests to add extra dimensions to the structure of the cube using the qb:measureType component. These extra dimensions will actually play the role of the measures of the cube. Each observation of the cube will then have a single measured value. The disadvantage of this approach is that it substantially multiplies the number of triples potentially leading to performance and storing issues in the triple store that are stored. It is difficult to create a generic tool that consumes data following both approaches. The Irish CSO and Digital Agenda currently don't use multiple measures while DCLG uses the qb:measureType property option. Finally, the Flemish government employs both approaches. However the qb:measureType approach seems to be the most extensible and flexible one, due to the fact that it allows the use of much metadata/attributes for every individual observation as needed.

The QB vocabulary offers the possibility to group a set of observations into a 'slice' where all but one or a small number of dimensions are fixed. The slice offers a mechanism for attaching metadata to that group of observations. The main example data collections examined in this paper, from DCLG, Irish CSO, Flemish Government and EU Digital Agenda, do not currently make use of slices. However, recent developments in our approach (mainly related to the browser developed in PublishMyData environment) have found slices beneficial in two main respects. Firstly, for large data cubes, selecting observations to display in a two-dimensional table can lead to SPARQL queries that are expensive to execute. If observations are already associated with two-dimensional slices, this provides a convenient index that simplifies and speeds up such queries. Secondly, for data cubes with many dimensions, it is often the case in practice that these cubes can be 'sparse': some combinations of dimension values do not have associated observations. In this case, it can sometimes be difficult for a user to navigate to populated parts of the cube. A user interface can present the user with a list of slices as a way of simplifying navigation to interesting, popular, or simply non-empty combinations of dimensions.

Moreover, the QB vocabulary allows two different practices for defining the allowed values of a dimension within a data set: (a) by connecting the qb:ComponentProperty with a qb:codeList property or (b) by defining the qb:ComponentProperty with a range of skos:Concept or a subclass of skos:Concept. Digital Agenda for example follows the first approach and connects qb:DimensionProperty with a codelist (it uses codelist <http://eurostat.linked-statistics.org/dic/geo#> from Eurostat). DCLG and Irish CSO in most cases do not associate dimension properties with a specific codelist but defines them with a range of skos:Concept, or a more specific class which is a subclass of skos:Concept. For example, the Irish CSO preserves data about 12 geographical hierarchical levels and defines a different concept scheme per geo level. This practice may be convenient but impedes the computation of aggregations as a complete codelist with levels is required.

3.2 QB vocabulary misuse

There are a few cases where the creation of linked data cubes is not consistent with what the QB vocabulary specifies. In such cases it is difficult to reuse generic tools for either exploiting or expanding data cubes.

For example, the RDF Data Cube vocabulary suggests the use of one qb:DimensionProperty for each of the cubes' dimensions. Digital Agenda follows a very particular approach for the definition of its data sets' dimensions where a "super-dimension" is defined to embrace the values of dimensions other than time and location. Precisely, a "super-dimension" named "breakdown" is used to represent several values of dimensions including, for example, dimensions labeled as "Individuals who are born in non-EU country", "Individuals with high formal education" or "Unemployed". This approach facilitates the creation of RDF out of a huge data warehouse with hundreds of dimensions. However this "super-dimension" approach also generates problems in (a) developing generic tools that consume RDF data cubes, and (b) combining data cubes.

Moreover, in a third party transformation of Eurostat's data the following practices have been observed:

- Measures are defined using *sdmx-measure:obsValue* that is declared as a *qb:DimensionProperty*.
- In cubes with multiple measures an extra *qb:DimensionProperty* is defined.
- Attributes such as frequency and unit are defined as *qb:DimensionProperty*.

3.3 Re-use of controlled vocabularies and code lists

It is very important in linked data cubes to follow the main principle of linked data and re-use whenever possible existing URIs that describe resources or classes and properties. This should be happened to defined dimensions, objects of dimensions, levels of dimensions, measures, unit of measures, etc. This is of great importance for the combination of different data cubes. If different but related concept schemes are used, it is important to be able to define relationships between them.

For example, the time dimension is very important in most data cubes. A common approach for the time dimension property of a cube is to use sdmx-dimension:timePeriod or sdmx-dimension:refPeriod or a subproperty of them. For example, Digital Agenda uses the <http://semantic.digital-agenda-data.eu/def/property/time-period> property, a subproperty of sdmx-dimension:timePeriod. Moreover, DCLG uses a subProperty of

<http://purl.org/linked-data/sdmx/2009/dimension#refPeriod> that is defined to have a range of <http://reference.data.gov.uk/def/intervals/Interval>. Finally, the Irish CSO does not use a time dimension in most of its data sets. However, when it does, it employs a resource of its own the name of which derives from the specific dataset (e.g. <http://data.cso.ie/census-2011/property/household-year-built>). Regarding the values of the time dimension of a cube, two different approaches are also used: (a) employing a predefined URI or (b) employing a literal value. For example the year 2014 could described as a resource e.g. <http://reference.data.gov.uk/id/gregorian-year/2014> or as a literal '2014'. DCLG and Digital Agenda standardises on URIs for time intervals provided by reference.data.gov.uk. These are clearly defined with start and end points to the time interval and allows use of commonly occurring but reasonably complex intervals such as 'government years' which in UK run from 1 April to 31 March. The use of URIs offers more precise definitions of the time interval as long as these URIs are predefined and provides with all the linked data advantages such as the facilitation of the identification, linking or comparability with other cubes. A challenge here is the ability to correctly order the values of the time dimension in time and not in lexical order. Regarding the second approach, the use of literal values for the time dimension facilitates the SPARQL querying of the cube using, for example, queries such as *"select observations made before 2014"* or *"select the most recent observation"*.

A geospatial dimension is also of high importance in most data cubes. A standard approach to define the geospatial dimension of a cube is to use sdmx-dimension:refArea property or a subproperty of the sdmx-dimension:refArea property. The Irish CSO for example uses sdmx-dimension:refArea property for the geospatial dimension. On the contrary, Digital Agenda uses <http://semantic.digital-agenda-data.eu/def/property/ref-area> which is sub-property of sdmx-dimension:refArea and DCLG uses <http://opendatacommunities.org/def/ontology/geography/refArea> also a sub-property of sdmx-dimension:refArea.

There is currently a need for constructing a commonly accepted codelist for the units of measures of cubes. The codelist will embrace the different units of measurements and be reused by different data sets. The lack of such commonly accepted codelist results in the adoption of different codelists for the unit values in different data sets. For example Digital Agenda uses units from a codelist of its own (http://semantic.digital-agenda-data.eu/codelist/unit-measure). In addition, DCLG and the flemish government use QUDT (http://www.linkedmodel.org/doc/qudt-vocab-units/1.1/index.html) which facilitates the conversion to other units. DCLG also uses DBPedia for currencies, and in particular <http://dbpedia.org/resource/Pound_sterling>. Finally, the Irish CSO doesn't define units for measures at all. There are many ways to define the unit for each of the cube's measures. If there is only one measure, then the unit can be defined at qb:DataSet level. While, if there are multiple measures the unit can either be defined at qb:MeasureProperty level using a qb:componentAttachment property or at qb:Observation level (at the latter case a separate observation is needed for each measure). The Irish CSO doesn't employ units while Digital Agenda defines the unit of a (single) measure at observation level. DCLG also defines units of measures at observation level. Finally, the Flemish government also defines units at the observation level. Regarding the use of the qb:componentAttachment property, the specification of the RDF Data Cube vocabulary is ambiguous. Precisely the vocabulary states that "*It is also possible to attach attributes to a qb:MeasureProperty in which case the attribute is intended to apply only to that property and not to the observations in which that property occurs*" (RDF Data Cube Vocabulary, Chapter 10) but also that "*Attributes can also be attached directly to the qb:MeasureProperty itself (e.g. to indicate the unit of measure for that measure) but that attachment applies to the whole data set (indeed any data set using that measure property) and cannot vary for different observations*." (RDF Data Cube Vocabulary, Chapter 6.5).

It is important to be able to present the values of codelists in a specific order. For example, age ranges should be presented in order of increasing age, not lexical order of the label. Moreover, a 'total' or 'all' item in a codelist should be presented as the last one. Sometimes there are additional standard orderings of codes used in datasets. Experiments have been done with using the <http://www.w3.org/ns/ui#sortPriority> as a predicate in a concept scheme to define how data should be ordered when presented.

The definition of machine-readable hierarchical relationships (existing e.g. in geospatial data) is very useful for enabling aggregations within codelists. Nevertheless such relationships are generally not widespread in codelists. For example the Irish Census and Digital Agenda don't define hierarchies. DCLG also doesn't currently define hierarchies within codelists although its data sets include geographical hierarchies. There are currently two approaches for defining hierarchical relationships: (a) using qb:HierarchicalCodeList or (b) adopting the SKOS or XKOS vocabularies. The qb:HierarchicalCodeList is introduced by the RDF Data Cube vocabulary and defines a set of root concepts in the hierarchy (qb:hierarchyRoot) and a parent-to-child relationship (qb:parentChildProperty). The SKOS vocabulary offers skos:broader and skos:narrower properties to enable the representation of hierarchical links. Moreover, XKOS, an extension of SKOS, also allows the modelling of hierarchies structured in levels. A hierarchy level can be defined using the xkos:ClassificationLevel concept. According to XKOS the levels of a hierarchy are organised as an rdf:List, which implies order, starting with the most aggregated level. Individual skos:Concept objects are related to the xkos:ClassificationLevel to which they belong by the skos:member property. Although XKOS seems to be a promising solution for the definition of machine-readable relationships, it is not currently commonly used.

3.4 Lack of data

In many cases linked data cubes have been published according to the QB vocabulary but some missing information hampers their exploitation from generic tools. For example, the unit of measure is often not available in the data cubes. This is the case for example in the Irish census data.

However, if one needs to perform OLAP operations such as moving the analysis details along a hierarchy (aka drill-down or roll-up) require computing aggregations of the measured fact across a dimension or a hierarchy. Mainly three types of aggregate functions as distinguished in the literature can be applied: Σ, applicable to data that can be added together, ϕ, applicable to data that can be used for average calculations, and c, applicable to data that is constant, i.e., it can only be counted. Considering only the standard SQL aggregation functions, we have that Σ = {SUM, COUNT, AVG, MIN, MAX}, ϕ = {COUNT, AVG, MIN, MAX} and c = {COUNT}. For example, let us consider a cube with three dimensions, namely stores, years, and products in which if we compute the SUM of sales of all products of a company for all years and stores we can remove the product dimension and thus have a view of sales based on only time and stores dimensions. The challenge in this type of operations is to select the aggregation function that is appropriate for the data at hand and compute the aggregations based on the initial linked data cube. The unit of measure is of vital importance towards this end.

3.5 Conceptual issues

An important challenge that hampers the development of tools that combine data cubes across the Web is the granularity of the cube. Different publishers specify cubes of different size. For example, the Irish Census of 2011 has defined 682 linked data cubes with one measure per cube while Digital Agenda only 2 cubes with more than 100 measures per cube. In such cases different approaches need to be followed in order to integrate data from two cubes and exploit them.

4 Discussion & Conclusions

During the last years the open data movement has been introduced evangelizing the need for certain data to be freely available for re-use. A major part of open data is structured as multi-dimensional data cubes. Linked data technologies have the potential to realise the vision of combining and performing analytics on top of previously isolated cubes at a Web scale.

Our objective in the OpenCube project has been to make data cubes more accessible and more powerful. Standardisation in the representation of the data means that analysis and visualisation tools can be applied successfully to a wide range of datasets. It also means that data from different datasets and publishers can be compared and combined.

The RDF Data Cube Vocabulary is a very valuable step towards this. As we have seen it allows a variety of different solutions to the detailed representation of data cubes. This is both a strength and a weakness: flexibility allows data publishers to choose the options that are best suited to their particular situation, but different approaches by different publishers make it difficult to produce generically applicable tools to work with the data.

Furthermore, the use of RDF Data Cube is an important step towards interoperability of statistical datasets, but is not the full story. The vocabulary provides mechanisms for carefully defining measures, dimensions and their values but for the greatest interoperability, common concept schemes and code lists across datasets and publishers is needed.

This is primarily a social problem rather than a technical one, and in many cases the sets of dimension values for a dataset may need to be specific to the characteristics of that data. However, often a data publisher could re-use an existing concept scheme or URI set rather than inventing their own near-duplicate of it. For this to succeed, the creators of such URI sets need to ensure they are well defined and documented, and to make them discoverable and re-usable by others.

Formal standarisation of even simple codelists is often a difficult and time consuming process. However, where such standardised codelists already exist, those responsible for the standard could greatly benefit the statistical community by making them easily available as SKOS concept schemes, or other data cube friendly formats, and of course committing to maintain the RDF representation. In other cases, data publishers may need to make their own concept scheme, but can use an existing formal standard as a starting point. The various linking mechanisms of Linked Data and SKOS can then be used to assert equivalence of identical or closely related concepts, supporting data users in joining data from different sources.

Another clear strand emerging from our work has been the importance of aggregating data, and the difficulties of doing this reliably. This is an established practice in the use of OLAP methods for business intelligence, but can be challenging in the more flexible data structures of the RDF Data Cube. It is another case where provision of high quality metadata by data publishers can greatly increase the value of the data for data users. A clear statement by the data owner of whether and how the data in a data cube can be meaningfully aggregated is an important first step.

Finally, an important strand of this work has been evaluation of new data cube tools by data users. A clear message from the evaluation work has been that, while the linked data approach is powerful in allowing automated and tool-supported combination of data from different sources, users often want to consume data in other formats and in a range of existing analysis and visualisation tools. Many popular and powerful data analysis tools exist and few (if any) of them are able to consume linked data directly. Therefore to get the maximum value from the use of RDF Data Cubes, we must not neglect the final step of delivering the user's selection of data in a format that suits them. The work within OpenCube to integrate data cube tools with 'R' (the R Project for Statistical Computing) is an important example of this. However there are many other tools and many other contexts where statistical data is used. The value generated by the data representation and interconnection work described in this paper can be greatly amplified by ensuring that the outputs can be translated for use by the most popular data consumption tools, whether simple charting packages or complex statistical analysis.

Acknowledgments. The work presented in this paper was partially carried out in the course of the OpenCube[5] project, which is funded by the European Commission within the 7th Framework Programme under grand agreement No. 611667.

[5] http://www.opencube-project.eu

References

1. Cyganiak, R., Hausenblas, M., McCuirc, E.: Official Statistics and the Practice of Data Fidelity. In: Wood, D. (ed.) Linking Government Data, pp. 135–151. Springer (2011)
2. Datta, A., Thomas, H.: The cube data model: a conceptual model and algebra for on-line analytical processing in data warehouses. Decision Support Systems 27(3), 289 – 301 (1999)
3. Bizer, C., Heath, T., Berners-Lee, T.: Linked data - the story so far. International Journal on Semantic Web and Information Systems 5(3), 1–22 (2009)
4. Kalampokis, E., Tambouris, E., Tarabanis, K.: Linked Open Government Data Analytics. Wimmer, M.A., Janssen, M., Scholl, H.J. (eds.) EGOV2013, LNCS, vol. 8074, pp.99-110. IFIP International Federation for Information Processing (2013)
5. Abello, A., Darmont, J., Etcheverry, L., Golfarelli, M., Mazon, J.-N., Naumann, F., Pedersen, T., Rizzi, S. B., Trujillo, J., Vassiliadis, P., Vossen, G.: Fusion cubes: Towards self-service business intelligence. Data Warehousing and Mining 9(2), 66-88 (2013)
6. R. Cyganiak and D. Reynolds, "The RDF data cube vocabulary: W3C recommendation," W3C, Tech. Rep., January 2014.
7. E. Tambouris, E. Kalampokis, K. Tarabanis (2015) Processing Linked Open Data Cubes, E. Tambouris, M. Janssen, H. J. Scholl, M. Wimmer, K. Tarabanis, M. Gascó, B. Klievink, I. Lindgren, and P. Parycek (Eds.): EGOV2015, LNCS 9248, IFIP
8. S. Capadisli, S. Auer, and A.-C. Ngonga Ngomo, "Linked SDMX data," Semantic Web, 2013
9. E. Kalampokis, A. Nikolov, P. Haase, R. Cyganiak, A. Stasiewicz, A. Karamanou, M. Zotou, D. Zeginis, E. Tambouris, and K. Tarabanis, "Exploiting linked data cubes with OpenCube toolkit," in *Proc. of the ISWC 2014 Posters and Demos Track a track within 13th International Semantic Web Conference (ISWC2014)*, M. Horridge, M. Rospocher, and J. van Ossenbruggen, Eds. CEUR-WS, 2014, vol. 1272.
10. P. E. R. Salas, F. M. Da Mota, K. K. Breitman, M. A. Casanova, M. Martin, and S. Auer, "Publishing statistical data on the web," International Journal of Semantic Computing, vol. 06, no. 04, pp. 373–388, 2012.
11. P. Salas, M. Martin, F. Mota, S. Auer, K. Breitman, and M. Casanova, "OLAP2DataCube: An Ontowiki plug-in for statistical data publishing," in *Developing Tools as Plug-ins (TOPI), 2012 2nd Workshop on*, June 2012, pp. 79–83
12. L. Ruback, M. Pesce, S. Manso, S. Ortiga, P. E. R. Salas, and M. A. Casanova, "A mediator for statistical linked data," in *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, ser. SAC '13. New York, NY, USA: ACM, 2013, pp. 339–341.
13. C. Mader, M. Martin, and C. Stadler, "Facilitating the exploration and visualization of linked data," in *Linked Open Data – Creating Knowledge Out of Interlinked Data*, ser. Lecture Notes in Computer Science, S. Auer, V. Bryl, and S. Tramp, Eds. Springer International Publishing, 2014, vol. 8661, pp. 90–107.
14. J. Helmich, J. Klımek, and M. Necasky, "Visualizing RDF data cubes using the linked data visualization model," in *The Semantic Web: ESWC 2014 Satellite Events*, ser. Lecture Notes in Computer Science, V. Presutti, E. Blomqvist, R. Troncy, H. Sack, I. Papadakis, and A. Tordai, Eds. Springer International Publishing, 2014, pp. 368–373.
15. Capadisli, S., Auer, S. Riedl, R.: Linked Statistical Data Analysis, ISWC SemStats (2013), http://csarven.ca/linked-statistical-data-analysis