Modeling Fiscal Data with the Data Cube Vocabulary

Jindřich Mynarz
Department of Information and Knowledge Engineering, University of Economics
W. Churchill Sq. 4
130 67 Prague, Czech Republic
jindrich.mynarz@vse.cz

Vojtěch Svátek
Department of Information and Knowledge Engineering, University of Economics
W. Churchill Sq. 4
130 67 Prague, Czech Republic
svatek@vse.cz

Sotirios Karampatakis
School of Mathematics
Aristotle University of Thessaloniki
Thessaloniki, Greece
and Open Knowledge Foundation Greece
Thessaloniki, Greece
sokaramp@auth.gr

Jakub Klímek
Department of Information and Knowledge Engineering, University of Economics
W. Churchill Sq. 4
130 67 Prague, Czech Republic
and CTU, Prague
Thákurova 9, Praha 6
Czech Republic
jakub.klimek@fit.cvut.cz

Charalampos Bratsas
School of Mathematics
Aristotle University of Thessaloniki
Thessaloniki, Greece
and Open Knowledge Foundation Greece
Thessaloniki, Greece
cbratsas@math.auth.gr

ABSTRACT

We present a fiscal data model based on the Data Cube Vocabulary, which we developed for the OpenBudgets.eu project. The model defines component properties out of which data structure definitions for concrete datasets can be composed. Based on initial usage experiments, simple validation constraints have been formulated.

CCS Concepts

•**Information systems** → **Resource Description Framework (RDF)**; •**Theory of computation** → *Data modeling;*

Keywords

Data Cube Vocabulary, SKOS, fiscal data, RDF

1. INTRODUCTION

Fiscal data is increasingly published as open data by various government institutions, from the EU to the municipality level. From the structural point of view they can be characterized as statistical data, consisting of numerical values (observations) indexed by discrete values of various dimensions. The dimensions typically refer to time (such as fiscal period), budget authority (municipality, regional government, ministry, etc.), thematic category (such as building or social service), spending type (e.g., salaries, subsidies, etc.), and a few others. The data can be modeled as multidimensional cubes and subsequently analyzed using data analytics tools: interactive visualization, data mining, etc.

SEMANTICS 2016: Posters and Demos Track
September 13-14, 2016, Leipzig, Germany

The Horizon 2020 project OpenBudgets.eu (2015-2017) aims to support different scenarios of exploiting fiscal data by journalists, corruption-fighting NGOs or local civic activists. While fiscal data alone only supports a limited scope of analytical tasks, they could benefit from being augmented with further data describing demographic, economic (e.g., GDP) or political (e.g., the ruling political party) features related to the values of dimensions in the data. This opens a way to efficient use of RDF technology, with the Data Cube Vocabulary (DCV) and Simple Knowledge Organization System (SKOS) RDF vocabularies playing a central role.

In the paper we share our experience from a practical use case in applying DCV to fiscal data. We first briefly describe both the underlying vocabularies and the newly developed OpenBudgets.org fiscal data model, conceived as a set of reusable DCV component properties that can be specialized for concrete datasets in the form of data structure definitions (DSDs). Compared to the existing fiscal data models, as reviewed in Klímek et al. [3], the data model is more generic and covers both spending and budget data. Next we report on initial usage experiments, carried out in both practical (project-internal) and educational (classroom) setting. Based on this experience, simple validation constraints have been formulated that make some assumptions about fiscal data testable.

2. DATA MODEL

DCV [1] is an RDF vocabulary designed for modeling multi-dimensional, in particular, statistical, data. It allows to assign quantitative values, *measures* (such as counts of physical objects or financial amounts), to combinations

of qualitative *dimensions* (usually related to time, space, 'theme' etc.), and be further specified by *attributes* (such as currency, for financial amounts). These three types of constructs are jointly called *components* and are modeled as RDF properties. The inventory of components used to model a particular *dataset* is called *data structure definition* (DSD), and the individual units of data coupling the values of components together are called *observations*.

DCV dimensions and (optionally) attributes have coded values, typically following code lists expressed as SKOS concept schemes. SKOS[1] is a vocabulary for structuring knowledge organization systems, potentially interrelated using thesaurus relationships such as 'broader', 'narrower' or 'related'.

The core data model of OpenBudgets.eu defines 20 components: 17 dimensions, 2 attributes and 1 measure, see Table 1. Some notions (`currency`, `taxesIncluded`) appear both as attributes and dimensions because they can be used either to qualify a measure or organize the measure in a data cube. Additionally we defined `obeu:OptionalProperty` as a subproperty of `qb:ComponentProperty` for representing optional properties, such as `location` (physical location affected by a payment) or `contract` (public contract for which a payment is made).

3. USAGE FEEDBACK

The next two subsections overview two ways of using the components for modeling real-world fiscal datasets: in an educational context and in the practical development activities of the OpenBudgets.eu project.

3.1 Feedback from educational use

Initially we have carried out small-scale usage testing in a classroom setting, in Winter 2015. The involved students were attending a Masters-degree course on linked data technology.[2] Five groups of students, having 2–3 members each, applied the core component collection each on a different fiscal dataset from either the EU or national (specifically, Czech, to avoid the language barrier) level, and covering both the budget and spending types. In parallel, they got acquainted with the structure and official documentation of the dataset, with the machine-readable version of the component collection[3] and the user documentation tailored for this task.[4]

The overall assignment, carried out stepwise, partly in the classroom and partly as homework, covered:

1. familiarization with the dataset in the CSV format (e.g., using a spreadsheet environment) as well as with its context at the original website from which it had been extracted
2. identification of relevant DCV components from the collection/s
3. assembly of a DSD from these components (in Turtle format)
4. manual assembly of a few sample observations (in Turtle format) conforming to this DSD as well as to a sample of the CSV dataset
5. bulk, automated transformation of the chosen dataset

In the first step, the students were assisted by a fiscal domain expert. In order to avoid the distraction of learning a complex ETL tool, the students did not use LinkedPipes ETL in the last step, but instead applied tarql[5] using its command line interface. The use of tarql did not require additional skills beyond SPARQL, which the students had already been accustomed to.

All five groups eventually produced RDF datasets corresponding to the original CSV files, however, some recurring errors have been identified:

- In the file containing the DSD and the reused components, these were not explicitly connected using the `qb:component` property. Similarly, in the file containing both the dataset entity (`qb:DataSet`) and the respective observations, the observation entities were not connected to the dataset entity using the `qb:dataSet` predicate. This indicates that users may underestimate the importance of explicit linking and the fact that RDF triples occurring together in a file or graph may not necessarily form a compact structure.
- Some components have been *instantiated* by observations. This indicates that the philosophy of the DCV vocabulary, where dimensions, measures and attributes are syntactic *predicates*, is not intuitive at the first sight and care must be paid to properly explaining it on examples.
- New components have been coined independently while they could have been subordinated under existing ones. For example, a new component for invoice number has been suggested, which could have become subproperty of *obeu-dimension:accountingRecord*.
- A mandatory (and obvious, for fiscal data) attribute for *currency* of financial amounts was often missing.
- Sometimes new, custom components have been created in the namespace of the OpenBudgets.eu components. This kind of "namespace hijacking" is a common bad practice unspecific to the respective domain, however, it might be worth emphasizing even in specialized usage guidelines.

Other kinds of mistakes were rather isolated cases; for example, the (numerical float) value of measure was transformed to a 'dimension' in one case, indicating deep misunderstanding of the DCV model by the student/s.

3.2 Feedback from project-level use

Next we aimed to leverage on practical ETL activities in the OpenBudgets.eu project, which, analogously to the students' case, produced RDF versions of fiscal datasets conforming to the core fiscal data model. A sample set of fiscal datasets was selected for the test phase, which was coming from different levels of administrative hierarchy, ranging from the budget of the EU to municipalities' budgets

[1] https://www.w3.org/2004/02/skos/

[2] The course description is available from http://isis.vse.cz/katalog/syllabus.pl?predmet=100613;lang=en.

[3] Technically divided into the 'budget' and 'spending' collection: http://openbudgets.eu/assets/deliverables/D1.2.pdf and http://openbudgets.eu/assets/deliverables/D1.3.pdf, respectively.

[4] http://openbudgets.eu/assets/deliverables/D1.4.pdf

[5] https://github.com/tarql

Table 1: Core component properties of the OpenBudgets.eu data model

Property name	Type	Description
`fiscalPeriod`; `fiscalYear`	dim.	The period of time, or, specifically, year (as subproperty) reflected in financial statements.
`date`	dim.	Date when expense was paid or revenue received.
`operationCharacter`	dim.	Distinguishes among expenditure and revenue.
`budgetLine`	dim.	Budget line from which the payment draws funds.
`budgetPhase`	dim.	Major event or stage in the budget cycle, e.g., Draft.
`paymentPhase`	dim.	Phase of the payment, e.g., Certified.

`classification`
`administrativeClassification`
`economicClassification`
`functionalClassification`
`programmeClassification`
	dim.	System of abstract properties, for which subproperties with concrete code lists should be created. They finer-grained (abstract) subproperties of `classification` characterize, in turn: the entity (e.g., department) responsible for managing the funds; type of expenditure incurred or source of revenues; classification by general government sector and purpose; budget lines grouping by common objective.

`currency`	dim.	Currency of a financial amount.
`taxesIncluded`	dim.	Indicates whether the reported amount includes taxes.
`organization`	dim.	Usually the the owner of the dataset (same throughout the dataset).
`partner`	dim.	The entity to which the payment was made or from which the revenue was collected.
`project`	dim.	Project associated with the payment.
`accountingRecord`	dim.	Link to an accounting record associated with expenditure or revenue.
`currency`	attr.	Currency of a financial amount.
`taxesIncluded`	attr.	Indicates whether the reported amount includes taxes.
`contract`	opt.	Public contract for which a payment is made.
`location`	opt.	Physical location affected by a payment.
`amount`	meas.	Monetary amount.

in EU member states. A short description of these samples is shown in Table 2. In contrast to the educational scenario, the LinkedPipes ETL (LP-ETL) [5] was used for transformation. The work has been carried out by several (mostly junior – Ph.D. level) researchers. Raw data, transformed data, and pipelines for transforming the datasets can be found on Github.[6]

Region	Datasets	Form	Years	Triples
EU	3	XML, CSV	2014	150k
Spain	11	CSV	2006-2016	445k
Czech R.	3	XLSX	2014	273k
Greece	66	CSV, XLSX	2002-2016	9.2M

Table 2: Sample datasets transformed

An example is the transformation of budget of Municipality of Thessaloniki, Greece. The raw data was in the CSV format. The budget of each year was described in two files, one for expenditures and one for revenues. There were three obvious dimensions corresponding to the year, organization and operation character, which we could immediately map on the data model. The measure (amount) was further associated with a currency attribute. The amounts were available for five different phases of budgets. OBEU data model includes only four phases of budget; however, the open nature of the data model allows its extension, which covered the additional phase. Values of some dimensions, such as administrative or economic classification, were in textual form in the original CSV. We however had code lists for them, created using SKOS in a previous stage. We thus lifted the value strings to the respective URIs using SPARQL. In total, each measure was eventually described using seven dimensions and an attribute.

The scope of the usage issues encountered in the resulting models, in this testing phase, overlapped with those identified in the classroom scenario, for example the instantiation of component properties or core namespace hijacking. They have been fixed before proceeding to later phases of processing the data, such as the application of visualization and data mining tools, which are the main focus of the current phase of OpenBudgets.eu.

[6]https://github.com/openbudgets/datasets

4. INTEGRITY CONSTRAINT DESIGN

In order to keep track of and systematically fix the common errors made by users of the OpenBudgets.eu data model we developed several integrity constraints that automate error detection. We used the errors recurrent in the student's work described in Section 3.1 and in the ETL work for the OpenBudgets.eu[7] as the source of requirements for the integrity constraints. We restricted the errors to those that can be detected automatically and are frequent enough to warrant the development of constraints detecting the errors. The integrity constraints are formalized as SPARQL 1.1 `CONSTRUCT` queries that match patterns of erroneous uses of the OpenBudgets.eu data model. Apart from these patterns the constraints leverage the background knowledge encoded in the Data Cube Vocabulary and the OpenBudgets.eu data model. The queries produce descriptions of the detected errors represented using the SPIN Modelling Vocabulary[8] as RDF data. Thanks to the machine-readable format of the errors, it is possible to transform them to the desired output format. In our case the errors are templated into better readable reports in HTML. In total, we implemented six integrity constraints that test assumptions concerning mostly the datasets' DSDs.

1. **Hijacked core namespace:** Tests if the validated dataset defines a term in the namespace of the core OpenBudgets.eu data model (i.e. `http://data.openbudgets.eu/ontology/`) that is undefined in the core data model. New dataset-specific terms must be defined in a different namespace.

2. **Missing mandatory component property:** The OpenBudgets.eu declares several properties (`obeu-attribute:currency`, `obeu-dimension:fiscalPeriod`, `obeu-dimension:operationCharacter`, `obeu-dimension:organization`, and `obeu-measure:amount`) as mandatory. This rule tests if these properties or their subproperties are provided in the validated dataset.

3. **Property instantiation:** Verifies that the validated instantiates only RDF classes and not properties. For example, properties may be erroneously instantiated due to typing errors (e.g., `qb:DataSet` vs. `qb:dataSet`).

4. **Redefinition of component property's code list:** If a core OpenBudgets.eu component property needs a custom code list in a specific dataset, a subproperty should be derived from the property instead of reusing it directly. This constraints checks if the validated dataset does not redefine the code lists to be used with the core component properties.

5. **Use of abstract property:** The OpenBudgets.eu declares several properties as abstract (e.g., `obeu-dimension:classification`), so that they should not be directly reused and instead their subproperties should be derived. This rule tests if the validated dataset does not reuse abstract properties.

6. **Wrong character case in DCV:** This constraint detects if the validated dataset contains non-existent terms from the DCV namespace (i.e. `http://purl.org/linked-data/cube#`) that differ only in character case from the existing DCV terms. Apart from reporting the non-existent terms, the constraint suggests existing DCV terms that may replace the non-existent ones.

The implemented integrity constraints are described in Klímek et al. [4] and are available as open source.[9]

5. CONCLUSIONS

The paper presents the experience from applying the Data Cube Vocabulary on modeling data from the fiscal domain, with special focus on recurring errors that could also be generalized to other domains. It indicates that users tend to make similar kinds of modeling errors even if working in different contexts, such as project development and educational assignment. The issues are addressed both at the level of documentation [2] and operationally in the form of specifically tailored integrity constraints. The described data model provides a foundation for ongoing data analyses and visualizations in the OpenBudgets.eu project.

Acknowledgements: The presented research has been supported by the H2020 project no. 645833 (OpenBudgets.eu).

6. REFERENCES

[1] R. Cyganiak and D. Reynolds. The RDF Data Cube Vocabulary. W3C recommendation, W3C, 2014.

[2] M. Dudáš, L. Horáková, J. Klímek, J. Kučera, J. Mynarz, L. Sedmihradská, J. Zbranek, and T. Dong. Openbudgets.eu deliverable 1.4: User documentation. Technical report, 2016.

[3] J. Klímek, J. Kučera, J. Mynarz, L. Sedmihradská, and J. Zbranek. Deliverable 1.1: Survey of modelling public spending data & knowledge elicitation report. Technical report, 2015.

[4] J. Klímek, J. Mynarz, P. Škoda, J. Zbranek, and V. Zeman. Openbudgets.eu deliverable 2.2: Data optimisation, enrichment, and preparation for analysis. Technical report, 2016.

[5] J. Klímek, P. Škoda, and M. Nečaský. LinkedPipes ETL: Evolved linked data preparation. In *The Semantic Web: ESWC 2016 Satellite Events - ESWC 2016 Satellite Events, Anissaras, Crete, Greece, May 29-June 2, 2016, Revised Selected Papers, to appear*, 2016.

[7] https://github.com/openbudgets/datasets

[8] http://spinrdf.org/spin.html

[9] https://github.com/openbudgets/pipeline-fragments/tree/master/obeu/obeu-model-integrity-constraints