How can Data Catalog Vocabulary (DCAT) be used to address the needs of databases?

Discussion Paper No. 5, November 2016, Joined-up Data Standards Project

Beata Lisowska

Data Scientist, Development Initiatives

Contents

Executive summary

The Development Data Hub is an example of one of many visualisation tools available on the web that aim to make data more accessible, easy to disaggregate and comparable in an intuitive way. As more such data tools are becoming available and as the World Wide Web Consortium (W3C) argues that data published on the web should always be coupled with metadata, this paper tests how easy it is to use one of the most widely used metadata standards, Data Catalog Vocabulary (DCAT), for such a purpose.

DCAT is a well-documented, flexible and practical metadata standard that is grounded in the solid foundations of Dublin Core. DCAT is an elegant standard to use for datasets published by a single source; however, it became more complicated when applied to the Development Data Hub or its underlying Data Warehouse.

This paper aims to find a practical approach to applying the DCAT standard to satisfy the needs of both a portal that provides dynamic visualisations and a database that provides the data to drive them. As we learn, this is a complex and tricky task. It would appear that a single instance of DCAT cannot handle the complexity of the data journey from its source to the final visual representation.

Why do we need DCAT to handle this problem? The transitions from data source to data warehouse to data series through datamart and finally to dataset cannot only be comprehensible from a human point of view. The logic needs to be encoded in a machine-readable way so that machines can point a data-user back to the original source of the transformed data and allow the discoverability and searchability of related datasets. This is in its heart a joined-up (meta)data standards problem.

Joined-up Data Standards will present this discussion paper at the W3C and VRE4EIC-organised workshop on ‘Smart Descriptions & Smarter Vocabularies’ in Amsterdam on 30–1 December 2016.

Introduction

The Development Data Hub is Development Initiatives (DI)'s flagship online resource for the discovery of financial and resource flow data. The tool brings together multiple datasets and through interactive visualisations allows the user to understand how resources designated for development and poverty eradication are spent.

With the click of a button it is possible to visualise where the poorest 20% of people are in the world or unbundle official development assistance (ODA) data by sector, recipient, donor or financial instrument. The Data Hub allows the user to compare resource flows across sectors, countries and channels. How? The tool collates data from a variety of sources and combines it to produce dynamic visualisations. These complex visualisations are accompanied by dynamically created downloadable datasets.

As with the vast majority of online databases and data portals, the Data Hub would benefit from an additional machine-readable layer of context that could direct the user to the relevant information on the data sources behind the visualisations. Since the Data Hub is dynamic, so should be the metadata that provides the information on when the data was published, who published it, when it was last updated, where it can be downloaded and how it was generated.

At the moment this information can be found in the static, human-readable methodology section of the resource page, where all the sources are presented in a list.

The World Wide Web Consortium (W3C)'s 'Data on the Web Best Practices' names providing metadata as a 'fundamental requirement when publishing data on the Web' and advises that the metadata should be provided in both human- and machine-readable format. Machine-readable format is a crucial requirement, as this allows computer applications to process it.

The benefits of machine-readable metadata are manifold but critically, they:

- Increase the discoverability of datasets
- Make it easier for users to search for what they are looking for across multiple platforms.

A range of metadata standards already exists to help data producers publish metadata in a human and machine-readable format. One of the most prominent, developed by the W3C, is the Data Catalog Vocabulary (DCAT) that has gained popularity largely due to its flexible design. It is employed by all the major software engines used for open data portals, such as CKAN, DKAN, Socrata and OpenDataSoft.

This standard is growing in popularity due to the ease with which it can be adapted to meet the challenges of data publishers. For example, the European Commission's DCAT application profile (DCAT-AP) is used to publish public sector datasets in Europe.

Development Initiatives is a leading curator of value-added joined-up data and is committed to improving the interoperability of all development-related and humanitarian data. Can DI adhere to the W3C principle of publishing metadata in a human and machine-readable format? Can it adopt DCAT as the metadata standard for both its Development Data Hub and the underlying databases that join-up data collected from a wide range of sources? The aim of this paper is to explore how this could work.

The challenge

DI discovers and collects empirical and processed data from a variety of sources. These range from global datasets – maintained by institutions such as the World Bank, International Monetary Fund, UN Statistics Division and Organisation for Economic Co-operation and Development (OECD) – to national statistics and new collections of emerging data manually curated by its analysts.

The relevant data from these sources are loaded into the Development Data Warehouse, which uses a collection of generic data models to integrate, where possible, data from these disparate sources into standardised, joined-up database schema.

This is used to create datamarts containing purpose-built joined-up datasets, each potentially containing data derived from a range of sources that drive context-specific visualisations designed by DI's analysts for the Development Data Hub.

Figure 1 presents a simplified architecture of the system described here.

This growing complex of interconnected data serving a range of digital products poses three problems.

1. Firstly, how does DI, and how do Data Hub users, keep track of what data is available and whether it is up to date?
2. Secondly, the intellectual credibility of DI's work depends on metadata that explains the provenance and methodology of its analysts' calculations. In paper reports you can find this in small-print footnotes, but how do you replicate this for dynamically produced datasets that have been generated from an interactive visualisation?
3. Thirdly, the joined-up 'raw' databases in the warehouse will, in future, become a public good with an open API. How will third-party developers wanting to make use of this repository access the metadata they will need to accompany the data they extract?

Is DCAT the answer to these problems?

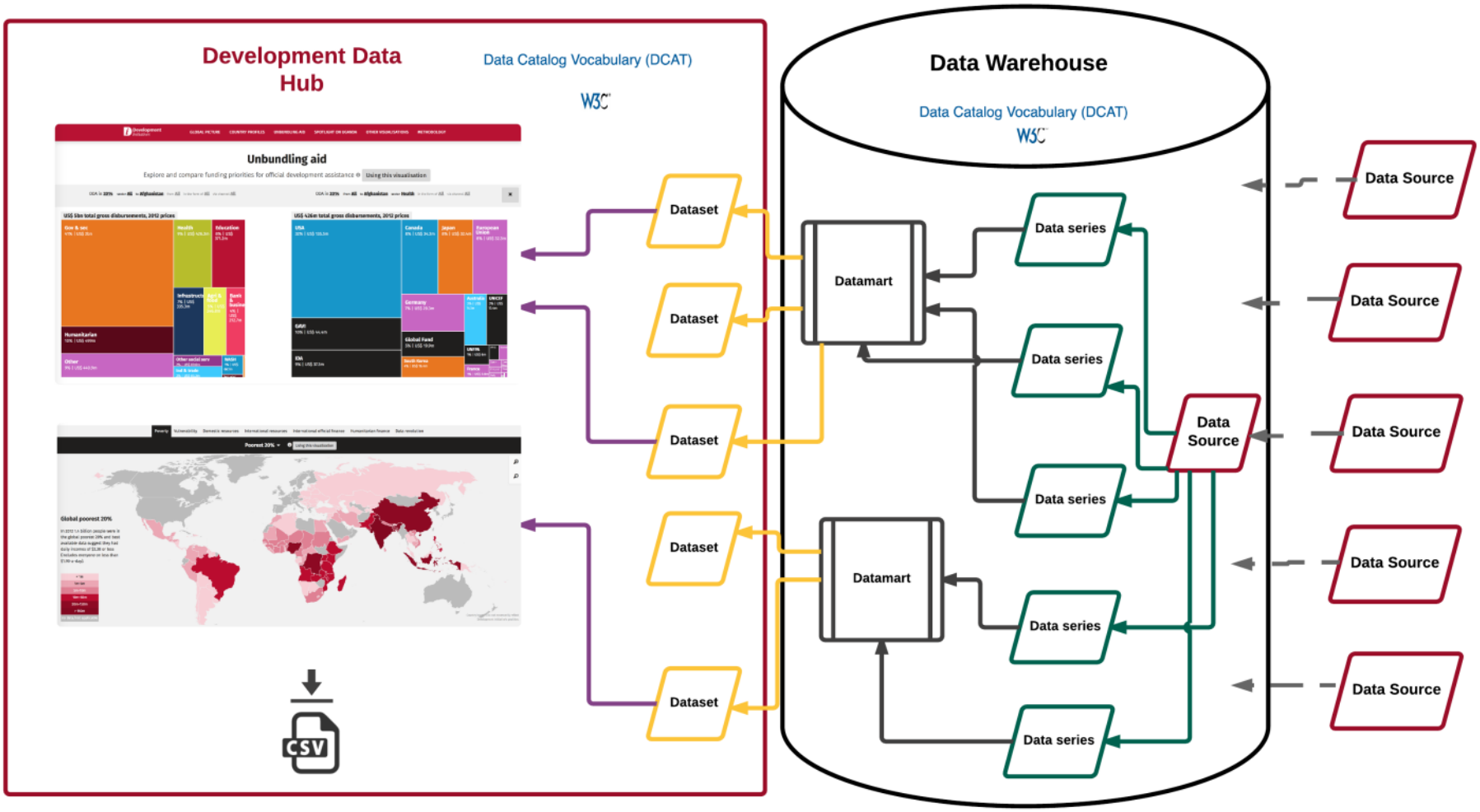

Figure 1: An overview of the flow of data through DI's Data Warehouse and Development Data Hub

Step 1: A catalog for the Data Hub

One of the ways to use the DCAT for the Development Data Hub is to treat the 'front end', the Data Hub, as a catalog of data in itself (Figure 1, red container). This way each 'dataset' that builds a visualisation can be described by dcat:Dataset class (Figure 2) and benefit from a range of properties that can describe the what, the who, the why and how of a given dataset.

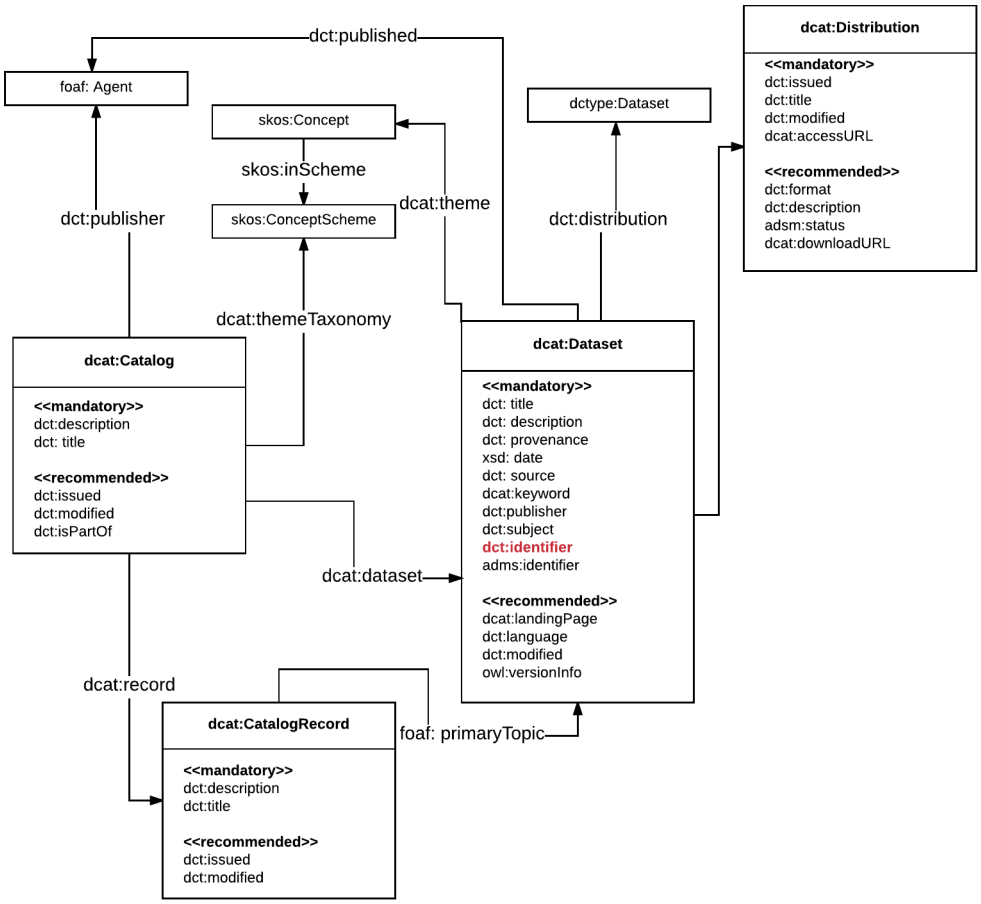

Figure 2: The model of DCAT that could theoretically be used to provide metadata for data sources in the Development Data Hub

Not every dataset is used for each given visualisation (depending on user selections) so if DCAT is used to catalogue **all** the datasets stored in the datamarts for the Development Data Hub, how can DI select the relevant subsets? One approach could be to use the property of the dataset called dct:Identifier (see Table 3). The datasets could be called by their unique dct:Identifier and matched to the corresponding visualisation (Figure 3).

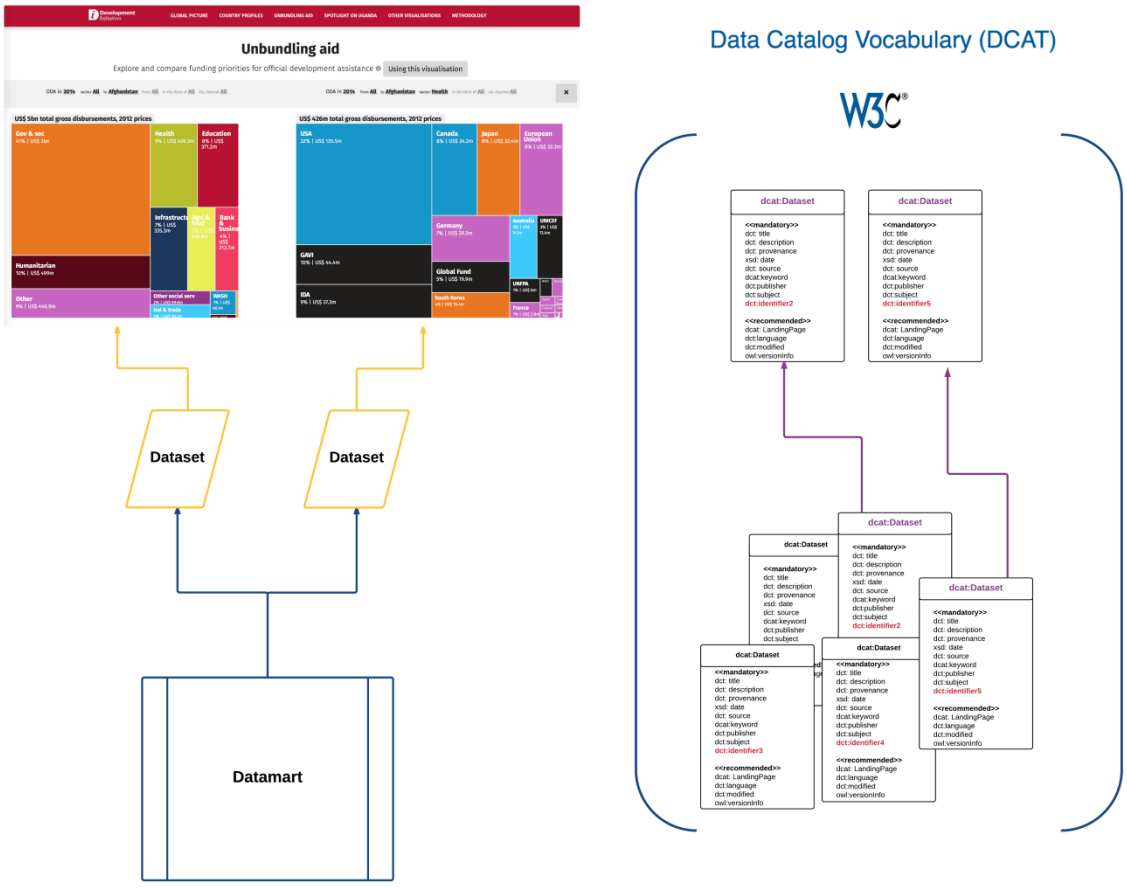

Figure 3: The theoretical proposal for retrieving the correct DCAT class datasets using dct:Identifier

Step 2: A catalog for the Data Warehouse

Step 1 applies the DCAT standard in a traditional use-case scenario where datasets and catalogs are used in an intuitive way. However, 'Step 1…' does not describe the Data Warehouse, which is the real heart of the Development Data Hub. For instance, the Data Hub's datasets are dynamically updated (quarterly for World Bank and OECD data) whenever more current data is discovered and loaded into the Data Warehouse.

The Data Warehouse is where the external data sources are curated into data series and disseminated to datamarts and finally to datasets that are used by the Data Hub's visualisations. As far as we are aware, DCAT has never before been used to capture the complex journey of data through a data warehouse (black container in Figure 1).

If the Data Warehouse is treated as a DCAT model then each 'data source' can be viewed as a catalog. Following this logic, the class dcat:Dataset corresponds, in the Data Warehouse, to the data series. The proposed solution broadly deals with the needs of the Data Warehouse and most importantly can map tit using dct:Source (Table 3); the original data source could be linked to the correct place, such as OECD DAC data or World Bank World Development Indicators.

The bigger challenge

DCAT defines a dataset flexibly as a ‘collection of data, published or curated by a single agent, and available for access or download in one or more formats’. For the Development Data Hub, this means that a dataset as defined by DCAT could either be a data series such as a World Development Indicator on ‘Literacy rate, adult total (% of people aged 15 and above)’ or a compound dataset produced by the Data Warehouse to build a dynamic visualisation.

The above solution provides a two-step (1 and 2) answer to the question posed in this paper. The challenge lies in merging these two parts into a single, coherent whole: using DCAT to describe the relationships between and across the Data Warehouse and the Data Hub.

Since both the Data Hub and the Data Warehouse can maintain their own DCAT models, then making links between them should not, theoretically at least, be a problem. The two DCAT models could be joined-up through the use of dct:Source. This is the same approach as proposed for the Data Warehouse to indicate the original source of data (such as World Development Indicators).

Conclusions

The exponential number of published datasets creates an ever-increasing and more complex data environment for a user to navigate. Data on its own, without contextual information or links to other similar sources, often proves difficult to analyse or interpret. However, this need not be the case. As this paper touches on, the data community is beginning to embrace the added value of metadata and the power of machine-readable metadata formats. The datasets that are coupled with metadata standards are searchable, discoverable, contextual and essential for any data user. Most encouragingly, the standards are already out there, ready to be used.

However, as this paper shows, the practical application of metadata standards, such as DCAT, can provide a challenge if applied to complex systems. Even though DCAT provides an elegant, clean and flexible standard for publishing metadata, in its basic form, it cannot handle the complexity of both a data warehouse and a dynamic tool such as the Development Data Hub (depicted by Figure 1) in one instance.

This is a two-fold problem. Firstly, the majority of data publishers are secondary data producers, which means that the journey of a single data point from its origin to its final destination is sometimes not clear to a data user. Metadata should provide a machine-readable map to make this information available and traceable across platforms and data producers. This can be achieved through joining-up machine-readable links between standards and DCAT is equipped to provide this through its many properties. This is one of the reasons why DCAT as a standard is favoured by open data portals.

This brings us to the second problem: data is stored in data warehouses that can be complex. A data warehouse drives the data published on the web and as such should also be comprehensively described by a metadata standard. This is, as this paper shows, a rather tricky endeavour.

Appendix

Fundamental principles of the Data Catalog Vocabulary (DCAT)

DCAT is a resource description framework (RDF) vocabulary that has risen in popularity due to its flexibility and intuitive design. As a result, a variety of vocabulary profiles or implementations were created, such as: DCAT-application profile (DCAT-AP), Asset Description Metadata Schema (ADMS) and Project Open Data Metadata Schema. These profiles exist as proof of how flexible DCAT is and that, as such, it can be used to provide metadata to different types of data.

DCAT is grounded in the solid foundations of Dublin Core, SKOS (Simple Knowledge Organization System), and FOAF (Friend of a Friend). These make it, in principle, possible to cross-map different DCAT implementations to one another, ensuring the interoperability between them.

Namespaces

To ensure compliance with the elemental DCAT model, other DCAT-based standards reuse existing 'namespaces' (Table 1). This ensures that this DCAT instance is interoperable and comparable with other DCAT profiles. These namespaces are, in simple terms, languages used by the metadata standards.

Table 1: Namespace reference for the classes and properties adopted by DCAT

Prefix	Namespace
adms	adms: http://www.w3.org/ns/adms#
dcat	dcat: http://www.w3.org/ns/dcat#
dct	dct: http://purl.org/dc/terms/
foaf	foaf: http://xmlns.com/foaf/0.1/
owl	owl: http://www.w3.org/2002/07/owl#
rdfs	rdfs: http://www.w3.org/2000/01/rdf-schema#
schema	schema: http://schema.org/
skos	skos: http://www.w3.org/2004/02/skos/core#
spdx	spdx: http://spdx.org/rdf/terms#
xsd	xsd: http://www.w3.org/2001/XMLSchema#
vcard	vcard: http://www.w3.org/2006/vcard/ns#

DCAT classes

DCAT defines RDF schema using classes such as: dcat:Catalog, dcat:Dataset, dcat:Distribution (Figure 4). These represent in order: catalog, dataset in a catalog, and accessible form of a dataset. Where catalog is defined as the repository of datasets, dataset refers to data published in a given dataset, and distribution describes the physical format of a dataset. Descriptions of all DCAT classes can be found in Table 2.

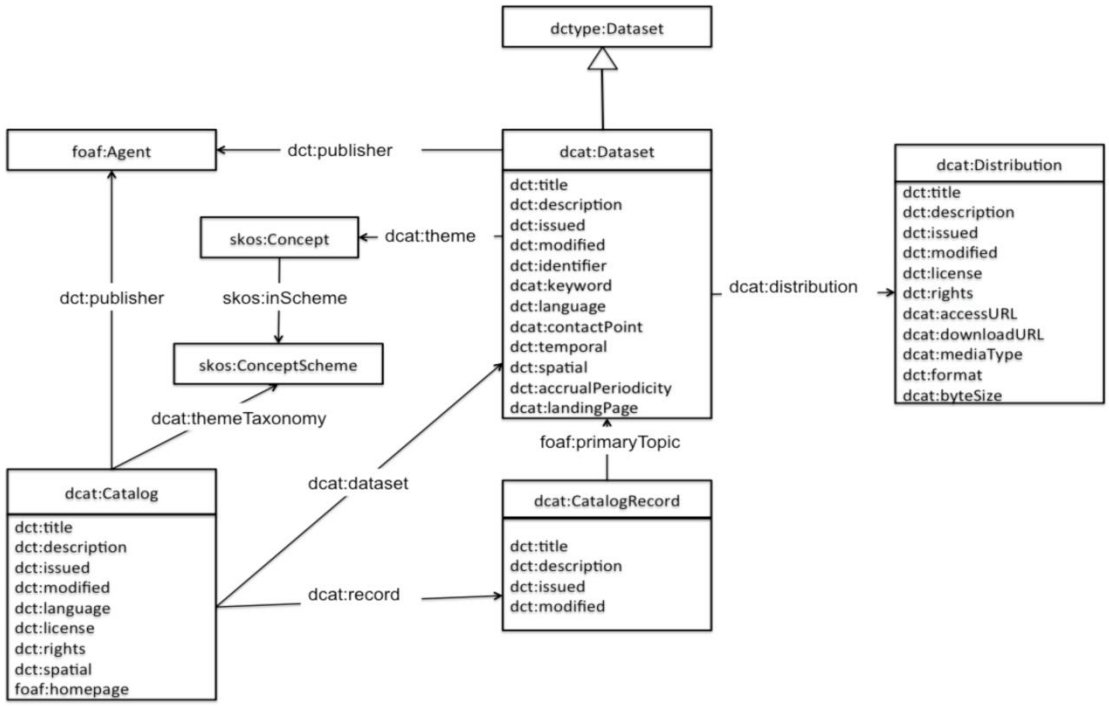

Figure 4: The basic DCAT model class and properties overview

Table 2: Class uniform resource identifier (URI), description and reference for DCAT model

Class URI	Description	Reference
dcat:Catalog	Catalog or repository that hosts the datasets	http://www.w3.org/TR/2013/WD-vocab-dcat-20130312/#class-catalog
dcat:Dataset	Data published in a dataset	http://www.w3.org/TR/2013/WD-vocab-dcat-20130312/#class-dataset
foaf:Agent	Organisation or a person that publishes data or is associated with the dataset	http://www.w3.org/TR/vocab-org/
skos:Concept	Subject of a dataset	http://www.w3.org/TR/2013/WD-vocab-dcat-20130312/ - class-category-and-category-scheme
dcat:CatalogRecord	Description of a dataset's entry in the catalog	http://www.w3.org/TR/2013/WD-vocab-dcat-20130312/#class-catalog-record
dcat:Distribution	Physical format of a dataset	http://www.w3.org/TR/2013/WD-vocab-dcat-20130312/#class-distribution
skos:ConceptScheme	Scheme where the concept/subject of a dataset is defined in a broader sense	http://www.w3.org/TR/2013/WD-vocab-dcat-20130312/#class-category-and-category-scheme

Class URI skos:ConceptScheme is the least intuitive class for a non-technical user; however, it is one of the most descriptive. This class provides the hierarchical representation of how the catalog is divided by subjects and grouped by categories. In a machine-readable form this is achieved using a SKOS. To use a library

Properties of classes

Each class can be further defined by a range of properties that in essence are the fields where metadata can be found. The properties answer questions (referring to class dataset) such as what sort of data does the dataset contain (dcat:Description), when was it issued (dct:Issued), and how often is it updated?

Class distribution refers to the format in which the dataset can be accessed; usually it contains such properties as (dct:AccessURL), format (dct:Format), size of the file (dct:bytSize), and license and rights (dct:License and dct:Rights).

In the standard DCAT implementation, each of class properties should be used for any given dataset. The full list of properties per class can be traced on Figure 4.

DCAT-AP significantly increased the number of properties per class to fit the metadata needs in the context of Action 1.1 of the European Commission's Interoperability Solutions for European Public Administrations (ISA) programme: 'Improving semantic interoperability in European eGovernment systems'.

DCAT was designed with a focus on the metadata fields that are available in all or most catalogs. However, depending on the metadata needs, this subset of properties is either too limited or not descriptive enough. DCAT-AP addresses this problem by grouping class properties as mandatory, recommended or optional.

Dataset properties

From a data science point of view, the elemental set of metadata information that should always accompany a dataset is:

- Who published the data?
- When was it published?
- Where can you access the original files?
- Do you have the rights to 'play and publish'?
- Why was this data collected?
- How was it collected?
- What format is the data in?

The DCAT-AP introduced the concept for classification of the DCAT class properties according to how crucial it is for the user of the data. The classification is divided into three tiers: mandatory, recommended and optional.

Mandatory denotes properties that must always be included; recommended suggests that the information should be included; and optional indicates that the property may be included but the sender of the information is not obliged to provide it. For the purpose of this theoretical exercise, the datasets properties for the Development Data Hub DCAT are divided into recommended and mandatory properties.

Table 3: Properties of class dataset for DCAT (recommended properties fields are highlighted in blue and mandatory in red)

Property	URI	Description
Description	**dct:Description**	This property contains a free-text account of the dataset. This property can be repeated for parallel language versions of the description.
Title	**dct:Title**	This property contains a name given to the dataset. This property can be repeated for parallel language versions of the name.
Dataset distribution	**dcat:Distribution**	This property links the dataset to an available distribution.
Keyword/tag	**dcat:Keyword**	This property contains a keyword or tag describing the dataset.
Publisher	**dct:Publisher**	This property refers to an entity (organisation) responsible for making the dataset available.
Theme/category	**dcat:Theme, subproperty of dct:subject**	This property refers to a category of the dataset. A dataset may be associated with multiple themes.
Identifier	**dct:Identifier**	This property contains the main identifier for the dataset, eg the URL or other unique identifier in the context of the catalog.
Other identifier	**adms:Identifier**	This property refers to a secondary identifier of the dataset, such as MAST/ADS], DataCite], DOI, EZID or W3ID.
Provenance	**dct:Provenance**	This property contains a statement about the lineage of a dataset.
Source	**dct:Source**	This property refers to a related dataset from which the described dataset is derived.
Landing page	**dcat:Landingpage**	This property refers to a web page that provides access to the dataset, its distributions and/or additional information. It is intended to point to a landing page at the original data provider, not to a page on a site of a third party, such as an aggregator.
Language	**dct:Language**	This property refers to the language of the dataset. It can be repeated if there are multiple languages in the dataset.
Update/ modification date	**dct:Modified**	This property contains the most recent date on which the dataset was changed or modified.
Version	**owl:Versioninfo**	This property contains a version number or other version designation of the dataset.