International Journal of Geo-Information

MDPI

Article

Semantic Specification of Data Types for a World of Open Data

Xiaogang Ma *, John S. Erickson, Stephan Zednik, Patrick West and Peter Fox

Tetherless World Constellation, Rensselaer Polytechnic Institute, 110 8th Street, Troy, NY 12180, USA; erickj4@rpi.edu (J.S.E.); zednis2@rpi.edu (S.Z.); westp@rpi.edu (P.W.); pfox@cs.rpi.edu (P.F.)

* Correspondence: max7@rpi.edu; Tel.: +1-518-276-4384; Fax: +1-518-276-4464

Academic Editors: Constanze Curdt, Christian Willmes, Georg Bareth and Wolfgang Kainz

Received: 10 December 2015; Accepted: 8 March 2016; Published: 16 March 2016

Abstract: Data interoperability is an ongoing challenge for global open data initiatives. The machine-readable specification of data types for datasets will help address interoperability issues. Data types have typically been at the syntactical level such as integer, float and string, *etc.* in programming languages. The work presented in this paper is a model design for the semantic specification of data types, such as a topographic map. The work was conducted in the context of the Semantic Web. The model differentiates the semantic data type from the basic data type. The former are instances (e.g., topographic map) of a specific data type class that is defined in the developed model. The latter are classes (e.g., Image) of resource types in existing ontologies. A data resource is an instance of a basic data type and is tagged with one or more specific data types. The implementation of the model is given within an existing production data portal that enables one to register specific data types and use them to annotate data resources. Data users can obtain explicating assumptions or information inherent in a dataset through the specific data types of that dataset. The machine-readable information of specific data types also paves the way for further studies, such as dataset recommendation.

Keywords: semantics; ontology; persistent identifier; linked data; faceted browser

1. Introduction

Global open data initiatives have received support from both public and private sector organizations in recent years. Changes can be seen in government policies, funding agency requirements, community guidelines, and technologies and in facilities for data management, curation, and sharing. The efforts on mechanisms of data publication [1], data cataloging [2], data citation [3], and alternative metrics [4] are incubating a new socio-technical system that promotes both the culture and the practice of open data. Within such a system, data are going to be shared and reused across the boundaries of nations, sectors, disciplines, repositories, and formats, as well as between levels of details. Data interoperability arises as a major challenge in those cross-boundary activities, which poses requirements for methods and technologies to make data discoverable, accessible, decodable, understandable, and usable [5].

The motivation of the research presented in this paper is to promote the decodability, understandability, and usability of data. Consider the scenario of a researcher wishing to use scientific data from an open data repository. Prior to retrieving a file from the data repository and using it, they will need to know the format, structure, parameters, and meaning of the data, and perhaps also the tools and services that can be used to process the data. In the world of open data, the researcher often receives no direct support or help from the data producers, which indicates that the metadata of the retrieved data may be the only source to obtain the information needed. Among the various metadata elements available, such as those in the Dublin Core Metadata Elements [6] and the DataCite

Metadata Schema [7], the elements describing data types are the most relevant ones for providing such information.

Data typing has been a research topic in computer programming for decades, whereby a data type is regarded as a collection of computational entities that share common properties. Data types in programming languages support three main uses: (1) naming and organizing concepts; (2) coordinating consistent interpretation of bit sequences in computer memory; and (3) providing information about data to the compiler [8,9]. There are primitive data types (e.g., integer, float, character, string, and Boolean), composite data types (e.g., array, union, set, and object), abstract data types (e.g., queue, stack, tree, and graph), as well as data types derived from the above types, such as utility types that address specific real-world uses. Knowledge of those data types can provide part of the information a researcher needs to work with the retrieved data, but are insufficient to fully address the requirement of understanding the data. For example, a researcher retrieves a table and knows that it is about thermodynamics of a chemical by reading the table name. The researcher reads words and numbers in the table but is not able to understand the meaning of those records because there is only an acronym in the title of each column, without further definitions. Moreover, the relationships between those columns are not clear to the researcher. Can we extend the content coverage of data types so that they can present an unambiguous, useful model of what the data represent? Under this scope, the specification of data types will be able to help the researcher understand the meaning of the data in a given science context.

The aim of this paper is to present our work of a conceptual model for the semantic specification of data types, as well as the implementation of the model in an existing production data portal for a decadal international science program: the Deep Carbon Observatory [10]. In this work, we regard a data type as the representation of particular qualities or features that a group of datasets shares, such as thermodynamics of chemicals and minerals, volcanic gas composition, or geologic contexts. Our model allows people to add domain specific meanings to a data type, to register the data type as an object in a data portal, and to annotate a dataset by associating it with one or more data types. Each registered data type has a unique identifier that is resolvable on the Web, and the information describing the data types is machine-readable and is accessible on the Web. The data type model adds new features to the data portal and enables better data curation and efficient data reuse. The remainder of the paper is organized as follows: Section 2 introduces the context of this work (*i.e.*, the Semantic Web) and details of our model design. Section 3 describes the implementation of the model in a data portal and the new functions created for the portal. Section 4 compares this work with relevant studies and discusses directions for future work. Finally, in Section 5, we provide a concluding discussion of the work presented in this paper.

2. Model Design

2.1. Semantic Web and Linked Open Data

The context of this work is the Semantic Web, which extends the core principles of the World Wide Web to make the meaning of data machine-readable [11]. Where information services on the original Web stopped at the text level and fell short of well-organized knowledge structures for concepts mentioned in the text, the Semantic Web encourages structure and meaning by enabling the development and use of ontologies. An ontology is the formal specification of a shared conceptualization of a domain [12]. Encoded in languages such as the Web Ontology Language (OWL), each ontology includes a list of concepts and a group of interrelationships between those concepts. The fundamental data structure of ontologies is the Resource Description Framework (RDF), which uses a triple form "Subject, Predicate, Object." For example, the triple "vivo:Dataset rdf:type owl:Class" asserts that vivo:Dataset is a class. The "vivo," "rdf," and "owl" here are namespace prefixes of ontologies or schemas. Each prefix is an abbreviation of the corresponding namespace identifier, *i.e.*, a Uniform Resource Identifier (URI). For example, "owl" represents the URI of the OWL

2 Schema Vocabulary [13]. Each triple is a detailed assertion that refines the definition of a concept or a relationship. Each subject, predicate, and object has a unique URI, and each URI can be dereferenced (*i.e.*, looked up) on the Web through the Hypertext Transfer Protocol (HTTP).

Ontologies provide the conceptual structure for data encoded and exchanged via the Semantic Web. Because ontologies are also encoded in the form of triples, the ontologies used in a given dataset may be loaded into the RDF database (a "triple store") in which the dataset is maintained. These properties provide certain advantages, including making data integration inherently easy. For example, the triple "dco:data_001 rdf:type vivo:Dataset" asserts that dco:data_001 is an instance of the class vivo:Dataset. In a query, if one sets a condition to find all instances of vivo:Dataset, dco:data_001 will be a record in the query result. It can be seen that, by following a best practice of ontology re-use, open data is more easily shared and reused within and among domains. URIs, HTTP, structured data such as those encoded in RDF, and links between them through URI-based names form the four core principles of Linked Data, of the foundation for practical publication and use of data in the Semantic Web [14,15]. Berners-Lee [15] further defined Linked Open Data as Linked Data released under an open license, and envisioned a five-star deployment scheme for Linked Open Data, following a five-step sequence: (1) on the web; (2) machine-readable data; (3) non-proprietary format; (4) RDF standards; and (5) linked RDF.

2.2. Research Approach and Implementation Methodology

The designed conceptual model for data type was deployed in the data portal of the Deep Carbon Observatory (DCO) community to enable data type registration and data resource annotation. The DCO science covers four broad themes related to carbon: Extreme Physics and Chemistry, Reservoirs and Fluxes, Deep Energy, and Deep Life, and the research is undertaken by a network of around 1700 scientists from more than 400 organizations and 40 countries. The DCO endorsed data policies for establishing the framework for the long-term stewardship of carbon data and information [16]. Since DCO datasets range from the highly regular/tightly specified to the very *ad hoc*, there are significant needs for specified data types. Moreover, scientists are also interested in knowing the origin or provenance of a data type, and this provenance representation is implemented in the DCO data portal. Provenance in the Semantic Web is the information about entities, activities, and agents involved in the production of something [17]. Such information is useful for the assessment of the quality, reliability, and trustworthiness of a resource.

In our previous work, we have already developed ontologies and built a data portal, and have made them a part of the Linked Open Data [18]. For this work, our intention is not to collect a full list of specific data types for that data portal. Instead, we want to build a service so that the users can register data types and use them to annotate datasets. The conceptual model of data type will be the core of that service, which in turn will be part of the data portal. Therefore, we need to make the model of data type consistent with the existing ontologies in the data portal, and we should also make concepts in the model general enough so they can also be reused in other places. From the perspective of semantic modeling and encoding, we can treat data type as an entity. There should be a class for data type, and a number of properties describing the attributes of each data type instance. There should also be a number of other properties describing relationships between data type and other entities (e.g., datasets), agents (e.g., creator of a data type instance), or activities (e.g., creation of a data type instance).

The model is an extension to the existing ontologies used in the DCO data portal, and it reuses a few properties and classes from those ontologies. The prefixes, full names, and namespace URIs of those ontologies are listed in Table 1. Details of the developed model will be given in the next few sections.

Table 1. Ontologies and schemas reused in the model for data type specification.

Prefix	Full Name	Namespace URI
owl	Web Ontology Language	http://www.w3.org/2002/07/owl#
dctype	DCMI Type Vocabulary	http://purl.org/dc/dcmitype/
dco	DCO Ontology	http://info.deepcarbon.net/schema#
prov	PROV Ontology	http://www.w3.org/ns/prov#
skos	Simple Knowledge Organization System	http://www.w3.org/2004/02/skos/core#
foaf	FOAF Ontology	http://xmlns.com/foaf/0.1/
vivo	VIVO Ontology	http://vivoweb.org/ontology/core#
bibo	Bibliographic Ontology	http://purl.org/ontology/bibo/
xsd	XML Schema Datatype	http://www.w3.org/2001/XMLSchema#
rdf	Resource Description Framework	http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs	Resource Description Framework Schema	http://www.w3.org/2000/01/rdf-schema#

2.3. Concepts of Basic Types and Specific Types

Among the various ontologies available in the Semantic Web, there are already defined classes for types of data resources. For example, the Dublin Core Metadata Initiative (DCMI) Type Vocabulary [19] defines a list of types, including Collection, Dataset, Event, Image, Interactive Resource, Moving Image, Physical Object, Service, Software, Sound, Still Image, and Text. Each DCMI Type is defined as an rdfs:Class. In other ontologies there are also similar classes, such as bibo:Image, vivo:Video, and vivo:Dataset. A resource in our data portal can be asserted as an instance of one of those classes. For instance, "dco:data_001 rdf:type vivo:Dataset." In other ontologies, there are also similar classes for resource types such as vivo:Dataset and vivo:Video in the VIVO Ontology and bibo:Code in the Bibliographic Ontology (see Table 1 for links to these ontologies).

In this paper, we regard data as a general concept that covers not just instances of vivo:Dataset but also instances of other classes, including those defined in the DCMI Type Vocabulary and other ontologies. We call those resource type classes as basic data types because each of them categorizes the nature of a certain type of resource. For a data resource instance, we can learn its basic data type by reading the triple with the predicate rdf:type, such as the above example "dco:data_001 rdf:type vivo:Dataset." Nevertheless, the basic data type offers only limited information for understanding the meaning of a resource. That is part of the driving force for the work on the semantic specification of data types, and we call them specific data types.

The specific data type provides more information about a data resource. For example, we can use the triple "dco:data_001 dco:hasDataType dco:volcanicGasComposition" to annotate a specific data type. Here, the dco:volcanicGasComposition is not a keyword or label. Instead, it is an instance of a specific data type class dco:DataType, and there are a group of properties describing it. Figure 1 shows the properties associated with the class dco:DataType in our designed model for data type specification. The diagram in the figure demonstrates that specific data types can be used to annotate instances of vivo:Dataset, and in practice it can also be used to annotate other data resources such as instances of dctype:Image, dctype:Sound, and more.

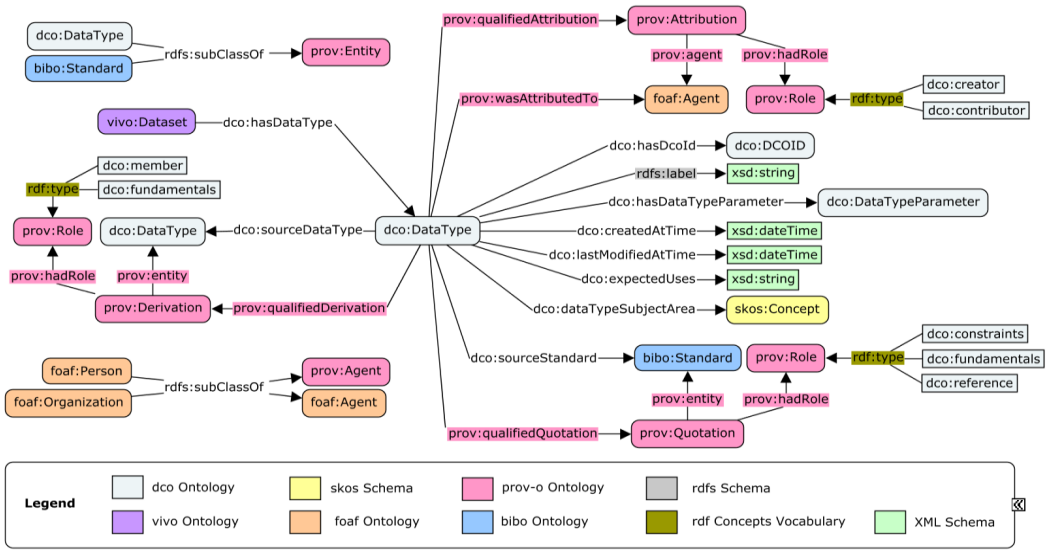

Figure 1. A conceptual model for the specification of data types.

2.4. Specification and Provenance

The properties and classes shown in Figure 1 reflect our current consideration on the semantics of data types and cover two primary parts: the specification of a data type and the provenance of it. We first give an introduction to the specification part, beginning with the property dco:hasDcoId at the mid-right part of the diagram in Figure 1. By using dco:hasDcoId, each data type is assigned a unique and persistent identifier called a DCO ID, which is similar to the Digital Object Identifier (DOI) of a journal paper. By using the properties rdfs:label, dco:createdAtTime, and dco:lastModifiedAtTime, a data type has its label, date of creation, and the date of modification if it has been modified. The property dco:expectedUses allows restrictions or suggestions on the uses of the data type to be recorded. A data type may consist of a number of parameters, such as the field names in a spreadsheet. Such parameters will be instances of the class dco:DataTypeParameter, and the property dco:hasDataTypeParameter connects a data type to its parameters. Using the property dco:dataTypeSubjectArea, a data type can be annotated with a number of keywords, which are instances of the class skos:Concept. A data type can be based on one or a few existing standards. The property dco:sourceStandard is used to connect a data type to the standards. Additionally, the property dco:sourceDataType can be used if a data type is derived from one or more existing data types. The creator(s) of a data type can be either a person or an organization, which both are instances of the class foaf:Agent. The property prov:wasAttributedTo connects a data type to its creator(s).

When we described the source data types, source standards, and creators of a data type above, we partly talked about the components of provenance. The developed data model allows us to record provenance at different levels of detail. For example, we may use only the property prov:wasAttributedTo to record the person who creates a data type, and we can also use more triples, as shown in the top right part of Figure 1, to show the detailed role of a person, such as creator and contributor. Similarly, the model also allows us to collect more detailed provenance information about the source data types and source standards. All the information recorded, including the specification and the provenance, will be machine-readable and will be stored in a triple store. End users can browse the information through the user interface of the DCO data portal and can also query the triple store and use the retrieved results in their applications.

3. Implementation and Results

The DCO data portal adapts the VIVO platform [20] for metadata management and the Handle System [21] for assigning unique identifiers (*i.e.*, the DCO ID) to all objects. The data portal also

uses Drupal [22] to develop user-friendly front end webpages for data resource navigation. Before the development of the model for data type, the DCO data portal already reused several ontologies in the DCO Ontology, such as the FOAF Ontology and the Bibliographic Ontology (see Table 1 for links to those ontologies). In order to link those ontologies to the provenance parts in the designed model for data type, we asserted a few existing classes as subclasses of corresponding classes in the PROV-O Ontology. For example, we added assertions "dco:DataType rdfs:subClassOf prov:Entity" and "foaf:Person rdfs:subClassOf prov:Agent" (Figure 1).

With the conceptual model of data type, we developed functions of data type registration, data type browsing and dataset annotation in the DCO data portal. For the data type registration, we used the default user interface of the VIVO platform, which follows the general workflow of creating an instance for any class. Once a data type instance is created, the data portal will assign a unique DCO ID for it. Then, on the VIVO profile of the data type, a user can fill in records for the properties describing the data type and the links that connect the data type with other objects. Figure 2 show a part of the profile of the registered data type "Thermodynamics of chemicals and minerals."

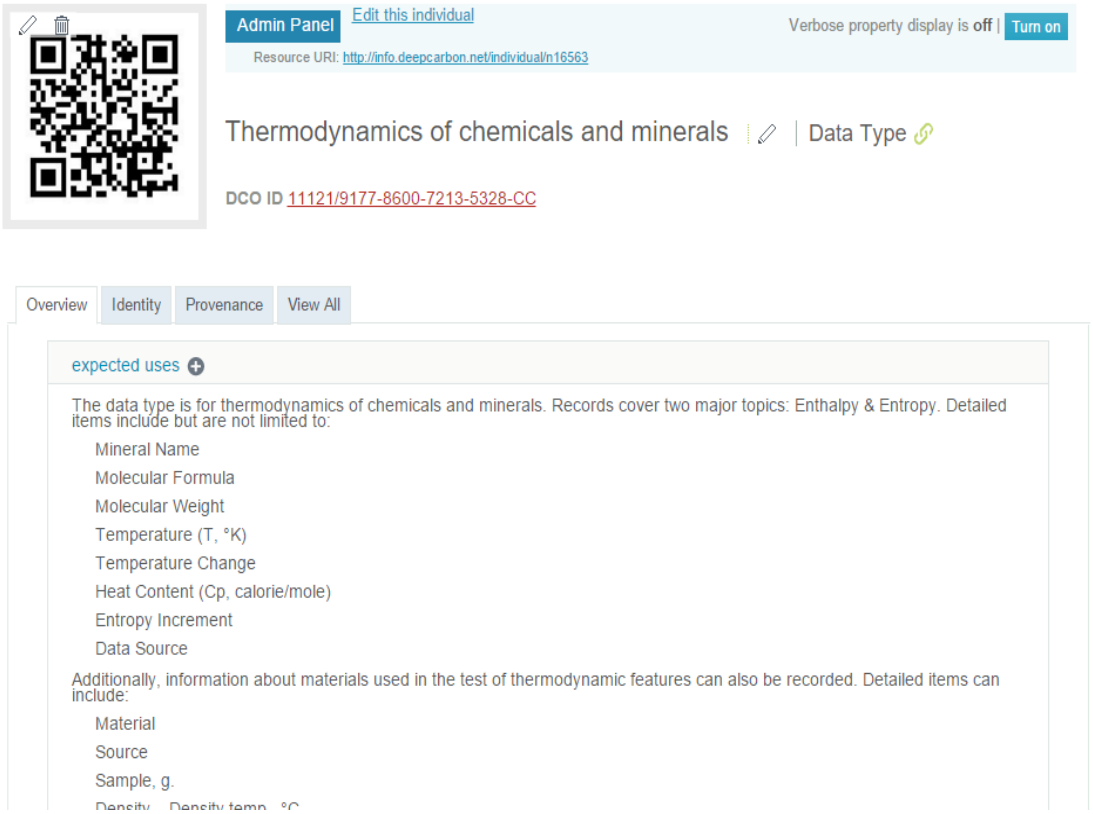

Figure 2. Screenshot of the profile of a registered data type.

We also developed a faceted browser for all the registered data types by adapting the Elasticsearch [23]. Figure 3 shows a screenshot of the faceted browser with a few data types we registered as test examples. On the left of the user interface there is a list of facets, which are related to the corresponding properties of registered data types in the portal. A user can search among the data types by choosing records in those facets. A feature of the browser is that, once the chosen records in a facet are changed, all records in other facets as well as the data type results will change correspondingly. The user can make selections in several facets and/or enter text-based search terms to search for one or more certain data types. Figure 3 shows the two resulted data types that have "Mineralogy" as research area (*i.e.*, keywords).

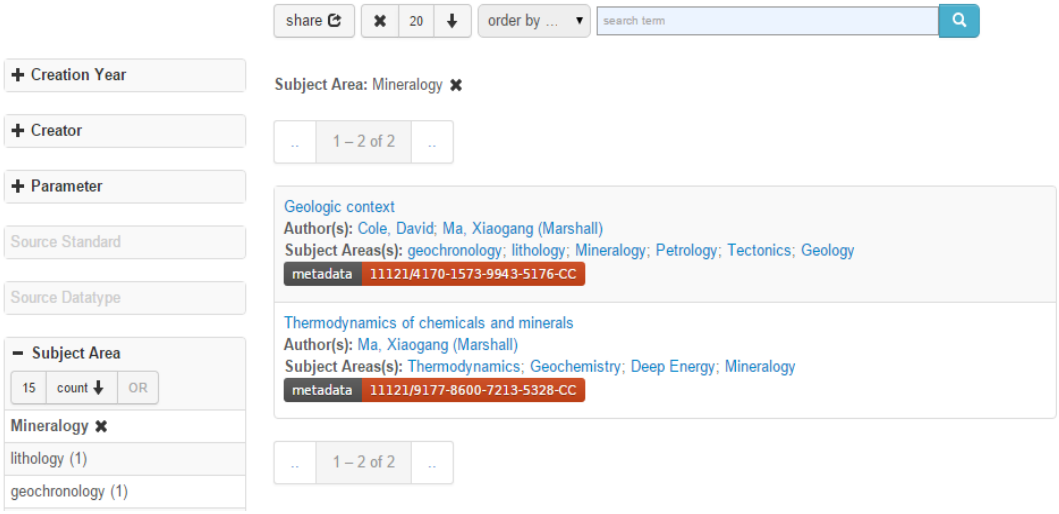

Figure 3. A faceted browser for registered data types.

We can use the registered data types to annotate data resources such as datasets. The only work a user needs to do is to fill in data type records for the property "dco:hasDataType" on the VIVO profile of a dataset. We also developed a faceted browser for datasets and listed "Data Types" as a facet in that browser. Figure 4 shows a part of returned datasets when "Thermodynamics of chemicals and minerals" is selected in the data type facet.

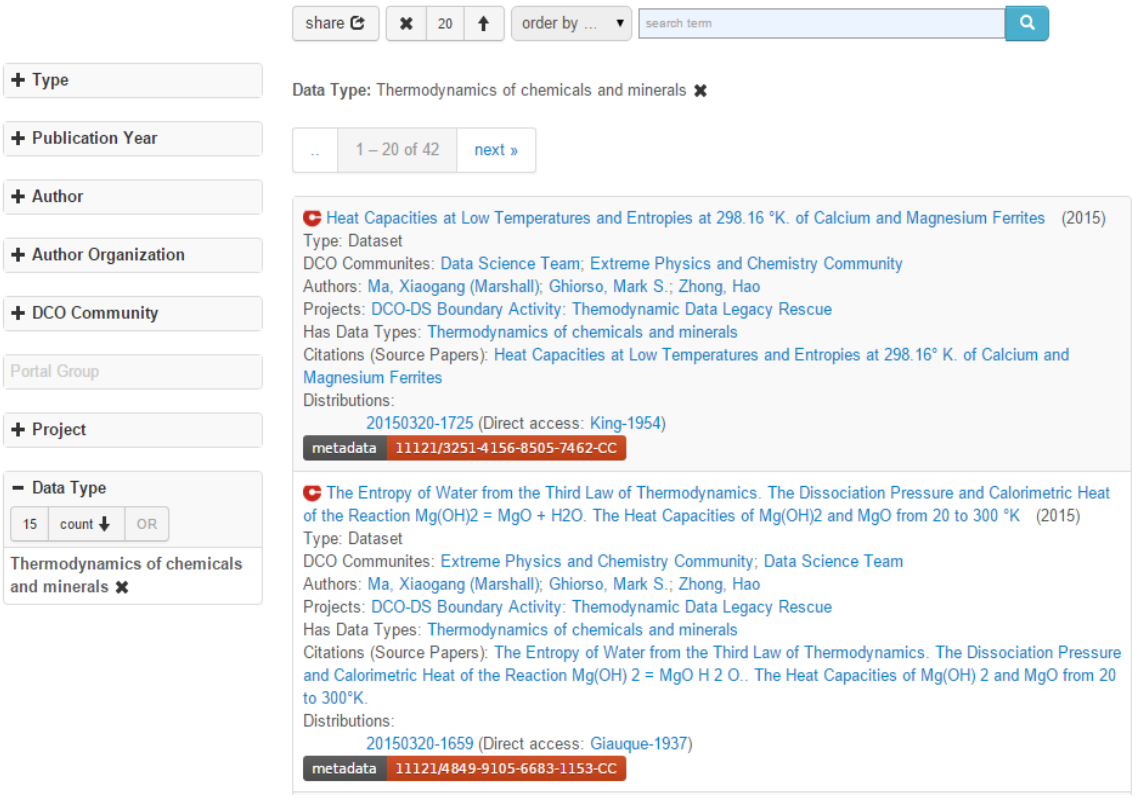

Figure 4. A faceted dataset browser with "Data Type" as a facet.

4. Discussion

Science today is increasingly facilitated by open data. As Fox [24] defined, "data science is doing science with someone else's data." Our work on the conceptual model of data type and the implementation of it in a data portal enriches the semantic description of datasets. The information in such description is not only human-readable but also machine-readable, which will provide valuable help to people who access and use datasets that are retrieved from the world of open data.

There have been many works on machine-readable models of data types. In the RDF Schema (RDFS, follow the namespace URI in Table 1 for more details), there is a class rdfs:DataType, and, in the concepts vocabulary of RDF, there are a few instances of it, such as rdf:HTML, rdf:langString, rdf:PlainLiteral, and rdf:XMLLiteral, which show that the work still focuses on the syntactic part of data types. The ISO standard ISO/IEC 11179-3:2013(E) [25] defines a data type as a "set of distinct values, characterized by properties of those values and by operations on those values." This definition, as well as the definitions of primitive and composite data types in that standard, is compatible with definitions of similar concepts in programming languages [8,9]. In the world of open data, a convention is to publish metadata together with data to describe the structure and contents of the data. Examples can be seen in netCDF headers [26] and Data Packages [27]. Most recently, W3C released the recommendation of a metadata vocabulary to support annotating, discovering, and displaying tabular data on the Web [28]. The recommendation provides metadata items for describing objects at several levels of detail, such as groups of tables, inter-relationships between tables, single tables, and individual columns within a table. Much of the work in that recommendation can be adopted to extend the model in this paper to a finer scale, especially the part surrounding the class dco:DataTypeParameter. Previous works on markup languages for harmonizing heterogeneous datasets, such as the Ecological Metadata Language [29], the Earth Science Markup Language [30] and the GeoSciML [31] can help provide use cases from the geoscience domain on how to represent data structures in a machine-readable format.

There have also been works that annotate datasets with domain specific data type information. For example, the EarthChem data portal [32] proposed a hierarchical list of data types to be used for tagging a registered dataset. However, the data types in EarthChem are specified at the text level, *i.e.*, as keywords. The work presented in this paper goes a step further from the "keyword" level by enabling the semantic specification of data types through a conceptual model. Currently, we do not have a full list of data types that meet all requirements of the DCO community, and we do not intend to register all the data types just by ourselves. Instead, the functionality we built for data type registration is open to the DCO community and any user can register specific data types of interest. Besides data type registration, we can also organize the explicit relationships among registered data types, *i.e.*, through the property "dco:sourceDataType." We can also organize the implicit relationships between data types. For example, the keywords used to describe data types can provide clues on the categories or disciplines of data types.

Our work was initiated with the adoption of the output of the Data Type Registry (DTR) working group of the Research Data Alliance (RDA) [33,34]. Each DTR is a self-contained portal for data type registration and curation. A registered data type is assumed to be resolvable to some useful information about that type. According to the vision of the DTR working group, there will be multiple DTR instances, and each governed by its own project, group, or community. All those DTR instances reuse some common basic types, which are called "primitives." Those primitives will be registered in a type registry presumably managed by the Corporation for National Research Initiatives (CNRI). So we can expect a two-level hierarchical federation of the DTR. The higher level is a list of primitives, and the lower level is the specific data types defined within a DTR.

Our differentiation of basic data types and specific data types are comparable to the thoughts of primitives and specific data types in the DTR working group. However, we adopted a different technological approach to realize the data type registration. From the point of view of ontology engineering, both primitives and specific data types in the DTR design are at the instance level, *i.e.*, they both are registered data types. In our work, the basic data types are at the class level, *i.e.*, they

are classes in ontologies, and data resources are instances of them. The specific data types in our work are at the instance level, *i.e.*, they are all instances of the class dco:DataType. If we put the specific data types at the class level, then we need to update the ontology frequently to include the new data type classes created. Using a single class dco:DataType and making all specific data types as instances of it significantly reduces the efforts needed to update the ontology and to maintain the framework of the data portal.

Focusing only on the case of the DCO data portal, because the portal is underpinned by ontologies, by making the model as a part of the DCO ontology, we deployed it in the DCO data portal quickly and smoothly. This shows the advantage of the conceptual model for data portals based on Semantic Web technologies. Mapping to the PROV-O Ontology allows the data type information to be connected with the broad provenance graph, such as the two example assertions "dco:DataType rdfs:subClassOf prov:Entity" and "foaf:Person rdfs:subClassOf prov:Agent" described in the previous section. We should note that, in these two example assertions, the former was fine because the DCO Ontology was developed by ourselves and we can make updates to it; however, for the latter assertion, we had the issue of "ontology hijacking" [35]. That is, newly developed ontologies re-defining the semantics of existing concepts resident in other ontologies. To reduce the negative impact in our work, we had those "hijacking" assertions only work for the DCO data portal and did not use them for other purposes.

Our work is just a first attempt in applying semantic technologies to enhance the meaning of and relationships between data types and datasets. In addition, we propose some possible future work that could further enhance the semantics of data types. First, we can extend and update the conceptual model to improve its ability to represent domain-specific meaning. We can do this by working with domain scientists within the DCO community, and the broader geoscience community, in the development of use cases specific to their science domains. A work of particular interest is to extend the specification of the class dco:DataTypeParameter and the relationships between parameter instances. A few existing works in the Semantic Web community (e.g., see [28]) can be adopted to accomplish this. We can even seek the opportunity to push the model to a more general level and build a data type ontology that can be used in various domains of studies outside of the DCO community. Second, the "Data Types" facet in the dataset browser can be enriched with more features to help users find datasets of interest. For example, a visualization gadget may be added to show the interrelationships among registered data types. We can develop a way to compute the similarity [36] between a researcher's interests and data types based on the researcher's profile in the data portal. Then, the data portal will be able to recommend data types of interest to that researcher. Third, once a certain number of data types are registered, methods can be developed to use the keywords in their description to explore the implicit relationships among those data types. Fourth, to leverage the value of data types and formalize their use and reuse, the possibility of setting up an Application Program Interface that serves (1) machine-readable information about the structure and contents of registered data types and (2) structured metadata for citation, both through the unique identifier of each data type, is worth exploring. In other words, it is an effort to promote the data type to be a first-class object in the world of open data.

5. Conclusions

Science is, in large, driven by data. The global open data has created great opportunities for science, but presents challenges in data interoperability. The clear identification of a meaningful data type is a key factor in solving data interoperability across scientific domains. Conventionally, a data type is often treated at the syntactic level, such as integer, float, Boolean, string, *etc.* Syntactic definitions of data types do not associate sufficient domain-specific semantic meaning to the data types. In this work, we describe the application of Semantic Web technologies for the specification of data types. With this approach, a data type can convey complex meaning, such as who creates the data type, the source standard that the data type derives from, the operations that can be done on datasets

of that data type, typical scientific domains, software programs and/or instruments that use the data type, and more. The implementation of our model in a production scientific data portal enables data producers to register data types and use them to annotate data resources. The data type information is both human and machine-readable. For the data users, they can receive explicating assumptions or information inherent in a dataset through the records of specific data types associated with that dataset. In this way, they can quickly see and understand the details within a dataset without even downloading it.

Acknowledgments: This research was supported by a sub-grant from the NSF Research Data Alliance (NSF-1349002) and Alfred P. Sloan Foundation (No. APS: 2014-06-02 (RPI)). We sincerely appreciate the constructive discussion with members of the Research Data Alliance—Data Type Registration Working Group in the early stage of this work.

Author Contributions: Peter Fox provided overview and led the work. Xiaogang Ma, John Erickson and Patrick West designed the model of semantic data type. Stephan Zednik and Patrick West developed the faceted data type browser. All authors contributed to the manuscript writing.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Klump, J.; Bertelmann, R.; Brase, J.; Diepenbroek, M.; Grobe, H.; Ouml, C.K.H.; Lautenschlager, M.; Schindler, U.; Sens, I.; Wächter, J. Data publication in the open access initiative. *Data Sci. J.* **2006**, *5*, 79–83. [CrossRef]
2. Maali, F.; Erickson, J.; Archer, P. Data Catalog Vocabulary (DCAT)—W3C Recommendation 16 January 2014. Available online: http://www.w3.org/TR/vocab-dcat/ (accessed on 11 February 2016).
3. Starr, J.; Gastl, A. isCitedBy: A metadata scheme for DataCite. *D-Lib Mag.* **2011**, *17*. [CrossRef]
4. Lin, J.; Fenner, M. Altmetrics in evolution: Defining & redefining the ontology of article-level metrics. *Inf. Stand. Q.* **2013**, *25*, 20–26.
5. Ma, X.; Asch, K.; Laxton, J.L.; Richard, S.M.; Asato, C.G.; Carranza, E.J.M.; van der Meer, F.D.; Wu, C.; Duclaux, G.; Wakita, K. Data exchange facilitated. *Nat. Geosci.* **2011**. [CrossRef]
6. International Organization for Standardization (ISO). *ISO15836: Information and Documentation—the Dublin Core Metadata Element Set*; International Organization for Standardization (ISO): Geneva, Switzerland, 2003; p. 8.
7. DataCite Metadata Working Group. DataCite Metadata Schema for the Publication and Citation of Research Data (Version 3.1). Available online: http://120.52.72.45/schema.datacite.org/c3pr90ntcsf0/meta/kernel-3/doc/DataCite-MetadataKernel_v3.1.pdf (accessed on 11 February 2016).
8. Mitchell, J.C. *Concepts in Programming Languages*; Cambridge University Press: Cambridge, UK, 2002; p. 529.
9. Donahue, J. On the semantics of "Data type". *SIAM J. Comput.* **1979**, *8*, 546–560. [CrossRef]
10. Deep Carbon Observatory Data Portal. Available online: https://info.deepcarbon.net (accessed on 11 February 2016).
11. Berners-Lee, T.; Hendler, J.; Lassila, O. The semantic web. *Sci. Am.* **2001**, *284*, 34–43. [CrossRef]
12. Gruber, T.R. Toward principles for the design of ontologies used for knowledge sharing. *Int. J. Hum. Comput. Stud.* **1995**, *43*, 907–928. [CrossRef]
13. The OWL 2 Schema Vocabulary. Available online: http://www.w3.org/2002/07/owl# (accessed on 11 February 2016).
14. Bizer, C.; Heath, T.; Berners-Lee, T. Linked data—The story so far. *Int. J. Semant. Web Inf. Syst.* **2009**, *5*, 1–22. [CrossRef]
15. Berners-Lee, T. Linked Data. Available online: http://www.w3.org/DesignIssues/LinkedData.html (accessed on 11 February 2016).
16. DCO Open Access and Data Policies. Available online: https://deepcarbon.net/page/dco-open-access-and-data-policies (accessed on 11 February 2016).
17. Groth, P.; Moreau, L. PROV-Overview: An Overview of the PROV Family of Documents. Available online: http://www.w3.org/TR/prov-overview/ (accessed on 11 February 2016).
18. Ma, X.; Chen, Y.; Wang, H.; Erickson, J.S.; West, P.; Fox, P. Deep Carbon Virtual Observatory: A cyber-enabled platform for linked science. In Proceedings of the SciDataCon2014, New Delhi, India, 2–5 November 2014.

19. DCMI Type Vocabulary. Available online: http://dublincore.org/documents/2000/07/11/dcmi-type-vocabulary/ (accessed on 10 June 2015).
20. VIVO. Available online: http://vivoweb.org/ (accessed on 10 June 2015).
21. HDL.NET® Information Services. Available online: http://handle.net/ (accessed on 10 June 2015).
22. Drupal. Available online: https://www.drupal.org/ (accessed on 10 June 2015).
23. Elastic. Available online: https://www.elastic.co/about (accessed on 10 June 2015).
24. Fox, P. Progress in Open-World, Integrative, Transparent, Collaborative Science Data Platforms. Keynote Presentation at ISWC2013 Conference, Sydney. 2013. Available online: http://tw.rpi.edu/web/doc/ISWC2013_Sydney_Fox20131024a.ppt (accessed on 11 February 2016).
25. International Organization for Standardization (ISO). *ISO/IEC 11179–3:2013(E) Information Technology—Metadata Registries (MDR)—Part 3: Registry Metamodel and Basic Attributes*; International Organization for Standardization (ISO): Geneva, Swizerland, 2013; p. 227.
26. Rew, R.; Davis, G. NetCDF: An interface for scientific data access. *IEEE Comput. Graph. Appl.* **1990**, *10*, 76–82. [CrossRef]
27. Pollock, R.; Keegan, M. Data Packages. Available online: http://dataprotocols.org/data-packages (accessed on 11 February 2016).
28. Tennison, J.; Kellogg, G. Metadata Vocabulary for Tabular Data. Available online: https://www.w3.org/TR/2015/REC-tabular-metadata-20151217 (accessed on 11 February 2016).
29. Michener, W.; Brunt, J.; Helly, J.; Kirchner, T.; Stafford, S. Nongeospatial metadata for the ecological sciences. *Ecol. Appl.* **1997**, *7*, 330–342. [CrossRef]
30. Ramachandran, R.; Graves, S.; Conover, H.; Moe, K. Earth Science Markup Language (ESML): A solution for scientific data-application interoperability problems. *Comput. Geosci.* **2004**, *30*, 117–124. [CrossRef]
31. Sen, M.; Duffy, T. GeoSciML: Development of a generic geoscience markup language. *Comput. Geosci.* **2005**, *31*, 1095–1103. [CrossRef]
32. EarthChem. Available online: http://www.earthchem.org/ (accessed on 10 June 2015).
33. Broeder, D.; Lannom, L. Data type registries: A research data alliance working group. *D-Lib Mag.* **2014**, *20*. [CrossRef]
34. Research Data Alliance. Available online: https://www.rd-alliance.org/groups/data-type-registries-wg.html (accessed on 10 June 2015).
35. Hogan, A.; Harth, A.; Polleres, A. SAOR: Authoritative reasoning for the web. In *The Semantic Web*; Domingue, J., Anutariya, C., Eds.; Springer: Berlin, Germany, 2008; pp. 76–90.
36. Zheng, J.; Fu, L.; Ma, X.; Fox, P. SEM+: Tool for discovering concept mapping in earth science related domain. *Earth Sci. Inf.* **2015**, *8*, 95–102. [CrossRef]