An OBI ontology Datum Proof Sheet

Toward a harmonized treatment of categorical and scalar data entities in user interfaces

Damion M. Dooley, William W.L Hsiao
Department of Pathology
University of British Columbia
Vancouver, Canada
damion.dooley@bccdc.ca

Emma Griffiths
Department of Molecular Biology and Biochemistry
Simon Fraser University
Burnaby, Canada

***Abstract*—There are numerous past and current examples of ontology-driven projects that provide auto-generated user interfaces for managing entities and relations, each presenting its own varied and complex data model. Our Datum Proof Sheet application aims to simplify the application development landscape by building community consensus about the way basic categorical, textual and numeric datum fields should be described within the OBOFoundry community of ontologies. The proof sheet shows selected datums (grouped under the context of an OBI "data representational model" item) as form inputs on an HTML page, enabling an application ontology's contents to be presented to end users (ranging in our case from epidemiologists to software developers) for review without necessarily having a working application to showcase them in. The basic relations and cases necessary for presenting datums in a user interface are mostly satisfied by OBI's design, but we introduce a few extra elements to bring more clarity to datum specifications, and to provide user interface term labels and definitions that may differ from those that ontologists prefer in the "backend".**

Keywords—ontology; user interface; datum; categorical; measurable; scalar variable

I. Introduction

Our open-source Datum Proof Sheet application, viewable at tinyurl.com/uiproofsheet as part of the under-development Genomic Epidemiology Ontology (GenEpiO) [1], is motivated by the different ontology needs of software developers, end users, and data exchange administrators – people who are usually unfamilar with the Web Ontology Language (OWL) or savy to Basic Formal Ontology (BFO) / Ontology for Biomedical Investigations (OBI) design principles. One need is to show ontology domain stakeholders what particular datums would look like in a data entry form visa vis field labels, definitions, units, or default pick-list options, in order for feedback to be gathered and those features to be finalized. In this way we avoid forcing non-technical users to vet an ontology on paper, or within the more complex environment of an ontology editor like Stanford's Protégé, and we can begin the ontology development process well before any related application is built.

Another motivation is to test the use of OBI categories for describing datums, namely the "data representaitonal model", "has value specification", "categorical measurement datum",

IRIDA project funding is graciously provided by Genome Canada, Genome BC, and the Genomics R&D Initiative (GRDI) with additional support from Simon Fraser University and Cystic Fibrosis Canada

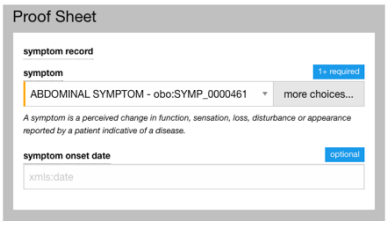

Fig. 1. Proof sheet of a symptom record

and "has measurement unit label" classes. Our own experience of trying to determine the accepted use of these terms by searching the literature for examples was frustrating insofar as working implementations lacked some of the terms or employed alternative usage. It should be a simple or at least well-defined process to establish OBI-conformant datum definitions independent of (but also as a natural precursor to) a more advanced content management system. We aim to reduce the training requirements for software implementers to adopt the ontology-defined aspects of input variables into their own interfaces.

Although the Web Ontology Language (OWL) was designed to provide reasoning in a world of entities described by subject-predicate-object triplets, it also inherits ways to describe "primitive" data – allowing entities to have properties with string, numeric and date/time values whose datatypes (like integer, decimal, string, and date) are imported from the XML namespace/schema. Using "data properties", Protégé's editor allows one to associate some datum with an XML schema datatype, and to place constraints on an associated value, if numeric, or its length, or character pattern, if a string.

As well, a unit (meters, seconds, etc.) can be tacked on to an entity "value specification" for scalar datums by way of the "has measurement unit label", but it is left to the ontologist to create or import unit terms. Unfortunately, there currently is no single standardized unit ontology, but rather a variety – for example, Quantities, Units, Dimensions and Data Types Ontologies (QUDT) [2], Ontology of units of Measure and related concepts (OM) [3], and the Units Ontology (UO) [4]. QUDT and OM illustrate how units are a microcosm of ontological complexity that are under-utilized if left as atomic terms. They enable a compound unit to be decomposed into specifications for its numerator and divisor, which can then enable unit analysis or unit conversion, e.g. OM's "Compound units", or QUDT's "Quantity Dimensions". Our work stops

short of naming best practices here except to recommend the specification of a datum's preferred unit and scale (e.g. degree vs. kelvin, or centimetre vs metre). This anticipates the ability to transform incomming data to the favoured unit for presentation or storage.

While a unit specification can be inherited down from superclass to subclass (a feature we employ), we have found that OBI lacks the ability to treat datatypes the same way because there is no relation that allows one to make claims about an entity's data type independently of an (instance of a) stored value. A discussion paper on OBI data prototypes by James Overton [5] shows the "has specified value" (previously "has measurable value") in use to specify a literal and its data type directly, as shown in this example:

```
Individual: 20g-specification
  Types: 'value specification'
  Annotations: rdfs:label "20g specification"
    Facts: 'has measurement unit label' 'gram',
    'has specified value' "20"^^xsd:float
```

It is not apparent how one can make a claim that a more abstract class can be associated with a primitive data type, such that underlying classes can take on that same data type or a subclass of it. For this reason we propose a new relation, "has primitive value spec" that points directly to a primitive URI, decimal, integer, string or date-time data type, and which allows subclasses to inherit the same.

Our proof sheet application testing to date has been on the GenEpiO ontology which supports the IRIDA (www.irida.ca) project. There many clinical and environmental measurables, and process/event related named points in time (e.g. "exposure start" and "symptom onset") have been placed under OBI categorical, scalar and time measurement categories. All categorical items like the disease or symptom hierarchies have a basic data type of "URI", meaning that to select a categorical value is to select some vocabulary item that must have a globally accessible URI (a URI enables an entity to be a categorical datum, and an ontology is at the very least a data dictionary of such things). Categorical measurables like "Symptom" are marked as an OBI "categorical value specification", but we propose an additional term under "value specification" called "categorical tree specification" that allows us to list, often in a hierarchy of finer-grained differentiation, particular pick list choices for a categorical variable, with choices usually imported from other ontologies. Unlike existing OBI "categorical value specification" members, a member of the categorical tree specification class is not itself selectable (except perhaps as an uninformative case), but any of its descendent (subclass) entities is a selectable value for the datum.

II. Implementation

The first stage of our ontology Datum Proof Sheet application is a python script (see jsonimo.py in https://github.com/GenEpiO/genepio/tree/master/proofsheet which loads an OWL ontology and all of its include files into memory, uses Sparql 1.1 queries to extract datum field specifications, and writes this user-interface related content to a JSON-LD file. On a Mac Powerbook i7 with 16Gb RAM the process to generate the resulting 780Kb JSON-LD data structure for our roughly 1,400 class, 11,500 axiom ontology took about 17 seconds.

Finally, a javascript-driven HTML page application reads in the JSON-LD file, and extracts the hierarchy of grouped datum fields and categorical pick-lists, and presents them via a menu-driven field rendering engine with the help of the Zurb Foundation website layout and form library. This interface enacts various data-validation checks that follow from the data type and range specifications laid out within an OWL file. Integer, date and string constraints (including regular expression patterns which for example exhaustively match all known e-coli k-antigen patterns) are implemented such that users can see if validation performs to their expectations.

When it comes to user interfaces, the logical formality present in term definitions often needs to be replaced by abbreviated "plain english" language, and for this reason we have introduced new "UI preferred label" and "UI preferred definition" annotations that are especially useful when terms imported from a 3rd party ontology already have "label" and "preferredLabel" etc. entries that don't match the data-entry needs of an application (e.g. "age" vs. "age since birth measurement datum"). These annotations promote coloquial term consolidation across applications that share ontology.

III. Discussion

We encourage feedback on our proposed "has primitive data spec" relation and "categorical tree specification" entity - there may be previously established terms from other projects within the community that effect the same result, or use cases we're yet to hear of.

References

[1] E. Griffiths, "IRIDA's Genomic Epidemiology Application Ontology (GenEpiO): Genomic, Clinical and Epidemiological Data Standardization and Integration." Online (March 2016) http://www.slideshare.net/EmmaGriffiths12/iridas-genomic-epidemiology-application-ontology-genepio-genomic-clinical-and-epidemiological-data-standardization-and-integration

[2] R. Hodgson and P. J. Keller, "QUDT-quantities, units, dimensions and data types in OWL and XML." Online (September 2011) http://www.qudt.org

[3] M. van Assem, H. Rijgersberg, and J. Top, "Ontology of units of measure and related concepts." Semantic Web 4, no. 1 (2013): 3-13.

[4] G. V. Gkoutos, P. N. Schofield, and R. Hoehndorf, "The Units Ontology: a tool for integrating units of measurement in science." *Database*2012 (2012): bas033.

[5] J. A. Overton, "OBI data prototype," Online (August 2013) https://svn.code.sf.net/p/obi/code/trunk/src/examples/development/data-prototype.pdf