An UMBEL Extension for Attributes

by Mike Bergman - Monday, February 16, 2015

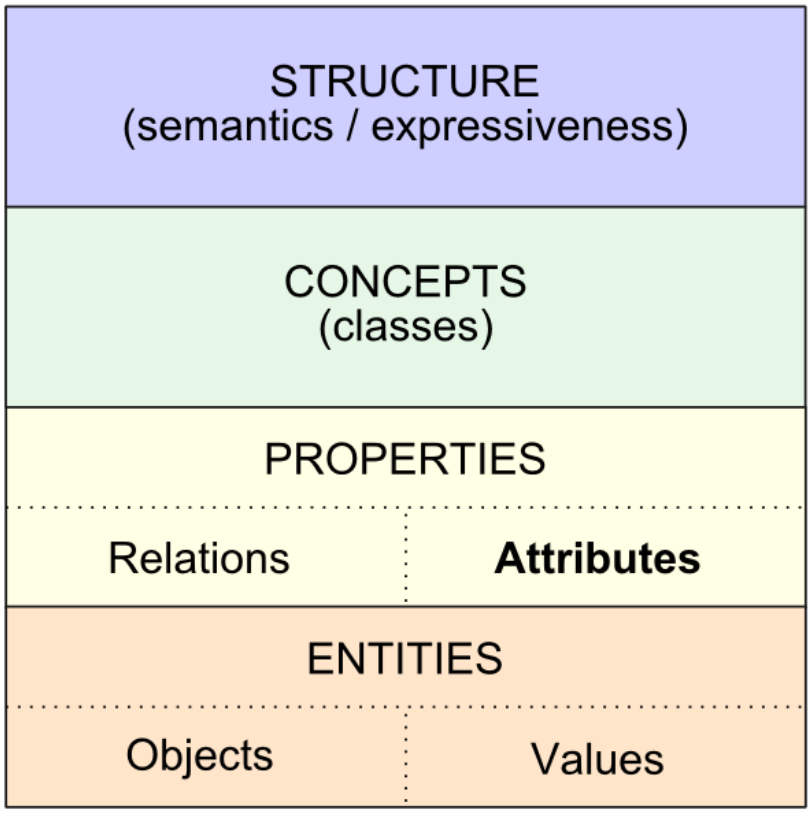
http://www.mkbergman.com/1838/an-umbel-extension-for-attributes/

The Attributes Ontology is Designed for Efficient Data Mapping

The semantic Web does not yet have the complete infrastructure for supporting data interoperability. Most ontology mapping or alignment efforts have focused on concepts, or the class structure of the schema. Comparatively little has been done on instance mapping or predicate (property) mapping [1]. Yet these considerations should reside at the heart of how semantic Web technologies can assist data interoperability.

We began the UMBEL (*Upper Mapping and Binding Exchange Layer*) vocabulary and ontology as a reference structure for concepts, a means to help match the discussion of topics and things across the Web. As such, UMBEL is part of a fairly robust library of upper ontologies that are meant to provide the grounding references for what information is *about*. Domains as diverse as biomedicine, banking, oil and gas, municipal governments, retail, marine organisms and the environment -- among many others -- have effectively leveraged upper ontologies to get diverse datasets and vocabularies to relate to one another. This is much welcomed, to be sure, and a good indicator of how semantic technologies can begin to approach getting data to interoperate.

Here is one way to look at the data interoperability space from a semantic technologies perspective (as initially informed by Pietranik and Nguyen [2]):



The Ontology Stack

The overall semantics of the structure -- indeed, how the structure itself is defined -- comes from which ontology languages and vocabularies are used. From an expressiveness standpoint, particularly in conceptual relations or domain schema, there are a variety of standards and specifications from which to choose [3]. We also have pretty good reference ontologies for many domains and what is called the upper levels. We are also starting, through efforts such as Wikipedia (DBpedia and Wikidata), schema.org, Freebase and OKKAM, to get referencable datasets of entities and their attributes, sometimes organized by type.

Reference groundings for properties, on the other hand, have received virtually no attention [4]. SIO, the Semanticscience Integrated Ontology, is one attempt to provide a reference structure for properties in the science domain. The approach is exemplary, but still lacks the scope required of a general grounding vocabulary. QUDT, the Quantities, Units, Dimensions and Data Types Ontologies, provides a standard vocabulary for measurement quantities, but lacks the scope to capture non-quantitative measures for describing things. Both SIO and QUDT should inform and contribute to a still-needed broader treatment of how to describe entities. That is the purpose of the Attributes Ontology in the forthcoming new release of UMBEL.

Attributes within the Semantic Technology Stack

The properties in RDF triples *(s - p - o)* relate two things, the subject and object, to one another. One

pragmatic way to understand properties, which are the predicates or verbs of these triple statements, is that they fall into two broad categories. The first category are the properties between or among different things; they are ***extrinsic*** to the subject at hand. These relations stipulate hierarchical relationships (subClassOf, fatherOf, daughterOf), mereological relationships (partOf, isComponent), role relationships (isBossOf, hasTeacher, isKeyInfluencer) or approximation relationships (isLike, isAbout, relatesTo). Both subjects and objects are concepts or identifiable things (entities).

However, the second category of properties, attribute properties, has a different nature. Attribute properties -- attributes for short -- are characteristics of an entity or entity type (class). They describe the entity at hand in the nature of key-value pairs. The key is the attribute, and the value is the literal value or object reference. In broad terms, attributes are the specifics of what is contained in a data record for a given instance. Multiple instances, or records, make up what is known as a dataset.

Attribute properties are ***intrinsic*** or descriptive properties. The combination of possible attributes for a given entity constitutes the intensional definition of that object. This use of the term attribute is consistent with its research sense as a descriptive characteristic of an object or its computing sense as being a factor of a given object. In the spirit of this inclusive sense of how attributes describe a given thing, we also include annotations and metadata as part of the attributes category of properties as well. All attribute properties provide a description or characteristic for the entity at hand.

Here are some example key-value pairs about me, the entity Mike Bergman, to illustrate the diversity of how attributes may describe things:

hair : red	location : 41°41?18?N 91°35?12?W	occupation : CEO	species : homo sapiens	maritalStatus : married
college : Pomona College	dateEntered : 02/16/2015	avocation : flyfishing, cooking	sex : male	blog : http://mkbergman.com
mood : happy	country : USA	lastBook : Zen and the Art of Motorcycle Maintenance	height : 6'3"	children : Erin,Zak
spouse : Wendy	city : Iowa City, IA		graduateSchool : Duke University	address : 380 Knowling
cat : Snuffles				fullName : Michael K Bergman

Example Key-Value Attributes

The infoboxes in Wikipedia are another example of such attribute types and values. Note that the values may vary widely as to units or quantities or even links to other things. Also note it really does not matter what order the value pairs are presented and some values refer to other objects (shown as links).

Virtually any data format or data serialization in existence can be expressed in such key-value pairs. Further, related types of entities have related attributes, such that attribute relationships are an alternative way to describe typologies. My attributes, as a human, are quite similar to attributes for other humans, and somewhat close to other mammals. But my attributes are very different than those for a worm or an automobile.

Even simple attributes can pose a challenge for mapping, absent a grounding framework. My name, for example, is Michael Kermit Bergman, which is often provided as Michael Bergman, Mike Bergman, M K Bergman, mkbergman or Michael K Bergman, and the fields that can capture those variants can capture one to four name parts, all called something different. References, rules, semsets (synonyms, jargon and aliases), and coherent organization are needed to ground all of these variants into a common form.

Attribute properties may be quantitative (with a quantitative, measurable value), qualitative, or descriptive or annotative. In many cases, the actual value of an attribute is a literal or numeric value, but it may also be an object, as when the value is a member of an enumerable set or its own defined entity. Describing something as having a color characteristic of *red*, for example, may result in a literal assignment of the string "red" or it may refer to another object definition where red is specified as to its chromatic properties. Further, if my idea of *red* was in context with my own personal record (as above), then the referent is more properly something like red hair. Semantics (and, thus, context) matter in data interoperability. I will describe more the rationale and importance of the relation-attribute property split in a following article [5].

The purpose of semantic technologies is to overcome some 40 categories of semantic heterogeneity, as I most recently discussed in [6]. One interesting aspect is the large number of semantic differences that may be ascribed to attributes, as this table from [6] shows (see the yellow entries):

Class	Category	Subcategory	Examples	Type [7]
LANGUAGE	Encoding	Ingest Encoding Mismatch	For example, ANSI *v* UTF-8	Concept
		Ingest Encoding Lacking	Mis-recognition of tokens because not being parsed with the proper encoding	Concept
		Query Encoding Mismatch	For example, ANSI *v* UTF-8 in search	Concept
		Query Encoding Lacking	Mis-recognition of search tokens because not being parsed with the proper encoding	Concept
	Languages	Script Mismatch	Variations in how parsers handle, say,	Concept

			stemming, white spaces or hyphens	
		Parsing / Morphological Analysis Errors (many)	Arabic languages (right-to-left) *v* Romance languages (left-to-right)	Concept
		Syntactical Errors (many)	Ambiguous sentence references, such as *I'm glad I'm a man, and so is Lola* (Lola by Ray Davies and the Kinks)	Concept
		Semantics Errors (many)	River *bank v* money *bank v* billiards *bank* shot	Concept
CONCEPTUAL	Naming	Case Sensitivity	Uppercase *v* lower case *v* Camel case	Concept
		Synonyms	United States *v* USA *v* America *v* Uncle Sam v Great Satan	Concept
		Acronyms	United States *v* USA *v* US	Concept
		Homonyms	Such as when the same name refers to more than one concept, such as Name referring to a person *v* Name referring to a book	Concept
		Misspellings	As stated	Concept
	Generalization / Specialization		When single items in one schema are related to multiple items in another schema, or vice versa. For example, one schema may refer to "phone" but the other schema has multiple elements such as	Concept

		“home phone,” “work phone” and “cell phone”	
Aggregation	Intra-aggregation	When the same population is divided differently (such as, Census *v* Federal regions for states, England *v* Great Britain *v* United Kingdom, or full person names *v* first-middle-last)	Concept
	Inter-aggregation	May occur when sums or counts are included as set members	Concept
Internal Path Discrepancy		Can arise from different source-target retrieval paths in two different schemas (for example, hierarchical structures where the elements are different levels of remove)	Concept
Missing Item	Content Discrepancy	Differences in set enumerations or including items or not (say, US territories) in a listing of US states	Concept
	Missing Content	Differences in scope coverage between two or more datasets for the same concept	Concept
	Attribute List Discrepancy	Differences in attribute completeness between two or	Attribute

			more datasets	
		Missing Attribute	Differences in scope coverage between two or more datasets for the same attribute	Attribute
	Item Equivalence		When two types (classes or sets) are asserted as being the same when the scope and reference are not (for example, Berlin the city *v* Berlin the official city-state)	Concept
			When two individuals are asserted as being the same when they are actually distinct (for example, John Kennedy the president *v* John Kennedy the aircraft carrier)	Attribute
	Type Mismatch		When the same item is characterized by different types, such as a person being typed as an animal *v* human being *v* person	Attribute
	Constraint Mismatch		When attributes referring to the same thing have dif ferent cardinalities or disjointedness assertions	Attribute
DOMAIN	Schematic Discrepancy	Element-value to Element-label Mapping	One of four errors that may occur when attribute names (say, Hair *v* Fur) may refer to the same attribute,	Attribute

		Attribute-value to Element-label Mapping	or when same attribute names (say, Hair *v* Hair)	Attribute
		Element-value to Attribute-label Mapping	may refer to different attribute scopes	Attribute
		Attribute-value to Attribute-label Mapping	(say, Hair *v* Fur) or where values for these attributes may be the same but refer to different actual attributes or where values may differ but be for the same attribute and putative value.Many of the other semantic hete rogeneities herein also contribute to schema discrepancies	Attribute
	Scale or Units	Measurement Type	Differences, say, in the metric *v* English measurement systems, or currencies	Attribute
		Units	Differences, say, in meters *v* centimeters *v* millimeters	Attribute
	Precision		For example, a value of 4.1 inches in one dataset *v* 4.106 in another dataset	Attribute
	Data Representation	Primitive Data Type	Confusion often arises in the use of literals *v* URIs *v* object types	Attribute
		Data Format	Delimiting decimals by period *v* commas; various date formats; using exponents or	Attribute

			aggregate units (such as thousands or millions)	
DATA	Naming	Case Sensitivity	Uppercase *v* lower case *v* Camel case	Attribute
		Synonyms	For example, centimeters *v* cm	Attribute
		Acronyms	For example, currency symbols *v* currency names	Attribute
		Homonyms	Such as when the same name refers to more than one attribute, such as Name referring to a person *v* Name referring to a book	Attribute
		Misspellings	As stated	Attribute
	ID Mismatch or Missing ID		URIs can be a particular problem here, due to actual mismatches but also use of name spaces or not and truncated URIs	Attribute
	Missing Data		A common problem, more acute with closed world approaches than with open world ones	Attribute
	Element Ordering		Set members can be ordered or unordered, and if ordered, the sequences of individual members or values can differ	Attribute

Sources of Semantic Heterogeneities

We can see that attribute heterogeneities may apply to the attribute itself (the key in a key-value pair), as to what it may contain and what it may refer to, as well as to the actual values and their units and measures. These aspects are important, in that they are the very ones we mean when we talk of data.

Rationale for an Attributes Ontology

When we combine the descriptions of things, we need ways to overcome these sources of semantic heterogeneities. As with concepts, it would be extremely helpful to have a similar attributes vocabulary, and one which is organized according to some logical attribute schema. This combination of vocabulary and schema defines what constitutes an attributes ontology. It can also be a reference grounding for how to relate data from different datasets to one another. Providing this grounding is the driving rationale for UMBEL's new Attributes Ontology.

Benefits

In addition to this overarching rationale in data interoperability, a reference Attributes Ontology brings with it a number of benefits:

- ***More efficient basis for interoperability*** -- the main advantage of a grounding reference is that it allows a spoke-and-hub design for data mapping, which is tremendously more efficient than pairwise mappings. In a spoke-and-hub design, where the reference ontology is the common node at the hub, only *n* - 1 routes are necessary to connect all sources, meaning that it scales linearly with the number of sources and attributes. Without a grounding reference, these same mapping capabilities would require $\frac{n(n-1)}{2}$ routes in a pairwise (point-to-point) approach, that also scales poorly as a quadratic function. A system of ten datasets would require 9 composite mappings in the reference grounding case, but 45 in a pairwise approach. And, of course, datasets themselves contain tens to thousands of attributes, compounding the map scaling problem further;
- ***Higher quality mappings*** -- a single target schema promotes schema enhancements, and toolsets can be justified to automate many processes, leading to;
- ***Faster integration*** -- these efficiencies lead to faster and more cost-effective mappings;
- ***Better ability to combine data values across records*** -- which means the approach can be seen as suitable for any content input type (structured, semi-structured or unstructured) or with any form of semantic heterogeneity;
- ***Faceted browsing and querying*** -- because the nature of the attributes and their values are mapped to a logical schema of attribute relationships, each attribute concept can be the basis of filtering and retrievals, powerfully supporting faceted browsing and querying;
- ***Infer attribute properties*** -- the logical basis of the attribute schema itself means that relationships and connections may be inferred, and semantics enable different perspectives and language to capture all aspects of the schema. This means the full capabilities of semantic search and querying can be brought to contributing data;
- ***Highest common denominator*** -- these capabilities mean that source datasets can be lifted and made consistent with a higher standard of testable values and inferences. The rich history of RDFizers points to the usefulness of RDF and related characterizations to bridge between multiple, native data formats [8]. The knowledge of the data already characterized in the system can inform the proper expression of new source data; and
- ***Better data integration, interoperability*** -- ultimately, of course, all of these factors lead to a complete approach to data interoperability, which leads to being able to finally achieve the objectives of "schema matching" or "data mapping."

Use Cases

These benefits can be realized in any data integration or interoperability setting. However, the benefits are particularly strong for these use cases:

- ***Combining records across datasets*** -- the *sine qua non* of data integration;
- ***Checking validity of values*** -- having an internal knowledge base of logic, schema, attributes listing and validated values against which to test data updates or new incoming data; and
- ***Establishing an EIA or MDM capability*** -- creating the internal infrastructure for truly responsive enterprise information integration or master data management. These are the reusable information and knowledge assets that are the grease for any data integration effort. The knowledge bases become assets in and of themselves. The budget sinkholes of most enterprise integration efforts can be turned around to become competitive assets in their own right.

As we have noted many times, these uses also benefit from the incremental and open world ability to expand the scope of the data integration at any point in time [9].

Description of the Attributes Ontology

We have recognized the importance of the attributes category going back to the first introduction of SuperTypes in UMBEL v.0.80 in 2010 [10]. We noted then that many of the concepts in UMBEL were devoted to how to describe things and the units or quantities associated with their values. We could also see the potential value in having a reference for mapping data characteristics and values.

The first creation of the Attributes SuperType -- also introduced in UMBEL v.0.80 in 2010 -- aggregated into one place related OpenCyc concepts regarding these descriptors. Working with this category over time surfaced, again, the underlying coherence and use of OpenCyc. We found that UMBEL (via its OpenCyc extraction) already had a strong, logical undergirding to support an organized representation of attributes. Once we understood these patterns, we were able to go back to OpenCyc and better capture other aspects of its attribute structure that we had earlier overlooked. We then added a few aggregate categories to UMBEL to provide a cleaner organization. UMBEL now understands and organizes some 2000 different descriptive attributes.

Over a period of years we did research on exemplars in these areas, with the limited results as first mentioned, notably QUDT and SIO, and also DERA [4]. We also enlisted input from the semantic Web mailing list and were not able to find a suitable extant reference structure [11]. We find it perplexing more work has not been done in this area. We do abhor a vacuum!

Nonetheless, we were able to combine the 2000 attributes infrastructure of OpenCyc into the following upper level of the Attributes Ontology structure:

```
AttributeValues

StringObject

StringDatatype_Unlimited
```

List_Information

FrequentlyAskedQuestionsList

MailingList

AlphabeticalList

Index_List_Information

BullettedFormat

UnitOfMeasure

UnitOfDistance

InternationalUnitOfMeasure

UnitOfMeasure_Common

NaturalLanguage

Encrypted

AuthenticationSource

Persistence

Distribution

Uniform_PersistenceDistribution

UnitOfMeasureConcept

Ratio

CollectionType

Phase

EmptyCollection

Preference

Quantity

AttachmentAttribute

WrittenInfo

StructuredInfo

VisualInfo

AudioInfo

LogicalFieldAttribute

TruthValue

AttributeTypes

DescriptiveAttributes

Definition_PCW

VisualPattern

SpatialThingTypeByShape

ShapeAttributes

Color

Name

Title

EnumeratedAttributes

EconomicalQuantity

DispositionalQuantity

MentalQuantity

PhysicalQuantity

Quality

SocialQuantity

MeasurableQuantity

TotallyOrderedQuantityType

QuantityType

NonAspectualQuantity

EnvironmentalQuantity

ActionAttributeLevelQuantity

EmotionalQuantityType

LocationAttributes

OrientationAttributes

GeographicalPlace

MappableAttributes

ContactLocation

PopulatedPlace

TimeAttributes

HistoricTemporalThing

Time_Quantity

EventAttributes

TimeInterval

TemporalThing

IdentificationAttributes

ContactLocation

ReferenceWork

IDString

`UniqueID`

`SituationAttributes`

`Situation`

Upper Structure of the Attributes Ontology

Note the structure above roughly splits into two parts. The first, `AttributeValues`, captures the various ways and measures that may be applied to actual values. We foresee a key mapping to QUDT in this part. The second part of the structure, `AttributeTypes`, organizes the nature of various attributes into similar, logical categories.

We have also added some experimental predicates to the UMBEL vocabulary for mapping domains, ranges and specific external properties to reference attributes. See the ongoing specification in the UMBEL Annex L documentation for other pertinent details.

Though the Attributes Ontology has a bit more structure, it too is a module that segregates out specific attributes into its own files. About 2000 of the UMBEL reference concepts are tagged as attributes; about two-thirds of those, or 1275, are specific attributes that are assigned to the Attributes Ontology, which is also the container for the attributes module.

To our knowledge, the Attributes Ontology (AO) will be the first publicly released attempt to provide an explicit modeling framework for data attributes and values. We expect there to be hiccups and improvements to be made as we work with the system. We expect quite a few release iterations, and experimentation and change. We will retain an experimental designation of the new UMBEL properties and the Attributes Ontology itself until we gain better working comfort with the system.

The Additional UMBEL Entities Module

This new UMBEL Attributes Ontology is being accompanied by the creation of another UMBEL component, the Entities Module. This new module, designed in a similar way to the Geo Module that was released in version 1.05, tags all entities as such and places another 12,000 instances into a separate module. A hierarchy of about 15,000 entity types (and their descriptions and relationships) remain in UMBEL core.

Like the Geo Module, itself comprised of entity instances, the Entities Module may be invoked or not for a given use of UMBEL. The ability to filter on entities and SuperTypes is also a powerful new feature. The fact that there is major disjunction among the SuperTypes also adds to the power of queries and retrievals.

Thus, with the attributes module that is now part of the Attributes Ontology, there are now three separate but invokable modules in addition to the UMBEL core. The Geo, Entities or Attributes modules may be included or not in any given UMBEL deployment.

Pending Releases

After five years of sporadically intense thinking, Structured Dynamics is extremely pleased to first formally express our ideas about how to manage and model data and its attributes using the underlying machinery of semantic technologies. We welcome use and commentary on our approach and the Attributes Ontology.

We willl be releasing UMBEL v.1.20 by the end of March with various improvements, including the Entities Module and Attributes Ontology noted above. We are also updating the UMBEL documentation and have added Annexes K and L that describe the Clojure-based UMBEL generation process and the specifics underlying the Attributes Ontology [12]. Shortly thereafter we expect to provide a new minor release that will provide mappings between the UMBEL Attributes Ontology and DBpedia and schema.org properties.

For the time being, we will be focused on refining our use of UMBEL for data interoperability, specifically for attributes. However, we note that the ontology structure used in this article also flags roles and relations as another possible gap. This gap is likely to be the next major focus in UMBEL's research agenda.

---

[1] For example, the relative status of various ontology mapping efforts are covered, among others, in Fei Wu and Daniel S. Weld, 2008. "Automatically Refining the Wikipedia Infobox Ontology," *WWW 2008*, April 21–25, 2008, Beijing, China; Lorena Otero-Cerdeira, Francisco J. Rodríguez-Martínez, and Alma Gómez-Rodríguez, 2015. "Ontology Matching: A Literature Review," *Expert Systems with Applications* 42, no. 2 (2015): 949-971; and Marcin Pietranik and Ngoc Thanh Nguyen, 2011. "Attribute Mapping as a Foundation of Ontology Alignment," N.T. Nguyen, C.-G. Kim, and A. Janiak (Eds.): *ACIIDS 2011*, LNAI 6591, pp. 455–465, 2011. Also, I also discuss the relative poor state of mapping predicates between entities in many articles. See, for example, commentary on sameAs in M.K. Bergman, 2011. "Making Connections Real," in *AI3:::Adaptive Information*, January 31, 2011.

See also reference [4] and the follow-on discussion in [5].

[2] The basic approach to this stack diagram was suggested by a figure in Marcin Pietranik and Ngoc Thanh Nguyen, 2011. "Attribute Mapping as a Foundation of Ontology Alignment," N.T. Nguyen, C.-G. Kim, and A. Janiak (Eds.): *ACIIDS 2011*, LNAI 6591, pp. 455–465, 2011.

[3] W3C standards exist for RDF, RDFS and OWL; also, Common Logic and conceptual graphs provide higher-order capabilities. We use OWL 2 in our efforts. Some rationale for this choice is provided in M.K. Bergman, 2010. "Metamodeling in Domain Ontologies," in *AI3:::Adaptive Information*, September 20, 2010.

[4] One relevant effort, but which has not yet posted details or an ontology, is Fausto Giunchiglia and Biswanath Dutta, 2011. "DERA: A Faceted Knowledge Organization Framework," *Technical Report # DISI-11-457, University of Trento*, March 2011; submitted to the *International Conference on Theory and Practice of Digital Libraries 2011 (TPDL'2011)*.

[5] When posted, the reference to the follow-on article will be listed here.

[6] First posted in M.K. Bergman, 2014. "Big Structure and Data Interoperability," in *AI3:::Adaptive Information*, August 18, 2014.

[7] Concept is the shorthand used for the schema or classes or TBox. Attribute is the shorthand used for instance data or entities and their ABox. I segregate class-relation properties (predicates) from instance-describing properties (attributes).

[8] There are more than 100 converters of various record and data structure types to RDF. These converters — also sometimes known as translators or 'RDFizers' — generally take some input data records with varying formats or serializations and convert them to a form of RDF serialization (such as RDF/XML or N3), often with some ontology matching or characterizations. See this listing of known RDFizers.

[9] See M. K. Bergman, 2009. "The Open World Assumption: Elephant in the Room," in *AI3:::Adaptive Information*, December 21, 2009. The open world assumption (OWA) generally asserts that the lack of a given assertion or fact being available does not imply whether that possible assertion is true or false: it simply is not known. In other words, lack of knowledge does not imply falsity. Anothe way to say is it that everything is permitted until it is prohibited. OWA lends itself to incremental and incomplete approaches to various modeling problems.

[10] See this earlier (2010) version of Annex G: UMBEL SuperTypes Documentation to the UMBEL specifications.

[11] See this thread on the linked open data (LOD) mailing list from July 2014.

[12] See further the UMBEL Annex K: UMBEL Generator and UMBEL Annex L: Attributes Ontology and Version 1.20 to the UMBEL specifications (still being completed).
