

Conceptual and Practical Distinctions in the Attributes Ontology

by Mike Bergman - Tuesday, March 03, 2015

<http://www.mkbergman.com/1842/conceptual-and-practical-distinctions-in-the-attributes-ontology/>



Reasons for the Relations and Attributes Split

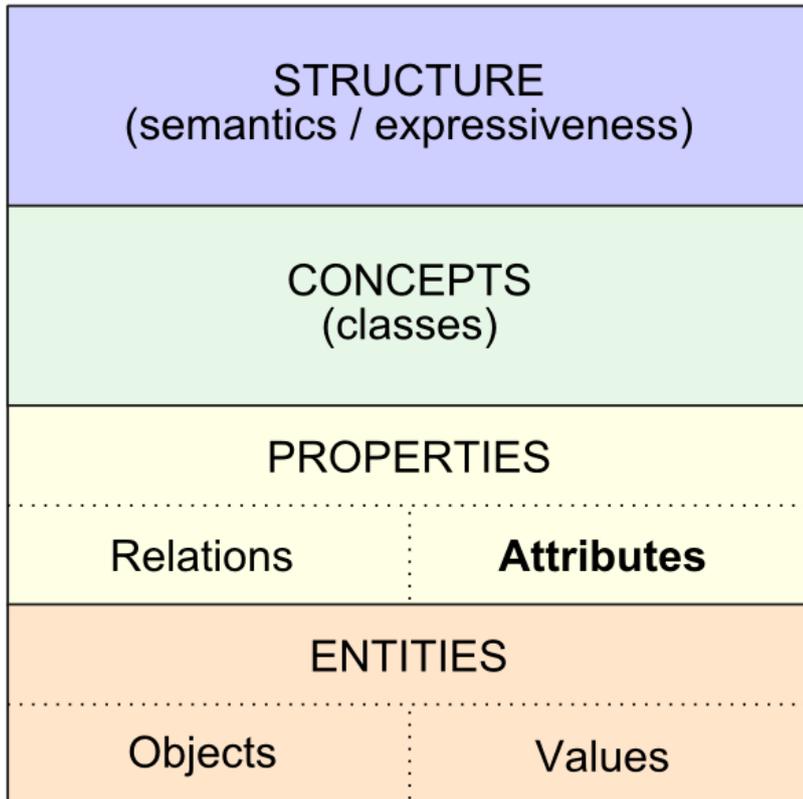
Logicians going back at least as far as [Charles S. Peirce \[1\]](#)-- and computer scientists as early as the [entity-relationship \(ER\) model](#) from the mid-1970s [\[2\]](#)-- have made the relations-attributes distinction for predicates relating to data objects. There are both conceptual and practical bases for these distinctions. This article elaborates upon the relations-attributes distinction in the [UMBEL](#) Attributes Ontology. I also try to precisely define my terms because terminology is overlapping and confusing amongst competing data models and standards.

In one of its first communiques in 1999 regarding the [Resource Description Framework](#), its sponsor, the [W3C](#) (World Wide Web Consortium), noted the RDF data model was a member of the entity-relationship modeling family [\[3\]](#). Though for its own reasons the W3C chose to label the relationships between things as "properties" in RDF [\[4\]](#), for practical modeling reasons the E-R model distinction into attributes and relations provides additional explanatory power. It also provides a more useful and tractable means for modeling the connections between things, an essential requirement for efficient data interoperability. Understanding these distinctions is an important basis for understanding the structure and design of UMBEL's Attributes Ontology.

Recap of the UMBEL Attributes Ontology

My [prior article](#) introduced the Attributes Ontology (AO), a new module and extension about to be released for [UMBEL](#) (*Upper Mapping and Binding Exchange Layer*). UMBEL is both a vocabulary and an ontology reference structure for concepts. UMBEL's role is to help match the discussion of topics and things across the Web.

The spoke-and-hub design [\[5\]](#) that UMBEL provides for concepts led to an obvious question of how the relationships ("properties") of things can be similarly organized and managed. Referring again to the [prior article](#), we put forward the following stack (as initially informed by Pietranik and Nguyen [\[6\]](#)) about how to look at the data interoperability space from a semantic technologies perspective:



The Ontology Stack

The prior article looked across the stack and noted the importance of relying on open standards. It noted the growing availability of public datasets and knowledge bases to inform our understanding and structure of the various layers in this stack. In this article, however, we mostly concentrate on the Properties layer, and its split between relations and attributes.

Definitions within the Attributes Ontology

Our symbols and the referents in our language are, as Peirce pointed out, a consensual process of society to converge upon the meaning of the words in our language. This is always the tricky thing about language: We think we know what the words or terms mean because of the converging process, but in our individual interpretations we still may have slight differences about what exactly these words or terms encompass and mean. Further, when one is dealing with semantics and meaning, being explicit about the terms used is all the more essential.

Thus, here are the exact definitions as used within UMBEL and the Attributes Ontology:

- **Annotation** -- is a note or *metadata* that may be assigned to ("associated with") a *subject*, *property*, *object*, *record* or *dataset*; annotations are almost always *values*
- **Assertion** -- is a statement in the form of a triple (*subject - property - object/value*), sometimes also referred to as a "fact"

- **Attribute** -- an inherent characteristic of an *entity* that helps further describe or define that *entity*. Every attribute either has a *value* or an assignable member of a set, with such members being *objects*. All entities within a given *entity type* share the same potential attributes
- **Class** -- a set of one or more members that share the same potential *attributes* or *relations*; *concepts* and *entity types* are both classes [7]
- **Concept** -- something that is conceived in the mind that is not a nameable, discrete, tangible thing. Concepts may have *metadata* assigned to them, but not *attributes* as defined herein [19]
- **Dataset** -- is a combination of one or more *records*, transmitted as a single unit (though it may be split into parts due to size)
- **Entity** -- a nameable and tangible thing, which is often independent or separate; can be part of something else when an identifiable thing by name, as "lung" is to "animal body". As used herein, distinct from *concept*. When assigned a proper name, also known as a *named entity* [8]
- **Entity Type** -- a collection of *entities* that have the same potential *attributes* and possible *relations* and are thus members of the same *class*
- **Individual** -- in RDF and Cyc [7] an "individual" is synonymous with an *instance* or *entity*; we try not to use it because of general terminological confusion [9]
- **Identifier** -- a unique string assigned to every *object* in the system. In RDF and UMBEL and the Attributes Ontology this identifier is a [URI](#). The identifier is a reference pointer, and should not be confused with the name or defining label for the *object*; it is a sign pointing to the *object*, but not the *object* itself
- **Instance** -- is a single *entity*, also called the *individual* in RDF
- **Metadata** -- is "data about data", usually in the form of descriptive *annotation*. Metadata provides reifying information about an *object* such as provenance, author, date created or used, version information and the like
- **Object** -- an *entity*, *concept* or *property* that can be referred to via an *identifier* of some sort; in the case of the Attributes Ontology, an object has a [URI identifier](#)
- **Property** -- a relationship between a *subject* and a *value* or *object*, which may either be of the form of a *relation*, *attribute* or *annotation*
- **Record** -- is an *instance* with one or more associated *attributes* and *values*
- **Referent** -- the *object* referred to by an *identifier*
- **Relation** -- is the relationship between two different *entities* or *concepts* (or between an *entity* and a *concept*), which is often hierarchical, mereological or some other form of mutual relationship between the two things [18]
- **Subject** -- is either a *concept*, *entity* or *property* (also collectively known as a "resource" in RDF), and is the item currently under consideration or focus; it is equivalent to the linguistic *subject*
- **Value** -- a string, literal or data value that provides the numerical quantity, or quality or utility of a *subject* in relation to the meaning of its associated *attribute*; a value has no meaning or context absent its paired *attribute*.

To illustrate why defining terms is so important, let's look at three of the terms above, and how they are described in other nomenclatures or terminological systems:

<i>UMBEL/AO Terminology</i>	<i>Terminology Used Elsewhere</i> [10]	
<i>entity type</i>	<ul style="list-style-type: none"> • concept • kind 	<ul style="list-style-type: none"> • collection • type

	<ul style="list-style-type: none"> • set 	<ul style="list-style-type: none"> • class
<i>entity</i>	<ul style="list-style-type: none"> • object • instance • exemplar • element 	<ul style="list-style-type: none"> • member • record • individual • dependent variable
<i>attribute</i>	<ul style="list-style-type: none"> • property • predicate • relationship • feature • facet 	<ul style="list-style-type: none"> • dimension • characteristic • field • header • independent variables

Some Comparisons of Terminology Differences

So, please try to keep in mind the definitions listed in the bullets above when looking at the distinctions in UMBEL or the Attributes Ontology.

Pragmatic Reasons for a Properties Split

Fundamentals matter. If someone were to ask you how the world was organized, what would you say? Does it even matter?

If it does matter, as it does for many with responsibility for getting divisions or people or accounting systems or companies to work with one another -- that is, data interoperability -- then you have to eventually grapple with the "how the world is organized" question. In my early training in plant systematics, we learned about '[lumpers and splitters](#)', taxonomists with different world views that emphasized similarities or distinctions in how they cataloged the world. Other perspectives and worldviews provide similar dichotomies or spectra in how to define and characterize the world. The challenge is how to get multiple parties to buy into an approach to data interoperability that is somehow grounded at a very fundamental level with a basis that all parties can agree as a foundation.

For a decade, I have believed that one part of that foundation should be grounded in RDF [\[11\]](#). I like the simplicity of the RDF *subject - property - object* 'triple' statement. I like that RDF is an open standard. I like the ability of simple RDF statements to be combined into more complicated and sophisticated vocabularies and languages, and then ontologies. I like RDF's applicability to any form of information, its expression in multiple serializations, and its ability to represent virtually any form of data in the wild. Though I think I understand the critics who want more expressiveness still, such as in concept models or higher-order logics, I think we are still working out the basics of human concepts and languages in a machine-understandable context. That is why I continue to try to work in the RDF and OWL sandbox, even though I suspect they will eventually be supplanted by more capable constructs.

Yet one of the things I don't like about RDF is the semantics and terminology of the 'property' construct. I think it conflates relationships between things, which help us to organize and understand connections between objects in the world, with how we describe those things. While OWL provides some improvements in that we now can distinguish between data, object and annotation properties [\[12\]](#),

those distinctions do not really get at the fundamental conflation of the 'property' construct. I further suspect that one of the reasons we have yet to see ABox or instance data-level ontologies to help the data interoperability question -- what I first discussed in the [introduction to the Attributes Ontology](#) -- is this very same conflation.

Fortunately, the extensibility of RDF and OWL via ontologies gives us a method for cleaving this conflation apart.

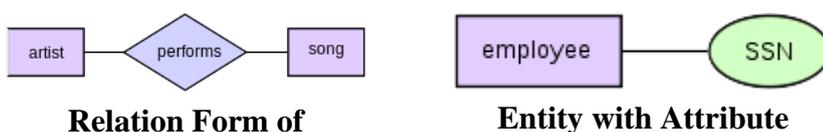
If we assume we can tease apart the fundamental nature and coverage of 'properties', what are the basic conceptual splits that represent this construct? From the aspect of data interoperability, I think we can see three: *relations*, *attributes*, and *annotations*.

The first conceptual split for 'properties' is *relations*. At the most fundamental level, we have things in the world and relationships or connections between those things. When we say things like dogs are a kind of mammal or Lassie is a dog, we are categorizing things by type. When we say that hair or toenails are parts of a mammal we are relating parts of an animal to a whole. When we say that mammals have hair but birds have feathers we are drawing distinctions between the two animal types. These kinds of statements tend to place the objects in our world in relation to one another. By so doing, we provide an organized view of the things in the world and give those very same things *context*. In all of these cases, our statements specify a relation between things that, combined with other relations, provide a schema or conceptualization of how things in the world relate to one another.

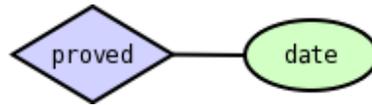
Some of these organizing principles are mental and intellectual constructs for how we group things together, such as dogs and people as mammals or mammals and birds as animals. Some of these organizing principles are ideas or concepts such as truth, beauty and conflict, a richness of terminology that gives us further explanatory power for how to place and give context to the things in our world. Relations between things are thus ultimately contextual in nature; they help to place our understanding of things in connection to other things. This is the portion of RDF 'properties' that we call *relations*, and they are explicitly *excluded* from our Attributes Ontology.

On the other hand, we look to separate the existence of some things different from other things by the nature of their characteristics, what we can observe and describe about that given thing. So, we describe shapes, sizes, weights, ages, colors and characteristics of things with increasingly nuanced vocabularies. We note that grasses have linear or simple leaves, oaks have serrated or wavy-shaped leaves, and carrots have branched or compound leaves. We distinguish hair color, eye color, place of birth, current location and a myriad of factors. Each one of these factors becomes an *attribute* for that object, with the specific values (simple v wavy v compound) distinguishing instances from one another. *Attributes* are the second conceptual split for 'properties'.

These same distinctions were described by Chen in his attempt to find a common ground across network, relational and entity set models [2] in his E-R model. These are represented in pictorial form in the Wikipedia [entity-relationship model](#) article as follows:



Property



Relation with Attribute

E-R Modeling Representation of Properties

Further, in a later elaboration of where his E-R modeling ideas arose, Chen was able to correlate these relationships to natural language [\[13\]](#), which I have updated to reflect the terminology herein:

<i>Word Sense</i>	<i>AO Component</i>
common noun	concept / entity type / entity
proper noun	entity
transitive verb	relation
intransitive verb	attribute
adjective	attribute
adverb	attribute (property)

Mapping of Word Senses to Attributes Ontology Components

Note that attributes may also apply to the relation-type of property.

The third conceptual split for 'properties' is *annotations*, or metadata or "data about data", which can apply to anything. Annotations give us a way to describe the circumstances and provenance of the item at hand. Annotations capture the circumstances or conditions or contexts or observations for the thing at hand. Where did we discover or find it? When did we find or elaborate upon it? By whom or when was it found or elaborated? What is our commentary about it? While these are all external elaborations of the thing at hand, and not intrinsic to the nature of the thing, they are all characterizations about a given thing. In these regards, annotations have as their focus a given object, similar to what is true for attributes. As a result, we have included annotations in the Attributes Ontology as well.

Thus, with respect to RDF 'triples', we can now map the three parts of the assertion statement as follows:

<i>subject</i>	<i>property</i>	<i>object</i>
concept	relation	object
entity type	attribute	value

entity		
--------	--	--

Mapping of an RDF 'Triple' to Attributes Ontology Components

This mapping sets the overall context for how the Attributes Ontology relates to the basic RDF building blocks.

From the Theoretical to the Pragmatic

These kinds of distinctions are not new. In philosophy, related distinctions have been drawn about intrinsic *v* extrinsic properties [\[14\]](#) or intensionality *v* extensionality [\[15\]](#). For conceptual models with specific reference to ontologies, Wand et al [\[16\]](#) in 1999 were making the distinction between *intrinsic properties* (akin to what we term attributes herein) and *mutual properties* between things (what we term relations). Unfortunately, at that time, the conventions of RDF had not yet become prevalent and the idea of annotation properties had not yet emerged (from OWL). These later distinctions are important, but the Wand et al discussion still is helpful to elucidate the same pragmatic and theoretical considerations.

More recently, the DERA initiative from the University of Trento has embraced these same distinctions [\[17\]](#). Unfortunately, no ontology supporting these viewpoints has yet been made public.

We are thus pretty much in virgin territory. While having a sound conceptual and theoretical basis is essential, which apparently we do, the real reasons for carving out an attribute perspective on RDF properties are pragmatic. Since attributes are the properties of an entity, we can better interoperate entity data by concentrating on those aspects that let us match data in one set of records to similar data in totally different records. By building a new vocabulary and structure upon RDF, we can provide a more sophisticated handling of 'properties' than RDF or OWL alone can provide in their native forms.

Specifically, an attribute focus, expressed in an Attributes Ontology, which conforms with open standards and is designed explicitly as a reference grounding, gives us these advantages:

1. A clear focus for organizing the attributes that describe objects, the first step in providing a reference grounding
2. With the assistance of Cyc [\[7\]](#), a logical and coherent basis for organizing a resulting knowledge graph of attribute types
3. A semantic underpinning that enables us to overcome [semantic heterogeneities](#) across all aspects of data expression
4. Immediate access to semantic approaches for inferencing and concept (semset) matching between different data vocabularies
5. A computable structure over which data values in external datasets can be mapped
6. Suitability for graph analytics techniques for clustering, relating or analyzing instance data
7. Because of its conceptual affinity with E-R modeling, a bridge to many other domains and expertise, through vehicles such as UML
8. A logical split of properties that makes RDF a more tractable data model, including for relations [\[18\]](#)

9. An attribute knowledge base for informing artificial intelligence techniques, and
10. A semantic grounding to help us overcome the data interoperability bottlenecks.

Making these distinctions operational, in part, is the purpose of the Attributes Ontology.

As a relatively cutting-edge effort, we expect some false steps and likely hiccups as we move to put in place this reference structure. However, we think this effort to be both innovative and essential to ongoing use of semantic technologies to tackle the decades-long challenges of data interoperability.

Other Related Concepts

Besides the other links mentioned in this article, here are some additional articles on Wikipedia that provide other and varied perspectives on the concepts and terminology used herein:

- [Attribute \(computing\)](#)
- [Attribute-value pair](#)
- [Attribute-value system](#)
- [Concept map](#)
- [Conceptual model](#)
- [Entity-relationship model](#)
- [Description logic](#)
- [Formal concept analysis](#)
- [Functor](#)
- [Intensional definition](#)
- [Intension](#)
- [Intensional logic](#)
- [Intrinsic and extrinsic properties \(philosophy\)](#)
- [Level of measurement](#)
- [Predicate \(mathematical logic\)](#)
- [Property \(philosophy\)](#)
- [Qualitative property](#)
- [Quantity](#)
- [Theory of relations](#)
- [Variable and attribute \(research\)](#)

The Entity Segregation

The next article in this series introducing the Attributes Ontology will discuss the related basis for segregating out 'entities' in UMBEL. The entities and attributes work closely with one another to aid data mapping and interoperability.

[1] For more discussion of Peirce in relation to semantic Web issues, see [9]. According to the [ER model entry](#) in Wikipedia, it is in accord with philosophic and theoretical traditions from the time of the Ancient Greek philosophers: Socrates, Plato and Aristotle (428 BC) through to modern epistemology, semiotics and logic reflecting the views of Peirce, [Frege](#), [Russell](#) and [Carnap](#).

[2] Peter Chen, 1976. "[The Entity-Relationship Model - Toward a Unified View of Data](#)", in *ACM Transactions on Database Systems* 1 (1): 9–36, March 1976.

[3] Ralph R. Swick and Henry S. Thompson, eds., 1999. [The Cambridge Communiqué](#), a *World Wide Web (W3C) Note*, 7 October 1999.

[4] Patrick Hayes, 2004. "RDF Semantics," a *W3C Recommendation*, February 2004. See <http://www.w3.org/TR/rdf-mt/>.

[5] A [spoke-and-hub design](#) (n-1) for data mapping is tremendously more efficient than the most common approach of pairwise mappings (a quadratic function). For example, ten datasets would require 9 composite mappings in a spoke-and-hub design versus 45 in a pairwise approach. And, of course, datasets themselves contain tens to thousands of attributes, compounding the map scaling problem further

[6] 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.

[7] UMBEL is based on the [OpenCyc](#) version of the [Cyc knowledge base](#); Cyc uses the term "collections" to refer to RDF classes.

[8] You sometimes see entities defined as "self-contained"; that is not strictly followed here. What is more important is being a nameable thing with attributes.

[9] M. K. Bergman, 2012. "[Give Me a Sign: What Do Things Mean on the Semantic Web?](#)," in *AI3::Adaptive Information* blog, January 12, 2012. See especially the discussion of the toucan.

[10] Some of the terminology sources are E-R, concept maps, RDF, Cyc, descriptive logics, data modeling (of various types), OWL, etc. Most particularly, sometimes entities are referred as including concepts; the Attributes Ontology does not.

[11] I have been but one of many arguing for the benefits of RDF. For my views, see M. K. Bergman, 2009. "[Advantages and Myths of RDF](#)" in *AI3::Adaptive Information* blog, April 8, 2009.

[12] See Pascal Hitzler, Markus Krötzsch, Bijan Parsia, Peter F. Patel-Schneider, Sebastian Rudolph, eds., 2012. [W3C Recommendation](#), *World Wide Web Consortium*, 11 December 2012, for the annotation - object - datatype property distinction.

[13] Peter Pin-Shan Chen, 1997. "[English, Chinese and ER diagrams](#)," in *Data & Knowledge Engineering* 23, no. 1 (1997): 5-16.

[14] Stanford Encyclopedia of Philosophy, 2012. "[Intrinsic vs. Extrinsic Properties](#)", online article, first published January 5, 2002; substantive revision December 23, 2012.

[15] At least for [Carnap](#), he thought "*...the full meaning of a concept is constituted by two aspects, its intension and its extension. The first part comprises the embedding of a concept in the world of concepts as a whole, i.e. the totality of all relations to other concepts. The second part establishes the referential meaning of the concept, i.e. its counterpart in the real or in a possible world*".

[16] Yair Wand, Veda C. Storey, and Ron Weber, 1999. "[An Ontological Analysis of the Relationship Construct in Conceptual Modeling](#)," in *ACM Transactions on Database Systems (TODS)* 24, no. 4 (1999): 494-528, December 1999. Also see, Jeffrey Parsons and Yair Wand, 2003. "[Attribute-Based Semantic Reconciliation of Multiple Data Sources](#)," in *Journal on Data Semantics* LNCS 2800.

[17] 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). DERA, in fact, stands for domain, entities, relations and attributes, but is mostly derived from the work of [S.R. Ranganathan](#), the 20th century Indian library theorist and mathematician. DERA appears to have evolved into later projects, but the provenance is unclear.

[18] This provides a logical invitation to a similar 'relations ontology' for capturing the role and relationship aspects of RDF properties. Such relations stipulate topographical relationships, hierarchical relationships (subClassOf, fatherOf, daughterOf), mereological relationships (partOf, isComponent), role relationships (isBossOf, hasTeacher, isKeyInfluencer) or approximation relationships (isLike, isAbout, relatesTo).

[19] Actually, this is a bit more complex than the definition. In OWL modeling, a "concept" may also act as an instance ("individual" in RDF terminology) through what is known as metamodeling; see M. K. Bergman, 2010. "[Metamodeling in Domain Ontologies](#)" in *AI3::Adaptive Information* blog, September 20, 2010.

PDF generated by *AI3::Adaptive Information* blog