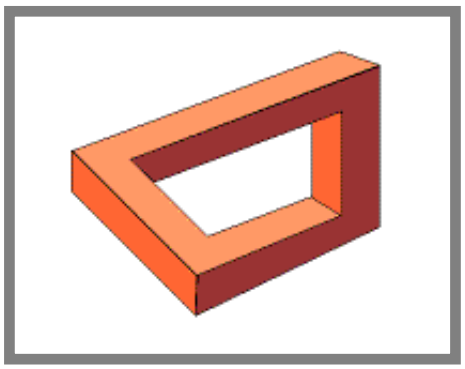
Ontologies as the 'Engine' for Data-Driven Applications

by Mike Bergman - Wednesday, June 10, 2009

http://www.mkbergman.com/492/ontology-best-practices-for-data-driven-applications-part-3/



Ontology Best Practices for Data-driven Applications: Part 3

In my *Intrepid Guide to Ontologies* from a couple of years back, I noted that "Ontology is one of the more daunting terms for those exposed for the first time to the semantic Web." And, for sure, if one starts to peruse the current discussions ranging from the Ontolog Forum to major academic symposia (not meaning to single anyone out), it is clear that the idea of developing "ontologies" is often freighted with much weight, hot air, and (by implication) cost.

This is both a shame because, firstly, it is unnecessary and not often true. And, secondly, because the whole pragmatic idea of what an ontology is and what it can do has often gotten lost in the shuffle.

To be sure, there have been massive standards efforts and EU-funded mega-projects devoted to ontologies. There are certainly cases where coordination of specific domains such as petroleum or integration with a complicated supplier base such as in airline manufacture warrant these massive, complicated ontology development efforts.

But, from my vantage, these extremes overshadow the vast majority of more prosaic, pragmatic applications of ontologies. Remember, ontologies are merely a means of describing a conceptual view of the world [1]. If one defines that "world" within focused and appropriate scope, it is surprising (we believe) how much mileage can be extracted from these suckers.

As we see a breakthrough of interest in semantic Web and linked data principles applied to the enterprise, as wonderfully described in the recent seminal PricewaterhouseCoopers quarterly Technology Forecast, also notably with a prominent focus on ontologies, I think it is time to direct all guns on prior bad assumptions and bad anecdotes. To wit:

- Ontology development need not be a comprehensive, self-contained definition of a "big picture." Ontologies can be focused, limited, and grow and change as needed
- Ontology development need not be expensive. Whoever is selling six-figure ontology

development to businesses ought to be taken out and shot. Start small and focused; frankly, a simple spreadsheet taxonomy or quick conversion of existing XML or metadata or vocabulary standards is A-OK to get started
- Ontology development is not massive and static: rather, it is small and flexible and incremental as more is brought in and more is learned
- Ontology development is not some imperative for conceptual "truth"; rather, it is a very adaptable means for stating, testing and refining stuff
- Ontology development is certainly no massive relational schema; by its nature it is malleable with nary a whiff of "lock-in", and
- Most importantly, ontology development is a way of "driving" applications and user interfaces and reports.

In fact, it is the last point that no one is discussing today, but it is the most important of the lot: Ontologies, properly crafted, can be the 'engines' for data-driven applications.

It is this latter point that is a true paradigm shift and one of the most exciting prospects of ontologies.

Manifest Uses

Ontologies, for sure, are a formal representation of conceptual relations, a "world view." But, that world view need not carry with it the freight of trying to describe all human knowledge. It can (and should) be restricted to an understandable scope (domain) and purpose. In that vein, what does such a "world view" need?

Let's first talk about scope. We don't need a "global ontology" that is accepted by everybody on Earth. What we need are focused ontology(ies) for describing things within a given problem space (whose data may reside in a single dataset or aggregate of datasets). We need to communicate how this system describes the things within its domain and how it understands the concepts and attributes associated with its problem space and data. This communication is published as the ontology. Rather than a global, comprehensive schema, we simply need these well constructed bricks, one by one.

Then, the ontology itself needs to be understandable and manageable. Ontologies should be readable by machines, but too many see ontologies solely through the lens of machines. I believe that to be a mistake. While importantly needing to be designed for machine ingest, I believe the real purpose of ontologies is for humans. How do we label things? How do we describe and define things? How do we find things? How do we organize things? How can these understandings be brought before us in the software that we use?

These types of questions lead us to the pragmatic and pull us back from the abstract. If we keep foremost the simple idea that ontologies are merely structures for how to organize (schema) and describe (vocabularies) our problem space at hand, then we can actually get on with cutting the bull and getting real stuff done.

Let's take as an example our structWSF Web services framework that I will be announcing and demoing for the first time at SemTech 2009 next week. We developed a simple and flexible ontology to describe what a "Web services framework" should be. Then, we developed and implemented the software to make

it happen. This means that an ontology development task can be seen as a specification task, too.

Pragmatic Applications of Ontologies

So, OK, what do these exhortations mean? Without respect to any particular scope or domain, let me then list below some important functional areas to which ontologies -- properly and pragmatically designed -- can contribute.

Conceptual Relationships

The traditional lens for viewing ontologies is as a means to express conceptual relationships. We agree.

However, ontologies need not have large and nuanced predicate (relationship) vocabularies in order to be useful. Relatively simple but powerful structures with hierarchical or part-of relationships can be very effectively employed for inferencing or faceted searching. From a pragmatic standpoint, let's first agree on what "things" (nouns) there are in our domains, then let's worry about how they relate (verbs).

The idea here has long been known as successive approximation: Let's first get ourselves into the right country, then right province, then right city, then right neighborhood, then right house, and then right room. Only then should we worry about the condition of the paint or the age of the floors.

Endless harangues about "true" conceptual relations are a hindrance, not a help, to this perspective. It is much better (and faster, cheaper and more pragmatic) to put forward simple but coherent relationships than to worry about what all of that "really" means. From a business perspective, isn't being able to utilize the assertion that the hip bone is connected to the thigh bone more important than having to await a full explication of all of the muscles and ligaments and tendons that might comprise them?

Once such simple relationship structures are embraced, then amazing inferencing power comes to the fore. If one searches on thigh bone, inferencing can also bring forward the hip because of its relationships to the leg.

Integrating Instance Data

OK, so now at least we have a coherent scaffolding of concepts and their straightforward relationships. That is, is concept A a "bigger" one (class or super set) than concept B? (Other simple relations could be substituted.) If so, we now see a bit of an organizing "world view" begin to emerge.

So, now we begin to bring in external data. But that data and its schema describe themselves differently. In one realm it is "foo"; in the other, "bar".

While this different terminology for the same "thing" or related things may not be known at the outset, it is discoverable. And, when discovered, it is quite easy to associate the idea or concept of "foo" in one dataset to "bar" in another. In this manner, through learning and accretion, we are able to associate more and more similar things to one another.

We did not need to begin with some global, cosmic view to begin relating this data to one another. We

only needed the right framework and structure that allowed this association to evolve as the learning occurred.

And, oh, by the way: this very same process is akin to documenting the organization's institutional memory.

Orienting to Other Knowledge and Domains

Being able to relate and "classify" or "organize" some things to other things also means that we are now beginning to create a roadmap for how "stuff" in a broad sense relates to other "stuff." For example, if I develop a detailed understanding of the hip bone, I can now bring that body of knowledge into the context above to relate this new information to the thigh and to the leg.

Frankly, at this juncture, while perhaps ultimately important, it is helpful merely to know that Domain A (hip) is somehow related to Domain B (thigh). Think of the issue more like trying to get into the right map vicinity on the globe, and not whether individual streets intersect.

Again, the mindset here is one of letting ontologies and their concepts get related knowledge bases into the same ballpark. Whether we are trying to match Little League ball players with Major League ballplayers is beside the point: accept that both are playing baseball, then decide the importance and specifics of the relationship in a later step.

Again, "ballpark" is more helpful than no connection whatsoever. Silly statements about "ontological commitments" really mean nothing. Ontologies, like any other tool, can play different roles at different times. When helping to get like-related things into the same ballpark, ontologies are easy and quite effective.

(As an aside, it is useful to note here that our efforts with the UMBEL upper-level subject ontology are solely premised on this "roadmap" purpose. In and of itself, UMBEL is not a very complicated explication of the world. But it does provide a comprehensive set of 20,000 subject concepts for orienting quite disparate datasets and information to one another. This very same approach could be replicated and then applied to the granularity of individual domains, kind of like zooming in on Google Maps, to provide similar benefits at smaller scale for domain-specific roadmaps. In fact, that is a common approach we apply in our own client ontologies, which we then also make sure we tie into UMBEL for global orientation.)

Mapping to Other Schema

OK, so with this foundation now built, we can next raise the bar a bit further. Once one begins to express these "world views" formally as an ontology, even with reduced ambitions as presented above, one still ends up with a formal specification of that conceptualization. And, that means, we now also have a basis with standard languages for mapping two disparate or separately developed ontologies to one another.

This is powerful. Through such mapping, we end up, in the memorable phrase of my colleague, Fred Giasson, "exploding the domain."

Moreover, we also have found a means for stitching together datasets with disparate schema to one another. Voilà: We now have met the Holy Grail of data interoperability.

In my opinion, this is the money shot from all of this effort. But, again, if we set the deployment threshold to the unrealistic levels that some ontology pundits suggest, this payoff is unlikely to happen. We are not trying to state absolute, universal truth about anything nor to be unrealistically comprehensive. All we are trying to do is make defensible assertions that one portion of a world view is similar or related to a portion of a different world view.

Now, does that sound that scary? No, of course not. It is merely a reasonable and pragmatic means for relating two structures together.

A key aspect to this mapping ability is to enrich the description of our concepts with what we call "semsets." Semsets are a listing of related terms and phrases that provide synonyms, aliases, jargon and related context for alternative ways to describe or bound the concept at hand. This terminological "grist" is the basis for relatively straightforward natural language processing techniques to suggest matches between concepts in different ontologies (which might also be combined with other ontology components such as preferred labels, descriptions or structural placements in the schema).

Like many of the points above, these semsets can be built incrementally and over time as new jargon and terminology is discovered.

Linked Data, with Federated and Comprehensive Data

These techniques of mapping datasets and their ontology structures can be leveraged still further with the proper application of linked data practices. Via linked data, we place our data into Web-accessible (HTTP) networks and give them Web-scalable identifiers (URIs). This means we can now integrate and interoperate with much external public Web data and break down our own internal data silos.

Our instance records can be fleshed out with supplementary sources to provide more comprehensive attibutes and characterizations. Uniformity of treatment and coverage is promoted. Data interoperability is finally at hand.

A key best practice to this, of course, needs to be the recognition that not all data or information is public and not all users have the same roles or should have the same access to different sets of data. Thus, to embrace global mechanisms for data interoperability, there must also be local methods for enforcing access, privacy and confidentiality.

Properly designed ontologies can fulfill this requirement, as well. By organizing information into datasets and setting profiles for access and CRUD (*create - read - update - delete*) rights, an effective environment for data sharing and federation is established.

Context- and Instance-sensitive Data Display

To this point, we have taken almost an exclusively data- or schema-centric view of ontologies. But, as structures, pure and simple, their structural nature can be exploited in other ways. It is here, frankly, that

less is spoken of the potential for ontologies than in the more "conceptual" areas noted above.

The first of these new areas is in instance-sensitive data display. Each instance record is associated with an instance type in a governing ontology. Detecting this type means that context-sensitive display templates can then be invoked.

Detecting that something refers to a city, for example, can invoke a template providing a map, population figures, area size, city governance method and the like. In contrast, detecting an instance as a camera might invoke an entirely different display template focusing on product features or price or store and purchasing locations. Such instance-type displays are common; they are known as "infoboxes" within Wikipedia articles, as one example.

But this power of data display templates can be generalized further. What if we detect our instance represents a camera but do not have a display template specific to cameras? Well, the ontology and simple inferencing can tell us that cameras are a form of digital or optical products, which more generally are part of a product concept, which more generally is a form of a human-made artifact, or similar.

By tracing this inferencing chain from the specifc to the more general we can "fall back" until a somewhat OK display template is discovered, even in the absence of the better and more specific one. Then, if we find we are trying to display information on cameras frequently, we only need take one of the more general, parent templates and specifically modify it for the desired camera attributes. We also keep presentation separate from data so that the styling and presentation mode of these templates is also freely modifiable.

This parallel set of display structures to the domain ontology provides a highly reusable and leveraged data presentation framework. For 30 years organizations have struggled with report generators and all sorts of complicated systems for responsive reporting and data display. When driven by ontologies, this challenge is greatly simplified.

Driving User Interfaces

The careful reader of the above will note that our ontologies now have a number of interesting characteristics, all of which can be leveraged within the user interface. For example, we have:

- Human-readable labels for our "things"
- Alternative labels in our semsets that can characterize those same "things"
- A readable description of each "thing"
- An organized and logical schema for how each "thing" relates to other things.

This very information, when indexed in a supplementary full-text search engine with faceting capabilities (such as the Solr engine we use), can be leveraged in the user interface for these types of desired UI capabilities:

- Attribute labels and tooltips
- Navigation and browsing structures and trees
- Menu structures

- Auto-completion of entered data
- Contextual dropdown list choices
- Spell checkers
- Online help systems
- Etc.

This is absolutely mindblowing power!

We can now design generic tools that do patterned functions. Then, based on the data at hand and the ontologies that describe them, we can now see completely modified and tailored interfaces. And all of this is done without modifying a single line of application code!

Applications in this brave new world now consist of assembling a proper suite of generic tools, and then spending the bulk of our time on describing and characterizing our data via ontologies and refining templates for displaying or reporting the types of specific instances within our current problem space.

Conclusions

All of the points made above are doable and being done today. Properly designed ontologies can readily deliver all of the aspects noted above. Later parts in this ongoing series will address many of those aspects in greater detail.

Ontologies are not magic. Properly done -- an important emphasis -- ontologies are the pivot point for faster and more adaptable ways of doing business. A simple, pragmatic mindset can help.

Our perspective is that ontologies are really the "flour that gets backed into the cake". While viewable and definable as their own structures, properly constructed ontologies actually should exist everywhere within applications and contribute everywhere ***to*** applications. This is what we mean by "data-driven applications."

To be sure, we are suggesting a paradigm shift from 30 years of IT frustrations: schema no longer must be fragile; reports no longer must be costly and delayed; and data can finally be made interoperable.

We will continue to give you our best thinking on these topics over the coming weeks and how they might be important to you.

Sound too good to be true? Read the material above again. And, then, we welcome getting your call.

This post is part of an occasional **AI3** series on ontology best practices.

[1] As used in knowledge representation or information science, 'ontology' is most often defined using Tom Gruber's "explicit specification of a conceptualization." See Thomas R. Gruber, 1993. "A Translation Approach to Portable Ontology Specifications," in *Knowledge Acquisition* **5(2):** 199-220; see http://tomgruber.org/writing/ontolingua-kaj-1993.pdf.
