

## Semantic APIs

The experience with RDF, Linked Data, XTM and other semantic technologies proves Steve Newcomb's contention that “new” formats (with their own semantics) don't displace old ones, they just add to the semantic stew. In part because no one wants to pay to migrate legacy data and in part because systems that use older formats persist. You do know that **all credit card transactions** are handled by systems written in COBOL. Yes?

The argument that we are facing a growing amount of data, swimming in a semantic stew with ever more ingredients, doesn't surprise anyone. I am often amazed that we (including myself) even bother mentioning it. No one would argue against it. Maybe that is why we do it, it is a safe starting point.

But it doesn't get us any closer to a solution that could be the basis for a standard and/or successful commercial technologies.

Consider the new service by [Factual](#) called [Resolve](#):

The Internet is awash with data. Where ten years ago developers had difficulty finding data to power applications, today's difficulty lies in making sense of its abundance, identifying signal amidst the noise, and understanding its contextual relevance. To address these problems Factual is today launching Resolve — an entity resolution API that makes partial records complete, matches one entity against another, and assists in de-duping and normalizing datasets.

The idea behind Resolve is very straightforward: you tell us what you know about an entity, and we, in turn, tell you everything we know about it. Because data is so commonly fractured and heterogeneous, we accept fragments of an entity and return the matching entity in its entirety.

It looks quite good, for geographic locations. That is handling a sub-set of geographic locations using a sub-set of its API for any given country. Resolve is going to be quite useful for some purposes, but not every semantic integration use case involves geographic locations.

So, what can we do that doesn't toss more ingredients into the semantic soup and at the same time is more broadly applicable than data sets that happen to share common characteristics?

I think Factual is on the right path with its data API, except that Factual is managing the data behind the API. A real bottleneck if you are interested in “big data” spread across the global information space.

Oct 28, 2011

I don't think we can escape contributing some ingredients into the semantic soup but I do think we can make ours a spice.

Data sets, at least a large number of them, have APIs of various kinds for accessing those data sets. My suggestion is that we create a standard for the creation of **semantic APIs**, a number of which could be produced by WG 3 both as proof of concept as well as continuing work on the **semantic API** standard.

A **semantic API**, at a minimum, repeats the terminology of the dataset, with at least one mapping to another terminology, with the causes/reasons for the mapping being cited. Semantic APIs should be capable of noting mappings to other, related data sets.

Any dataset admits to one or more such APIs with different target mappings. It may or may not be the case that different target mappings can be interchanged based on the mapping from the source dataset. It is certainly possible that different target mappings can be interchanged on the basis of a common source but I don't think authors of target mappings should be so constrained. The intent of such a standard being to ease the use of data, not to impose what seem to us useful data practices.

The use of the semantics defined by the Semantic API standard should not require any particular methodology or format. Users can choose anything from blind mappings of enterprise integration software to common logic. How or why the data is put to some purpose is beyond our purview.

I would suggest that the **semantic API** side be written in XML but that the target side of a mapping be capture in its native format, if at all possible. I would also like to see the development of a semantic API for several public data sets to be issued as technical reports when the standard is completed.

Patrick Durusau  
Covington, GA

PS: At least for some public datasets in the United States, programming APIs have been developed for common access. Those may be a starting point for developing semantic APIs. A lot of the ground work has been done and with the “shim” of a semantic API, it may be possible to demonstrate interoperability between such projects, without changing their internal APIs. (This is the 1990's or was that the 1980's(?) notion of data wrappers with a semantic twist.)

Oct 28, 2011