'Structs': Naïve Data Formats and the ABox

by Mike Bergman - Thursday, January 22, 2009

http://www.mkbergman.com/471/structs-naive-data-formats-and-the-abox/

Writing and Sharing Data Can be Lightened Up

Ever since I first started to learn in earnest about ontology, something has been gnawing at me. The term seemed to be (shall I say?) an obtuse one whose obscurity was not the result of subtle precision or technicality, but rather one of fuzziness. As I introduced my Intrepid Guide to Ontology two years ago, I noted:

The root of the [ontology] term is the Greek *ontos*, or *being* or the *nature of things*. Literally and in classical philosophy, ontology was used in relation to the study of the nature of being or the world, the nature of existence. Tom Gruber, among others, made the term popular in relation to computer science and artificial intelligence about 15 years ago when he defined ontology as a "formal specification of a conceptualization."

Since then, I have continued to find ontology one of the hardest concepts to communicate to clients and quite a muddled mess even as used by practitioners. I have come to the conclusion that this problem is not because I have failed to grasp some ephemeral nuance, but because the term as used in practice is indeed fuzzy and imprecise.

What Isn't an Ontology?

Even two years ago, I noted more than 40 different types of information structure that have at one time or another been labelled as an example of an "ontology":

- Tag cloud
- Controlled vocabulary
- Thesauri
- Collaborative tagging
- Folk taxonomy
- Directory
- Subject Map
- Semantic Web
- Cladistics
- Social bookmarking
- Tags
- Tagging
- Taxonomy
- Folksonomy
- Classification
- Categorization
- RDF
- Metadata
- Ontology
- Microformats
- Data dictionary
- OPML
- XOXO
- OWL
- Subject Trees
- Information Architecture
- Data Reference Model
- Topic Maps
- Concept Maps
- Synsets
- Glossary
- WordNet
- Metadata
- Facets
- Structure
- Dublin Core

- Markup languages
- Systematics
- Phylogeny
- Typology

Since then, I could add even more terms to this list.

Lack of precision as to what ontology means has meant that it has been sloppily defined. As I have harped upon many times regarding semantic Web terminology, this is a sad state of affairs for the semWeb endeavor that has *meaning* at the core of its purpose.

I'm pretty sure that the original intent in embracing the concept of ontology within the realm of knowledge representation was not to see this term so broadly misused or mis-applied. I suspect, as well, that if we could sharpen up our understanding and remove some of the fuzziness that we could improve communications with the lay public across many levels of the semWeb enterprise.

The Useful Distinction of the TBox and ABox

Recently, I have been looking to the semantic Web's roots in description logics. One of my writings, Thinking 'Inside the Box' with Description Logics, looked at the conceptual distinctions between the so-called 'TBox' and 'ABox'. That is, a knowledge base is a logical schema of roles and concepts and the relationships between them (the *TBox*), which is populated by the actual data (instances) asserting memberships and attributes ("facts") (the *ABox*).

By analogy, in a conventional relational database system, the database or logical schema would correspond to the TBox; the actual data records or tables would correspond to the ABox. Often, the term *ontology* is used to cover both ABox and TBox statements (which, I argue, only makes the understanding of the 'ontology' concept more difficult).

My recent writing, Back to the Future with Description Logics, discussed at some length the advantages of keeping the TBox and ABox separate. This current article now expands on those thoughts, particularly with respect to the definition and understanding of ontology.

The starting point for this new mindset is to return to the ideas of data records or data tables *v.* the logical schema that is prevalent in relational databases.

So Many Structs, So Little Time

The last time I took a census, about a year ago, there were more than 100 converters of various record and data structure types to RDF [2]. 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. That last census listed these converters:

- RDF Serialization formats:
- REST-style Web service APIs:
  - Google Base

    - RDF/XML
    - N3
    - Turtle
  - Automatically recognized ontologies:
    - SIOC
    - SKOS
    - FOAF
    - AtomOWL
    - Annotea
    - Music Ontology
    - Bibliographic Ontology
    - EXIF
    - vCard
    - Others
- (X)HTML pages
- HTML header metadata
  - Dublin Core
- Embedded microformats
  - eRDF
  - RDFa
  - hCard
  - hCalendar
  - XFN
  - xFolk
- Syndication Formats:
  - RSS 2.0
  - Atom
  - OPML
  - OCS
  - XBEL (for bookmarks)
- GRDDL [1]
    - Flickr
    - Del.icio.us
    - Ning
    - Amazon
    - eBay
    - Freebase
    - Facebook
    - raw HTTP
    - Etc.
- Files (multitude of file formats and MIME types, including):
    - MS Office
    - OpenOffice
    - Open Document Format
    - images
    - audio
    - video
    - Etc.
- Web services:
    - BPEL
    - WSDL
    - XBRL
    - XBEL
- Data exchange formats
    - iCalendar
    - vCard
- Virtuoso VADs
- OpenLink license files
- Third party metadata extraction frameworks:
    - Aperture
    - Spotlight
    - SIMILE RDFizers