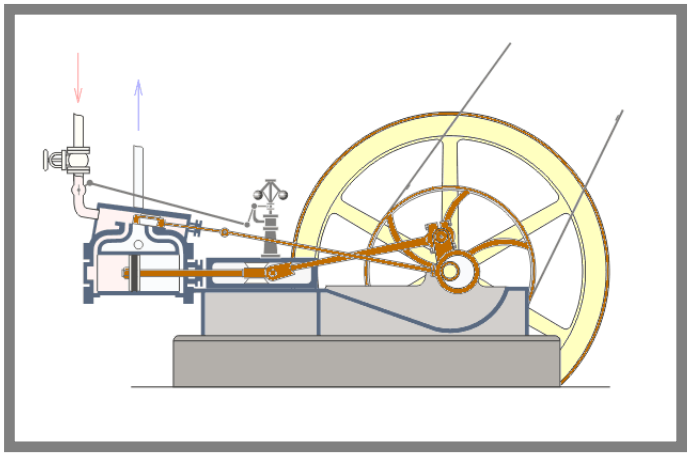
Big Structure and Data Interoperability

by Mike Bergman - Monday, August 18, 2014

http://www.mkbergman.com/1782/big-structure-and-data-interoperability/



A Critical Fit with the Semantic Web and AI

In the first parts of this series we introduced the idea of *Big Structure*, and the fact that it resides at the nexus of the semantic Web, artificial intelligence, natural language processing, knowledge bases, and Big Data. In this article, we look specifically at the work that Big Structure promotes in data interoperability as a way to clarify what the roles these various aspects play.

By its nature, data integration (the first step in data interoperability) means that data is being combined across two or more datasets. Such integration surfaces all of the myriad aspects of semantic heterogeneities, exactly the kinds of issues that the semantic Web and semantic technologies were designed to address. But resolving semantic differences can not be fulfilled by semantic technologies alone. While semantics can address the basis of differences in meaning and context, resolution of those differences or deciding between differing interpretations (that is, ambiguity) also requires many of the tools of artificial intelligence or natural language processing (NLP).

By decomposing this space into its various sources of semantic heterogeneities -- as well as the work required in order to provide for such functions as search, disambiguation, mapping and transformations -- we can begin to understand how all of these components can work together in order to help achieve data interoperability. This understanding, in turn, is essential to understand the stack and software architecture -- and its accompanying information architecture -- in order to best achieve these interoperability objectives.

So, this current article lays out this conceptual framework of components and roles. Later articles in this series will address the specific questions of software and information architectural design.

Data Interoperability in Relation to Semantics

Semantic technologies give us the basis for understanding differences in meaning across sources, specifically geared to address differences in real world usage and context. These semantic tools are essential for providing common bases for relating structured data across various sources and contexts. These same semantic tools are also the basis by which we can determine what unstructured content "means", thus providing the structured data tags that also enable us to relate documents to conventional data sources (from databases, spreadsheets, tables and the like). These semantic technologies are thus the key enablers for making information -- unstructured, semi-structured and structured -- understandable to both humans and machines across sources. Such understandings are then a key basis for powering the artificial intelligence applications that are now emerging to make our lives more productive and less routine.

For nearly a decade I have used an initial schema by Pluempitiwiriyawej and Hammer to elucidate the sources of possible semantic differences between content. Over the years I have added language and encoding differences to this schema. Most recently, I have updated this schema to specifically call out semantic heterogeneities due to either *conceptual* differences between sources (largely arising from schema differences) and value and *attribute* differences amongst actual data. I have further added examples for what each of these categories of semantic heterogenities means [1].

This table of more than 40 sources of semantic heterogeneities clearly shows the possible impediments to get data to interoperate across sources:

Class	Category	Subcategory	Examples	Type [2] [4]
LANGUAGE	Encoding	Ingest Encoding Mismatch	For example, ANSI *v* UTF-8 [3]	Concept
		Ingest Encoding Lacking	Mis-recognition of tokens because not being parsed with the proper encoding [3]	Concept
		Query Encoding Mismatch	For example, ANSI *v* UTF-8 in search [3]	Concept
		Query Encoding Lacking	Mis-recognition of search tokens because not being parsed with the proper encoding [3]	Concept
	Languages	Script Mismatch	Variations in how parsers handle, say, stemming, white spaces or hyphens	Concept
		Parsing / Morphological Analysis Errors (many)	Arabic languages (right-to-left) *v* Romance languages (left-to-right)	Concept
		Syntactical Errors	Ambiguous	Concept

		(many)	sentence references, such as *I'm glad I'm a man, and so is Lola* (Lola by Ray Davies and the Kinks)	
		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 “home phone,” “work phone” and “cell phone”	Concept
	Aggregation	Intra-aggregation	When the same population is divided differently (such as, Census *v* Federal regions for	Concept

			states, England *v* Great Britain *v* United Kingdom, or full person names *v* first-middle-last)	
		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 more datasets	Attribute
		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	Concept

			scope and reference are not (for example, Berlin the city *v* Berlin the official city-state)	
			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 different 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, or when same attribute names (say, Hair *v* Hair) may refer to different attribute scopes (say, Hair *v* Fur) or where values for these attributes may be the same but refer	Attribute

		Attribute-value to Element-label Mapping	to different actual attributes or where values may differ but be for the same attribute and putative value.Many of the other semantic heterogeneities herein also contribute to schema discrepancies	Attribute
		Element-value to Attribute-label Mapping		Attribute
		Attribute-value to Attribute-label Mapping		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 aggregate units (such as thousands or millions)	Attribute
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*	Attribute

			currency names	
		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

Ultimately, since we express all of our content and information with human language, we need to start there to understand the first sources in semantic differences. Like the differences in human language, we also have differences in world views and experience. These differences are often conceptual in nature and get at what we might call differences in real world perspectives and experiences. From there, we encounter differences in our specific realms of expertise or concern, or the applicable domain(s) for our information and knowledge. Then, lastly, we give our observations and characterizations data and values in order to specify and quantify our observations. But the attributes of data are subject to the same semantic vagaries as concepts, in addition to their own specific challenges in units and measures and how they are expressed.

From the conceptual to actual data, then, we see differences in perspective, vocabularies, measures and conventions. Only by systematically understanding these sources of heterogeneity -- and then explicitly addressing them -- can we begin to try to put disparate information on a common footing. Only by reconciling these differences can we begin to get data to interoperate.

Some of these differences and heterogeneities are intrinsic to the nature of the data at hand. Even for the same putative topics, data from French researchers will be expressed in a different language and with different measurements (metric) than will data from English researchers. Some of these heterogeneities also arise from the basis and connections asserted between datasets, as misuse of the *sameAs* predicate shows in many linked data applications [5].

Fortunately, in many areas we are transitioning by social convention to overcome many of these sources of semantic heterogeneity. A mere twenty years ago, our information technology systems expressed and stored data in a multitude of formats and systems. The Internet and Web protocols have done much to overcome these sources of differences, what I've termed elsewhere as climbing the data federation pyramid [6]. Semantic Web approaches where data items are assigned unique URIs are another source of making integration easier. And, whether all agree from a cultural aspect if it is good, we are also seeing English become the *lingua franca* of research and data.

The point of the table above is not to throw up our hands and say there is just too much complexity in data integration. Rather, by systematically decomposing the sources of semantic heterogeneity, we can anticipate and accommodate those sources not yet being addressed by cultural or technological conventions. While there is a large number of categories of semantic heterogeneity, these categories are also patterned and can be anticipated and corrected. These patterned sources inform us about what kind of work must be done to overcome semantic differences where they still reside.

Work Components in Data Interoperability

The description logics that underly the semantic Web already do a fair job of architecting this concept-attribute split in semantics. The concept split is known as the TBox (for terminological knowledge, the basis for T in TBox) and represents the schema or taxonomy of the domain at hand. The TBox is the structural and intensional component of conceptual relationships. The second split of instances is known as the ABox (for assertions, the basis for A in ABox) and describes the attributes of instances (individuals), the roles between instances, and other assertions about instances regarding their class membership with the TBox concepts [7].

The semantic Web is a standards-based effort by the W3C (World Wide Web Consortium); many of its accomplishments have arisen around ontology and TBox-related efforts. Data integration has putatively been tackled from the perspective of linked data, but that methodology so far is short on attributes and property-mapping linkages between datasets and schema. There are as yet no reference vocabularies or schema for attributes [8]. Many of the existing linked data linkages are based on erroneous *owl:sameAs* assertions. It is fair to say that attribute and ABox-level semantics and interoperability have received scarce attention, even though the logic underpinnings exist for progress to be made.

This lack on the attributes or ABox-side of things is a major gap in the work requirements for data interoperability, as we see from the table below. The TBox development and understanding is quite good; and, a number of reference ontologies are available upon which to ground conceptual mappings [9]. But the ABox third is largely missing grounding references. And, the specialty work tasks, representing about the last third, are needful of better effectiveness and tooling.

For both the TBox and the ABox we are able to describe and model concepts (classes), instances

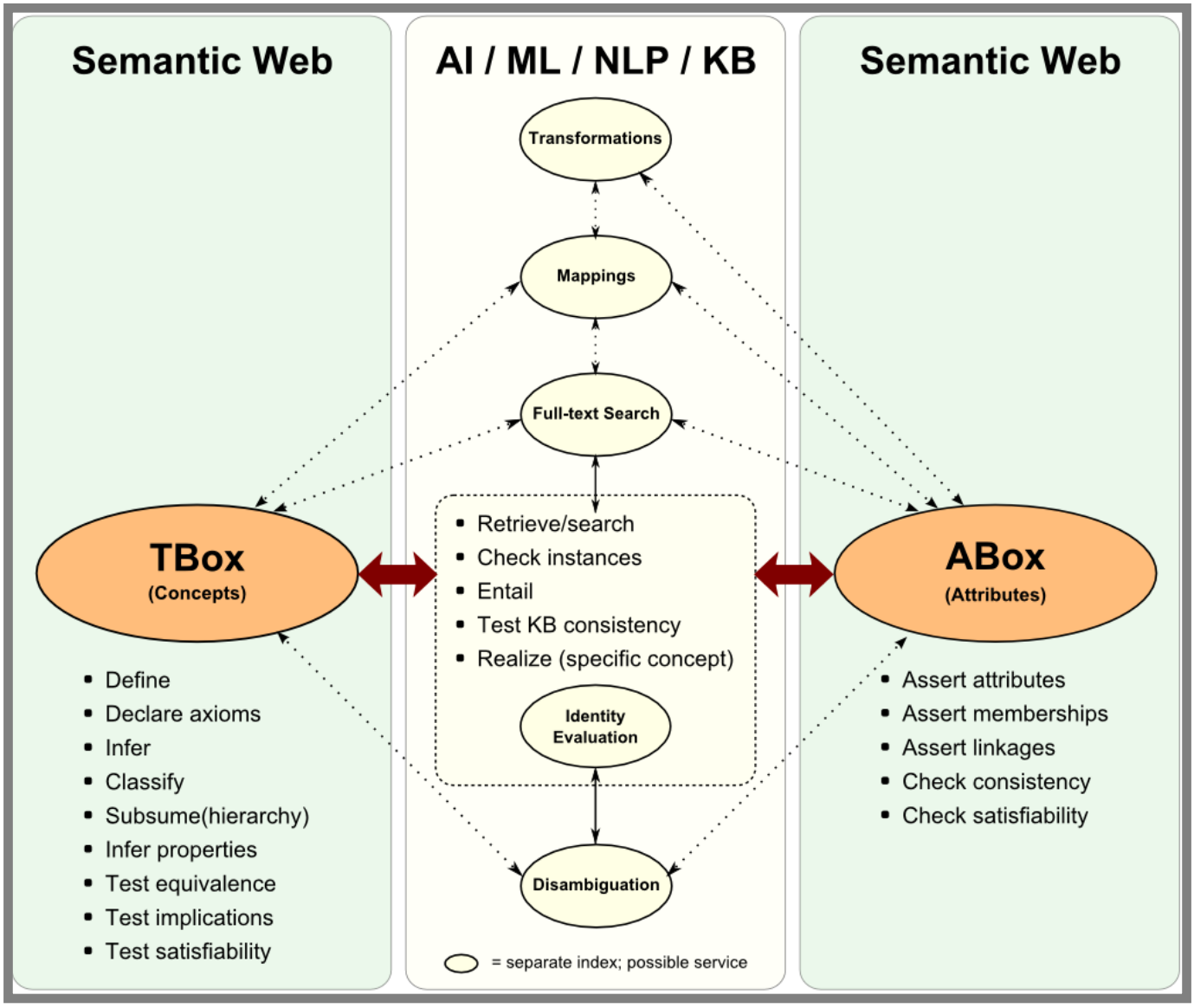
(individuals), and are pretty good at being able to model relationships (predicates) between concepts and individuals. We also are able to ground concepts and their relationships through a number of reference concept ontologies [9]. But our understanding of attributes (the descriptive properties of instances) remains poor and ungrounded. Best practices -- let alone general practices -- still remain to be discovered.

TBox (concepts)	Specialty Work Tasks	ABox (data)
• *Definitions* of the *concepts* and *properties* (relationships) of the controlled vocabulary • *Declarations* of *concept axioms* or *roles* • *Inferencing* of relationships, be they transitive, symmetric, functional or inverse to another property • *Equivalence testing* as to whether two classes or properties are equivalent to one another • *Subsumption*, which is checking whether one concept is more general than another • *Satisfiability*, which is the problem of checking whether a concept has been defined (is not an empty concept) • *Classification*, which places a new concept in the proper place in a taxonomic hierarchy of concepts • *Logical implication*, which is whether a generic relationship is a logical consequence of the declarations in the TBox	• *Mappings* are the core of interoperability in that concepts and attributes get matched across schema and datasets • *Transformations* are the means to bring disparate data into common grounds, the second leg of interoperability • *Entailments*, which are whether other propositions are implied by the stated condition • *Instance checking*, which verifies whether a given individual is an instance of (belongs to) a specified concept • *Knowledge base consistency*, which is to verify whether all concepts admit at least one individual • *Realization*, which is to find the most specific concept for an individual object • *Retrieval*, which is to find the individuals that are instances of a given concept • *Identity relations*, which is to determine the equivalence or relatedness of instances	• *Membership assertions*, either as *concepts* or as *roles* • *Attributes assertions* • *Linkages assertions* that capture the above but also assert the external sources for these assignments • *Consistency checking* of instances • *Satisfiability checks*, which are that the conditions of instance membership are met

• *Infer property assertions* implicit through the transitive property	in different datasets] • *Disambiguation*, which is resolving references to the proper instance	

Work Tasks for a Data Interoperability Framework

Across the knowledge base (that is, the combination of the TBox and the ABox), the semantic Web has improved its search capabilities by formally integrating with conventional text search engines, such as Solr. Instance and consistency checking are pretty straightforward to do, but are often neglected steps in most non-commercial semantic installations. Critical areas such as mappings, transformations and identity evaluation remain weak work areas. This figure helps show these major areas and their work splits:



Work Splits Between the Semantic Web and AI

As we discussed earlier on the recent and rapid advances of artificial intelligence [10], the combination of knowledge bases and the semantic Web with AI machine learning (ML) and NLP techniques will show rapid improvements in data interoperability. The two stumbling blocks of not having a framework and architecture for interoperability, plus the lack of attributes groundings, have been controlling. Now that these factors are known and they are being purposefully addressed, we should see rapid improvements, similar to other areas in AI.

This re-embedding of the semantic Web in artificial intelligence, coupled with the conscious attention to provide reference groundings for data interoperability, should do much to address what are current, labor-intensive stumbling blocks in the knowledge management workflow.

Putting Some Grown-up Pants on the Semantic Web

The semantic Web clearly needs to play a central role in data integration and interoperability. Fortunately, like we have seen in other areas [11], semantic technologies lend themselves to generic functional software that can be designed for re-use in most any knowledge domain, chiefly by changing the data and ontologies guiding them. This means that reference libraries of groundings, mappings and transformations can be built over time and reused across enterprises and projects. Use of functional programming languages will also align well with the data and schema in knowledge management functions and ontologies and DSLs. These prospects parallel the emergence of knowledge-based AI (KBAI), which marries electronic Web knowledge bases with improvements in machine-learning algorithms.

The time for these initiatives is now. The complete lack of distributed data interoperability is no longer tolerable. High costs due to unacceptable manual efforts and too many failed projects plague the data interoperability efforts of the past. Data interoperability is no longer a luxury, but a necessity for enterprises needing to compete in a data-intensive environment. At scale, point-to-point integration efforts become ineffective; a form of reusable and transferable master data management (MDM) needs to emerge for the realiites of Big Data, and one that is based on the open and standard protocols of the Web.

Much tooling and better workflows and user interfaces will need to emerge. But the critical aspects are the ones we are addressing now: information and software architectures; reference groundings and attributes; and education about these very real prospects near at hand. The challenge of data interoperability in cooperation with its artificial intelligence cousin is where the semantic Web will finally put on its Big Boy pants.

---

[1] See Charnyote Pluempitiwiriyawej and Joachim Hammer, 2000. A Classification Scheme for Semantic and Schematic Heterogeneities in XML Data Sources, *Technical Report TR00-004*, University of Florida, Gainesville, FL, 36 pp., September 2000. See https://cise.ufl.edu/tr/DOC/REP-2000-396.pdf. I first cited this report and extended it to cover languages (see [3]) in M.K. Bergman 2006. Sources and Classification of Semantic Heterogeneities, *AI3:::Adaptive Information* blog, June 6, 2006. See http://www.mkbergman.com/232/sources-and-classification-of-semantic-heterogeneities/). This most recent version added the examples and expanding the listing a bit further, to where it is no longer faithful to the original 2000 paper.

[2] 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). This distinction is not use in standard TBox-ABox splits; its rationale will be described in a further article.

[3] See M.K. Bergman, 2006. Tutorial: Internet Languages, Character Sets and Encodings, *BrightPlanet Corporation Technical Documentation*, March 2006, 13 pp. See http://www.mkbergman.com/wp-content/themes/ai3v2/files/2006Posts/InternationalizationTutorial060323.pdf.

[4] See [7]. Also the TBox portion, or classes (concepts), is the basis of the ontologies. The ontologies establish the structure used for governing the conceptual relationships for that domain and in reference to external (Web) ontologies. The ABox portion, or instances (named entities), represents the specific, individual things that are the members of those classes. Named entities are the notable objects, persons, places, events, organizations and things of the world. Each named entity is related to one or more classes (concepts) to which it is a member. Named entities do not set the structure of the domain, but populate that structure. The ABox and TBox play different roles in the use and organization of the information and structure.

[5] M.K. Bergman 2009. When Linked Data Rules Fail, *AI3:::Adaptive Information* blog, November 16, 2009. See http://www.mkbergman.com/846/when-linked-data-rules-fail/.

[6] M.K. Bergman 2006. Climbing the Data Federation Pyramid, *AI3:::Adaptive Information* blog, May 25, 2006. See http://www.mkbergman.com/229/climbing-the-data-federation-pyramid/.

[7] M.K. Bergman 2008. Thinking 'Inside the Box' with Description Logics, *AI3:::Adaptive Information* blog, November 10, 2008. See http://www.mkbergman.com/466/thinking-inside-the-box-with-description-logics/.

[8] See the thread on the W3C semantic web mailing list beginning at http://lists.w3.org/Archives/Public/semantic-web/2014Jul/0129.html.

[9] Examples of upper-level ontologies include UMBEL, the Suggested Upper Merged Ontology (SUMO), the Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE), PROTON, Cyc and BFO (Basic Formal Ontology). Most of the content in their upper-levels is akin to broad, abstract relations or concepts (similar to the primary classes, for example, in a Roget's Thesaurus) than to "generic common knowledge." Most all of them have both a hierarchical and networked structure, though their actual subject structure relating to concrete things is generally pretty weak. See further the Wikipedia entry on upper ontologies.

[10] M.K. Bergman 2014. Spring Dawns on Artificial Intelligence, *AI3:::Adaptive Information* blog, June 2, 2014. See http://www.mkbergman.com/1731/spring-dawns-on-artificial-intelligence/.

[11] M.K. Bergman 2011. Ontology-driven Apps Using Generic Applications, *AI3:::Adaptive Information* blog, March 7, 2011. See http://www.mkbergman.com/948/ontology-driven-apps-using-generic-applications/.
