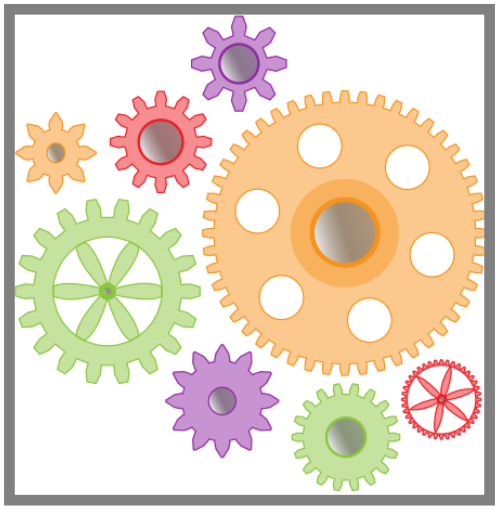
The Primacy of Search in the Semantic Enterprise

by Mike Bergman - Monday, February 11, 2013

http://www.mkbergman.com/1618/the-primacy-of-search-in-the-semantic-enterprise/



Part 5 in the Enterprise-scale Semantic Systems Series

We become such captives to our language and what we think we are saying. Many basic words or concepts, such as "search," seem to fit into that mould. A weird thing with "search" is that twenty years ago the term and its prominence were quite different. Today, "search" is ubiquitous and for many its embodiment is Google such that "to google" is often the shorthand for "to search".

When we actually do "search", we submit a query. The same extension that has inveigled its tendrils into search has also caused the idea of a query to become synonymous with standard text (Google) search.

But, there's more, much more, both within the meaning and the execution of "to search".

Enterprises, familiar with structured query language (SQL), have understood for quite some time that queries and search were more than text searches to search engines. Semantic technologies have their own structured query approach, SPARQL. Enterprises know the value of search from discovery to research and navigation. And, they also intuitively know that they waste much time and don't often get what they want from search. U.S. businesses alone could derive $33 billion in annual benefits from being better able to re-find previously discovered Internet information, an amount equal to $10 million per firm for the 1,000 largest businesses [1]. And those benefits, of course are only for Internet searches. There are much larger costs arising from unnecessary duplicated effort because of weaknesses in ***internal*** search [1].

The thing that's great about semantic search -- done right -- is it combines conventional text search with structured search, adds more goodies, and basically overcomes most current search limitations.

Many Kinds of Search

The Webster definition of "search" is to, *"*look into or over carefully or thoroughly in an effort to find or discover something*."*

There are two telling aspects to this definition. One, search may be either casual or careful, from "looking" into something to being "thorough". Second, search may have as its purpose finding or discovery. Finding, again, implies purpose or research. Discovery can range from serendipity to broadening one's understanding or horizons given a starting topic.

Prior to the relational systems, network databases represented the state-of-the-art. One of E.F. Codd's stated reasons in developing the relational approach and its accompanying SQL query language was to shift the orientation of databases from links and relationships (the network approach) to query and focused search [2]. By virtue of the technology design put forward, relational databases shifted the premise to structured information and direct search queries. Yet, as noted, this only represents the purposeful end of the search spectrum; navigation and discovery now becomes secondary.

Text search and (text) search engines then came to the fore, offering a still-different model of indexing and search. Each term became a basis for document retrieval, leading to term-based means of scoring (the famous Salton TF/IDF statistical model), but with actually no understanding of the semantic structure or meaning of the document. Other term-based retrieval bases, such as latent semantic indexing, were put forward, but these were based on the statistical relationships ***between*** terms in documents, and not the actual meaning of the text or natural language ***within*** the documents.

What we see in the early evolution of "search" is kind of a fragmented mess. Structured search swung from navigation to purposeful queries. Text search showed itself to be term-based and reliant on Boolean logic. Each approach and information store thus had its own way to represent or index the data and a different kind of search function to access it. Web search, with its renewal of links and relationships, further shifted the locus back to the network model.

State-of-the-art semantic search , as practiced by Structured Dynamics, has found a way to combine these various underlying retrieval engines with the descriptive power of the graph and semantic technologies to provide a universal search mechanism across all types of information stores. We describe this basis more fully below, but what is important to emphasize at the outset is that this approach fundamentally addresses all aspects of search within the enterprise. As a compelling rationale for trying and then adopting semantic technologies, semantic search is the primary first interest for most enterprises.

Unique Advantages to Semantic Search

The first advantage of semantic search is that all content within the organization can be combined and searched at once. Structured stuff . . . documents . . . image metadata . . . databases . . . can now all be characterized and put on an equivalent search footing. As we just discussed in text as a first class citizen, this power of indexing all content types is the real dynamo underneath semantic search.

The universality of search means that being able to search all available content is awesome. But, being able to add the dimensions of relationships between things means that the semantic graph takes information exploration to a totally new level.

The simplest way to understand semantic search is to de-construct the basic RDF triple down to its fundamentals. This first tells us that the RDF data model is able to represent any thing, that is, an object or idea. And, we can represent that object in virtually any way that any viewer would care to describe it, in any language. Do we want it to be big, small? blue, green? meaningful, silly? smart, stupid? The data model allows this and more. We can capture how diverse users describe the same thing in diverse ways.

But, now that I have my world populated with things and descriptions of them, how do they connect? What are the relationships between these things? It is the linkages -- the connections, the relationships -- between things that give us context, the basis for classifying, and as a result, the means to ascertain the similarity or adjacency of those things. These sorts of adjacencies then enable us to understand the "pattern" of the thing, which is ultimately the real basis for organizing our world.

The rich brew of things ('nouns") and the connections between them ("verbs") starts to give our computers a basis for describing the world more akin to our real language. It is not perfect, and even if it were, it would still suffer from the communication challenges that occur between all of us as humans. Language itself is another codified way of transmitting messages, which will always suffer some degree of loss [3]. But in this comment we can also glean a truth: humans interacting with their computer servants will be more effective the more "human" their interfaces are. And this truth can also give us some insight into what search must do.

First, we are interested in classifying and organizing things. The idea of "facets", the arrangement of search results into categories based on indexed terms, is not a new one in search. In conventional approaches, "facets" are approached as a kind of dimension, one that is purposefully organized, sometimes hierarchically. In Web interfaces, facets most often appear as a listing in a left-hand column from which one or more of these dimensions might be selected, sometimes with a count number of potential results after the facet or sometimes with a checkbox or such by which multiple of these facets might be combined. In essence, these facets act as structural or classificatory "filters" for the content at hand. This is made potentially more powerful when also combined with basic keyword search.

In semantic search, facets may be derived from not only what types of things exist in the search space, but also what kinds of attributes (or properties) connect them. And, this all comes for free. Unlike conventional faceting, no one needs to decide what are the important "dimensions" or any such. With semantic search, the very basis of describing the domain at hand creates an organization of all things in the space. As a result of semantic search, this combination of entities and properties leads to what could be called "global faceting". The structure of how the domain is described is the sole basis required to gain -- and make universal to the information space -- these facets.

Whoa! How did that happen? All we did is describe our information space, but now we have all of this rich structure. This is when the first important enterprise realization sets in: how we describe the information in our domain is the driving, critical factor. Semantic search is but easy pickings from this baseline. What is totally cool about the nature of semantic search is that slicing-and-dicing would put a sushi restaurant to shame. Every property represents a different pathway; and every entry (node) is an entry point.

Second, because we have based all of this on an underlying logic model in descriptive logics, we gain a huge Archimedes' lever about our information space. We do not need to state all of the relationships and

organizations in our information space. We can infer them from the assertions already made. Two parents have a child? That child has a sibling? Then, we can infer the second child also has the same parents. The "facts" that one might assume about a given domain can grow by 10x or more when inference is included.

Now we can begin to see where the benefits and return from semantic search becomes evident. Semantic search also enables a qualitatively different content enrichment: we can use these broad understandings of our content to do better targeting, tagging, highlighting or relating concepts to one another. The fact that semantic search is simply a foundation to semantic publishing is noteworthy. We will discuss this topic in a later part to this series.

SD's Approach: RDF Triple Store + Solr + OWLAPI

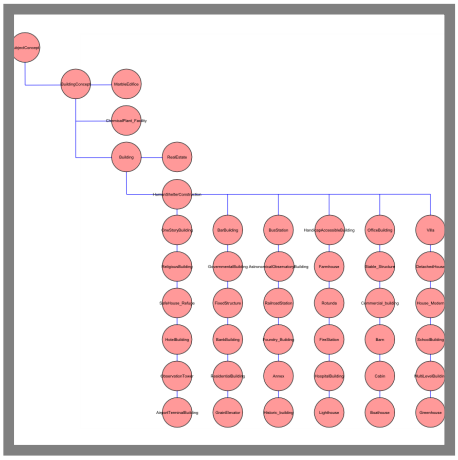
In recognition of the primacy of search, we at Structured Dynamics were one of the first in the semantic Web community to add Solr (based on Lucene) full-text indexing to the structured search of an RDF triple store [4]. We later added the OWL API to gain even more power in our structured queries [5]. These three components give us the best of unstructured and structured search, and enable us to handle all kinds of search with additional flexibility at scale. Since we historically combined RDF and Solr first, let's discuss it first.

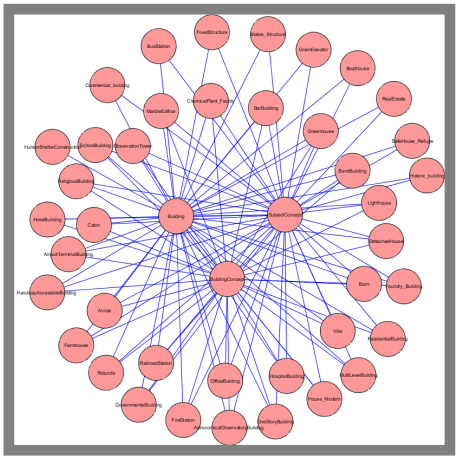
We first adopted Solr because traditional text search of RDF triple stores is not sufficiently performant and makes it difficult to retrieve logical (user) labels in place of the URIs used in semantic technologies. While RDF and its graph model provide manifest benefits (see below), text search is a relatively mature technology and Solr provided commercial-grade features and performance in an open source option.

In our design, the triple store is the data orchestrator. The RDF data model and its triple store are used to populate the Solr schema index. The structural specifications (schema) in the triple store guide the development of facets and dynamic fields within Solr. These fields and facets in Solr give us the ability to gain Solr advantages such as aggregates, autocompletion, spell checkers and the like. We also are able to capture the full text if the item is a document, enabling standard text search to be combined with the structural aspects orchestrated from the RDF. On the RDF side, we can leverage the schema of the underlying ontologies to also do inferencing (via forward chaining). This combination gives us an optimal search platform to do full-text search, aggregates and filtering.

Since our initial adoption of Solr, and Solr's own continued growth, we have been able to (more-or-less) seamlessly embrace geo-locational based search, time-based search, the use of multiple search profiles and ranking and scoring approaches (using Solr's powerful extended disMax *edismax* parser) and other advantages. We now have nearly five years of experience of the RDF + Solr combination. We continue to discover new functionality and power in this combination. We are extremely pleased with this choice.

On the structured data side, RDF and its graph model have many inherent advantages, as earlier described. One of those advantages is the graph structure itself:





A distinguishing characteristic of ontologies compared to conventional hierarchical structures is their degree of connectedness, their ability to model coherent, linked relationships

Another advantage over conventional structured search (SQL) with relational databases is performance. For example, as Rik Van Bruggen recently explained [6], RDBMs searches that need to obtain information from more than one table require a "join." The indexes in all applicable tables need to be scanned recursively to find all the data elements fitting the query criteria. Conversely, in a graph database, the index needs only be accessed once to find the starting point in the graph, after which the relationships in the graph are "walked" to traverse the graph to find the next applicable data elements. The need for complete scans is what makes “joins” expensive computationally. Graph queries are incredibly fast because index lookups are hugely reduced.

Queries that experienced DBAs with relational databases would never attempt because of the excessive need for joins are trivial in a graph search.

Various graph databases provide canned means for traversing or doing graph-based operations. And that brings us to the second addition we added to the RDF triple store: inclusion of the OWL API. While it is true that our standard triple store, Virtuoso, has support for simple inferencing and forward chaining, the fact that our semantic technologies are based on OWL 2 means that we can bring more power to bear with an ontology-specific API, including reasoners.The OWL API allows all or portions of the ontology specification to be manipulated separately, with a variety of serializations. Changes made to the ontology can also be tested for validity. Most leading reasoners can interact directly with the API. Protégé 4 also interacts directly with the API, as can various rules engines. Additionally, other existing APIs, notably the Alignment API with its own mapping tools and links to other tools such as S-Match can interact with the OWL API.

Thus, besides the advantages of RDF and graph-based search, we can now reason over and manipulate the ontologies themselves to bring even more search power to the system. Because of the existing integrations between the triple store and Solr, these same retrieval options can also be used to inform Solr query retrievals.

Shaking Hands with the Enterprise

On the face of it, a search infrastructure based on three components -- triple store + Solr + OWL API -- appears more complex than a single solution. But, enterprises already have search provided in many different guises involving text or SQL-based queries. Structured Dynamics now has nearly five years experience with this combined search configuration. Each deployment results in better installation and deployment procedures, including scripting and testing automation. The fact there are three components to the search stack is not really the challenge for enterprise adoption.

This combined approach to search really poses two classes of challenges to the enterprise. The first, and more fundamental one, is the new mindset that semantic search requires. Facets need to be understood and widely embraced; graphs and graph traversals are quite new concepts; full incorporation of tagging to make text a first-class citizen with structured search needs to be embraced; and, the pivotal role of ontologies in driving the whole structural understanding of the domain and all the various ways to describe it means a shift in thinking from dedicated applications for specific purposes to generic ontology-driven applications. These new mindsets require concerted knowledge transfer and training. Many of the new implementers are now the subject matter experts and content editors within the enterprise, rather than developers. Dedicated effort is also necessary -- and needs to be continually applied -- to enable ontologies to properly and adaptively capture the enterprise's understanding of its applicable domain.

These are people-oriented aspects that require documentation, training materials, tools and work processes. These topics, actually some of the most critical to our own services, are discussed in later parts to this ESSS series.

The second challenge is in the greater variability and diversity of the "dials and knobs" now available to the enterprise to govern how these search capabilities actually work. The ranking of search results can now embrace many fields and attributes; many different types of content; and potentially different contexts. Weights (or "boosts" in Solr terms) can be applied to every single field involved in a search. Fields may be included or excluded in searches, thereby acting as filters. Different processors or parsers may be applied to handle such things as text case (upper or lower), stemming for dealing with plurals and variants, spelling variants such as between British and American English, invoking or not synonyms, handling multiple languages, and the like.

This level of control means that purposeful means and frameworks must be put in place that enable responsible managers in the enterprise to decide such settings. Understanding of these "dials and knobs" must therefore also be transferred to the enterprise. Then, easily used interfaces for changing and selecting options and then comparing the results of those changes must be embedded in tools and transferred. (This latter area is quite exciting and one area of innovation SD will be reporting on in the near future.)

The Productivity Benefits

There are actually many public Web sites that are doing fantastic and admirable jobs of bringing broad, complicated, structured search to users, all without much if any semantic technologies in the back end. Some quick examples that come to mind are Trulia in real estate; Fidelity in financial products; Amazon in general retail, etc. One difficulty that semantic search has in comparison to the alternatives is that first-blush inspection of Web sites may not show many large differences.

The real advantages from semantic search comes in its productivity and flexibility. Semantic search frameworks are easier to construct, easier to extend, easier to modify and cheaper to build. Semantic search frameworks are inherently robust. Adding entirely new domains of scope -- say from moving from a department level to the entire enterprise or accommodating a new acquisition -- can be implemented in a fraction of the time without the need for rework.

It will be necessary to document the use case experience of early adopting enterprises to quantify these productivity and flexibility benefits. From Structured Dynamics' experience, however, these advantages are in the range of one to two orders of magnitude in reduced deployment and maintenance costs compared to RDBMs-based approaches.

The Tie-in with Semantic Publication

Another hot topic of late has been "semantic publishing" that is of keen interest to media and content-intensive sites on the Web. What is interesting about semantic pubishing, however, is that it is completely founded on semantic search. All of the presentation or publishing of content in the interface (or in an exported form) is the result of search. Remember, due to Structured Dynamics' semantic technology design with its structWSF interfaces, all interaction with the underlying engines and system occur via queries.

We will be talking much about semantic publishing toward the conclusion of this series. We will cover content enrichment, new kinds of products such as topic pages and semantic forms and widgets, and the fact that semantic publishing is available almost for "free" when your stack is based on semantic technologies with semantic search, SD-style.

NOTE: This is part of an ongoing series on enterprise-scale semantic systems (ESSS), which has its own category on this blog. Simply click on that category link to see other articles in this series.

---

[1] M.K. Bergman, 2004. “Untapped Assets: The $3 Trillion Value of U.S. Enterprise Documents,” *BrightPlanet Corporation White Paper*, December 2004, 41 pp. Published on this blog at http://www.mkbergman.com/82/untapped-assets-the-3-trillion-value-of-us-enterprise-documents/.

[2] See, for instance, the Wikipedia entry on the historical development of databases.

[3] M.K. Bergman, 2012. "What is Structure?," AI3:::Adaptive Information blog, May 28, 2012.

[4] F. Giasson, 2009. "RDF Aggregates and Full Text Search on Steroids with Solr," Fred Giasson's blog, April 9, 2009.

[5] M.K. Bergman, 2010. "A New Landscape in Ontology Development Tools,", AI3:::Adaptive Information blog, September 7, 2010.

[6] See, for example, Rik Van Bruggen, 2013. "Demining the 'Join Bomb' with Graph Queries," Neo4J blog, January 28, 2013.
