An Ontologies Architecture for Ontology-driven Apps

by Mike Bergman - Monday, December 05, 2011

http://www.mkbergman.com/989/an-ontologies-architecture-for-ontology-driven-apps/

Ontology Modularization and Roles within an OSF Instance

For some time now, Structured Dynamics (SD) has been touting the unique advantages of ODapps, or ontology-driven applications [1]. ODapps are modular, generic software applications designed to operate in accordance with the specifications contained in one or more ontologies. The relationships and structure of the information driving these applications are based on the standard functions and roles of ontologies (namely as domain ontologies), as supplemented by UI and instruction sets and validations and rules. When these supplements are added to standard ontology functions, we collectively term them adaptive ontologies [2].

To further the discussion around ODapps, today we are publishing two new documents, using the semantic technology foundation of the open semantic framework. OSF is a comprehensive, open source stack of SD and external tools that provides a turnkey environment for enterprises to adopt semantic technologies and approaches. OSF has been designed from the ground up to be an ontology-driven application framework.

The first new document, posted on Fred Giasson's blog, provides a detailed discussion of the dozen or so roles ontologies can play within an OSF installation. Fred's document is geared more to specific properties and configurations useful to deploy this framework; that is, the "drivers" in an ODapp setting. The second new document -- this one -- is more of a broad overview of the modularization and architecture of the constituent ontologies that make up an OSF installation. Both documents have also been posted to SD's open content TechWiki [3], which now has about 360 technical articles on understanding and implementing an OSF installation, importantly including its ontologies.

OSF Constituent Ontologies

As presently configured, an OSF installation may typically utilize most or all of the following internal ontologies:

- The SCO Ontology (Semantic Component Ontology)
- The WSF Ontology (Web Service Framework Ontology)
- The AGGR Ontology (Aggregation Ontology)
- The irON Ontology (Instance Record and Object Notation Ontology)
- One or more domain ontologies, to capture the concepts and relationships for the purposes of a

given OSF installation, and

- Possibly UMBEL (optional) or other upper-level concept ontologies, used for linkages to external systems.

(**Note:** the internal wiki links to each of these ontologies also provides links to the actual ontology specifications on Github.)

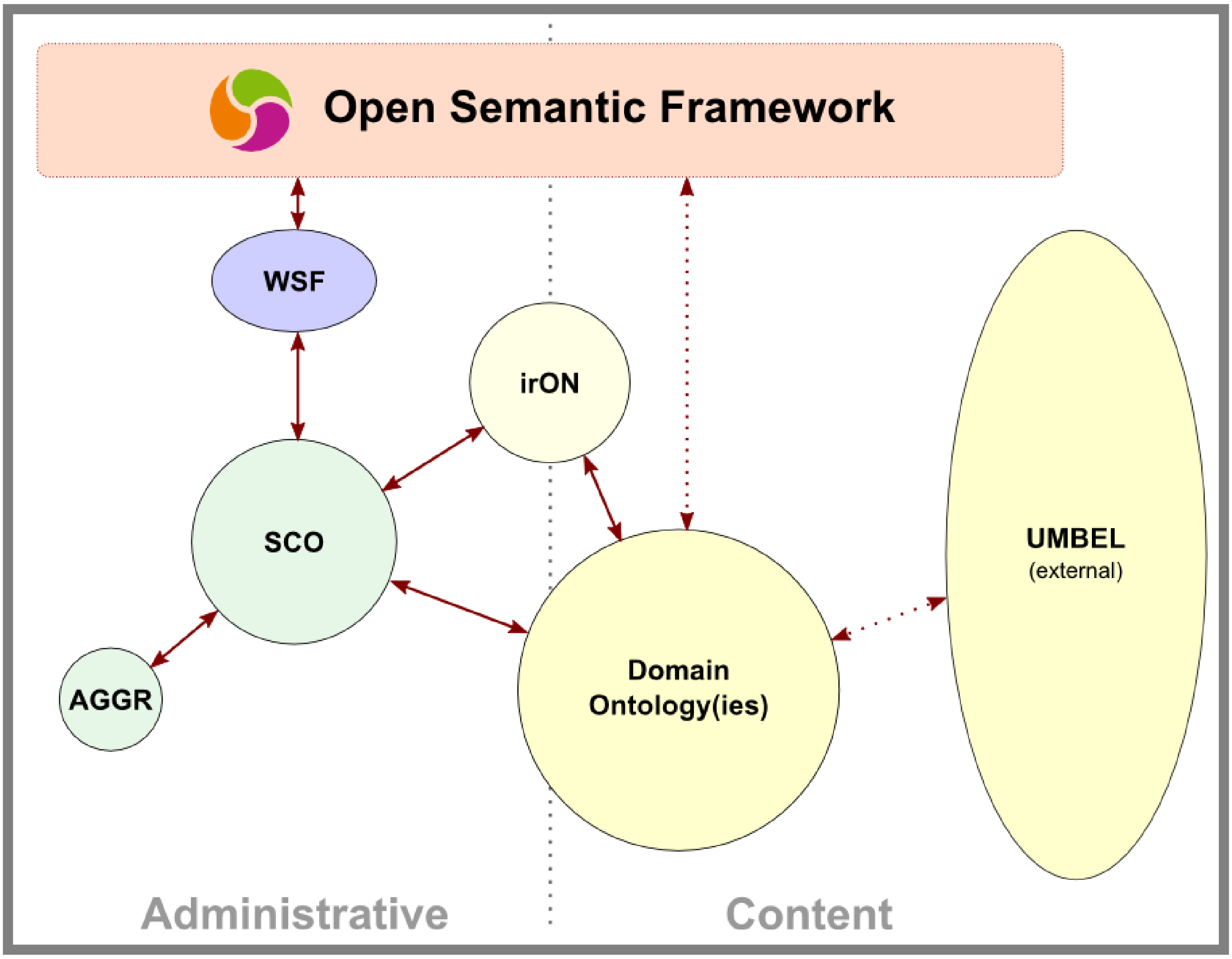
Depending on the specific OSF installation, of course, multiple external ontologies may also be employed. Some of the common external ones used in an OSF installation are described by the external ontologies document on the TechWiki. These external ontologies are important -- indeed essential in order to ensure linkage to the external world -- but have little to do with internal OSF control structures. That is why the rest of this discussion is focused on internal ontologies only.

The OSF Ontologies Architecture

The actual relationships between these ontologies are shown in the following diagram. Note that the ontologies tend to cluster into two main areas:

1. Content (or domain) ontologies, which tend to embody more of the traditional ontology functions such as information interoperability. inferencing, reasoning and conceptual and knowledge capture of the applicable domain; and
2. Administrative ontologies, which govern internal application use and user interface interactions.

This ontology architecture supports the broader open semantic framework:



(click for full size*)*

The WSF ontology plays a special role in that it sets the overall permission and access rights to the other components and ontologies. The UMBEL ontology (or other upper-level ontologies that might be chosen) is also optional. Such vocabularies are included when interoperability with external applications or knowledge bases is desired.

Summary of OSF Roles

We can further disaggregate these ontology splits with respect to the specific dozen or so ontology roles discussed in Fred's complementary piece on ontology roles in OSF. These dozen roles are shown by the rows with interactions marked for the various ontologies:

	S C O	A G G R	W S F	i r O N	D o m a i	U M B E L

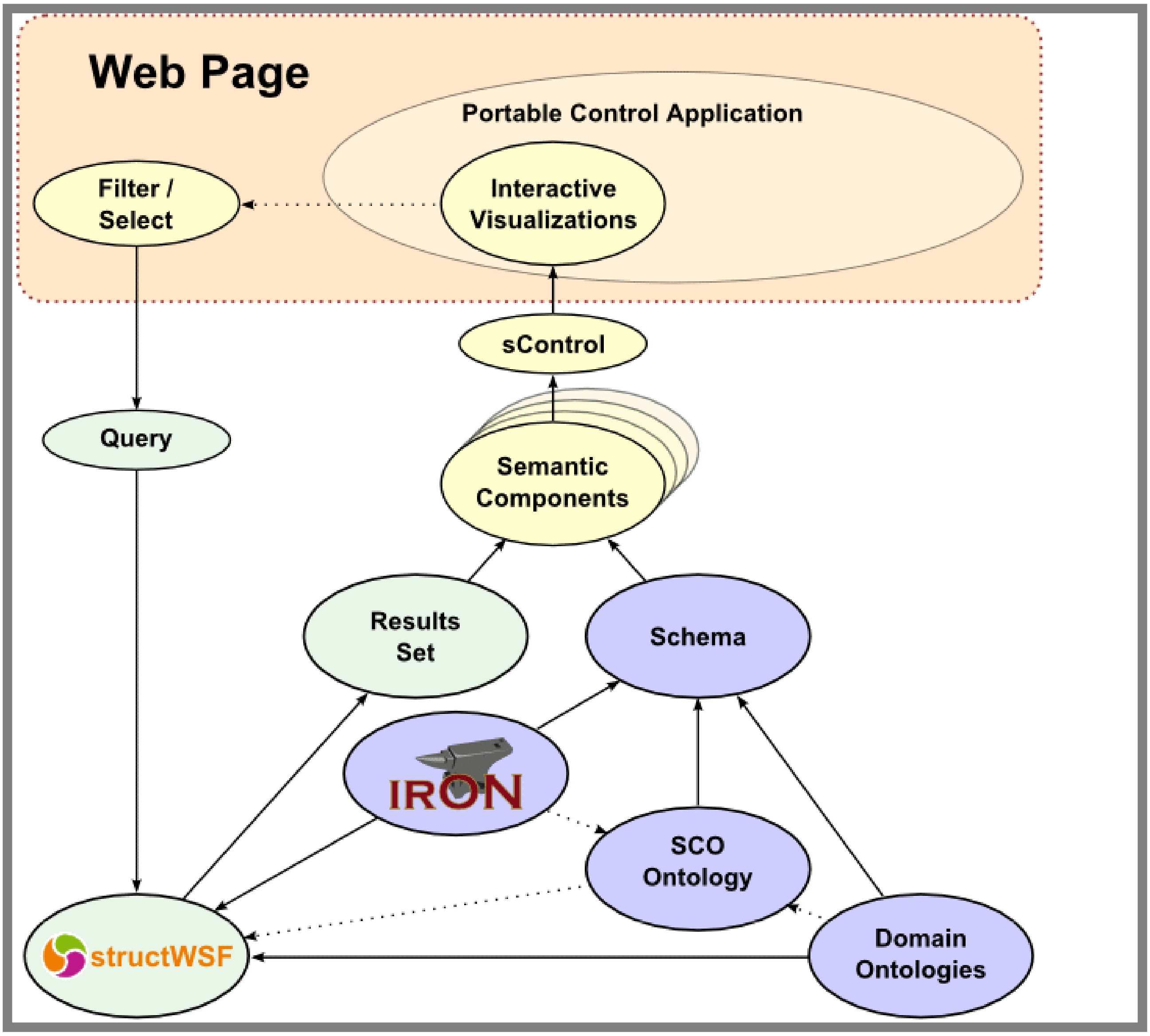
					n	
Define record descriptions				?		
Inform interface displays				?	?	?
Integrate different data sources				?	?	?
Define component selections	?			?	?	?
Define component behaviors	?	?				
Guide template selection				?	?	?
Provide reasoning and inference				?	?	?
Guide content filtering (with and without inference)		?	?			
Tag concepts in text documents				?	?	?
Help organize and navigate Web portals					?	?
Manage datasets and ontologies			?			
Set access permissions and registrations			?			

One of the unique aspects of adaptive ontologies is their added role in informing user interfaces and supporting specific semantic tools. Note, for example, the role of the content ontologies in informing interface displays, as well as their use in tagging concepts (via information extraction). These additional roles are the reason that these ontologies are shown as straddling both content and administrative functions in the first figure.

See Fred's piece to learn more about these dozen roles.

Interactions Are More Complex than Arrows

Naturally, a simple drawn arrow between ontologies (first figure) or a checkmark on a matrix (table above) can hide important details of how these interactions between ontologies and components actually work. In an earlier article, we discussed how the whole workflow takes place between users and user interface selections affecting the types of data returned by those selections, and then the semantic components (widgets) used to display them. This example interaction is shown by the following animation:



(click for [full size](#))

The blue nodes show the ontology interactions. These, in turn, instruct how the various components (yellow) and code (green) need to operate. These interactions are the essence of an ontology-driven app. The software is expressively designed to respond to specifications in the ontology(ies) used, and the ontologies themselves embrace some additional properties specific to driving those apps.

Possible Future Directions

ODapps are a relatively new paradigm, from which we continue to learn more about uses and potentials. We have wanted to write the first versions of these two new documents for some time, but have held off as we learned and exploited further the latent potentials in this design. As it stands, we see further potentials in this approach, and will therefore be likely adding new ontologies and capabilities to the general system for some time.

Some of the areas that look promising to us include:

- A generalized statistical ontology, especially as it can inform data displays in the semantic components
- Even more capable widgets in business intelligence (BI) uses, with a concomitant expansion of the vocabulary (predicates and classes) in some of the underlying ontologies
- More aggregation and summation functions supported by the AGGR ontology, and
- Still further improved permissions and access layers in the WSF ontology.

These potentials arise from the native power of the design basis for ontology-driven apps. Conceptually, the design is simplicity itself. Operationally, the system is extremely flexibile and robust. Strategically, it means that development and specification efforts can now move from coding and programmers to ontologies and the subject matter users who define and depend on them. With these advantages, who can argue with that?

[1] For the most comprehensive discussion of ODapps, see M. K. Bergman, 2011. " Ontology-Driven Apps Using Generic Applications," posted on the AI3:::Adaptive Information blog, March 7, 2011. You may also search on that blog for 'ODapps' to see related content.

[2] See M.K. Bergman, 2009. "Ontologies as the 'Engine' for Data-Driven Applications", AI3:::Adaptive Information blog, June 10, 2009, for the first presentation of these topics, but the specific term adaptive ontology was not yet used. That term was first introduced in "Confronting Misconceptions with Adaptive Ontologies" (August 17, 2009). The dedicated treatment of these topics and their interplay was provided in M.K. Bergman, 2009. "Ontology-driven Applications Using Adaptive Ontologies", AI3:::Adaptive Information blog, November 23, 2009. The relation of these topics to enterprise software was first presented in M.K. Bergman, 2009. "Fresh Perspectives on the Semantic Enterprise", AI3:::Adaptive Information blog, September 28, 2009.

[3] Slight revisions of these documents have been posted to the TechWiki as Role and Use of Ontologies in OSF and OSF Ontologies Modularization and Architecture, respectively.