ICD Wiki – Framework for Enabling Semantic Web Service Definition and Orchestration

Dean Brown, Dominick Profico
Lockheed Martin, IS&GS, Valley Forge, PA

Abstract – As Net-Centric enterprises grow, the desire to rapidly define and build reusable services and create new business processes through the combination of services and workflow will grow. Semantic Web Services is one approach to facilitate automated mediation between services based on semantic understanding of the services. Lockheed Martin is investigating the use of a wiki with an underlying RDF data model to provide a collaborative framework to define services, document services, manage ontology models, and quickly build composite services.

I. Net-Centricity, SOA, and Web Services

> ***Net-Centric:*** *Participating as a part of a continuously-evolving, complex community of people, devices, information and services interconnected by a communications network to achieve optimal benefit of resources and better synchronization of events and their consequences.*[i]

A. Net-Centricity

The following two principles are what makes Net-Centric different from how we usually build systems:

- Openness. Information systems can communicate across traditional system and enterprise boundaries in an open-ended ways.
- Dynamic Interaction. The capability to dynamically change the interaction and organizational scope at run-time versus at system development time.[ii]

Service-oriented architectures (SOAs) implemented with web services provides an open, standards-based approach to implementing capabilities that can be dynamically linked together to implement a business process. The movement towards SOA and web services allows service providers to provide high-value capabilities and services without necessarily knowing the service consumer. To optimize the value of these services, service providers need to design and build services that are reusable (as agnostic as possible to a specific implementation) and as stateless as possible (scalable and more independent).

B. Composite Web Services

As the enterprise inventory of reusable web services grows, the desire to build Composite Web Services that leverage these services to quickly support new business processes and user-desired functionality grows. Composite Services logically chain multiple web services together, ideally using an execution language like WS-BPEL or Google Mashup that can execute the service without requiring software compilation by software developers.

However, based on the heterogeneous nature of web services, linking web services together where data formats, names, units, and message formats are different requires an integrator knowledgeable about the specifics of each service; which is usually a software developer.

II. Semantic Web Services

> *The vision of Semantic Web services is to describe and annotate the various aspects of a Web service using explicit, machine-understandable semantics, enabling the automatic location, combination, and use of Web services.*[iii]

A. Semantic Web Service Overview

The goal of our research on the IntegrationWare IRAD project is to make service orchestration more in the spirit of the net-centric and Web 2.0 paradigm by allowing Composite Services to be built quickly and easily by end-users in a familiar environment in an intuitive, drag-and-drop user interface. Semantic Web Services provide the foundation to performing automated data-level mediation (matching dissimilar data names, formats, units) between services.

This is done by requiring the web service providers to perform a one-time mapping of their web service to a set of ontology models, as well as documenting additional information regarding the functionality of their services. The ontology models either pre-exist (developed specifically for a particular domain), or are modified as needed to support the web services. Once the web service-to-ontology model mappings are complete, the mapping is converted into a machine-readable format that will be used to facilitate the discovery of services and automated data mediation between the services.

B. Ontology Models

In order to create semantic web services, several different ontology models need to be created: at least one domain model, a units model, a transformation model, and a schema model.

The domain model(s) documents the entities, their relationships, and their properties that are relevant to a particular domain. For example, for the intelligence community domain model, some entities could include ISR assets, sensors, products, reports, tasking, geospatial locations, collection plans, observations,… Ideally, the domain model is built prior to performing the web service mapping to help identify the desired set of web services to be built or obtained. However, as new web services are used that don't map to existing entities and properties, then the domain ontology model will grow.

The units model documents units for values such as distance, area, volume, mass, temperature,… and how to transform between them.

The transformation model documents known "generic" transformations between different data types that aren't units of measurement and aren't associated with a particular entity. For example, a transformation service that can transform from lat/long coordinates to MGRS coordinates would show a relationship between lat/long and MGRS.

The schema model documents the message syntaxes to support message transformations between services.

C. Web Service Mapping to Ontology Models

Our primary short-term goal in developing Semantic Web Services is to support data-level mediation between web services. Because web services can be developed by a wide range of producers that don't build their services with a common data interface model in mind, many services that reference the same data can have different data formats (strings, ints, doubles,…), different data names (asset_id versus AssetId), different data units (meters versus feet, MGRS versus lat/long), and different data structures or groupings of data.

The mapping of web service interface elements (data inputs and outputs) to ontology model object properties unambiguously "defines" that data element in terms of "what" it is (domain model), the data format (schema model), and data unit (unit model) in a machine readable format (see Figure 1). This facilitates automated data transformations to address all these data matching issues.

Once the mapping has been completed, the mapped relationships are converted into a machine-readable format.

D. Design-Time Service Composition

When multiple web service data interface elements are mapped to the same ontology model object property, they are declared to be semantically "equivalent"; meaning a mapped web service output element can be mapped to an equivalent web service input element regardless of data type, name, unit, or data structure if the appropriate

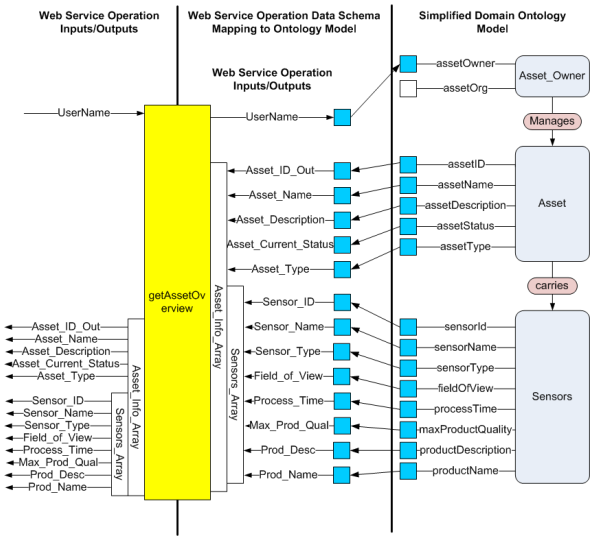

Figure 1. Web Service to Domain Ontology Mapping Example

mapping services are supported (ex: we know how to transform meters to feet).

For example, suppose a user wanted to create a composite service that computes how far an ISR asset is off plan from it's current position. This might require the composition of two specific web services like GetAssetInfo (to get position of identified asset in lat/long/elev) and GetISR_AssetPositionOffset (to take the position of the identified asset (in MGRS coordinates and elev) and a plan ID) to compute the offset (see Figure 2).

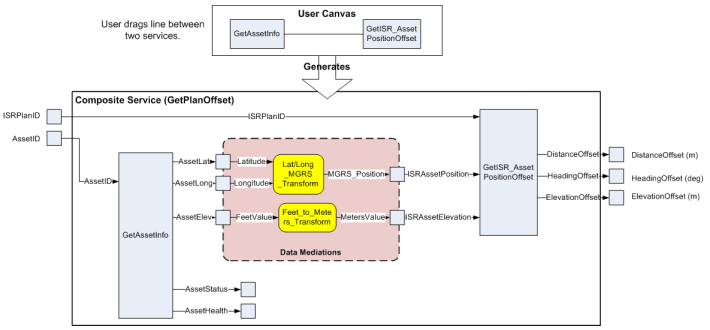

Figure 2. Example Composite Service Generation

If both services are mapped to a domain model (which identifies AssetElev and ISRAssetElev as equivalent), a units model (which knows how to convert feet to meters), and a transformation model (which knows how to convert lat/long to MGRS), then the user can simply drag each service to the canvas and connect them together. The underlying system will use the ontology model mappings to determine what outputs from the first service map to

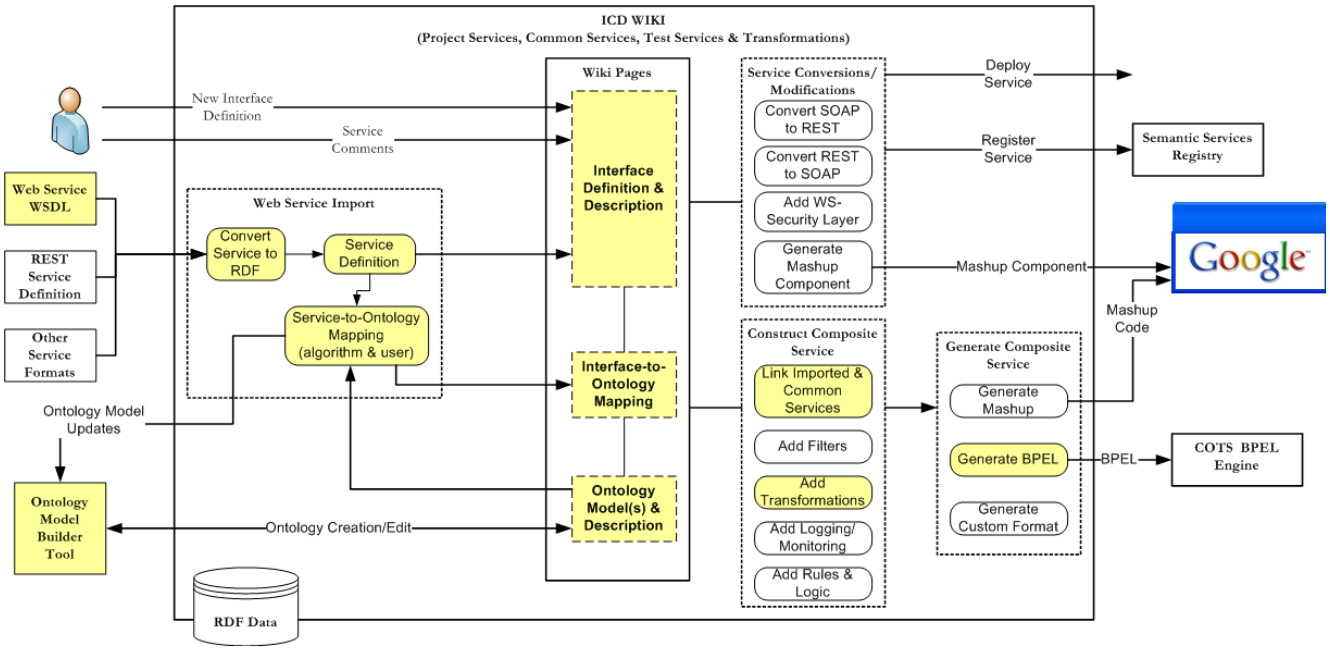

Figure 3. ICD Wiki Functional Diagram

inputs to second service (equivalance). If data transformations are required, then the appropriate transformation services are automatically identified and inserted between the user linked services.

III. ICD Wiki

A. Service Design, Discovery & Composition Framework

We provide a framework to build semantic web services that is intuitive for users by using the Wiki paradigm. The ICD wiki framework allows users to import web services, perform the service-to-ontology model mappings, and generate/convert/modify services for export. It also allows users to "define" desired web services and to provide feedback regarding the usability of existing web services.

See Figure 3 for a functional diagram of the ICD wiki framework vision. The yellow shaded boxes show where we have done development so far. The following sections provide more detail regarding the implemention of the ICD Wiki.

B. Service Import Process

Importing existing service definitions is a key component of increasing the usefulness and adoption of the ICD Wiki system. The wiki provides a user interface designed to compliment existing common user interfaces on the Web. Through this interface, a user is able to select a Web Service Description Language (WSDL) file to be imported into the ICD Wiki Semantic Store. This WSDL file can be either located locally on the user's workstation or any network reachable URL.

The import process places copies of all WSDL and related schema files onto a "Resource Bus", making all files available via a standard URL reference. Co-locating each of the required files simplifies the task of inspecting the interface files and validating references.

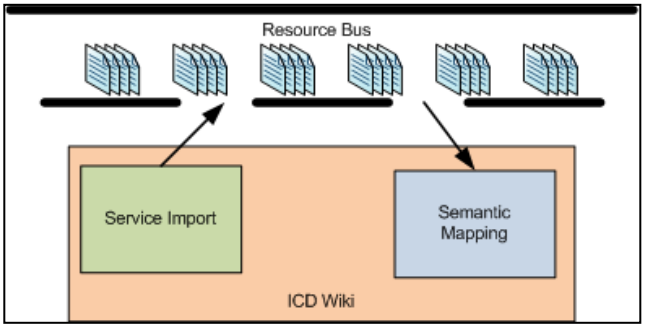

Figure 4. ICD Wiki / Resource Bus Interaction

After replicating the necessary files to the Resource Bus, the primary file is inspected to determine the format and version. WSDL v1.1 files are converted to WSDL v2.0 in order to facilitate the transformation and mapping stages. Conversion of the WSDL takes place via an Extensible Stylesheet Language Transformation (XSLT). The XSL file utilized to perform this transformation was acquired from the W3Civ. WSDL v2.0 specifications require no additional conversion before being transformed into semantic representations.

C. Semantic Service Transformations

There are multiple aspects to converting a service specification into a semantic representation suitable for storage in the ICD Wiki Semantic Store. The OWL-S definition allows for a service specification to be represented semantically, but it does include the ability to represent the underlying syntactic structure of the messages passed into and out of the service's respective operations. Semantic representation of the syntactic structure is required in order to accurately determine how service interfaces can be mapped to one-another. In order to support this level of detail we developed a small model to represent XML Schemas.

This model represents each of the most commonly used XSD elements as semantic entities. We are then able to create individuals of those class elements which represent the imported schema. Once the full schema for the service has been re-represented semantically, that semantic data is inserted into our semantic data store. In addition to automatically transforming the syntactic schema into a semantic one, the system, at this stage, provides the user with the ability to assign "units" to entities in the semantic representation of the service schema. Early in our design process, it was determined that support for unit conversion among service operations would be an integral part of enabling service composition. A simple model was built to represent the abstract concept of units and unit transformations, both simple and complex. If a user chooses to include unit designations for various elements in the schema representation, those units will be taken into consideration during any future composition sessions and used to facilitate additional transformations where appropriate.

At this stage, additional user input is required in order to fully understand the relationship between the imported schema and the known domain-specific models in the ICD Wiki thus far. Automated determination of this relationship is not available at this time in the prototype, so we present the user with an interface to allow point-and-click mapping of the imported elements to one or more domain models. A single entity can be mapped to multiple entities across multiple domain models, further enhancing the intrinsic knowledge within the semantic data store.

Mapping from complex objects in the syntactic schema to entities and entity –types with in the domain models facilitates the systems ability at later stages to determine "assignability" between two service interfaces.

"Assignability" is a determination made by applying semantic rules to the ICD Wiki service domain model and other domain models to generate an entailment. This entailment is used to ensure that an output message for a service's operation is "assignable" to the input message of another service, during service composition. Entities are assignable in many ways, and can be chained together to support the integration of services without those services supporting the exact same mapping.

D. Service Composition

Composition of services into larger, more robust services is one of the primary drivers of the ICD Wiki concept. Utilizing an off-the-shelf product called mxGraphv, we have a built a canvas-style, drag and drop interface for service composition. The available set of services are retrieved from the semantic store for inclusion in the new composite service. The composition tool makes use of common Web 2.0 tenets to allow a user to drag a service from the available service pool and place it on the canvas. Once on the canvas, a user can draw linkages between service inputs and outputs. During this process, the underlying architecture will continually check for assignability between the services linked.

Assignability is the determination "IF" two services can be linked together. During composition, this is enough information to allow the user to connect two services without being burdened with type and meaning mapping. The necessary transformations and conversions are added during the composed service generation phase.

After a user has completed laying out the composed service as desired, the canvas will be examined and the generation of the necessary work flow will be begin. The current system supports work flow generation as Business Process Execution Language files. Built in to the generated workflow code is all of the necessary unit and type transformations to combine the services. The data is not transformed into a semantic format during execution of the workflow, but rather the semantic data is used to determine what values can be assigned to what parameters, so a direct assignment is done between parameters inside the workflow. Multiple assignments and transformations may be necessary to progress from one service parameter to another, but these complexities are completely hidden from the user. These composed services are made available within the ICD Wiki for further composition and integration as needed by additional users.

E. Wiki Page Generation

Generation of pages inside the ICD Wiki takes place during the service and model import capabilities. During an import of either a model definition file or a service definition file, the necessary wiki pages are automatically created to support the human-readable aspect of the ICD Wiki concept.

Every wiki page in the ICD Wiki is capable of displaying a side-bar style component which shows all of the known semantic relationships between the entity represented on the current page and other entities in the semantic store. Using the sidebar, users are able to

explorer related entities in a traditional point-and-click, web format.

For imported services, a page is created to represent the service document as a whole, as well as pages for each

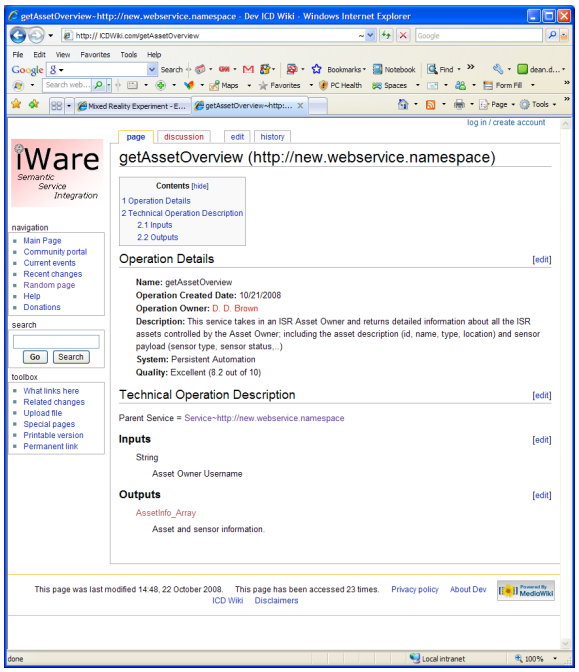

Figure 5. Operation Wiki Page Example

operation and input/output for those operations. The created pages contain little textual content, but instead there are custom wiki tags injected into the page text in order to support dynamic generation of various page sections, including the cross-linking of operations and types belonging to the service.

In addition to the semantic side bar outlined above, every imported model has a page from which a user can start exploring an imported model. This page provides access to the Domain Model Explorer.

F. Domain Model Explorer

The Domain Model Explorer is a tool built directly into the ICD Wiki system which supports a user in their exploration of the available domain models (see Figure 6). On each domain model's starting page, a "web" of semantic entities is displayed, allowing the user to find relationships amongst the various entities in the model. Further, each entity is accessible as a drill down point in order to find further relationships with in the model.

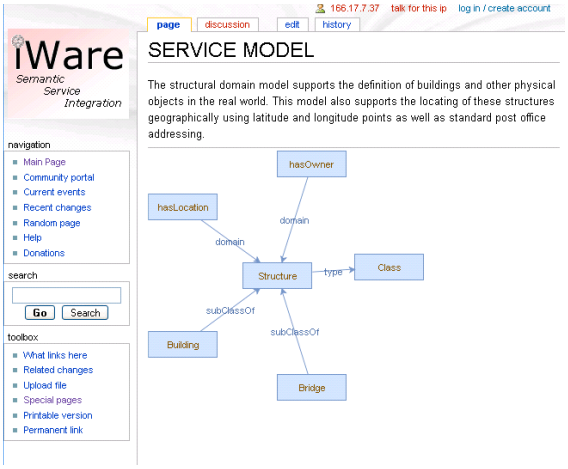

Figure 6. Domain Model Start Page

IV. Conclusion

We believe that the ICD wiki framework and semantic web services will allow all users to better leverage the growing list of available web services, intuitively define the services they want built, and provide feedback on the usability of all services..

V. References

[i] 'Net-Centric', *Wikipedia, The Free Encyclopedia*, http://en.wikipedia.org/wiki/Net-centric, (accessed 10/2/08)

[ii] The Essence of Net-Centricity, Hans Polzer

[iii] Dieter Fensel, Holger Lausen, Axel Polleres, Jos de Bruijn, Michael Stollberg, Dumitru Roman, John Domingue. *Enabling Semantic Web Services*. Berlin Heidelberg: Springer-Verlag, 2007

[iv] http://dev.w3.org/2006/wsdlConverter/wsdl11to20.xsl

[v] http://www.mxgraph.com/