Designing a Semantic Repository

Integrating architectures for reuse and integration

Cory Casanave

Cory-c (at) modeldriven.org

ModelDriven.org

May 2007

Overview

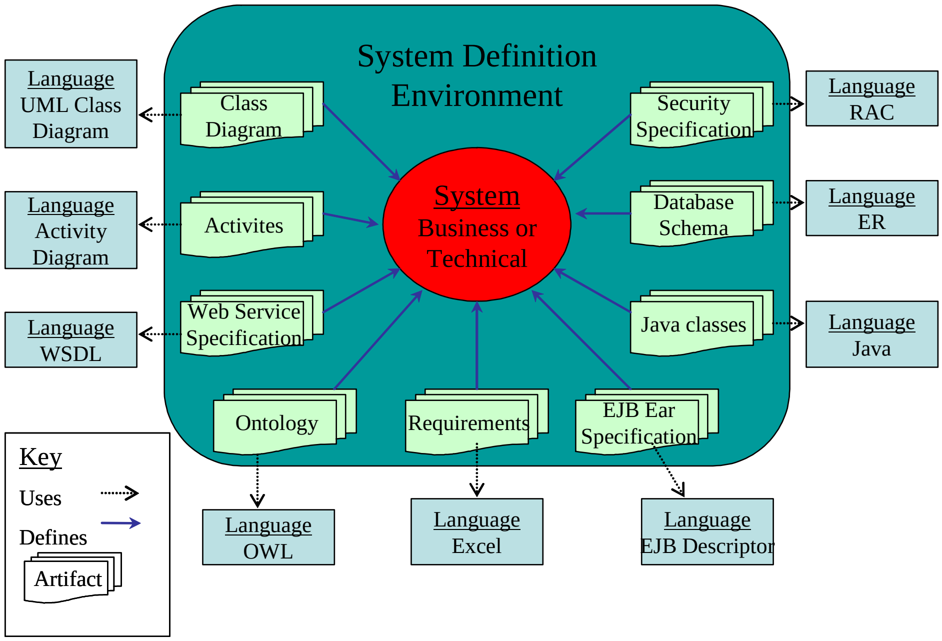
The Semantic Metadata infrastructure will provide a "smart" repository for architectures at multiple levels of abstraction, from multiple sources and with multiple views. This infrastructure will integrate the OMG-Meta Object Facility (MOF) and Resource Description Framework (RDF) and RDF-Schema as defined by W3C as part of the "Semantic Web" initiative and will integrate the specification and provisioning concepts of the OMG Model Driven Architecture (MDA). The semantic repository is an open source project at modeldriven.org.

Problem Statement

Information about government processes, information and I.T. systems currently exists in a diverse set of forms, formats, repositories and documents with little to no management and coordination. The result is that there is rampant redundancy in the developing and re-developing the same information, different models, architectures and studies about the same things. Understanding and integrating this wide Varity of information is almost impossible and thus generally such understanding and integration is not achieved.

Not only is this information expensive and time-consuming to develop, re-analyzing the same area results in inconsistent designs, lack of interoperability, redundant systems and missed opportunities for improvement.

A Semantic Repository will replace this fragmentation and loss of information with an architected approach to developing, integrating and managing information across the government.



A Semantic Repository will provide this architected approach by bringing together a suite of open source technologies to provide for architecture creation, integration and management. This suite will utilize Semantic Web and Model Driven Architecture (MDA) technologies to derive value out of architectures through the automated production of specification, implementation and testing artifacts to automate the enterprise based on business requirements.

Central to A Semantic Repository are the concepts of a system and an architecture. A system is any complex set of resources and behaviors integrated for a purpose. An organization is a kind of system, a business process is a system, as is a computer application. Architecture is a specification for how a system may or will fulfill its purpose. Architectures exist at many levels, from the architecture for a new venture to the architecture of a physical part in a machine. Understanding the connections between various systems and architectures is the means for more effectively realizing the goals. Architectures are expressed in models and ontologies (the term “model” and “ontology” are used interchangeable as expressions of architectures but ontologies are also used for other purposes thus we generally use “model” to include specification of an architecture).

Systems defined by multiple languages

Modern enterprises, governments and computer systems are complex; they are complex in terms of what they do, how they do these things and on how they use various technologies. Due to this complexity systems are defined using a set of languages, each language is used to prepare various models that, together, define the system.

The following picture illustrates how a set of specifications in various languages may serve to define a system.

Our challenge, as system developers, is that these languages overlap and the specifications are **always** inconsistent. The same element in the system may be represented in multiple specifications in multiple languages. Understanding the whole system becomes difficult and error prone. The connection between these languages is often lost as the system definition evolves over time. In fact, it is rare that the entire system is understood by anybody or documented in one place.

With each language being independently conceived and designed from its own viewpoint, there is no way to understand or refer to the touch points between the languages – whether they are saying the same thing or are stating unique facts about the same thing. This lack of integration is a source of complexity and error in our systems and the reason behind many system design failures.

Compound the above with multiple systems from multiple vendors and contractors with multiple architectural approaches. *The fragmentation of information is profound.*

Languages and sources of information

A Semantic Repository will be used to manage, integrate and create architectures from multiple sources and using multiple tools based on different languages. There is no assumption that the Semantic Repository model or infrastructure will be required or suitable for all purposes. To this end, Semantic Repository must be able to deal with architectures expressed using differently languages and idioms. The languages of most concern for architectures such as The Federal Enterprise Architecture, UML, OWL, DoDOF, BPMN, E-R and XML as well as various proprietary tool formats.

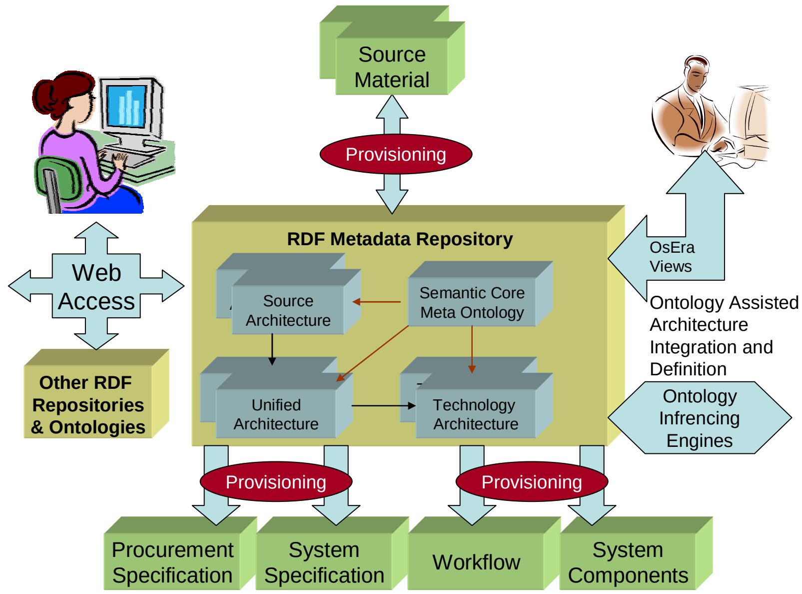
To this end A Semantic Repository will be required to understand the semantics embodied in architectures expressed using these formalisms and make sense of them *together*, as a consistent view of the enterprise.

Metadata Architecture

The A Semantic Repository provides for the definition, understanding, manipulation and management of architecture models. It fits into and supports the overall Semantic Repository Architecture and capabilities.

There is currently duplication of capability between the OMG-MOF infrastructure and the RDF infrastructure. Due to the requirement to interface with systems in both environments we intend on bridging these infrastructures such that our models may be used from either view. Ultimately one or the other will probably be selected as "core" and the other a mapping. But at this time they exist as peer infrastructures. As we refer to our metadata infrastructure we are including both the MOF and RDF interfaces to the models.

OsEra Metadata Infrastructure



The Metadata infrastructure is the hub into which the various capabilities a Semantic Repository integrates. This includes the 2-way synchronization with external sources of information - models, architectures, ontologies, requirements, processes, etc, - anything that helps define the enterprise and how it operates.

Architects use "Semantic View" tools to manage, design and integrate the information from the source material, creating a unified view of the enterprise from business requirements to system interfaces. Views may be provided by "internal" or external tools, integrating the specification and modeling environment. The architecture design and integration process is assisted with smart, ontology driven infrencing engines.

Architectures are not an end, but a means to an end – to improve the enterprise. Using MDA techniques, the architectures will help produce acquisition specifications, documentation, system specifications, DBMS systems, ontologies, component implementation and workflow systems.

Semantic Core

The infrastructure architecture, above discusses the technologies to manage architectures. The semantic core is for managing the semantics of these architectures. Semantic core is the reference ontology for architecture and also the "meta model" for the architectures that are represented in the Semantic Repository.

While a plethora of languages and notations have been used to specify architectures, there is a great degree of overlap in what those languages can say – their semantics. There is also inconsistency in how they are used and combined, essentially idioms of expression (For example, different architects use UML or OWL in very different ways).

The complicating factor is that different languages and idioms combine these concepts in very different ways and sometimes use very different ways to express similar concepts. In semantic core we have “sliced and diced” these concepts into smaller units we call “semantic components”. Each semantic component represents a specific fact in the architecture. Some semantic components are very small and atomic while others aggregate concepts into common packages and patterns.

The source languages have been analyzed to discover the underlying semantics and define these as semantic components, wherever possible having just one way to express a particular idea. The set of semantic components is the semantic core. No one graphical or textual language may have all the semantic components; languages tend to express a subset of these concepts for particular purposes. A particular syntax or file format is mapped to its semantic core representation when it is imported into or exported from the repository.

When an architecture model is referenced in the Semantic Repository it is mapped to a set of instances in the RDF store that are instances of the semantic core concepts.

Example;

For example, the computer languages UML, Java and WSDL all have a concept of an interface, but with some “extra sauce” attached to each. The semantic core provides a single concept of an interface, and separates the special sauce of each language.

Mapping architectures to the semantic core as a “hub” means that each language or idiom only needs to be mapped once to be integrated with all other semantic core enabled languages. It also means that architectures can be built that integrate the information found in multiple source models in multiple languages, something that could not be done with a point to point mapping between languages. The whole is more than the sum of its parts as patterns of information emerge.

As new languages and idioms are integrated into new semantic components are created to account for any new concepts introduced. In this way the Semantic Repository grows in its capability, it is not a fixed model that everything must adapt to – the Semantic Repository adapts to the architectures and languages it represents and integrates.

In the initial phases semantic core components will be represented with RDFS and a subset of OWL, as semantic core is more developed it will be defined in it’s self but still utilize RDF and OWL technologies.

This approach to metadata management has several advantages;

- Integration of information from diverse sources
- Translation between languages and views
- Semantic integration of architectures
- Support for OntoBots and provisioning from a single source or target
- Traceability between the source, derivation and use of information
- Independence from specific forms of diagrams, language syntaxes or technologies.