Response to the

OMG Ontology and Vocabulary Management Information RFI

Based on the

ModelDriven.org Enterprise Knowledge Base (EKB)

OMG Document: ontology/2008-08-01
August 25th, 2008

Table of Contents

Cover Letter & Summary

ModelDriven.org is pleased to submit this response to the OMG RFI for Ontology and Vocabulary management. As part of a project partially funded by the U.S. General Services administration's "OSERA" program (www.osera.gov) Model Driven Solutions has developed the "Enterprise Knowledge Base" (EKB) to the level of an operational prototype. The EKB directly addresses many of the needs outlined in the RFI, in particular:

- Management of models, Ontologies and artifacts across multiple versions
- Effectivity
- Provenance
- Asset management using standard tools (subversion)
- Integration with OMG based technologies, such as Eclipse
- Mapping between models and different standards

In addition, the EKB includes capabilities for the management of unstructured assets (arbitrary files) and tracking the versions and provenance of these assets.

More information on the EKB may be found on:
http://www.modeldriven.org/projects/EKB/

The source code for the EKB may be found on:
https://sourceforge.net/projects/enterpisekb/

However, the code is still being organized and is not ready for general use.

ModelDriven.org is very interested in industry feedback on the needs, approach and implementation choices of the EKB and in participating in related standards activities.

Answers to RFI Questions

Tooling

The ModelDriven.org EKB provides tooling and infrastructure for model, ontology and artifact management. Please see below on page: 9.

Tool Interoperability

There are multiple areas that would benefit from standardization – both at the level of Ontologies and supporting technologies.

Ontologies

As part of the EKB, we have developed a set of Ontologies for two purposes: To act as “hub” between different modeling languages, tools and methods, and, 2) to support the versioning, provenance and management of the information in the EKB.

The architecture ontology (AO), or ontology of architecture is a composite ontology for representing architectures. Architectures include models, specifications and designs of anything. The general subject area of these architectures includes logical domain models, business models and systems models and technology implementation models as practiced in business architecture, systems architecture and technology architecture. In that such models have been expressed in a variety of languages, standards, tools and idiom, the ontology of architecture provides away to integrate these diverse models into a common set of concepts. Languages and paradigms typically used to express architectures include: UML, EDOC, BPMN, OWL, XSD, SQL Schema, Organizational Structure, Goals and Requirements, the US-Federal Enterprise Architecture, etc. All of these are "input" to the ontology of architecture as sources of and requirements for architecture concepts. The ontology of architecture is part of a general theory of "shared concepts". Shared concepts identify "hub" identities for various expressions, models and opinions about the same thing. While well defined AO is not sufficiently formally defined and would benefit from grounding in upper Ontologies and more formal methods.

The concepts found in these "source" languages and paradigms frequently overlap and mix the same concepts in unique ways, sometimes adding a particular nuance. Part of the task of producing the ontology of architecture is to normalize these constructs into more independent and easily combinable concepts expressed in RDFS and OWL. These become the shared concepts of architecture.

The architecture ontology is a "meta ontology" in that it expresses concepts for expressing other kinds of concepts, concepts not found in RDFS or OWL - the language used for the architecture ontology. This may be confusing to some in that many of the concepts in the architecture ontology are also found in RDFS and OWL. For this reason there is overlap between RDFS and OWL and the architecture ontology. This overlap is intentional as RDFS and OWL are just one "input source" that needs to be normalized in the architecture ontology. Because RDFS and OWL are used to express the architecture ontology they also sometimes specialize RDFS and OWL resources. However, the role of AO as a meta-ontology should be clear in that architectures are an INSTANCE OF the architecture ontology and manipulated as instances. We can then reason about our architectures as first-class elements. That some of the same elements are also RDFS/OWL resources is a side-effect of expressing the ontology in this language.

The architecture ontology is not "one thing" or static. It is intended to be both modular and extensible. As new concepts are discovered they are integrated with the existing concepts and added to the ontology in one of the AO sub-ontologies. The ontology of architecture is then a controlled vocabulary of shared concepts for expressing architectural concepts using a formal language. Additional shared-concept ontologies (which may be more domain specific) are intended to build-on the architecture ontology.

Architecture Ontology sub-ontologies:

The ontology of architecture is made up of the following sub-ontologies, documented here: http://modeldriven.org/2008/ArchitectureOntology/doc/

Ontology	**Summary**
Architecture Ontology	The architecture ontology, or ontology of architecture is a composite ontology for representing architectures. Architectures include models, specifications and designs. The general subject area of these architectures includes logical domain models, business models and systems models and technology implementation models. ao.owl imports all of the other AO ontologies.
Metatype	The metatype ontology defines the concepts for defining other concepts. In particular the concepts for classes, properties, associations, behavior and constraints.
Base	The base ontology introduces the concepts of a statement and a concept. Everything in the knowledge base is a statement based on a concept defined as part of the base ontology.
Composition	Composition defines the whole/part relationship and the concepts of composites and parts.
Context	Context defines what statements in an arbitrary set of resources are applicable to a given concept or situation. Context defines the basic concepts of a context, something being in a context and contextual relationships. Context is used to produce the RDF graph applicable to a given situation, query, inference or view.
Interaction	The interaction ontology describes communications patterns between

	parties.
Set	The set ontology defines the basic concepts of sets, set membership, set relationships and operations. {Set is not complete}
Structure	The structure ontology defines hierarchical information views about things or types of things. Struct applies the whole/part paradigm to information in support of common structuring languages such as XML schema or SQL.
Shared Concepts	Shared concepts defines a shared concept as a "hub" identity for a concept that may be be mentioned in multiple models.
Composite Use	The composite use ontology implements the use of one composite pattern in another. This uses the role synthesis pattern used in collaboration modeling.
Relation	A relation is a statement with well defined semantics about a set of involved statements. Relation is the superclass of everything that connects statements, including properties, constraints, calculations and associations.
Proposition	The proposition ontology defines the types and roles of propositions - statements that can be true or false. This ontology uses concepts from the OMG SBVR standard. The proposition ontology also covers effectivity.
Happening	Base classes for things that take place over time.
Model	Models and model management
Actors	Actors, groups and organizations
Authority	Defines base for authorities and authorization (replaces Effectivity)
Map	Defined base classes for model mapping rules and relations
Map Hub	Defines specific shared concepts for UML, CX and BPMN into and out of the ontology of architecture without information loss. These extend the more general shard concepts to capture language-family-specific concepts.
Map Instance	Defines mapping rules for mapping into and out of the ontology of architecture.
Versioning	Defines artifact and model lifetimes, versions and how they have

	been changed over time by authorities.

Technologies

As part of the EKB we have developed a way to structure RDF information in more standard XML formats and both query and modify the knowledge base using this format. This is currently implemented in a "REST" interface to the knowledge base. As part of the development of the EKB we looked for XML standards for modifying and managing XML or RDF resources and found little. The only thing found was "query languages" such as SPARQL and xQuery which do not support managing the DBMS and have a format that is quite different from normal XML interactions (Neither SPARQL or xQuery are expressed in XML). Due to the need for an easy query interface as well as the ability to update and manage information through an XML centric interface, a new structure was defined that allows "actions" and "selectors" to be added to standard XML documents. These XML documents can then operate on the back-end RDF repository.

To support interactions using standard, structured, XML (in addition to RDF-XML) the EKB defined a "structure" ontology. The structure ontology creates one or more structures for RDF/OWL types such that they can more easily be used with standard XML tools. The structure ontology supports the XML centric interface.

ModelDriven.org would be interested in standards for repository information management and query similar to what has been built for the EKB.

Querying and Accessibility

The EKB support query using the RDF "SPARQL" language as well as the XML centric query and management interface referenced above.

Knowledge Management & Mapping

The "map" ontology, above, as well as the supporting infrastructure is in-development as a general way to map between models, similar in intent to the OMG "QVT" standard – but ontology oriented. Mapping is an area in need of additional research and standards. The lack of standard mapping capabilities is slowing down the adoption of MDA.

Of particular interest to this RFI is the following document: http://modeldriven.org/projects/EKB/Management%20of%20Change%20and%20Provenance%20in%20a%20RDF%20Repository.pdf

Standards of Practice

Standards of practice would seem to be an excellent area for white papers from the OMG.

Example Repositories

The EKB, as already mentioned, should be considered as a new player in the field.

About the EKB

The Enterprise Knowledge Base (EKB) is an open source project of ModelDriven.org and the OSERA program sponsored by the U.S. General Services administration (http:/www.osera.gov). It is currently an operational prototype and being structured for general open source availability.

Overview

Governance, planning, architecture and analytics at the enterprise and segment levels require a wide variety of information from multiple sources be brought together, managed, published and analyzed with accuracy and efficiency. In this white paper such information is collectively called Enterprise Information Assets (EIA).

The EKB combines knowledge management with dynamic configuration management. Dynamic Enterprise Architecture Change and Configuration Management means the ability to manage Enterprise Information Assets. Dynamic EA change and configuration management recognizes both internally and externally prompted change and provides for continuous capture of all information & artifacts , including governance information, and provides for informed decision making.

The enterprise Knowledge Management Challenges

The typical enterprise or government agency has acquired and accumulated a great deal of information, architectures and plans that are not well organized, integrated or maintained. Much information, some of it costly, is lost, forgotten or simply not applied at the right time. Plans and architectures are done and redone due to changes in contractors, tools, methodologies, technologies and management.

The information that is retained is frequently not “linked” or consistent. The key drivers in an IT plan are not the same key drivers in a human capital plan, funding is not consistently applied to enterprise needs and architectures don’t match reality. Decisions are made on inconclusive data after millions have been spent on analysis. *What information we have is not effectively transformed into knowledge.*

The standard practice in industry and government is to deliver information in static and unstructured *documents*. While documents are a crucial part of the knowledge management landscape this approach has severe limitations. Information in documents tends to be unstructured and stove piped. If, for example, an enterprise architecture is delivered as a document it is nearly impossible to reuse, reference or revise information in that document. Initiatives that should use, extend and revise this information don’t, because it is locked up in this document. Best practices in information and knowledge management are moving to various kinds of more structured and well defined information, such that it can be used, managed, referenced and extended for multiple purposes. Technologies such as XML, Semantic Web, Ontologies, Models, Metadata Repositories and DBMS systems store and manage information in a more flexible way. In bridging from the culture of documents to the culture of knowledge management we must recognize both and help them to work together more seamlessly while encouraging a transition to well defined and structured information.

While the technologies and methods for managing operational data in DBMS systems are well established, the same level of maturity has not emerged for management of governance and architectural information. This kind of information is less suited to the ridged structure of a DBMS, yet still needs some structure and management. It is this kind of information for which knowledge management is the right approach when applied to the dynamic requirements of EA.

While the cost of information loss is disturbing, even more troubling is that this is the kind of information that could help The enterprise achieve greater efficiency, transform to a more effective enterprise, improve its value to citizens, integrate with other agencies and achieve a more mature enterprise. There is, of course, also process and culture changes required – many of these substantial.

The Enterprise Knowledge Base (EKB)

The EKB is an open-source project to address knowledge management challenges in support knowledge, artifact and model management. The EKB provides the ability to search and retrieve information assets for any given concept. Through semantic metadata capture, information assets will be categorized and contextualized to ease search and retrieval. This affords information suppliers and consumers the capability to easily determine what information is available, the subject of the information, and how to obtain the information. Additionally, when information components are visible, managed, and accessible within the the enterprise community, information suppliers and consumers will be more inclined to share and reuse common information rather than re-inventing it.

A goal of the EKB is to integrate both the traditional “document centric” information and semantically enabled structured information (such as XML, Ontologies and Models), making it available, accessible and manageable in this knowledge management platform.

A great deal of information is developed inside and outside of the enterprise in the form of documents, spreadsheets, presentations, models and other forms of files. These artifacts typically go through multiple revisions with the contribution of multiple authors. Some information is official while other artifacts are temporary or simply informative. As this information is collected and developed it is generally put on local or networked disk drives under some directory structure relating to the source of the information (such as a project). While this serves the few who are directly involved with the artifact, it is not effective as a “corporate memory” because such artifacts are hard to find and manage. Furthermore, as artifacts are developed they go through multiple revisions– and the progression of these revisions and the ability to retrieve past versions can be critically important for the maintenance of a reliable record of changes, who made them and why.

The basis of the EKB knowledge management vision is that it is a place for all enterprise knowledge to be stored, managed, published and located. Any information that is worth developing is worth placing in the EKB for the full lifecycle of that information. Placing information in the EKB does not give it “Status” or “Approval”, it simply provides management and versioning of that artifact with the “hooks” for such approvals and status as we shall see, below.

Utilities and Features

The EKB is an operational, open source prototype using ontologies, semantic web standards, OMG modeling and metadata standards to integrate, transform and repurpose enterprise information. The knowledge base supports various kinds of information in a variety of tools and formats. The EKB provides a means for defining and shared concepts that are used for the categorization of information assets. The EKB platform includes a set of utilities that leverage “shared concepts” and provide the following “ontology model driven” capabilities:

1) User Interface: The user interface consists of a simple web based forms interface, allowing information to be entered, categorized and related.
2) Query: Queries can be performed in 2 ways:
    a. The RDF query interface provides a user friendly layer over the underlying technologies (SPARQL) to query the knowledge base, locating information for browsing, editing or analysis.
    b. XML based queries can search and modify the knowledge base and return structured XML documents. The XML based query and knowledge management interface is not currently standards based.
3) Upload/Download: The upload/download utility will accept data in any format to provide configuration management and categorization of that data in the repository – essentially enhanced document management. Data in supported artifact formats (e.g., EDOC, DRM, BPMN, UML, OWL, RDF) will be able to be mapped to the knowledge repository directly. Artifacts linked to the knowledge repository will automatically update that repository when checked in and reflect any changes to the repository when checked out. This simple “check in/check out” paradigm for linked artifacts presents a very simple interface to the leading-edge capability underneath that maps between the data and file formats using shared concepts.
4) Check-in/Check-out using off-the-shelf configuration management tools based on “Subversion” – a popular open source configuration management system.
5) Mapping Facility: The mapping facility implements the generic infrastructure for mapping between ontologies and for import/export of external artifacts in the OMG standard XML Metadata Interchange (XMI) format. The mapping facility is be component oriented and will map import export components to source and target requirements.
6) Ontology of Architecture: Information that is relevant at the enterprise level comes from and is used by a variety of sources that use a variety of tools, standards, methodologies and formats. The problem is that this information is inter-related, there are concepts shared between them. A business case should

reference an enterprise initiative and effect the budget. A business process should be connected to both goals and the underlying systems that help automate that process. System data structures are reflections of business requirements. The EKB implements "shared concepts[1]", a way to normalize and connect the same information expressed in different ways and using different terminology. The ontology of architecture is a set of shared concepts used in enterprise and systems architecture, so we can better integrate information from these many sources – and repurpose that information for other usages.

7) Adapting information: Once shared concepts are defined for a language, an "adapter" can be developed to map between the XMI representations of each artifact to instances of the shared concept ontology.

Key Benefits of the EKB

The EKB can improve core enterprise capabilities, support the development enterprise-level information services and mission critical applications and serve as a basis for information modeling standards. Some of the key benefits include:

- Complete and current information and metadata about that information will be captured and stored for use by the entire agency.
- Supports an information sharing environment because information is made generally available and accessible to any interested party through a simple, user-friendly interface; and provides a platform to migrate internal agency information assets from internal resources to cross-agency information assets where appropriate.
- Provides efficient discovery features so that information consumers can obtain information, formulate answers to business questions and exploit knowledge for better business decisions.
- Reduces the need for labor intensive collection and reconciliation activities to satisfy requests for information.
- Increases stakeholder collaboration by providing a central store for information assets; enhances the ability to implement standards and best practices by providing insight into existing resources.
- Provides for document management and versioning enhanced with the capability to categorize, group and approve information assets in multiple ways, across multiple dimensions, using an ontology.

[1] Shared Concept. An abstract identifier that links two or more terms in order to define their meaning. For example, the "morning star" and the "evening star" both refer to the planet Venus. See http://www.w3.org/2000/10/swap/Primer

- Provides for structured knowledge management using semantic web technologies and a common “ontology of architecture” so that models, architectures and governance information can be connected and managed as a dynamic configuration of knowledge.
- Provides for moving information between documents, models and other “static” artifacts to the ontology of architecture.
- Supports dynamic EA change and configuration management by providing the repository for all relevant information as well as internally and externally prompted change.

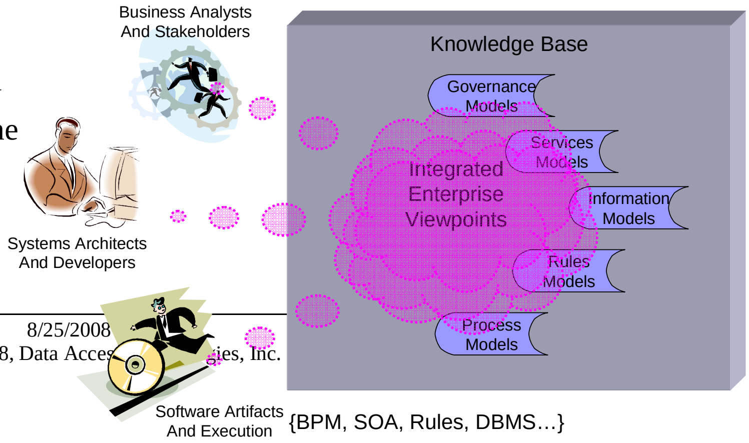
The combination of the Information Model, EKB platform and supporting governance structure results in the enterprise vision for its dynamic configuration management of EA related knowledge. This vision is intended to enhance knowledge throughout the the enterprise and achieve the following goals:

- Support collaboration, sharing and reuse: By enabling and supporting a culture of shared information assets, services, and processes across The enterprise.
- Enable strategic thinking and information driven decision making: By providing a flexible, model driven tool to manage information assets and developing an integrated information model that aligns to the enterprise business architecture.
- Improve the value and quality of information: By facilitating agency level comprehension of information and providing the capability to communicate within The enterprise and/or with partnering agencies about information with a clear understanding of its meaning and confidence in its quality.
- Reducing duplicative efforts: Redundancy in information and processes in the enterprise is not only time consuming and expensive, it is a source of inconsistent and sometimes contradictory information.
- Reduce costs by avoiding redundancy and providing for better integration of initiatives across the enterprise and by reducing the cost of re-developing and re-purposing information.

By providing a means to explore the opportunities for the integration of business processes and information, the enterprise will become more adept at consistent planning and execution of strategic initiatives.

Stakeholder appropriate knowledge

The diagram to the right illustrates how information in the knowledge base in the form of governance models, SOA models, Information models, Rules Models and Process models is integrated and “projected” as a view



appropriate to different stakeholders such as business analysis, systems architects and developers.

Approach

The target EA Data Architecture is expressed as a set of shared metadata concepts using Web Ontology Language (OWL) and Resource Description Framework (RDF) in the ontology of architecture. These concepts will be synthesized from the current The enterprise data architecture, as expressed in previously developed EA artifacts, as well as the Federal Enterprise Architecture (FEA) and Data Reference Model (DRM) 2.0. These concepts will be made available for use via the EKB.

Model Integration

The target EA Data Architecture combines multiple source models to provide a holistic view of enterprise information, regardless of the source.

Aligning to the Federal Enterprise Architecture (FEA) Data Reference Model (DRM) 2.0

Unifying data architecture is essential in providing a foundation for consistent information. The FEA DRM 2.0 establishes a common data model for the purposes of streamlining information exchange processes within the Federal government and between government and external stakeholders..

Integrating Architectural Views

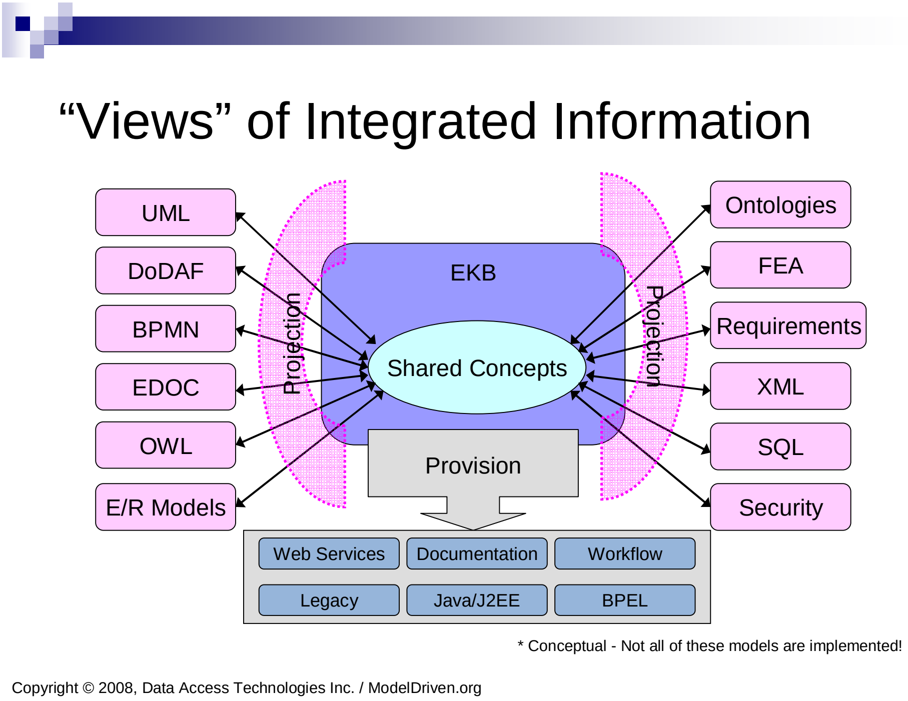
Enterprise, business and systems architectures integrate a number of viewpoints, each applicable to different needs and stakeholders. The primary viewpoints to be integrated in the first revision of the EKB build on industry standards and best practices. These viewpoints are:

- **Information & Data**: To be useful, information must be understood and shared. Information architectures focus on both the information that is shared between parties and the information that is stored in repositories and DBMS systems. The information architecture describes the semantics, context and structure of information based on DRM 2.0.

- **Service Oriented Architecture**: SOA provides a capability for people, organizations, systems and communities to work more effectively together by providing and using services. SOA in this context is both a business and technology concept – a way to understand our organizations, supply chains and communities as service providers and consumers. It also provides a way for our systems to work together more effectively using technologies such as web services. SOA relies in the information architecture to describe the structured information that enables service interactions and services are directly tied to the business processes that both enable and are enabled by services.

- **Business Processes**: Business processes describe how our organizations meet their mission and service responsibilities with well defined processes and activities. Business processes span the range from high-level “value chains” to detailed processes within a group. Business processes depend on and help develop information and both implement and use services.
- **Business Rules**: While rules have not yet been fully integrated into the EKB, the plan is to support rules as one of the primary views into the knowledge base. Rules effect and are effected by the other views, such as information, process and services.
- **Governance**: Governance is an integral part of the architectural process and both informs and is informed by the other architectural aspects. Governance is enterprise wide and business focused.

All structured information “understood” by the EKB (such as any XMI or RDF resource) is “normalized” into RDF using an isomorphic transformation – all of the original information is retained but the format is changed to RDF so it can be used within the knowledge base. A transformation is then made from this format to the “shared concepts” in the “ontology of architecture”. This allows the information to be transformed into other model types.



“Views” of Integrated Information

* Conceptual - Not all of these models are implemented!

The above demonstrates how shared concepts provides a “hub” for concepts in multiple modeling and implementation languages. These concepts can then be “provisioned” to executable components using MDA.

Examples

Document Approval Example

Scenario

In this scenario a fictitious performance evaluation (PE-2008) is required for the OMB. The requirement for PE-2008 is managed by Sue and provided by a contractor – Sam.

Steps involving the EKB

- Sue has previously set up a folder in the EKB for OMB deliverables, called The enterprise/OMB.
- Sue enters a new asset into the EKB user interface called "PE-2008" and categorizes this asset as "required".
- After many months of hard work, Sam checks in a document as "PE-2008".
- Sue reviews this document and has some issues; these are entered as a "corrective action" in the document.
- Jane, the supervisor, runs an artifact status report and sees that PE-2008 is required but not yet approved and notes the corrective action. Jane checks with Sue on the status.
- Responding to the issues, Sam checks in another version of PE-2008.
- Sue reviews the new version and categorizes it as approved.
- As part of the next OMB deliverable, all approved and required OMB deliverables are packaged and sent to OMB.

Model Integration Example

Scenario

In this scenario the DRM based information model stored in the EKB is used to find commonality between a new requirement for an "account" entity and existing account entities.

Steps involving the EKB

- Ralph, a systems architect, is developing a system that has a requirement for an "Account" in a vendor management application.
- Ralph searches the EKB and finds a "Vendor Account" that was defined in the "FMEA" project. This entity looks good but lacks some information Ralph needs.
- Ralph checks out the model that defines "Vendor Account" in UML and, in another UML file, makes a subtype of "Vendor Account" called "Vendor

Management Account". Ralph adds some attributes and relationships to this new entity.

- Ralph checks his UML model into the EKB
- When anyone looks at "Vendor Account" they will now see that is has a subtype "Vendor Management Account" defined as part of the vendor management application. They also see that this was entered by Ralph on 4/22/09 as part of his UML model.

Licensing

The ModelDriven.org EKB is licensed under GPL, commercial licensing is also available. Portions are also available under the OSERA public license and is available all government users at no charge as an open source product.

Status

The above capabilities are in prototype development within ModelDriven.org and are being prepared for wider distribution as open source. ModelDriven.org is currently seeking partners and members to help fund and further develop the EKB.

More information on the EKB may be found on:
http://www.modeldriven.org/projects/EKB/

The source code for the EKB may be found on:
https://sourceforge.net/projects/enterpisekb/

Roles of Actors using the EKB

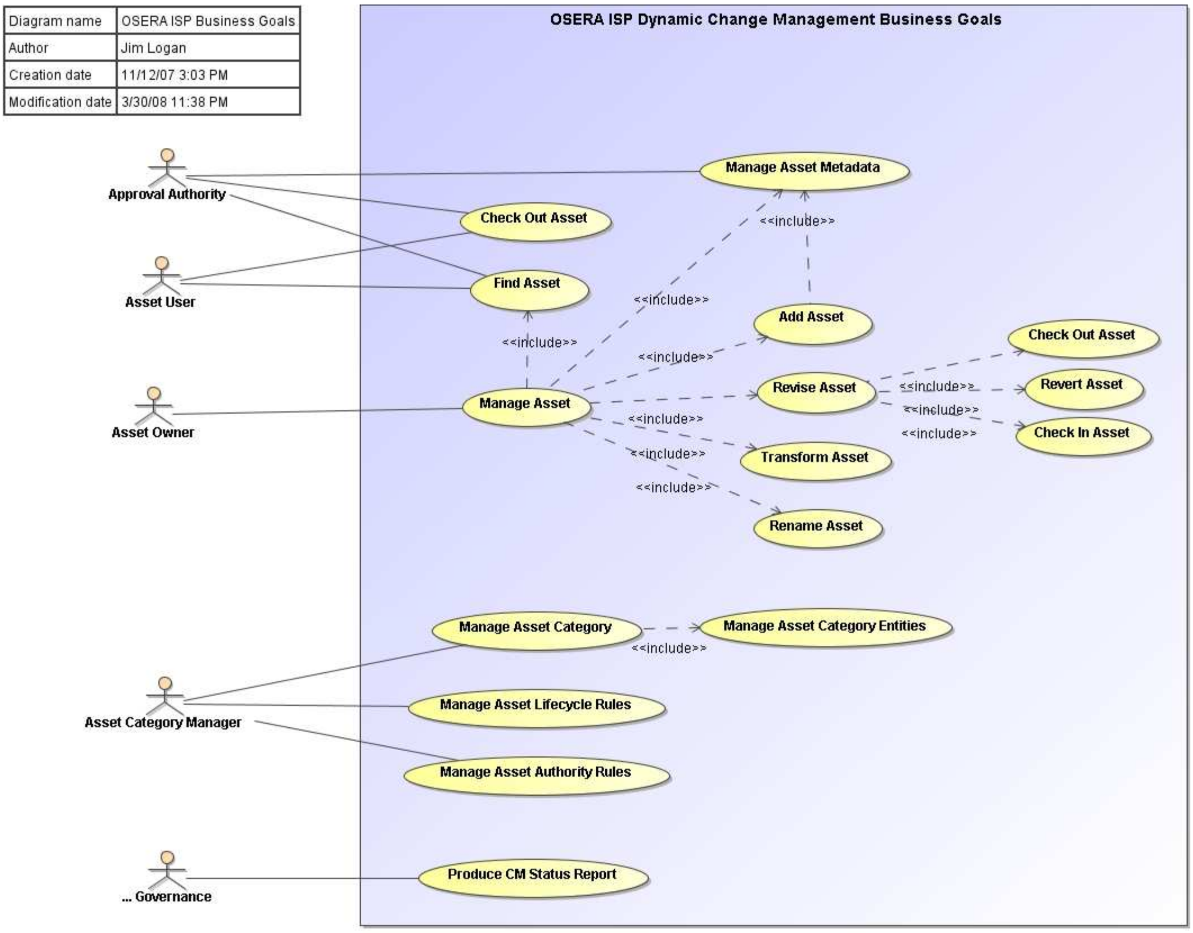


Diagram name	OSERA ISP Business Goals
Author	Jim Logan
Creation date	11/12/07 3:03 PM
Modification date	3/30/08 11:38 PM

The diagram, above, illustrates the roles of people using the EKB to manage information assets.

Asset User

The asset user is able to find and check out assets in the knowledge base. The asset user can narrow down the choice of assets based on category, approval and owner. Once located the assets can be downloaded to the users PC. An asset user may also subscribe to changes in an asset or asset category and be notified of changes by email.

Asset Owner

The asset owner is the primary “user” of the EKB asset management capability. The asset owners record assets in the knowledge base, revise them and can categorize them as well as provide other metadata. Asset owners are also able to search the knowledge base and retrieve assets.

The EKB keeps track of every version of an artifact and is able to provide a complete history of that artifact including who changed it, why and under what authority. Recording why a change was made does require the asset owner to record the reason (the EKB is smart but not that intuitive).

Approval Authority

The approval authority can mark assets as being accepted and/or the official position of The enterprise or a The enterprise business unit. Categorizing assets as approved may require process and authorization in accordance with governance policy.

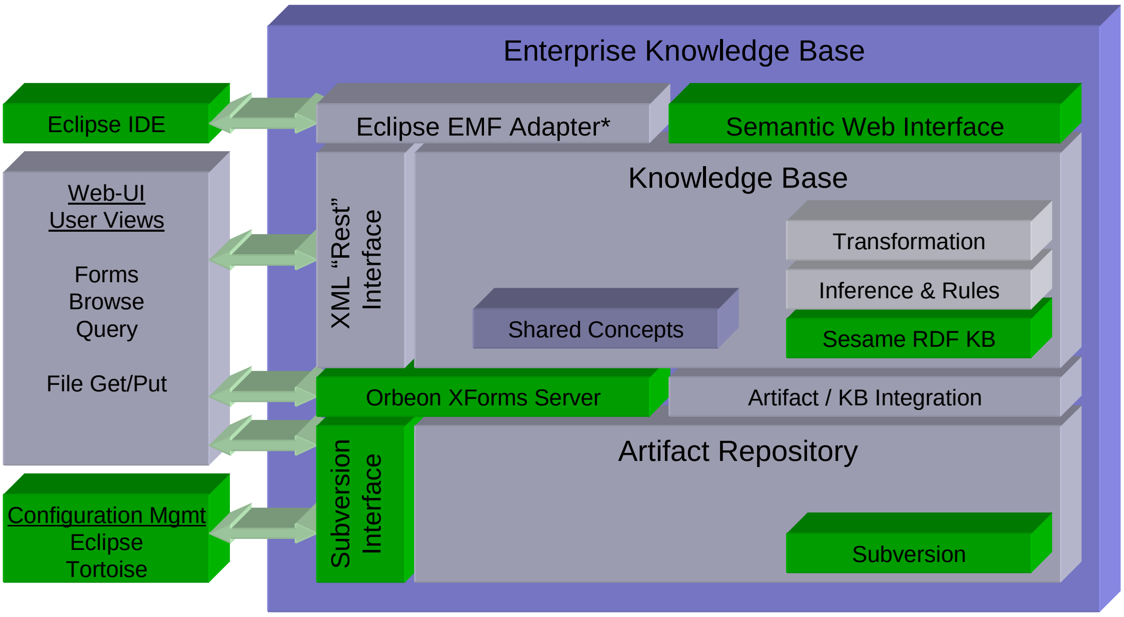
Asset Category Manager

The asset category manager “sets up” the knowledge base by adding new categories and approval rules – thus providing the basis for making the assts accessible as well as supporting the governance process as defined by the EA configuration management plan.

Governance

The governance role monitors the information in the knowledge base to make sure that the required information is in place and appropriate policies are being enforced.

Architectural Overview of the EKB



The diagram above illustrates the high-level EKB architecture which is composed of the following components:

- Artifact Repository – a Subversion based repository for files, such as documents, images and models. The artifact repository manages versioned files in a folder structure.
- Subversion – Subversion is an open source CM product from http://subversion.tigris.org/ which includes the subversion interface.
- Artifact / KB integration – Synchronizes the metadata about all assets with the knowledge base and synchronizes model files with models in the knowledge base.
- Oberon XForms Server – an existing user interface component for managing information in the EKB using XML from http://www.orbeon.com/
- Knowledge Base – the knowledge management part of the EKB which stores and manages all asset metadata and models.
- Transformation – the generic capability to transform between different model formats and languages.
- Inference and rules – RDFS (semantic web) inference with added support for EKB rules and transformations.
- Sesame RDF server – Semantic web RDF database that stores and manages information in the knowledge base. Sesame is from http://www.openrdf.org/

- Shared Concepts – the “Ontology of Architecture” used as a “hub” to integrate different representations of architectural information.
- XML Rest Interface – XML based web interface for the EKB that allows standard XML tools to utilize the EKB.
- Semantic Web Interface – Part of the Sesame server which allows generic semantic web access to the knowledge base, including SPARQL queries.
- Eclipse EML adapter – future plans to support direct access to the EKB from eclipse via the “EMF” API.
- Web UI – User interface for the EKB which includes forms for managing instance metadata and for browsing the EKB based on the Data Reference Model.
- Enterprise Knowledge Base – The core open source technology of the EKB from www.modeldriven.org
- Eclipse IDE – Standard and open source IDE for developers.
- Configuration Management / Tortoise SVN – Client for SVN which allows users to check in and check out files using windows explorer. Tortoise is from: http://tortoisesvn.tigris.org/

Technology references: http://ekb.modeldriven.org/references.html

About ModelDriven.org

ModelDriven.org is a community of government, commercial and university members who use, develop and integrate open source and commercial capabilities to enable agile business solutions based on model driven methods and technologies. ModelDriven.org is standards based, leveraging Model Driven Architecture® as defined by the OMG and the Semantic Web as defined by W3C.

ModelDriven.org is sponsored and is a division of Data Access Technologies, Inc. (www.modeldriven.com). As ModelDriven.org develops it is expected to be "spooled off" to an independent, not for profit, organization.

Who does ModelDriven.org serve?

ModelDriven.org serves its community. This community has both a user membership and a provider membership. The government and industry user community drives the agenda – it is their needs that ModelDriven.org and the provider community are there to address. Projects, such as the EKB, are sponsored and supported by members.

ModelDriven.org serves the open source community by being an active contributor to open source and sponsoring open source projects that help build the Model Driven vision. ModelDriven.org provides open source developers a way to focus efforts on problems that need to be addressed and a way to build quality software that will really make a difference.

ModeDriven.org provides commercial vendors with an outlet for their products and services that support open source and a venue for funded open source projects that are strategically important for both the provider and user communities.

Why Model Driven?

The model driven approach is the cornerstone of ModelDriven.org. Model Driven technologies include architecture, modeling and ontologies that capture knowledge about our business, our environment, our information, our processes and our supporting technologies. The Model Driven approach is able to integrate this knowledge into views of our enterprise or communities that provide the right information to the right stakeholders at the right time. Besides providing information, the Model Driven approach gets more value out of that information by being able to use it to support automated processes, technology development and re-purposing. Model Driven solutions are more technology independent, more agile and less expensive. Being able to utilize the Model Driven approach makes our organizations more agile, more efficient, and less costly to operate – in short it enables the transformation to a better enterprise.

Why the Semantic Web?

The Semantic Web is an initiative and set of standards of W3C to create a "web of data". ModelDriven.org uses the Semantic Web in support of architectures and models; models of our enterprise, our services, our processes and our technologies. The Semantic Web offers a series of advantages; it makes our architectures "web assets", directly accessible and connectible world-wide. The Semantic Web also provides capabilities to make our architectures and tools smarter using the power of ontologies and automated reasoning. One of the first projects of ModelDriven.org is to integrate MDA and Semantic Web worlds.

What are the Open Source technologies?

ModelDriven.org addresses three layers of Open Source assets;

- **Business Solutions -** Solution models, architectures and components including full applications for a variety of domains.
- **Enabling the Model Driven Approac**h - Modeling Tools, MDA Tools and Metadata Management leveraging the Semantic Web. The modeling layer provides for the full life-cycle of solutions from high level business modeling to runtime services.
- **Execution Frameworks and Infrastructure –** including Business Process Execution Engines, Service Oriented Architecture Technologies, Application Servers, Enterprise Service Bus, Asset and Service Discovery and Supporting Infrastructure.

Why Open Source?

Widespread adoption of open and interoperable solutions requires that these solutions be vendor neutral, technology neutral and pervasive. Since Open Source is free, it can be integrated into experimental and production efforts without excessive cost, commitment or lock-in. The ability to "try out" and then to utilize a new approach is critical for pervasive adoption. As an integration solution this must include government and the private sector at all levels – from the smallest department to world-wide initiatives. This is critical because the world wide initiatives must be able to embrace and include the hundreds and thousands of small business and governmental organizations that make up our society. Every individual and organization must have the ability to become part of an integrated "web of communities" at all levels. This ability to integrate, to dynamically form open communities and to embrace large and small organizations is what ModelDriven.org is pursuing, and we believe an Open Source model is the only path to achieve it.

Why commercial providers?

One of the barriers to Open Source has always been the effort it takes to assemble and use Open Source assets and the difficulty of mixing Open Source with commercial products. ModelDriven.org takes a "product like" orientation to Open Source – this gives our user community the benefits of tested, supported and integrated solutions without sunk costs or being locked into any vendor's infrastructure. On top of the Open Source foundation there are products and services from multiple vendors that work together to provide an enterprise solution. By making sure the underlying frameworks and information are fully open, users can take advantage of best-of-breed applications, tools and infrastructure.

Commercial vendors also provide the resource for funded Open Source and client specific development projects, ModelDriven.org is able to draw from the entire community – not just one vendor. Because of the commercial support opportunity, commercial vendors are more likely to donate assets to and participate in Open Source – to the benefit of the entire community.

What is the ModelDriven.org business model?

ModelDriven.org is a membership organization. Membership fees support the organization as well as development of Open Source assets. Members are able to direct part of their membership fee to support specific projects. In addition to membership fees ModelDriven.org makes a small fee on commercial products located through or purchased through ModelDriven.org. Major projects, like the EKB, are individually funded by stakeholders.

What are ModelDriven.org's current and anticipated projects and assets?

ModelDriven.org is driven by the community. The current projects and assets under development or in process of being adapted to open source are;

- **Enterprise Knowledge Base (Prototype) –** The Enterprise Knowledge Base (EKB) is an open source project of ModelDriven.org and the OSERA program sponsored by the U.S. General Services administration. It is currently an operational prototype and being structured for general open source availability.
- **ModelPro (In Progress) –** SOA-Provisioning enables the production of a web service based "Integration Architecture" predicated on high-level business service and process models. SOA-Provisioning brings the architecture to the SOA. SOA-Provisioning is in prototype development based on work done with The enterprise on the OsEra project (www.oOsEra.gov) , as well as open existing open source assets.
- **Ontology of Architecture** The Ontology of Architecture (OoA) is the set of common concepts used across architectures expressed as a formal ontology. The OaA is used to bridge and integrate architectures from multiple sources using multiple tools, standards and methodologies. The Ontology of Architecture is implemented in the EKB. OoA was formally called the "Semantic Core" and is in its second revision.
- **Component-X –** Component-X implements the OMG "EDOC" standard for a Component Collaboration Architecture. Component-X is used for collaboration modeling at the business and technology levels. Component-X has been donated by Data Access Technologies and is available now, but is not undergoing further development.
- **Business Blueprints (Planned) –** Business Blueprints are enterprise and systems architecture assets that help form the basis of business and solutions architectures. Blueprints will be available for a variety of domains. There are multiple blueprints

already in the public domain and acquisition of additional blueprints is currently under negotiation.