Enterprise Architecture Reference Modeling in OWL/RDF

Dean Allemang, Irene Polikoff, and Ralph Hodgson

TopQuadrant Inc,
141 Howard Drive, Beaver Falls, PA
{dallemang, irene, ralph}@topquadrant.com

Abstract. This paper describes the design of and the deployment options for the Federal Enterprise Architecture Reference Model Ontology (FEA-RMO). The goal of any reference model is to provide a basis or starting point for some design process. While this is a laudable goal, it poses an immediate problem for representation; how can a model be represented in such a way that it can be extended in certain ways (for application to a particular problem), but not without regard to the advice that it gives? Reference models are usually expressed in natural language. At their best, such models provide a starting point for designers, and a checklist for their designs, to see that they conform to industry best practices. At worst, reference models expressed in natural language become a source of busy work; designers do not use the models during the design process, instead they spend time after the fact writing up an explanation of how and why they are compliant with the reference framework they've never seriously considered. In this paper, we have used Semantic Web technologies (in particular, RDF and OWL) to represent a reference mode for enterprise architecture in the US government. The content of the model comes from the recent Federal Enterprise Architecture Reference Model effort. We use the capability of RDF to distribute structured information to allow the reference model to be extended (as intended in its design). We use OWL to maintain the consistency of those extensions. The model has been used as the basis for an implementation of an FEA registry, a web-based system for managing enterprise architectures based on the FEA. The work of representing the FEA as formal ontologies was funded in part by GSA.[1]

Keywords: Government Sector, Portals, Knowledge Management.

1 Introduction

Reference models have been developed for a number of areas, ranging from highly technical areas like networking and distributed processing ([10], [11]), social areas like library and archiving ([9]) and even culturally focused areas like Cultural Heri-

[1] We would like to thank Rick Murphy, Enterprise Architect and George Thomas, Chief Enterprise Architect of the Office of CIO of GSA for their vision, support and contribution to the use cases described in this paper.

Y. Gil et al. (Eds.): ISWC 2005, LNCS 3729, pp. 844 – 857, 2005.

tage and Museum management ([12]). In all these cases, the reference model represents some agreement on good practices which, if followed, will provide some specific value to designers of systems in each of these areas.

The reference models themselves are not system models; they are blueprints, templates, or starting points for system design. The intended use of a reference model is as an aid for a designer. It gets past the "blank slate" problem of a design from scratch, by providing a starting point. It also provides guidance to the design, so that the designer can reuse known and proven solution patterns. . The reference model also encourages independent design teams to conform to core principles that will facilitate future integration.

This role of a Reference Model in the design process poses very particular challenges: a reference model must be represented as a reusable asset, which is not a system design in its own right. The engineering of a reference model is therefore a problem of "design for reuse". In short, how can a reference model be represented in such a way that it simultaneously makes enough design commitments to advise a system designer (involved with a system that was not even conceived at the time of reference model development), while leaving that same designer enough latitude to customize a design to the particular needs of the problem at hand. This kind of engineering problem is called by the name "domain modeling for asset reuse" [13].

In this paper, we will use a particular reference model as a case study. In response to a presidential initiative for e-government, the US federal government has developed the Federal Enterprise Architecture (FEA, [1]) a reference model for enterprise architecture. The basic idea of the FEA is that each government agency supports services, functions, operations and technologies that are not unique to their agency. The government as a whole would run more smoothly if all the agencies were to "align" their operations. In 2004, the first full version of the FEA Reference Model (FEA RM) was released. Like other reference models, this is not an enterprise architecture in itself, but a model to guide enterprise architects in government agencies as they create their own, agency-specific, enterprise architectures. Like other reference models, it provides design guidance, while allowing for a certain latitude for the specific agencies.

Reference models are typically written in natural language, and are presented as some form of human-readable document. The reference models of the FEA are no exception.[2] This form of presentation has the advantage that the reference models can be read by anyone who can read PDF files; but it has the disadvantage that the process of reusing the reference model ("alignment") can only be verified by an interpretation process whereby an enterprise architect (or whoever has the job of making the alignment) determines what the reference architecture means, and argues for the particular alignment of their architecture to the model. This is a highly ambiguous and subjective task, and is prone to errors and even misuse.

A formal representation of a reference model addresses this problem by providing an unambiguous (or at least, less ambiguous) representation of the reference model,

[2] Some of the FEA models are available in XML as well as in natural language. However, XML alone (not being a graph representation) can not describe all the relationships within and between the models. RDFS and OWL layered on top of XML provide us with all the language constructs needed to represent FEA.

and allows for the definition of objective criteria for whether an architecture is actually conformant.

But a formal representation brings up a new issue: while some value can be gained from simple formal representations (e.g., the list of business areas, lines of business, and subfunctions found in the Business Reference Model), most reference models have more complex structure than simple lists, or even hierarchies. Furthermore, description of how an enterprise architecture aligns with such a reference model architecture requires more complex consistency checking than is usual even for a taxonomy.

Fortunately, 2004 also saw the adoption by the W3C of the OWL standard for representing *ontologies*, which are formal models that allow the sort of complexity required by enterprise architecture reference models. Furthermore, the OWL standard provides a formal semantic for the meaning of these models, which addresses the issues of fragility and ambiguity of informal models. Finally, OWL provides a framework for combining ontologies and checking their consistency, thereby providing the framework for a systematic discipline for determining how well proposed architectures match the reference model.

1.1 Federal Enterprise Architecture

The FEA has five models: the Performance Reference Model (PRM), the Business Reference Model (BRM), the Service Component Reference Model (SRM), and the Technology Reference Model (TRM) and the Data Reference Model (DRM) Each of these is, at its core, a taxonomic structure of enterprise architecture entities.

The idea of providing a reference model for enterprise architecture is that each agency should be able to use the reference model as a starting point for documenting its own enterprise architecture. The current draft of the FEA RM contains some ambiguity about just how the FEA RM can/may be adapted to apply to a particular agency's enterprise architecture. By examining the example of the DOD [14] and other vanguard agencies who are already applying the FEA RM, we have determined that the following types of adaptations to the FEA RM are being done:

- Some agencies are making extensions to the reference model (describing entities in more detail than given in the FEA RM)
- Additions to the model (adding more siblings to an entity in the FEA RM) are also being done
- Finally, some agencies are making deletions (leaving out entities from the model) and replacements (leaving an entity out while noting that it is being replaced by one or more new entities)

While each agency has considerable autonomy, there are many benefits to having a central reference model to which each agency refers.

1.2 The FEA RM Ontology

We have constructed a number of ontologies using the W3C standard Web Ontology Language OWL that reflects the structure of the published FEA RM. Collectively, these models make up the FEA RM Ontology, or FEA-RMO for short.

The FEA Reference Model Ontology architecture mirrors that of the FEA RM itself [1], that is, the Performance Reference Model (PRM) organizes the overall architecture, making reference to the other models as needed. The Business Reference Model (BRM) draws upon the Service Reference Model (SRM), the Data Reference Model (DRM) and the Technical Reference Model (TRM). In section 0, we describe how OWL was used to represent the various dependencies of these models upon one another, in particular, a recurring design pattern we call the "*Class-instance Mirror Pattern*" that is essential for representing the reference models.

Performance Reference Model

The PRM is organized into layers called Measurement Areas, Measurement Categories and Generic Indicators.

Business Reference Model

The BRM is organized into Business Areas, Lines of Business and Subfunctions.

Service Component Reference Model

The SRM is organized into Service Domains, Service Types and Components.

Technology Reference Model

The TRM is organized into core Service Areas, Service Categories, Service Standards and Service Specifications.

FEA Core Ontology

The FEA-RMO includes a model that is not explicitly called out in the FEA RM, where concepts and properties that are common to all the reference models are defined. This provides modularity to the ontology design that allows for simplified maintenance and integration of the models.

2 Use Cases for FEA-RMO

We have identified several use cases for FEA-RMO [2], but in this paper we will report on those that are supported by the current models and the semantic applications we have built around them. Many of the other use cases rely on modifications to government workflow beyond the technical scope of the FEA-RMO pilot.

2.1 Use Case: FEA Browser

Actor: Enterprise Architect, system developer, project manager. Anyone who needs to consult the FEA for any purposes.

Stakeholders: FEA Program Management Office (PMO), Agency that Enterprise Architect works for, Agency employees and administrators.

Goal: Gain better understanding of the FEA, understand how enterprise entities are aligned with the FEA RM.

Precondition:

A browsing system for the FEA RM. General purpose modeling tools like SWOOP [7] and Protégé [4] can provide some assistance for this purpose.

Steps:

1. User selects an entity from one of the reference models (e.g., a Line of Business from the BRM)
2. System displays information about that entity (e.g., quotes from the RM documents), and the entities adjacent to it (e.g., the Business Area that is comprised of it, the subfunctions that it is comprised of).
3. System displays a legend to orient the user in the appropriate RM document.
4. In the case of agency extensions (see next use case), system displays FEA core and agency extension entities separately.
5. User selects another related entity to browse, or uses a search function to search the FEA for a concept of interest.

Postcondition: none

Outcome:

User viewed and inspected any supported perspectives on the FEA.

2.2 Use Case: Agency Extension

Actor: Agency Enterprise Architect designing agency-specific modifications to the core FEA RM.

Stakeholders: FEA PMO, Agency that the Enterprise Architect works for

Goal: Maintain agency extensions separately from core model, while aligning the modifications with the appropriate parts of the model. Make these extensions available for the Browse use case.

Precondition:

Agency has accepted the FEA RM as a starting point for enterprise architecture modeling, and has access to the FEA.

Steps:

1. User identifies agency to which extensions belong, and validates identity credentials.
2. User selects an element in one of the reference models where a modification is to be made.
3. System provides guidance (based on the current reference model) for extensions, and records the extensions made by the user.

Postcondition:

Model and extensions are in a satisfactory state. The Browse use case is applicable in determining this postcondition.

Outcome:

Agency-specific modifications are stored independently (with appropriate access protections) of the reference model, but are available for browsing.

3 Application of Semantic Web Technology

There are two features of the FEA RM requirements that made RDF(S) and OWL Semantic Web technologies particularly appropriate. In fact, these requirements were so well suited to these technologies, that had we not used RDF and OWL, we would have needed to have invented similar knowledge representation technologies to have completed the project.

There are two other issues for which the current RDF and OWL standards posed some problems.

In this section we outline these features, the ways in which Semantic Web technologies were particularly well suited as solutions, and the ways in which we were able to work around the shortcomings.

3.1 Issue: Remote Composability

A major feature of a reference model is that it has been designed as a re-usable object. It is useful only inasmuch as it can be added to, changed, or otherwise edited. This makes the design of a reference model different from most engineering design.

In order to make use of a reference model, its users have to be able to reference any part of the model, and relate new items to those parts. This requires a robust system for extending and connecting to reference-able entities.

The match of this requirement to RDF is almost definitional. RDF is a graph-based modeling language, in which the identity of nodes and links (the modeling constructs) are done using URIs (making them web-worthy). The primitive operation in RDF is the graph merge; two graphs are made into one by unifying any references in one graph to nodes that are mentioned in the other.

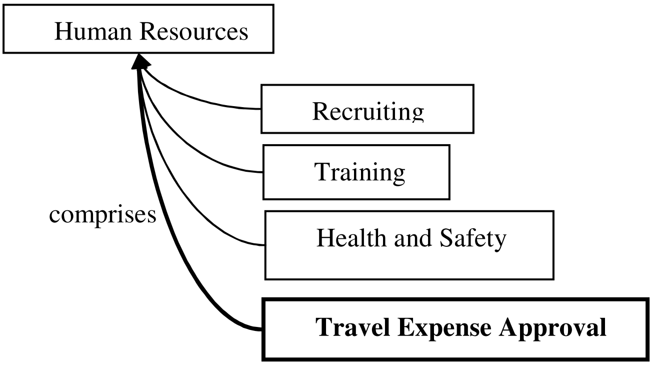


Fig. 1. Human Resources is comprised of several service types

The application of this to a reference model is perfectly straightforward. The reference model is expressed as an RDF graph, in which the parts of the model to be extended are published (as URIs). Then anyone in the world can express an extension to the model, simply by referring to it (importing it), and asserting new triples that refer to the nodes in the reference model.

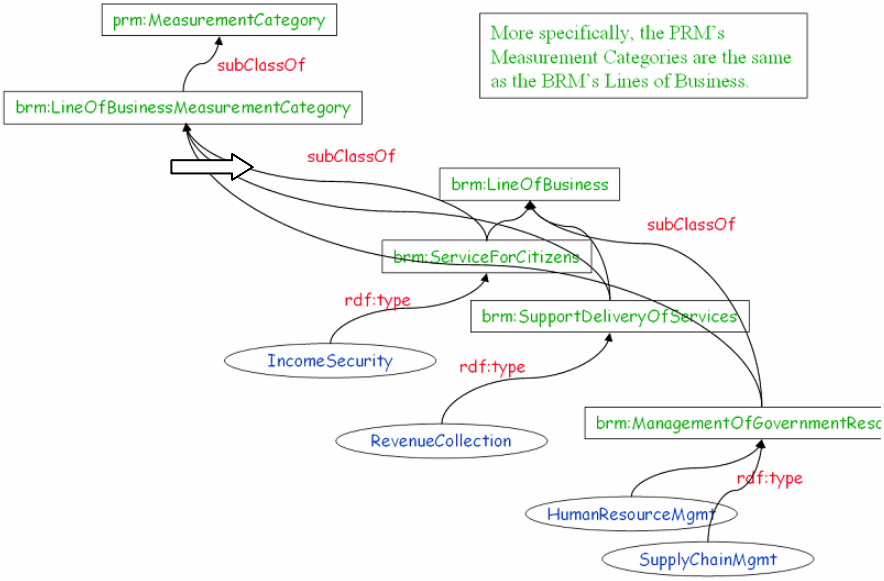
As an example, the SRM identifies a Service Type for Human Resources, which is comprised of several Service Components, like Health and Safety, Recruiting, Training, etc. Suppose that the Federal Junket Agency (FJA) wants to add more service types here, having to do with how they manage travel. In terms of RDF triples, this is simply a matter of storing new triples, shown in **bold** in the Figure 1.

RDF allows the FJA to store the triple (<Travel Expense Approval> <comprises> <Human Resources>) in a separate web page from the FEA RM triples, but have any RDF application treat it as appearing at the appropriate part of the SRM.

3.2 Issue: Inter-model Consistency

The FEA expresses certain constraints between the various models. For example, the PRM makes use of BRM constructs in a number of ways. We will take a fairly simple one as an example.

The PRM specifies that “[T]he PRM’s Measurement Categories are the same as the BRM’s Lines of Business.” This poses a problem for our second use case. If we simply construct our PRM model according to this rule, then when an agency adds new Lines of Business to the BRM, they will not be considered as Measurement Categories, and this rule will not be maintained. We could write a complex procedure to maintain the consistency between the models, but this adds a level of maintenance complexity that could severely restrict the applicability of the project.


prm:MeasurementCategory
subClassOf
brm:LineOfBusinessMeasurementCategory
More specifically, the PRM's Measurement Categories are the same as the BRM's Lines of Business.
subClassOf
brm:LineOfBusiness
subClassOf
brm:ServiceForCitizens
rdf:type
brm:SupportDeliveryOfServices
IncomeSecurity
rdf:type
RevenueCollection
brm:ManagementOfGovernmentResc
rdf:type
HumanResourceMgmt
SupplyChainMgmt

Our solution was to allow the OWL[3] reasoner to maintain this kind of connection between the models. We modeled the connection using OWL [4, 5], and allowed the reasoner to maintain consistency. In this case, the connection between the models was simply a few well-placed subClassOf triples between PRM and BRM entities, as shown by the arrow in the diagram above:

In particular, these subClassOf relations (between the appropriate subclasses of Lines of Business and the PRM Measurement Category) imply (according to the formal semantics of OWL) that any Line of Business will also be considered as a Measurement Category. This greatly simplifies maintenance of the models as agencies make changes; no special-purpose code is required, simply an OWL-compliant reasoner.

3.3 Issue: Removal of Triples

RDF's data model has been described by the slogan, "anyone can say anything about any topic." This feature was very well suited for our first issue, but it brings in a problem as well; it does not provide a direct way for anyone to *gainsay* what another has said. In the FEA case, it does not allow an agency to remove or replace triples from the core FEA-RMO.

One solution to this issue would be to simply insist that no agency be permitted to remove triples from the FEA-RMO; after all, the FEA RM can't provide much guidance, if agencies are permitted unrestricted deletions and additions to it. Nevertheless, it seems that the agencies are already reserving the right to remove triples as well as add them, so we need to support this capability in our modeling system.

Our solution to this problem is to introduce a new owl:ObjectProperty of our own called "replaces", that an agency can use to state explicitly that they are not using some element from the FEA-RMO. Returning to the Federal Junket Agency, the SRM actually already includes a consideration for Travel; but the FJA wants to replace this with a number of travel-related services, as shown below:

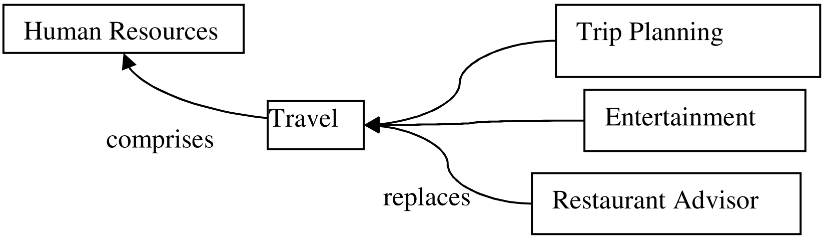


Fig. 2. Replacements for the Travel service

This solution can be made to work, but requires special-purpose code to interpret when displaying the agency modified model. On the other hand, this method does

[3] For this simple example, only RDFS reasoning was required; for more involved constraints, we required some reasoning from OWL, in particular, the use of inverses, transitive properties, and elementary reasoning about the owl:hasValue restriction.

have the advantage that it forces the agency to be specific about the parts of the FEA-RMO that it has chosen to ignore.

3.4 Issue: Individuals as Classes in OWL-DL

So far, we have only discussed the taxonomic portions of the FEA. There are more parts to the FEA, however, which involve referencing the taxonomies in various ways. For the most part, this does not raise any special issues. However, there is one issue that is relevant to modeling in OWL, and in particular, to modeling in OWL-DL. This is the issue that some entities in the taxonomy are occasionally referred to as individuals, and occasionally as classes. This double-usage is not allowed in OWL-DL.

Fortunately, the W3C is aware of this issue, and the Best Practices Working Group has published a Working Draft [3] to address exactly this point. We have used Pattern 2 from this draft (which we prefer to call the "*Class-Instance Mirror pattern*") over 200 times in the construction of the FEA-RMO.

As a simple example, consider once again the Human Resources aspects of the FJA. In the original diagram Human Resources is an instance, and it is related to several other instances with the '*comprises'* property, as already shown in Fig. 1. Now suppose that we want to refer to the *Class* of all service types that comprise Human Resources (say, to be used as described in section 0 or 0). We can't use the Human Resources instance; that isn't the Class of all things that comprise Human Resources.

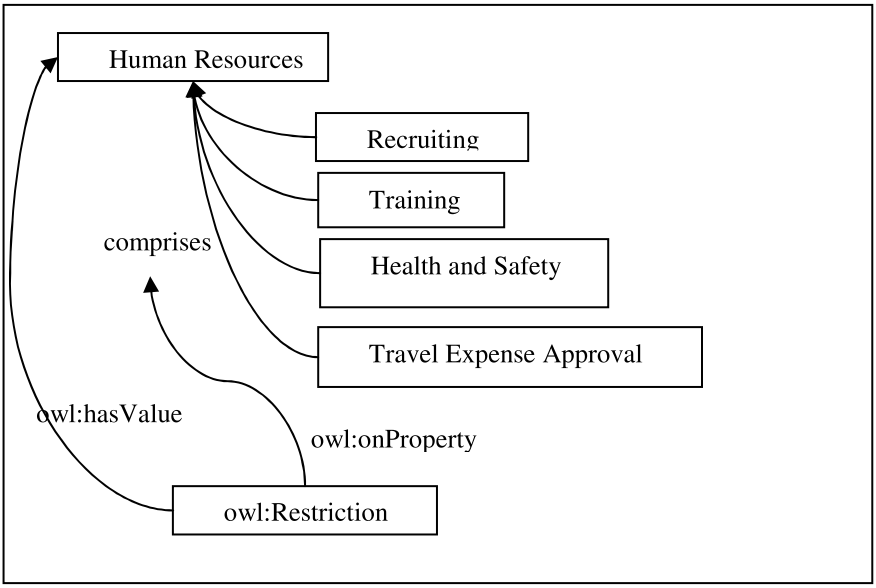


Fig. 3. Restriction onProperty comprises hasValue Human Resources

Fortunately, there is a simple OWL construct that does just this; the (**Restriction onProperty** *comprises* **hasValue** *Human Resources*). This is shown in Fig. 3. The OWL inference engine will maintain the instances of this class, to be exactly those instances which comprise Human Resources. Since this is maintained by the inference engine, it will remain consistent even when new service types are added.

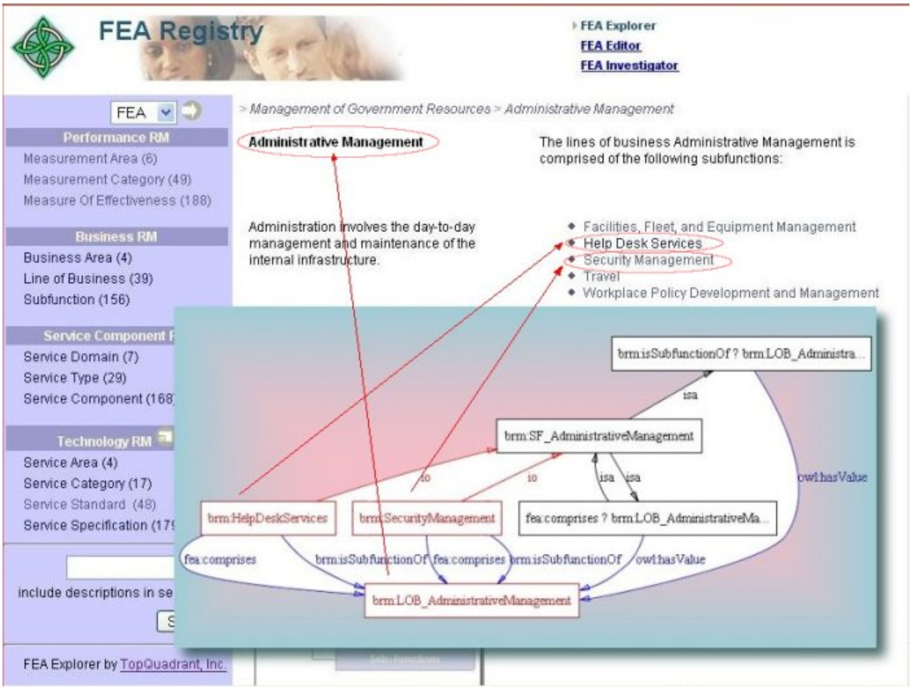
We call this the "Class-Instance Mirror pattern" to highlight the relationship between the instance ("Human Resources") and the Class (an anonymous Restriction).This class exists only to provide a handle on the instances related to the instance, figuratively providing a "mirror image" of the instance in the Class world.

The pattern relies on very simple OWL reasoning (the ability to assign values according to the owl:hasValue restriction, and the ability to classify instances according to the same restriction), so this reasoning can be done quite reliably and efficiently.

4 Implementation: A Portal for the FEA Registry

We used these models to produce a web portal that displays the parts of the FEA, and allows agencies to make extensions to the FEA-RMO. The portal is a pilot project; that is, while it is fully functional and available on the web [6], it has not been deployed for use by enterprise architects in the agencies. In this section, we will highlight some of its functionality as it pertains to the use cases given here and the issues discussed above. The system was implemented using the RDF Gateway product from Intellidimension.

The basic portal is a model-driven web application; every page is generated from information in the FEA RM (with agency additions, when appropriate). A sample page from the application is shown, along with a fragment of the model that generated it.

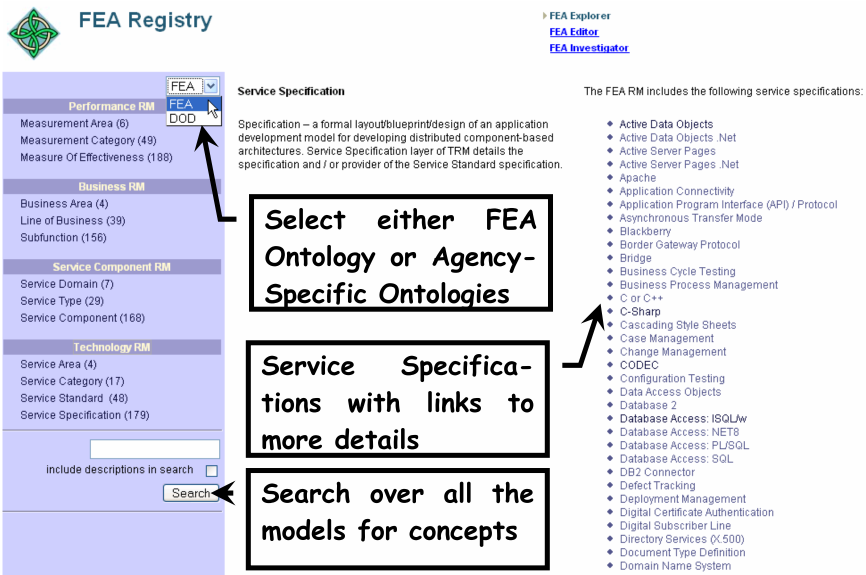


The lines of business Administrative Management is comprised of the following subfunctions:

Administration involves the day-to-day management and maintenance of the internal infrastructure.

- Facilities, Fleet, and Equipment Management
- Help Desk Services
- Security Management
- Travel
- Workplace Policy Development and Management

The registry portal allows users to browse all parts of the four models, either in the core FEA-RMO itself, or with agency extensions. Each entity in the portal links to another place in the model.



Service Specification

Specification – a formal layout/blueprint/design of an application development model for developing distributed component-based architectures. Service Specification layer of TRM details the specification and / or provider of the Service Standard specification.

The FEA RM includes the following service specifications:

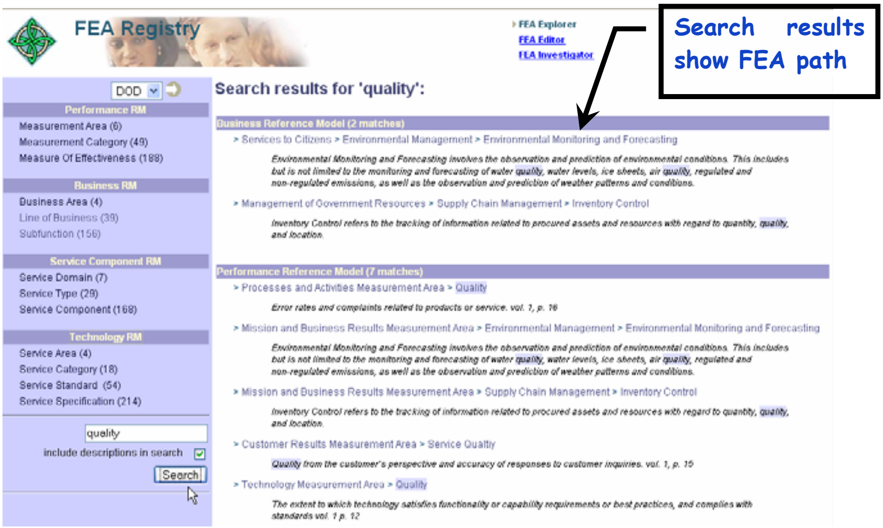
- Active Data Objects
- Active Data Objects .Net
- Active Server Pages
- Active Server Pages .Net
- Apache
- Application Connectivity
- Application Program Interface (API) / Protocol
- Asynchronous Transfer Mode
- Blackberry
- Border Gateway Protocol
- Bridge
- Business Cycle Testing
- Business Process Management
- C or C++
- C-Sharp
- Cascading Style Sheets
- Case Management
- Change Management
- CODEC
- Configuration Testing
- Data Access Objects
- Database 2
- Database Access: ISQL/w
- Database Access: NET8
- Database Access: PL/SQL
- Database Access: SQL
- DB2 Connector
- Defect Tracking
- Deployment Management
- Digital Certificate Authentication
- Digital Subscriber Line
- Directory Services (X.500)
- Document Type Definition
- Domain Name System

Select either FEA Ontology or Agency-Specific Ontologies

Service Specifications with links to more details

Search over all the models for concepts

A search capability uses simple regular expression matching, but shows results in the context of the model itself.



Search results show FEA path

Search results for 'quality':

Business Reference Model (2 matches)

> Services to Citizens > Environmental Management > Environmental Monitoring and Forecasting

Environmental Monitoring and Forecasting involves the observation and prediction of environmental conditions. This includes but is not limited to the monitoring and forecasting of water quality, water levels, ice sheets, air quality, regulated and non-regulated emissions, as well as the observation and prediction of weather patterns and conditions.

> Management of Government Resources > Supply Chain Management > Inventory Control

Inventory Control refers to the tracking of information related to procured assets and resources with regard to quantity, quality, and location.

Performance Reference Model (7 matches)

> Processes and Activities Measurement Area > Quality

Error rates and complaints related to products or service. vol. 1, p. 16

> Mission and Business Results Measurement Area > Environmental Management > Environmental Monitoring and Forecasting

Environmental Monitoring and Forecasting involves the observation and prediction of environmental conditions. This includes but is not limited to the monitoring and forecasting of water quality, water levels, ice sheets, air quality, regulated and non-regulated emissions, as well as the observation and prediction of weather patterns and conditions.

> Mission and Business Results Measurement Area > Supply Chain Management > Inventory Control

Inventory Control refers to the tracking of information related to procured assets and resources with regard to quantity, quality, and location.

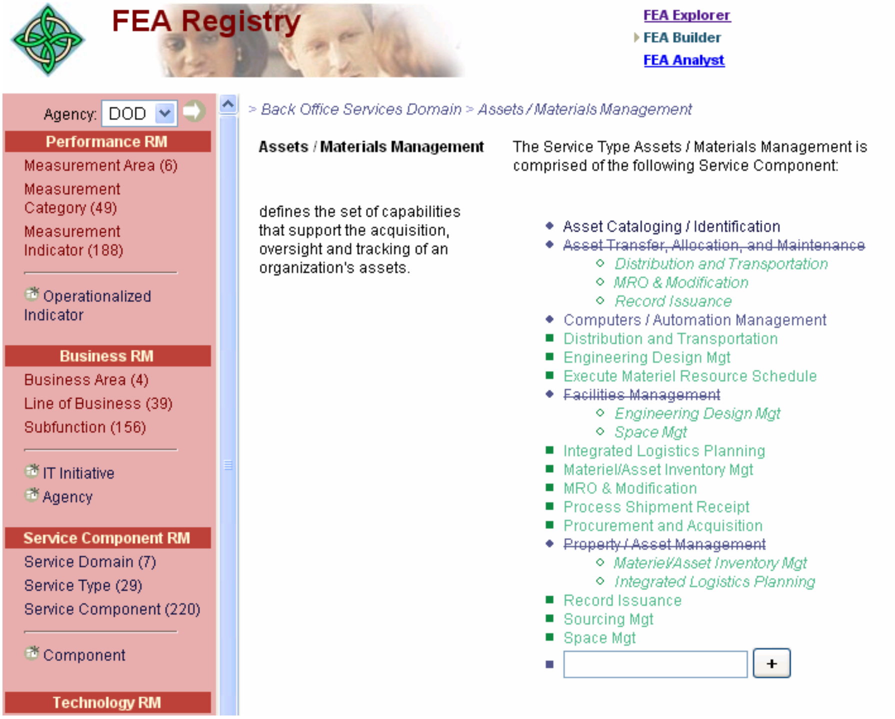
> Customer Results Measurement Area > Service Quality

Quality from the customer's perspective and accuracy of responses to customer inquiries. vol. 1, p. 15

> Technology Measurement Area > Quality

The extent to which technology satisfies functionality or capability requirements or best practices, and complies with standards vol. 1 p. 12

Finally, an agency can make additions or changes to some part of the model and review the changes using a number of reports.



FEA Registry

FEA Explorer
FEA Builder
FEA Analyst

Agency: DOD

Performance RM

- Measurement Area (6)
- Measurement Category (49)
- Measurement Indicator (188)
- Operationalized Indicator

Business RM

- Business Area (4)
- Line of Business (39)
- Subfunction (156)
- IT Initiative
- Agency

Service Component RM

- Service Domain (7)
- Service Type (29)
- Service Component (220)
- Component

Technology RM

> Back Office Services Domain > Assets / Materials Management

Assets / Materials Management

defines the set of capabilities that support the acquisition, oversight and tracking of an organization's assets.

The Service Type Assets / Materials Management is comprised of the following Service Component:

- Asset Cataloging / Identification
- ~~Asset Transfer, Allocation, and Maintenance~~
  - *Distribution and Transportation*
  - *MRO & Modification*
  - *Record Issuance*
- Computers / Automation Management
- Distribution and Transportation
- Engineering Design Mgt
- Execute Materiel Resource Schedule
- ~~Facilities Management~~
  - *Engineering Design Mgt*
  - *Space Mgt*
- Integrated Logistics Planning
- Materiel/Asset Inventory Mgt
- MRO & Modification
- Process Shipment Receipt
- Procurement and Acquisition
- ~~Property / Asset Management~~
  - *Materiel/Asset Inventory Mgt*
  - *Integrated Logistics Planning*
- Record Issuance
- Sourcing Mgt
- Space Mgt
- +

5 Conclusions and Lessons Learned

The entire development process for the FEA-RMO took just about three months, from project inception to delivery. This suggests to us that it is possible to deliver semantic technology solutions in short time frames. A key to this speedy development was a good starting point; the published FEA RM, though it was developed and delivered as a natural language publication, was highly structured and quite consistent. This allowed the modeling process to proceed smoothly and with minimal ambiguity.

Another key to the project's success was the availability of design patterns to help guide model development. We had to document some patterns ourselves [2], but others were available from the W3C [3]. As the craft of ontology engineering develops, more such patterns will be available, making this process simpler.

The role of RDF and OWL cannot be overstated here. RDF as a foundation technology provided a great deal of the functionality needed to support distribution of the models in a coherent and semantically consistent way.

The role of OWL was a bit more subtle. While the reasoning capabilities of OWL were essential in allowing the models to express the appropriate constraints between the elements, the actual reasoning capabilities required were considerably less than those specified in the OWL standard [4]. In fact, the only reasoning that was needed could be achieved with a very simple reasoner that can do RDFS reasoning, combined with a-box reasoning on inverses, transitive properties, and owl:hasValue restrictions. This reasoning can be handled quite easily by any of several technologies (Rete [15], Datalog [16], Prolog [17], etc.), and need not make use of tableaux algorithms. This suggests to us that perhaps other reasoning strategies could have considerable applicability in the semantic web.

Finally, this project suggests a whole area of applicability of semantic web technologies. The features of the FEA Reference Model that made RDF so appropriate (distribution of modifications, the need for modifications to be able to specify just what part of the model is being modified) applies to reference models in general, not just the FEA RM. We feel that the success of many reference model activities has been limited by the weaknesses of the delivery methods (as natural language documents). If our experience is any indication of the future, Semantic Web technologies could well bring a revolution to this field.

References

1. Federal Enterprise Architecture, http://www.feapmo.gov/
2. Allemang, Hodgson, Polikoff, Federal Reference Model Ontologies (FEA-RMO), White Paper, www.topquadrant.com. Feb. 2005.
3. Rector, A. Representing Specified Values in OWL: "value partitions" and "value sets" (ed) http://www.w3.org/TR/swbp-specified-values/
4. Patel-Schneider, Hayes, Horrocks (ed). OWL Web Ontology Language Semantics and Abstract Syntax, http://www.w3.org/TR/owl-semantics/
5. Brickley, Guha (ed). RDF Vocabulary Description Language 1.0: RDF Schema http://www.w3.org/TR/rdf-schema/
6. TopQuadrant, Enterprise Architecture, http://www.topquadrant.com/tq_ea_solutions.htm
7. Aditya Kalyanpur, Bijan Parsia, James Hendler "A Tool for Working with Web Ontologies," *In Proceedings of the International Journal on Semantic Web and Information Systems, Vol. 1, No. 1, Jan-Mar 2005*
8. J. Gennari, M. A. Musen, R. W. Fergerson, W. E. Grosso, M. Crubézy, H. Eriksson, N. F. Noy, S. W. Tu The Evolution of Protégé: An Environment for Knowledge-Based Systems Development. 2002.
9. Consultative Committee for Space Data Systems. Reference Model for an Open Archival Information System (OAIS). (CCSDS 650.0-R-1, Red Book, 1999) (http://ssdoo.gsfc.nasa.gov/nost/isoas/ref_model.html).
10. Zimmerman, H. "OSI reference model - the ISO model of architecture for open systems intercommunications, IEEE Transactions on Communications vol. COM-28 pp. 425-432 April 1980
11. Reference Model of Open Distributed Processing (RM-ODP). ISO/IEC 10746.
12. ICOM/CIDOC Documentation Standards Group, CIDOC Conceptual Reference Model, http://www.ville-ge.ch/musinfo/cidoc/oomodel/.
13. Simos, M. "Juggling in Free Fall: Uncertainty Management Aspects of Domain Analysis Methods," 512-521. Fifth International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems. Paris, France, July 4-8, 1994. Berlin, Germany: Springer-Verlag, 1995.
14. Army Enterprise Transformation Guide, http://www.army.mil/aeioo/aetg/activities.htm

15. Charles Forgy, "Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem", Artificial Intelligence, 19, pp 17-37, 1982
16. Ceri, S., Gottlob, G., Tanca, L.: What you always wanted to know about Datalog (and never dared to ask). IEEE Transactions on Knowledge and Data Engineering 1(1) (1989) 146-166
17. K. L. Clark and F. G. McCabe. PROLOG: A Language for Implementing Expert Systems. In J. E. Hayes, D. Michie, and Y.-H. Pao, editors, Machine Intelligence, volume 10, pages 455--470. Ellis Horwood, Chichester, 1982.