Engineering Enterprise Architecture: Call to Action

By Dennis Wisnosky

There are a plethora of Enterprise Architecture (EA) frameworks, methodologies, tools, views, languages, approaches and 'standards'. There are over thirty years of accumulated detritus with respect to how to build an EA, yet there are no accepted best practices. There is no underlying engineering discipline or mathematics. Instead, an EA is viscerally good or bad, useful or not. When two Architects cannot agree on a framework or a methodology, often a third one is born. When a community does reach agreement, the result is a stovepipe that itself is isolated from other communities.

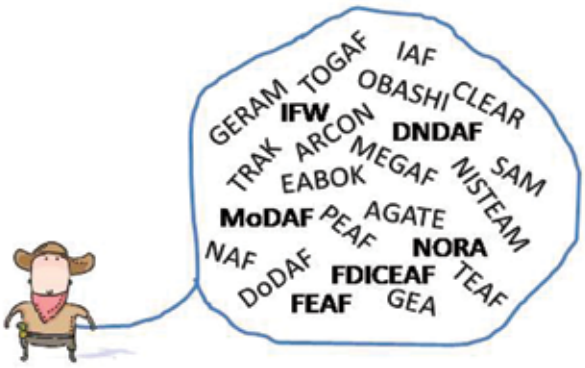


Figure 1.1: Just a few of the Enterprise Architecture frameworks utilized today

An EA is, or should be - a blueprint, a model, a schematic diagram, a recipe, or even a formula that will lead to a specific predetermined result. Sadly, sometimes even tragically, based upon the amount of time and money expended, many or perhaps most EA's, have no practical value. They are built and shelved.

It is time to change this paradigm. It is time to approach the building and use of EA within the context of an engineering, or at least a business discipline. The experience of the U.S. Department of Defense (DoD) over the last 2 years demonstrates that this can be done.

Why would we not want every Enterprise Architect be able to read and understand the meaning of any Enterprise Architecture? Why would we not want computers to be able to execute the End to End Business Process Models that EA can represent?

Background: The DoD Business Enterprise Architecture (BEA) provides a blueprint for DoD business operations prescribed by the Department of Defense Architecture Framework (DoDAF). The purpose of the BEA is to provide defense business system owners and program managers with the information they need to make informed decisions in support of the Department; which in turn supports our Warfighter by ensuring that the right capabilities, resources and material are delivered to them when they need it, where they need it, anywhere in the world.

More specifically, the BEA guides and constrains implementation of interoperable defense business system solutions and guides information technology investment management to align with strategic business capabilities. The BEA outlines and defines the Department's business transformation priorities, the business capabilities required to support those priorities, and the combinations of enterprise systems and initiatives that enable those capabilities.

Point: The BEA is a big deal! It is significant in purpose, cost, man power and maintenance. It is also one of many architectures within the Department. The components, Army, Navy, Air Force, Marine Corps etc… have their own architectures, and these architectures must comply and federate with the BEA.

Issue: While the DoDAF provides the framework or views from which to build architecture, it does not prescribe a standard methodology of how to model the architecture within the views. This lack of a common standard means that the same work is being done over and over again. Certified Enterprise Architects cannot understand one another's work; and if enterprise architectures (EA's) cannot be read by anyone but the people and programs that created them, imagine the waste and cost associated with trying to integrate and/or federate. The cost in time and money within the DoD is substantial. But, it should not and does not need to be this way.

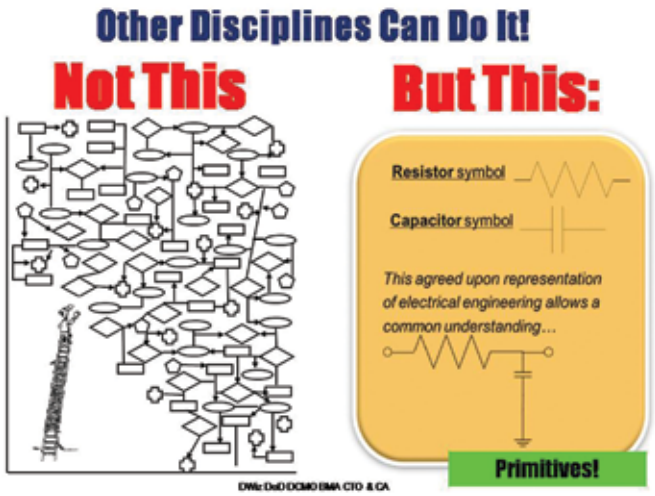


Figure 1.2: A standard EA representation that all can understand is a necessity!

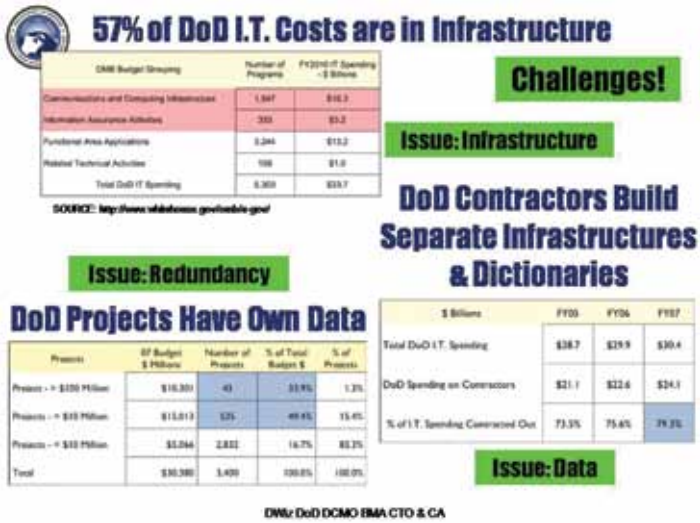


Figure 1.3: Cost in time and money within the DoD is substantial

Standards, Common Vocabularies and Primitives in Enterprise Architecture: About two years ago, the Department sought an answer – an entirely new way of building EA so that they would be:

- People readable
- Machine readable
- Executable, and
- Promote Long-term re-use of authoritative data

At that time the DoD Business Enterprise Architecture was basically a collection of DoDAF products that could be referenced by Services, Components and Agencies to assert compliance as required by the National Defense Authorization Act of 2005. It could serve this purpose, but not the purpose of truly guiding DoD business transformation. It could not be used to build an unambiguous blue print of an end to end business process guaranteed to deliver a specific business capability.

In looking for a standard way to achieve this goal, the Department studied the Business Process Management Notation (BPMN) standard published by the Object Management Group (OMG). BPMN is a graphical notation that is used to build business process models. OMG has been working on BPMN as an international standard for more than a decade.

After deciding that BPMN could be the standard way for the DoD to build business process models and use them as DoDAF OV-6C (process model view) in the BEA, the Department built a BPMN model, sent it to BPMN experts around the world and asked for their interpretation. Surprise! The results were inconsistent. The rich nature of BPMN allowed BPMN experts to interpret BPMN symbols and how they could be linked even for the same purpose – differently.

Rather than abandoning BPMN, or inventing still another methodology, the decision was made to analyze BPMN, reduce its symbol set and write rules that would precisely show how business process models should be built with these symbols. The primary goal was a set of symbols and rules that could not be misinterpreted – these symbols reflected the most base level possible, and for that reason were labeled 'Primitives'.

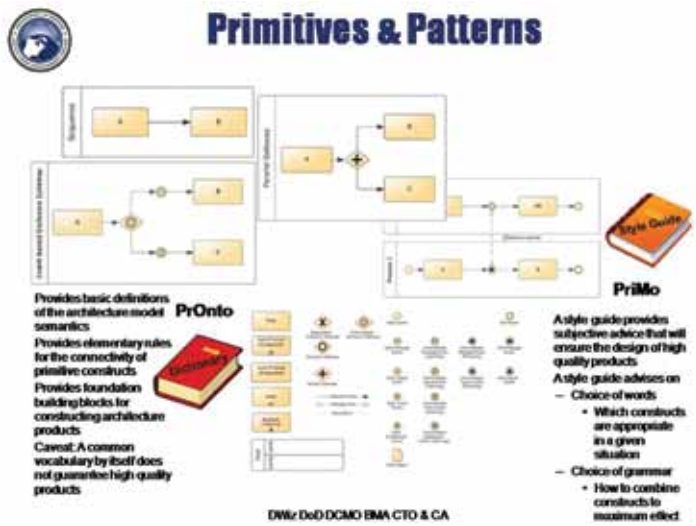


Figure 1.4: DoD Primitives – simplification and reuse of a widely recognized standard

The Primitives were tested initially in building a business process model for Close Air Combat Support. They were further tested in a project called Human Resources Enterprise Information Web (HR EIW), which will be described later in this paper in some detail. The resultant objective was that OMG would in turn embrace the Primitives as BPMN 2.0 Process Modeling Analytic Conformance Subclass. This goal was achieved in August 2010.

Standards, Common Vocabularies and Primitives in Life: The idea of using Primitive symbols and the rules for linking them to build an EA is the same as the idea of the Periodic Table of Elements. Each element is a specific pattern of atoms. The elements occur in nature and mankind has figured out how they are put together in nature – the construct of each atom. Of the 117 know elements, some stand alone such as Oxygen and Hydrogen. And, we know that two Hydrogen atoms and one Oxygen atom combine into a pattern, a molecule - H2O which is water. Any other pattern of Oxygen and Hydrogen is not water.

Engineering: Every Electrical Engineer (EE) knows their symbols. Resistors are basic building blocks of electronic circuits (see figure 1.2). In a schematic diagram – the EE equivalent of an EA, a resistor looks like the track of a snake through sand. Every EE in the world knows this symbol. The value of the resistor is measured in ohms and its power capacity is measured in watts. This is the resistor's metadata. Every EE in the world knows how to read and understand this metadata.

The second most common electronic symbol is a capacitor (see figure 1.2). The capacitor symbol looks like two roads stopping at an intersection. Capacitor metadata is farads for its value, and some types of capacitors have polarity. Every EE in the world knows the capacitor symbol and how to read and understand its metadata.

The third most common electronic symbol is the ground. Visualize an upside down Christmas tree as the symbol for ground. The ground in an electronic circuit is the reference point from which all measurements are made.

These symbols and about 130 others are all that are allowed and all that are necessary for EE's to design and build any and every electronic circuit. But, it takes more than the symbols. It also takes the rules, the mathematics which tell the how the symbols can be fit together, or the circuits simply would not work.

A circuit is a pattern. For example, the pattern made up of the resistor, capacitor and ground is called a filter. Filters are very important circuits. A very specific filter allows only the TV channel we want to see to be displayed to us and all of the others to be blocked, to be filtered out. The benefit of patterns is that when we identify the best pattern - the best filter for the ESPN channel for example, we can use that pattern over and over again. It does not need to be redesigned. Patterns that EE's use, literally make possible our modern world.

There are other far older symbols and patterns that are important to our lives. They occur in music. Music is based upon symbols that every musician in the world understands. There is a precise mathematics behind the symbols and a standard way of each constructing each line and writing patterns, and of course reading - music. In the vocabulary of music, notes are written on a standard 5 line Staff where each space represents a different

note. The pattern of notes is repeated every eighth note. Because of this standardization, a Russian Oboe player with the Moscow Symphony Orchestra can play equally as well with the Chicago Symphony Orchestra, even though he or she may not speak or understand a word of English. All that is necessary is an understanding of the common vocabulary, the primitives of the music discipline.

Discipline: From Merriam Webster, "orderly or prescribed conduct or pattern of behavior".

Just as Electrical Engineering, music, and the elements found in nature are supported by a discipline, EA Primitives - BPMN 2.0 Process Modeling Analytic Conformance Subclass is also supported by a discipline, the DoDAF 2.0 Architecture Primitives Series: Primo/Pronto.

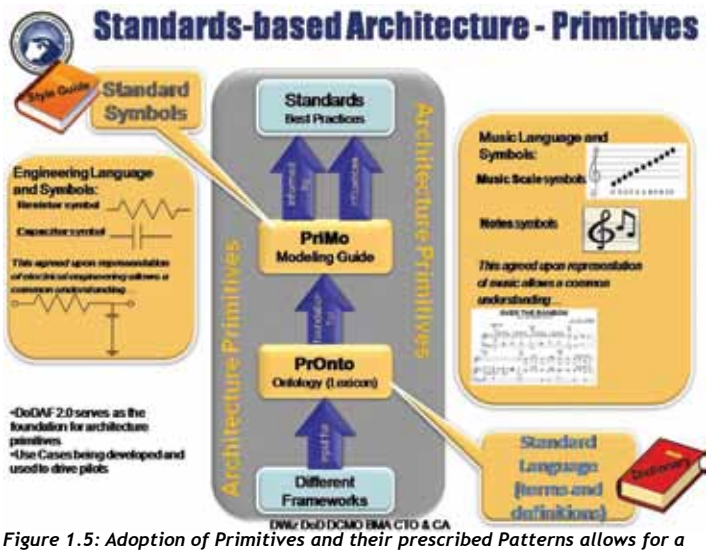


Figure 1.5: Adoption of Primitives and their prescribed Patterns allows for a standard EA representation

- **Primo/Pronto:** At this time, is comprised of two documents, and there will be more to come. The first, called Pronto, provides basic definitions of the architecture model semantics currently represented in the DoDAF AV-2; it specifies the elementary rules for the connection of primitive constructs, and notes the foundation building blocks for constructing architecture products. Because a common vocabulary by itself does not guarantee high quality products, the second, called Primo, is focused on the OV6-C. Primo provides prescriptive advice that will ensure the design of high quality products, by advising on choice of words, which constructs are appropriate in a given situation and how to combine constructs to maximum effect. The documents can be found at: http://cio-nii.defense.gov/sites/dodaf20/journal_exp3.html.

Primitives lead to Patterns: Primitives are nothing more than the symbols – the resistors, the notes, and the elements. The patterns are what can be done with those symbols in a repeatable way. One pattern could be the end to end business process model of how DoD does Identity Management. Another pattern could be the end to end business process model of how DoD does User Authentication. In the BEA, these patterns would be repeated. They would be reused over and over again, rather than reinvented for each new business system. Primo and Pronto show how to build these end to end business process models with Primitives and how to build and discover patterns within these models.

- The entire Department of Defense building Business Architectures based upon Primo and Pronto would ensure that EA's are people readable and machine readable – two of the Department's goals. The latter goals - that EA's be executable and promote long-term re-use of authoritative data, require the use of new technology (Web 3.0) and additional standards. These standards, maintained by the IETF and the W3C, come from technology invented by DARPA, and are known collectively as Web 3.0 or the Semantic Web.

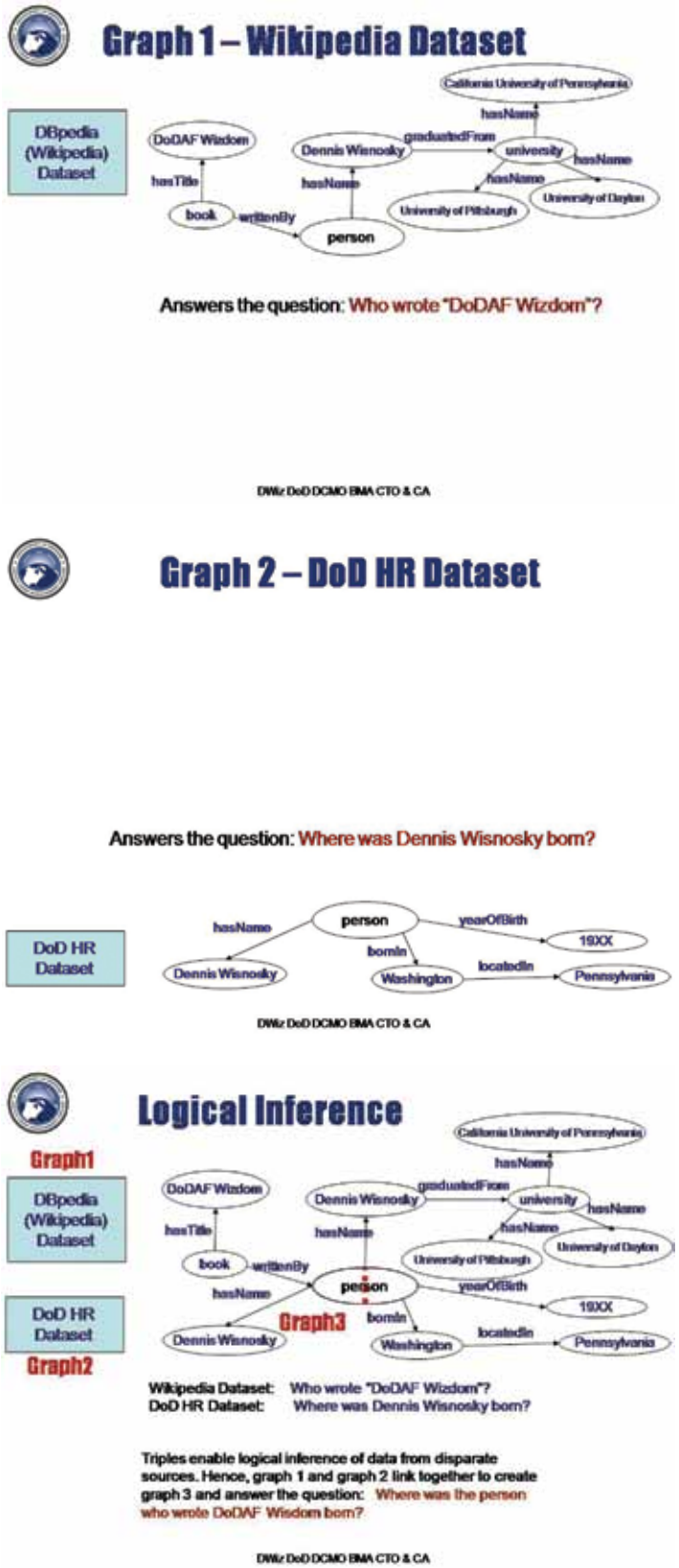
Web 3.0 - The Semantic Web: Web 3.0 has to do with tools and methods for finding and managing knowledge. Web 3.0 is based upon a standardization of the vocabulary of the web (URIs – Universal Resource Indicators), and a specification of how terms, objects, people, places, etc. are related in order to provide semantics for the terms. A document or file containing these relations is called an ontology – the description of the relationships of things, and the inference rules about them. The simple categorization or classification of these terms is a taxonomy – the skeleton of an ontology. All of the words that we use to tell us what we know about objects, people places, etc., make up the vocabulary of the domain in which the words are used. For examples, it is the business context that tells us that a particular tank is for fuel, not for fighting.

The purpose of this new technology is to associate and retrieve data in an unambiguous manner from any source that publishes its data in compliance with the standards. In the space of the World Wide Web (www), it is used to retrieve context specific data with respect to the query or question being asked, versus pages of keyword associations that may or may not contain the data being sought. In an organization it could be used to find the names and telephone numbers of all of the employees born within 25 miles of Washington, Pennsylvania even though there is no single data base with this information. The technology works by inference. It finds data that the ontology says is the same, or related in some exact way to the question being asked.

Data must be described in a totally new way for this to work. This new way must make data simple, understandable and yet still be structured enough to allow for query across a federated "data web". The World Wide Web Consortium (W3C) maintains two standards, one called RDF (Resource Description Framework) and another OWL (Web Ontology Language) that has become the basis of the Semantic Web. "Triples" is the RDF terminology for how these standards structure data. A triple is a statement expressed in the form of subject-predicate-object. Two examples are: Dennis is a man; and Dennis was born in Washington, Pennsylvania. "Dennis" is the subject in both statements; "is a" and "was born in" are the predicates; while "man" and "Washington, Pennsylvania" are the objects. Data structured in this way allows for logical inferences to be formed from disparate data sources. For example, based upon a biographical dataset on Dennis, it can be inferred that a leading Enterprise Architect was born in Washington PA.

Because these data stores can be visualized as a graph - a set of nodes and edges, each of the triples consists of two nodes connected by an edge. The nodes represent the subject and object of an assertion, and the edge represents a predicate or property. Where triples share physical instances (URIs) in the

object of one and the subject of another, they can be represented as a larger graph interconnecting the triples. A data store that contains the set of interlinked triples is called a triple store. See figure 1.6 below for a graphical example.



A real world example relevant to the DoD: There is an earthquake in Haiti. Service men and women with the following qualifications need to be deployed immediately. They need to speak Creole and English, they need to be deployable within 24 hours, and they need to have more than 12 months remaining on their Army, Navy, Marines or USAF commitment. Responding to this is difficult because this data is everywhere in the world and it is dynamic. It is best to go to the authoritative data sources (ADS) - the owners of the data to find answers to ad hoc questions like this, but in a situation like this time is critical. The utilization of this technology allows us to respond in real time versus hours or days – here is how.

BEA Solution Statement: Within the DoD, the key to being able to answer such questions as the ones stated in the example above, is the Business Enterprise Architecture. This is because the BEA contain the "Domain Vocabulary" which in turn is used to define the "Domain Ontology". The BEA establishes the context within which a term is used. This context unambiguously defines the meaning – the semantics of the term which is only truly known in the business context within an end to end business process model. For example, in the business process of 'Recruit a Service Member', the result could be an Airman, a Soldier, a Sailor, or a Marine. The fact that all four are service members can be inferred from our construction of a high level BPMN business process model. A more detailed model in the architecture could tell us which service a particular service member belongs to, and point us to the ADS which contains data about him or her.

Building the BEA based on the standards described here: BPMN – OMG Analytical Conformance Class, and the Semantic Web standards from W3C, will significantly advance the Department from today's world. We will have:

Visibility – that is, ability to pull and display data in real time or near real time from the authoritative source – no matter where that source is.

Agility – when assets, the data, are unambiguously described, the entire environment becomes plug and play. Graph based models are by definition infinitely extensible, linkable to one another and therefore straight forward to change.

Federation – domain vocabularies described in RDF/OWL and an ontology that describes how domains are related means that federation, linking architectures is virtually automatic.

Savings – today, a significant part of the DoD IT budget is spent on the meaningless process of converting one data format to another. In the industry, this is referred to as Extract, Transform, Load or ETL. Once a cottage industry, building translation maps and interfaces between systems is a major part the DoD IT infrastructure today. An ontology makes this unnecessary process unnecessary. There will be no worry about an exact syntax match for sailor or airmen. The ontology says that they are both service members. The authoritative sources can keep their data formats and their schemas as they want them. A query to the DoD semantic web will find the data. The amount of interface development will be reduced by orders of magnitude.

Executable Architecture – exciting as visibility, agility, federation and savings are, they are just the tip of the

iceberg. While it cannot be said definitively for all, research suggests that an end to end business process modeled using OMG standards can be executed directly by an OMG BPMN Executable format standard-based runtime engine, or even a properly constituted BPEL engine. Standardization of such execution within the DoD also depends on the processes using DoD standard messages which are built using the DoD Common Vocabulary, defined in the ontologies referenced above; as well as implementing standardized service interface contracts.

In this way, the business processes can be executed in a distributed fashion by DoD components and agencies while remaining demonstrably compliant with the BEA definitions of standard end-to-end business processes. This compliance is ensured because the components and agencies would be executing, at the highest level of their process automation, the processes defined in the BEA. The (web) services orchestrated by the processes would work with the standard processes, and in many cases, also work directly with the exposed DoD semantic web RDF data.

In other words, build the business process model, make it use the common vocabulary and standard messages to interact with services, and have the services use the authoritative data sources, and hit run. This is the ultimate goal.

Impossible? Consider that electronic circuits are compiled today from stored patterns, not designed a component at a time. Player Pianos and synthesizers play music from code that is the music. Airplane pilots need only to place their aircraft at the beginning of a runway and touch nothing else until the end of a trip. The processes that enable the complex products of the 21st century to work are based upon models and execute automatically, often with their human overseers not allowed to touch their controllers. These advances did not happen overnight. Their underlying standards and tools have been being refined for decades. Enterprise Architectures have also been being built for decades, but with virtually no advancement in the tools or standardization of methods used to build them.

The DoD is leading this change. The DoD built the original EA's thirty years ago and developed the IDEF's and IGES as well as factory automation schemas which still endure worldwide.

Implementing Semantic Web Technology

Nearly two years ago the decision was made to terminate a DoD program that had been in development for more than a decade. A main tenet of this program was to reach into ADS's in the HR systems of the Department and to populate a data warehouse that could be searched. One of the problems that could not be solved is the one described in this paper, how to be sure that a person called an airman, or a marine, or a sailor or a soldier was counted as a service person. Semantic technology seemed to offer a solution to this conundrum and more.

Permission was received to investigate this solution. The goal is/was to build an HR Semantic Application that:

- Reports near real-time, authoritative information on-demand,
- Supports enterprise information standards (Open; HRM ES)
- Supports IT flexibility/agility

A crawl-walk-run roadmap – that goes out for four years - was created, with detail for the first two years laid out in 90 day deliverables know as PoDs (Proof of Delivery). An operational goal of 18 months was also established. The official name for this initiative is the Human Resources Enterprise Information Web (HR EIW).

The underpinning of HR EIW is the Domain Ontology/BPM Methodology described in this paper: The methodology of creating the data that goes in HR EIW triple stores is based on end-to-end business process models. The HR EIW common vocabulary is built collaboratively by a team using a semantic collaboration Wiki, commonvocabulary.army.mil, an existing DoD capability. An example of an end-to-end business process model is for the DoD End-to-End Military Pay Process. This effort builds and captures the domain vocabulary in a domain ontology. This enables the team to guarantee that they have the meaning of the data in context.

PoDs are mandated deliverables every 90 days. Each PoD builds on the previous. We learn something from each PoD. PoD1 delivered in December of 2009, was to simply test the ability of the team to work together and demonstrate that a question could be answered by going to two different data stores and displaying the results in any arbitrary format.

PoD2, 90 days later, was to produce the same result, but to go to an actual RDF triple store which contained the data.

PoD3 produced even more robust results, from scrambled data in an Army database as well as a Marine Corps database. The result of this PoD was essentially the same as the project the original project, thus proving the validity of the Semantic Web approach.

PoD4, delivered in September was based on the real world use case described earlier in this article. There is an earthquake in Haiti and there is a need to find the men and women in uniform who can be deployed within 24 hours, who have at least a year to go in service, who speak either English and Creole or English and French. This data is in data stores around the world in native formats. The proof of this technology is that based upon the domain ontology, data was interpreted on the fly and presented as a virtual triple store, allowing the authoritative data to be queried in real time. Added benefits include being able to tell precisely where the ontology was either not adequate, or was not being followed.

Interoperability through MDI: This project has been the pathfinder for all BEA work in the future and for not only guiding and constraining investment, which is the legislated purpose of the BEA, but also for serving as a true TO-BE blueprint of how systems and services should be built to deliver needed capability. This is called: Model, Data, Implement (MDI).

Model – build the end-to-end business process model necessary to describe how to deliver your capability.

Data – build the common vocabulary necessary to implement the end to end, and identify your authoritative data sources, populate your data store – using the standards.

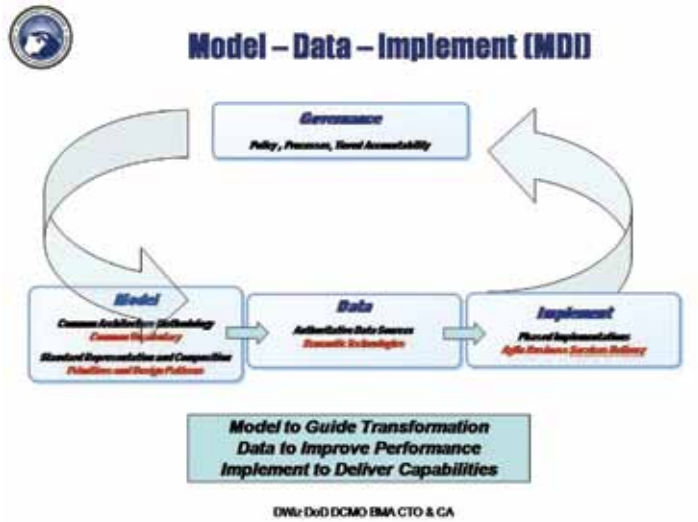


Figure 1.7: Model - Data - Implement

Implement – in a short period of time, either executing models directly, or generating runtime execution artifacts directly from the models and data.

Overriding MDI is an overarching comprehensive governance process. This is necessary because if one step is skipped or short cut in MDI, nothing works. The beauty of the 90 day PoD concept is that this is known quickly.

The true meaning of this message is: **Model** to guide transformation, **Identify** Data to improve performance, **Implement** to deliver capabilities, and **Govern** to ensure that it comes together.

Implications for the International Defense Cooperation Community: Engineering disciplines have a shared understanding of their world through standardized techniques and a common vocabulary. In Enterprise Architecture there are a multitude of ways to describe the same concepts and perspectives. As a result it is difficult to validate, communicate, and reuse architecture content, thus increasing cost, inefficiency and redundancy. But what if there was a unified discipline behind Enterprise Architecture? Think back to the music, electrical engineering and aircraft pilot examples, their fundamental discipline, thus accepted and adhered to standards, allow for the music of the great composers to be shared and played all over the world; they allow an engineer from the Europe to work in the US and vice versa; and they make possible international travel through flight. The implications, domestic and international, of a common discipline and standards around Enterprise Architecture are substantial; and with the incorporation of web 3.0, the capabilities provided by semantic technology, the implications for shared knowledge (not information) are even more significant. Stay tuned as we continue to explore this path within the Department of Defense, Business Mission Area.

Mr. Dennis E. Wisnosky is the Chief Technical Officer (CTO) and Chief Architect (CA) of the Department of Defense (DoD) Business Mission Area (BMA) within the Office of the Deputy Chief Management Officer (DCMO). Mr. Wisnosky is responsible for providing expert guidance and oversight in the design, development, and modification of the federated architectures supporting the Department's Business Mission Area. As such, Mr. Wisnosky is leading the transformation of architecture-driven business systems and services development, and deployment. He ensures that Business Process Models are based on a standardized representation, enabling the analysis and comparison of end-to-end business processes leading to the re-use of the most efficient and effective process patterns and elements throughout the DoD Business Mission Area.