Open Source eGovernment Reference Architecture
Osera.modeldriven.org
Caveat

OsEra and the Semantic Core is work in progress, not a ready to use capability
OsEra

• What we will cover
  – OsEra Overview
  – Model to Integrate – From business model to execution
  – Synthesis of MDA, Semantic Web and FEA
OsEra Stack

- **Tool Integration**
  - Component-X
  - System Architect
  - Eclipse
  - UML

- **Architecture Modeling**
  - Objectives
  - Information
  - Process
  - Rules

- **Model Repository**
  - FEA
  - EDOC
  - UML
  - Semantics Core

- **Provisioning**
  - Transformations
  - Import/Export
  - Model->Integrate

- **Enterprise Service Bus**
  - BPEL Processes
  - jBoss Application Server
  - Policy
  - Web Services

- **Semantic Web**
  - Architecture Publishing
  - Inference
  - RDF & OWL

**Eclipse Environment**

- Capture existing information and integrate with choice of tools
- Manage models and information from diverse sources across projects and communities
- FEA with real time metrics
- Eclipse is an open source “IDE”
- Runtime capabilities for deployment and integration of application components

**Capabilities**

- Provide tools for the entire integrated life cycle
- Integrate diverse information into a coherent enterprise view
- Import/Export external information and produce documentation and technical artifacts.
- Publish and Integrate enterprise intellectual capital on the web

**Additional**

- This is planned but not the current focus
- RDF & OWL
- Model Integration
- Inference
- Publish and Integrate enterprise intellectual capital on the web
- Eclipse is an open source “IDE”
- Runtime capabilities for deployment and integration of application components
- Capture existing information and integrate with choice of tools
- Manage models and information from diverse sources across projects and communities
- FEA with real time metrics
Agility Enablers & Cost Reduction

• Value Chain Analysis
  – Analyzing and restructuring business processes based on realized customer value

• Model Driven Executable Architecture
  – Executable enterprise architecture to realize business goals with systems and workflow automation

• Business Service Oriented Architecture (SOA)
  – An enterprise modernization strategy supporting business services, integration, reuse and common components across a system of systems integrated with SOA/ESB

• Enterprise Service Bus (ESB)
  – A technology platform to support the deployment and integration of SOA components.

• Semantic web with “semantic core” integration ontology
  – Make architectures a web resource able to be interconnected and analyzed
  – Integrate information into a coherent enterprise view

• Combined effect of architected, automated processes
Standards Base

- **OMG-MDA Specifications**
  - Enterprise Distributed Object Computing (EDOC)
  - Unified Modeling Language (UML)
  - Meta Object Facility (MOF)
  - XML Model Interchange (XMI)
  - Query View Transform (QVT)
  - Ontology Definition Metamodel (ODM)

- **W3C Standards**
  - XML, XML Schema, Web Services
  - Business Process Execution Language (BPEL)
  - Ws-Security, Ws-Policy
  - Resource Definition Framework (RDF)
  - Ontology Web Language (OWL)

- **Java Community**
  - J2EE set of standards
Model to Integrate

From business needs to executing solutions
eGoverment MDA Process

- Stakeholders
- Business Drivers
- As-Is Business Processes

- As-Is Systems

- Open Source eGov Reference Architecture

Program Management and Risk Assessment/Mitigation
Change Management, Configuration Management, and Communications

Computation Independent Model (CIM)
- Business Context
- Value Chains
- Roles and Collaboration
- Information Model

Platform Independent Model (PIM)
- Service-Oriented Architecture
- Data Model

Platform Specific Model (PSM)
- Web Services
- Components

Business Case
EA Report
Sequencing Plan

Stakeholders
- Business Drivers
- As-Is Business Processes

• Stakeholders
• Business Drivers
• As-Is Business Processes

• As-Is Systems

• Open Source eGov Reference Architecture

Program Management and Risk Assessment/Mitigation
Change Management, Configuration Management, and Communications

Copyright © 2006
Data Access Technologies, Inc.
Slide 8
Value Chains

Mission-Critical Value Chain

- Plan and Design
- Develop and Deliver
- Provide After Care

Support Services Value Chains

- Development of Government-wide Policy
- Marketing
- Acquisition

Shared Services Value Chains

- Financial Management Services
- I.T. Services
- Human Capital Services
Disciplines – Areas of Responsibility

- Financial Management
- Acquisition
- Solutions
- Property Management
- Policy
- Business Intelligence
- Human Resources
- Marketing
Collaborative Process Model

**Enterprise Role.** A major area of functional responsibility within the discipline of financial management.

**Work Role.** A role responsible for a specific functional area within an enterprise role, such as might be assigned to a single worker or supported by an IT system.

**Activity.** A specification of a business function in carried out the context of a work role.

**Subactivity.** A specification of a subfunction within necessary to carry out an activity.

**Protocol.** A defined conversation between two roles that may be extended over time. One role initiates and the other responds to the protocol, but information may flow both ways across the protocol.

**Information Flow.** An individual flow of information across a protocol or into or out of an activity.
Receivables Management Example

Related to Customer Orders

Related to Receivables
Information Model Example

A term in the vocabulary represents a **class** of things to be described.

Attributes specify descriptive information having simple types.

A class may be **specialized** into sub-classifications.

Entities may be described as having a unique **identity**.

A relation between terms is described by an **association** between classes.

This means “zero or more”.

This indicates a compositional (as opposed to referential) association.

This is a constraint that defines the sub-classification.

An un-shaded class is further detailed on a different diagram.

This means “one or more”.

This indicates “one or more”.
Business (CIM) view - Collaborating Roles with Processes
People, organizations And/or enterprise components play roles in Business Processes.
People, Organizations
And systems play roles

Components frequently help people play these roles

People, organizations and systems components work together to realize roles

Components are the peoples Automated assistant

Enterprise components help people and organizations play roles by automating and monitoring The business process

From the system perspective. People and organizations become part of the implementation Of the role

The “Enterprise Digital Assistant”
People, Components & Organizations Collaborating
“Lower” PIM View - Enterprise Component Internals

Enterprise Component

UI Client Tier

UI Server Tier

Business Logic

Enterprise Component

Adaptation

[Web] Service

DBMS

Data Managers

Containers

Browser

UI Framework

Legacy Systems

Enterprise Component

Enterprise [Web] Service
Each Work Component in the PIM implements a Work Role from the CIM.

Service Managers implement as system services the business services defined in the CIM.
Information Model

Note; Not expecting anyone
To really read this
Note; Not expecting anyone To really read this
Persistence Model

Association indicates a reference to an entity persisted elsewhere.
Enterprise Service Bus to Enable Target State

- Services driven from the business model
- Reusable Enterprise Services are independent & easily adapted and interconnected
  - Services communicate with each other like humans do with email
- Information systems become a lattice of cooperating components providing services
- SOA/Enterprise Service Bus using commercial standards
  - Industry best practice to avoid developing large monolithic applications

One-GSA Business Model

- Project Management Service
- Contracting Service
- Solution Provider Service
- Funds Management Service
Provisioning Model
Example of XML provisioned from model

```xml
<CustomerOrderEstablishment>
  <Inter-Work-RoleTransaction>
    <inter-work-roleTransactionID> ... </inter-work-roleTransactionID>
    ...
  </Inter-Work-RoleTransaction>
  <newOrder>
    <orderingCustomer>
      <customerID> ... </customerID>
    </orderingCustomer>
    <controllingSalesInstrument>
      <salesInstrumentId> ... </salesInstrumentId>
    </controllingSalesInstrument>
    <customerOrderAmount> ... </customerOrderAmount>
    ...
    <lineItems>
      ...
    </lineItems>
  </newOrder>
</CustomerOrderEstablishment>
```
Enterprise Service Bus

Logical SOA Tiers and Components

<table>
<thead>
<tr>
<th>Client</th>
<th>Presentation</th>
<th>Business</th>
<th>Intermediary</th>
<th>Resource</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Portlet</td>
<td>EJB, POJO,</td>
<td>JBI</td>
<td>Heritage Systems</td>
</tr>
<tr>
<td></td>
<td>or Web</td>
<td>or Servlet</td>
<td>Container</td>
<td>RDBMS</td>
</tr>
<tr>
<td></td>
<td>Container</td>
<td>Container</td>
<td></td>
<td>Other Apps</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Messaging, Rules, Process, Persistence, Transaction, Routing, Transformation, etc.</td>
<td>Technical Services</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Domain</td>
<td>Domain</td>
<td>Domain</td>
<td>BPM Repository</td>
</tr>
<tr>
<td></td>
<td>Objects</td>
<td>Objects</td>
<td>Objects</td>
<td>Rules Repository</td>
</tr>
</tbody>
</table>

* Complements of jBoss
Many BPEL Processes support the CIM
Net Effect of Enterprise MDA

- Clear path from needs to running technology
- Integrate business driven solutions with capital planning & the FEA
- Interoperable component architecture based on SOA
- Integrate legacy, COTS, GOTS and new development into a coherent solution
- Strategic evolution
- Reduced time, costs & risk
Integration of the Semantic Web with MDA

And now a word from another community!
Problems to be Addressed

• Infrastructure
  – Get ANY architecture or model or specification into a common form
  – Manage and publish these architectures as web resources
  – Relate and reuse across organizational and technological domains

• Semantics
  – Relate the meaning of the models
  – Understand and unify different ways to express architecture (E.G. Activity)
  – Understand and unify the concepts used in the architectures (E.G. Invoice)
Semantic Web Technologies

• RDF – Resource Description Framework
  – Describes (mostly) any kind of information as very atomic “triples” accessible as web resources
  – Has an XML syntax
  – Can be used for data or metadata infrastructure

• OWL – Web Ontology Language
  – A formal language for describing concepts in a domain – based on “description logics”
  – Extends and constrains RDF
  – Provides a vehicle for semantic integration

• You will get much more detail on this in later presentations
MDA Modeling Needs Ontologies

• The semantic web infrastructure provides a great way to
  – Publish models as web resources
  – Query over models
  – Analyze models and the intersection of multiple models
  – “Semantically ground” models

• RDF/Ontology based models are more resilient to change without “refactoring”

• Ontologies are better able to connect models that were not designed together – integrating and adapting architectures, processes, interfaces and information

• The open, distributed and federated “meta object facility” has yet to emerge as readily available and mainstream

• Semantic web infrastructure is picking up industry steam, tools and infrastructure are coming available
Ontologies Need MDA Modeling

• A vast amount of information exists in these environments.
• Use of these paradigms represents mainstream best practice – there are lots of practitioners.
• The structured modeling tools are more mature and suited to specific problems.
• Model Driven Architecture has started to bind structured models with the software development process, providing even more leverage
  – Doesn’t require changing the runtime infrastructure as some Semantic Web Approaches are suggesting – separation of technology concerns
• Ontologies can’t ignore this wealth of knowledge, tools, expertise and industry momentum.
Metadata Technology Layers

Modeling/SW Layer
- Eclipse UI
- UML, Business Models, Schema, Java, ...
- Eclipse-EMF
- EMF Adapter

Ontology Layer
- RDF/OWL Ontology Store
- Metadata DBMS
- Ontology Tools
- RDF Metadata Infrastructure API
- Query
- Inference & Rules

Software
- Eclipse Code Generators
- SW Eng Tools

Web Server
- MOF/RDF Mapping

Copyright © 2006 Data Access Technologies, Inc. Slide 35
Semantic Core

• The semantic core integrates the concepts of architecture as expressed in multiple languages such as UML, OWL, FEA, BPM, EDOC, XML, Requirements, Etc.
• This provides for a unification of the intellectual capital used to specify
  – Organizations
  – Systems
  – Information
  – Interfaces
  – Processes
  – … Anything we architect
• Making the organizations, processes and systems more agile and interoperable
Driver: Interoperability

- Interoperability of information and interfaces is a primary driver today. The cost and agility advantages are established, the issues known. We must enable a solution to these problems.
- Semantic core provides a missing link for enabling interoperability, this is our driving requirement.
  - Adaptation of similar information and interfaces across organizations, processes, and systems.
- Semantic Core combined with the capabilities of service oriented and model driven architectures provides a capability for wide scale, net centric interoperability.
The Basics

• Problems to be solved
  – Interoperability of organizations and technology
  – Collaboration
  – Architected business information and processes
  – Agile solutions based on the architectures

• The Approach
  – Architecture models grounded in an open and extensible semantic framework
  – Model Driven Architecture to generate technology components
  – Service Oriented Architecture as the infrastructure
“Meta” Integration Problem

Too many ways to talk about the same thing, redundant and conflicting semantics.
How this Effects the Enterprise

• Contractors, using different and incompatible tools
• Generate different architectures about the same things
• That then need to be aligned – but are never maintained
• Each project becomes an island, without reuse or interoperability
• The resultant complexity is expensive, and anti-agile
Adapting Systems with OsEra

OsEra

Semantic Core

Commerce Ontology

Provision

Adapter

Purchasing System
J2EE

Invoicing System
Microsoft .NET

UML

XML

Described in

Described in

Described in

Described in

Copyright © 2006
Data Access Technologies, Inc.
Slide 41
Integration Via Semantic Hubs

Ways To Specify Things
- UML
- XML

Dis-Integrated Information

Common Concepts
- Joint Behavior
- Monetary Trade

Semantic Hub
- Semantic Core
- Reference Ontologies

Concepts We Specify
- Buy
- Purchase

Mapping is not “one-one”
Library of common concepts Can grow over time
“Views” of Integrated Information

OsEra

Semantic Core

Provision

Web Services
Documentation
Workflow
Components
Java/J2EE
BPEL

UML
DoDAF
BPMN
EDOC
OWL
E/R Models

Ontologies
FEA
Requirements
XML
SQL
Security

Copyright © 2006
Data Access Technologies, Inc.
Slide 43
Semantic Components

- Library of component concepts
- Growing based on need
- Modular, not monolithic
- A construction set for languages
- A lattice of interoperable concepts
Interoperability of Systems

System A

System B

System of Systems

Shared Context

Hub Context

System C

System D
Importing Information

External Artifacts

E.G. Business Process
UML
DoDAF

Provision

Imported Source Model

Semantic Core
Meta Ontology

OsEra

Provision

Unified Model
Getting Value Out

- **OsEra**
  - High Level (Business) Model
  - Derivative Systems Model
  - Technology Model

- External Artifacts
  - Provision to E.G. SOA or DoDAF
  - E.G. Business Process
  - Semantic Core Meta Ontology
  - E.G. Web Services
Unifying Intellectual Capital

- Business Process (BPMN)
- System Architecture (UML)
- XML Vocabulary

OsEra
- Process Model: Order
- Component Model: PO
- Document Model: OR_ST_05
- Human/Automated Integration

Reference Ontology: Order Concept

Unified Architecture: Order

Copyright © 2006 Data Access Technologies, Inc.
Slide 48
Ontological Grounding

• Grounding our common concepts in Ontologies has multiple advantages
  – We can add “axioms” that help to more concretely define the concepts
  – Ontology tools can use this information to bridge like terms for the same concept or similar concepts
  – Other ontology aware components can assist architects in “grounding” their models
  – Adaptation components can help build “adapters” between different interfaces and information stores
  – Inconsistencies can be identified and resolve early
  – As ontologies advance, additional capabilities can be added
  – We can connect Ontological “hubs” – not requiring “one true solution”
Example Workflow

1. Import new or Legacy Spec and instances

2. For each term, Relate to reference Ontologies
   - Yes
   - Term Exists
   - Concept Exists
     - Yes
     - Make new term for existing Concept
     - no
     - Make New Concept Relate to existing Concepts
   - no
   - Grounding Complete

3. Use tools to compare/adapt grounded specs

4. Validate with Instances
   - Correct?
     - Yes
     - Generate Adapter Implementations
     - no
     - Extend/Correct Adaptation

5. Correct?
Some Semantic Core Concepts

• Concepts, Symbols & Types
• Roles and relations define logical connections between concepts
• Context
• Assertions about things and types of things
• Process & activities
• Purpose and objectives
• Constraints & conditions
Joining the “Stacks”

- Modeling & Architecture “Stack”
  - UML
  - Model Driven Architecture
  - Meta Object Facility
  - Business Process
- Semantic Web & Ontology “Stack”
  - RDF
  - OWL
- Current Project – MOF to RDF
  - Makes ANY MOF compliant model (UML, EDOC, E/R, Etc) an OWL Ontology
  - Provides foundation for grounding models
  - Reduces the gaps between the camps
  - Allows models to be published as semantic web ontology resources