Agile Development Meets Ontology as a Service: A Cyber-Security Example

Semantic Technology Conference
June 6-10, 2011

Mary Parmeelee
Artificial Intelligence Engineer
The MITRE Corporation
mparmelee@mitre.org
Introduction

• As OWL transitions from academia and research to widespread commercial application, there is a need for systematic methods that provide a clear path for large-scale implementation and deployment while quickly demonstrating added value to the customer and the end user.

• We introduce a methodology for the incremental implementation and deployment of OWL ontologies as services:
  – Applies service oriented computing and Agile Development process techniques
  – Enables simultaneous rapid deployment of new capability while incrementally building an inventory of loosely-coupled reusable ontologies.
  – As the ontology inventory grows, it gradually reduces development costs of future applications while simultaneously improving cross-application semantic interoperability.

• We demonstrate the methodology’s value by stepping through a real world cyber-security example from concept to application deployment, including a live demonstration of the resulting semantic application.
Assumptions

• Audience has a basic understanding of
  – Semantic Web Technology (e.g. OWL, RDF)
  – Ontology
  – Service Oriented Architecture
Agenda

• Background
• Conceptual Overview
• Related Work
• Case Study
• Application Demonstration
• Future work
Background: Agile Development

• From Scrum Alliance.org: [http://www.scrumalliance.org/pages/what_is_scrum](http://www.scrumalliance.org/pages/what_is_scrum)

“Scrum is an agile framework for completing complex projects. Scrum originally was formalized for software development projects, but works well for any complex, innovative scope of work.”

• Why Scrum for the OaaS Methodology?
  – Demonstrates value early and often (critical for lesser known technologies and approaches)
  – Designed for change (e.g. requirements, scope, resources, management)
  – Empowers the developer, the user and the customer
  – Minimizes rework: does more with less
Background: Service Oriented Computing

• From the SOA Glossary: http://www.soaglossary.com/service_oriented_computing.php

*Service-oriented computing* is an umbrella term used to represent a distributed computing platform. The major components of SOC are:

– A service-oriented architecture
– Service-orientation
– Service-oriented solution logic
– Services
– Service compositions
– Service inventory

• Why service-oriented computing for the OaaS Methodology?

– It enables ontologies to be integrated in to a widely adopted, well accepted and well understood technology
– SOC is intrinsically designed for distribution, sharing, reuse and composition of modular “things” called services.
– An existing large pool of production-ready tools and skilled technologists reduce the learning curve and lower the up-front costs of integrating ontologies into enterprises scale systems
OaaS & Related Work

- OaaS benefits overlap with ontology architecture. An ontology architecture is a federation of modular ontologies that form the structural and semantic framework of an information domain.
  - Loose coupling and modularization makes it easier to add, remove and maintain individual ontologies;
  - Modular ontologies are easier to reuse and process than large monolithic ontologies;
  - Component ontologies can be dynamically combined on demand at implementation time to meet application-specific requirements.

- Semantic Web Services (e.g. WSMO, WSMX, SAWSDL, OWL-S)
  - Are complementary to OaaS
  - Focus is on the implementation of services (e.g. discovery, composition, monitoring)
OaaS Conceptual Overview
Cyber-Security Example

• MITRE Corporation: leading development of a family of open security automation standards, many of which have been adopted by NIST’s Security Content Automation Protocol (SCAP) program
• Covers most of the cyber-security information and event management (CSIEM) lifecycle
• Standards Objective: provide a baseline of common content and best practices that support interoperability across disparate security-related tools and facilitate automation across security related processes
• NIST sponsors an associated SCAP tool validation program
• SCAP currently being adopted by IETF
• Federal government moving toward adoption of validated products to ensure baseline tool interoperability
Many of the standards require a controlled vocabulary (CV) both at the content level and the representation level. Some of the major standards are:

- CCE: Common Configuration Enumeration
- CEE: Common Event Expression
- CPE: Common Platform Enumeration
- CRE: Common Remediation Enumeration
- CVE: Common Vulnerability Enumeration
- CWE: Common Weakness Enumeration
- MAEC: Malware Attribute Enumeration and Characterization
- OVAL: Open Vulnerability and Assessment Language
- XCCDF: Extensible Configuration Checklist Description Format

Current State of Standards

• Developed independently
• Are at various stages of maturity (0 – 10 yrs)
• Are growing rapidly in size, number and complexity
• Vocabulary development, management and mapping processes are largely manual
• Vocabulary representations are mostly encoded in XML Schema, some XML instance documents, and Excel spreadsheets
• Funded by various government organizations with competing requirements
• In the early stages of addressing cross-standard interoperability, vocabulary reuse and alignment among standards
Agile Development Approach

Rapid Iterative Design and Development for Each Capability

Envisioning Phase
- **Requirements**
  - Rapidly collect user needs
  - Determine criteria and priorities
- **Architecture**
  - Coarse grained modeling
  - Rough estimate of scope

Sprint N: JIT Modeling, Development and Review

Sprint 2: JIT Modeling, Development and Review

Sprint 1: JIT Modeling, Development and Review
- Small incremental development cycles
- User-driven requirements
- Envisioning phase evolves with each Sprint
- Develop working capability with *every* Sprint
Collect User Needs/Set Criteria

• Chose the top three barriers to adoption of the family of standards
  1. Inconsistent, static representation of common concepts and unnecessary variation of common processes across existing standards impedes cross-standard interoperability (e.g. meaning of the term “platform”)
  2. Manual vocabulary management and implementation processes are too labor intensive and error prone.
  3. Rapid growth in size and complexity of the cyber-security domain exceeds the representation capability of current technologies and outpaces the production capability of current methods

• Set prioritized criteria for each barrier
  1. Make standards semantically interoperable
  2. Integrate and semi-automate vocabulary management processes
  3. Improve ability to represent complex semantics and semi-automate content production methods
Chose an OWL-Based Approach

- Defines structure and real world semantics in a machine interpretable way
- Determining meaning is semi-automatable and less time consuming
- Concepts and relations are defined in a machine interpretable specification rather than only human interpretable documentation
- Supports many-to-many term:meaning relations
  - Eliminates terminology wars
  - Provides common models that express the meaning of real world relationships instead of artificially constraining it
  - Complements XML-only technologies because its exchange syntax is grounded in XML. It can be embedded in XML-only systems to provide additional capability.
Why Not XML Schema?

• Schema provides only structural interoperability:
  – Schema defines structure, not meaning
  – Determining meaning in a schema is time consuming, manual, and prone to human error
    • All terms and relevant relationships must be clearly defined by a human in documentation
    • Documentation must be readily available to the user
    • The user manually reads and interprets the documentation

• Common Schemas are controlled vocabularies that try to artificially constrain semantics by:
  – Imposing a common structure and syntax
  – Forcing a one-to-one term:meaning relationship
  – This artificial constraint sparks terminology wars because real world term:meaning relationships are many-to-many
  – We need common models that express the meaning of real world relationships instead of artificially constraining it. OWL and Ontologies do this very well.
Semantic and Structural Interoperability

- **Entities of Interest**
- **Controlled Vocabulary Terms**
- **Ontology Concepts**
- **Fragmented Information**

**Model**
- Name & describe
- Scopes

**Define**
- Semantic Interoperability
- Structural Interoperability
Chose TopBraid Suite Tool

• Provides and end-to-end development to deployment solution
• Designed for Semantic Web Development
• Semantic Web standards compliant
• Easy-to-use rapid prototyping environment
• Thin Client
• REST web-service
Coarse Grained Model/Estimate Scope
## Current CCE Management

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>The required auditing for %SystemDrive% directory should be enabled.</td>
<td>(1) set of accounts</td>
<td>(1) defined by the object’s SA CL</td>
<td>CCE.25</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>The required auditing for the registry key HKEY_LOCAL_MACHINE\SOFTWARE should be enabled.</td>
<td>(1) set of accounts</td>
<td>(1) defined by the object’s SA CL</td>
<td>CCE.899</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>The required auditing for the registry key HKEY_LOCAL_MACHINE\SYSTEM should be enabled.</td>
<td>(1) set of accounts</td>
<td>(1) defined by the object’s SA CL</td>
<td>CCE.727</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>The required permissions for the directory %ALL% should be assigned</td>
<td>(1) set of accounts</td>
<td>(1) defined by the object’s DAC L</td>
<td>CCE.211</td>
<td>File Auditing - Must Have ACE</td>
<td>(OD 269)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>The required permissions for the directory %AllUsersProfile% should be assigned</td>
<td>(1) set of accounts</td>
<td>(1) defined by the object’s DAC L</td>
<td>CCE.39</td>
<td></td>
<td></td>
<td></td>
<td>%AllUsersProfile%</td>
</tr>
<tr>
<td>9</td>
<td>The required permissions for the directory %AllUsersProfile%\Application Data should be assigned</td>
<td>(1) set of accounts</td>
<td>(1) defined by the object’s DAC L</td>
<td>CCE.83</td>
<td></td>
<td></td>
<td></td>
<td>%AllUsersProfile%\Application Data</td>
</tr>
</tbody>
</table>
## CCE Content Distribution

### Downloads (MS Excel format)

<table>
<thead>
<tr>
<th>CCE v5 - Platform Groups-COMBINED FILE (2 MB)</th>
<th>DATE UPDATED</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCE v5 - AIX 5.3 (111 KB)</td>
<td>May 6, 2009</td>
</tr>
<tr>
<td>CCE v5 - HP-UX 11.23 (81 KB)</td>
<td>May 6, 2009</td>
</tr>
<tr>
<td>CCE v5 - Internet Explorer 7 (143 KB)</td>
<td>April 28, 2010</td>
</tr>
<tr>
<td>CCE v5 - Microsoft Office 2007 (375 KB)</td>
<td>April 28, 2010</td>
</tr>
<tr>
<td>CCE v5 - Red Hat Enterprise Linux 4 (94 KB)</td>
<td>May 6, 2009</td>
</tr>
<tr>
<td>CCE v5 - Red Hat Enterprise Linux 5 (113 KB)</td>
<td>April 28, 2010</td>
</tr>
<tr>
<td>CCE v5 - Sun Solaris 8 (106 KB)</td>
<td>May 6, 2009</td>
</tr>
<tr>
<td>CCE v5 - Sun Solaris 9 (109 KB)</td>
<td>May 6, 2009</td>
</tr>
<tr>
<td>CCE v5 - Sun Solaris 10 (89 KB)</td>
<td>April 28, 2010</td>
</tr>
<tr>
<td>CCE v5 - Windows Vista (241 KB)</td>
<td>April 28, 2010</td>
</tr>
<tr>
<td>CCE v5 - Windows 2000 (247 KB)</td>
<td>April 28, 2010</td>
</tr>
<tr>
<td>CCE v5 - Windows Server 2003 (329 KB)</td>
<td>April 28, 2010</td>
</tr>
<tr>
<td>CCE v5 - Windows Server 2008 (248 KB)</td>
<td>April 28, 2010</td>
</tr>
</tbody>
</table>

Also available in XML instance document (no schema)

[http://cce.mitre.org/lists/cce_list.html](http://cce.mitre.org/lists/cce_list.html)
Approximately 3 person months of total effort

The Agile Development process steps included

1. Collected CCE-specific user needs and acceptance criteria for a standards-management application
2. Rough model and scope
3. Developed 4 reusable cross-standard ontologies that were necessary for prototype implementation
   A. Discovered and developed 1 new reusable ontology as a result of Sprint 1 analysis. (evolved the envisioning phase)
   B. Developed a composite implementation-profile ontology to represent CCE specific vocabulary management concepts
4. Implemented the composite ontology in a Standards Management prototype
CCE Analyst Needs

1. A flexible, extensible and intuitive CCE vocabulary management environment
2. Track complex relations between CCE Entries, CCE submissions, references, resources, parameters, technical mechanisms and platforms
3. Support the CCE analysis process, from CCE submission to CCE publication
Ontology Development

- Common Ontology
  - Point of Contact
  - Dublin Core
  - Resource Manager
  - Geospatial
  - Content Curation

- Domain Ontology
  - Configuration
  - Software
  - Vulnerability

- Cont. Vocabulary
  - CCE XML
  - CPE XML
  - CVE XML
  - OVAL XML

- Align and Define
- Extract and Reuse
CCE Implementation Profile Ontology

Merges Common Configuration Enumeration (CCE)
and Content Curation

Combined with parts of: Point of Contact, Resource Manager, Dublin Core
## Common Configuration Concepts

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CCE-2682-3</td>
<td>The required auditing for %SystemDrive% directory should be enabled.</td>
<td>(1) set of accounts</td>
<td>CCE-25</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(2) events to audit</td>
<td>(2) applicability</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(3) defined by the object's SACL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CCE-2796-1</td>
<td>The required auditing for the registry key HKEY_LOCAL_MACHINE\SOFTWARE should be enabled.</td>
<td>(1) set of accounts</td>
<td>CCE-899</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(2) events to audit</td>
<td>(2) applicability</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(3) defined by the object's SACL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CCE-1840-8</td>
<td>The required auditing for the registry key HKEY_LOCAL_MACHINE\SYSTEM should be enabled.</td>
<td>(1) set of accounts</td>
<td>CCE-727</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(2) events to audit</td>
<td>(2) applicability</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(3) defined by the object's SACL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CCE-2463-6</td>
<td>The required permissions for the directory %ALL% should be assigned.</td>
<td>(1) set of accounts</td>
<td>CCE-211</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(2) list of permissions</td>
<td>(2) applicability</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(3) defined by the object's DACL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CCE-1845-8</td>
<td>The required permissions for the directory %AllUsersProfile% should be assigned</td>
<td>(1) set of accounts</td>
<td>CCE-39</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(2) list of permissions</td>
<td>(2) applicability</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(3) defined by the object's DACL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CCE-2620-3</td>
<td>The required permissions for the directory %AllUsersProfile%Application Data should be assigned</td>
<td>(1) set of accounts</td>
<td>CCE-83</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(2) list of permissions</td>
<td>(2) applicability</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(3) defined by the object's DACL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
CCE Ontology
PoC Component
## Common Configuration Concepts

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>The required auditing for %SystemDrive% directory should be enabled.</td>
<td>(1) set of accounts</td>
<td>(2) events to audit</td>
<td>CCE-25</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>The required auditing for the registry key HKEY_LOCAL_MACHINE \SOFTWARE should be enabled</td>
<td>(1) set of accounts</td>
<td>(2) events to audit</td>
<td>CCE-899</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>The required auditing for the registry key HKEY_LOCAL_MACHINE \SYSTEM should be enabled</td>
<td>(1) set of accounts</td>
<td>(2) events to audit</td>
<td>CCE-727</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>The required permissions for the directory %ALL% should be assigned.</td>
<td>(1) set of accounts</td>
<td>(2) list of permissions</td>
<td>CCE-211</td>
<td></td>
<td>File Auditing - Must Have ACE (OID 269)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>The required permissions for the directory %AllUsersProfile% should be assigned</td>
<td>(1) set of accounts</td>
<td>(2) list of permissions</td>
<td>CCE-39</td>
<td></td>
<td>%AllUsersProfile%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>The required permissions for the directory %AllUsersProfile%\Application Data should be assigned</td>
<td>(1) set of accounts</td>
<td>(2) list of permissions</td>
<td>CCE-83</td>
<td></td>
<td>%AllUsersProfile%\Application Data</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
DCE and RM Components
We demonstrated 4 of the 5 target capabilities with 3 person months of effort:

1. Improved capability for representing complex domain semantics (M:N) relations and multiple inheritance.
2. Vocabulary management capabilities that facilitate and semi-automate workflow processes
3. Ability to express vocabulary in a common representation language that is adaptive to rapid change with minimal reengineering
4. Improved ability to more consistently represent common concepts to facilitate cross-standard interoperability
Standards Management Tool Demo

Developed with TopBraid Suite
OaaS Summary
Future Work

1. Future Sprints
   A. Continue to demonstrate the reuse value of the ontology inventory by developing implementation profile ontologies for new ontology-based CSIEM applications
   B. Expand Standards Management application to support other standards and tasks
   C. Develop a new reference implementation that semi-automates some subset of the CSIEM process. (e.g. semi-automatic mapping of proprietary tool results to standard vocabularies)

2. Build Ontology Inventory with each Sprint
   A. Revisit the envisioning phase with each Sprint to refine the architecture and the underlying CSIEM process
   B. Apply lessons learned to develop best practices and common tools for standards development, management, and mapping
   C. Apply the OaaS methodology to other domains
Questions