OntoMaven

Maven-based Ontology Repositories

Adrian Paschke

Corporate Semantic Web (AG-CSW)
Institut für Informatik, Freie Universität Berlin
paschke@inf.fu-berlin
http://www.inf.fu-berlin/groups/ag-csw/

9th International Workshop on Semantic Web Enables Software Engineering
(SWSE 2013) @ ICSOC 2013
2. December 2013 | Berlin
Ontology Project Life Cycle Management

- Avoid manual steps
  - Reuse/import, rename, refactor, merge, test, deploy, …
  - Pass documentation and configuration information
- Work collaboratively but support local development
  - Remote and local repositories (even without Internet connection)
- Automatically process resources
  - Manage dependencies and development + deployment properties
- Share ontology resources across projects
  - Ontology artifacts including metadata, documentation, tests, …
OntoMaven – Apache Maven Extension

- **Ontology Artifacts**
  - Managed by `groupId`, `artifactId`, and `version`

- **Ontology Repositories**
  - **Local** and **remote** repositories used to store artifacts (ontologies, plug-ins, resources, test suites etc.)

- **OntoMaven Plug-ins**
  - Implement the ontology project life cycle management functionalities
  - **Goals** provide interfaces bundled to life cycle **phases**

- **Project Ontology Model**
  - POM.xml – Declarative project description and configuration
<project>
  <modelVersion>...</modelVersion>
  <groupId>...</groupId>
  <artifactId>...</artifactId>
  <version>...</version>
  ...
</project>
OntoMaven – Selected Plug-Ins

1. Import Dependency Management
2. Documentation
3. Versioning
4. Testing with Test Suites
5. …
Import Ontology

Parse Ontology

exist?

yes

Import and Dependency Management Plug-In

no

Return Result List (List of imported and not imported Ontologies)

Create XMLCatalog

Save Ontology in Local Repository

Return Result List (List of local ontologies and XMLCatalog with local references)

save?

yes

no

end

Return Result List (List of imported and not imported Ontologies)
Ontology Artifacts

Archiva Repository Manager

Dependencies
Ontology Artifacts in Remote Repository

<profiles> <profile>
  <id>2</id>
  <activation> <activeByDefault>true</activeByDefault> </activation>
  <repositories> <repository>
    <snapshots> <enabled>true</enabled> </snapshots>
    <id>snapshots</id>
    <name>OntoMaven Snapshot Repository</name>
    <url>http://www.corporate-semantic-web.de/repository/snapshots/</url>
    ... 
  </repository>
</repositories>
</profile> 
<dependencies> <dependency>
  <groupId>xfront.com.owl.ontologies</groupId>
  <artifactId>Camera-OWL-Ontology</artifactId>
  <version>1.0-SNAPSHOT</version> <type>owl</type>
</dependency> </dependencies>
Report Plug-In – Documentation

POM.xml

<description>here's the description of an ontology </description>
<organization>
  <name>Corporate Semantic Web, Freie Universitaet Berlin</name>
  <url>http://www.corporate-semantic-web.de</url>
</organization>
<inceptionYear>2013</inceptionYear>
/licenses> <license>
  <name>GPL-3.0</name>
  <url>http://www.gnu.org/licenses/gpl.txt</url>
</license></licenses>
<developers> <developer>
  <name>Adrian Paschke</name>
  <email>paschke@inf.fu-berlin.d</email>
  <organization>Corporate Semantic Web</organization>
  <organizationUrl>http://www.corporate-semantic-web.de/</organizationUrl>
</developer>
<roles> <role>developer</role> </roles> </developers>
The Documention of Anonymous-2

Description: Camera OWL Ontology

Author: Roger L. Costello

Acknowledgements: Many thanks to the following people for their invaluable input: Richard McCullough, Yuzhong Qu, Leo Sauermann, Brian McBride and Jim Farrugia. Modified as a Jena example by Ian Dickinson.

Ontology Format is: RDF/XML

Profile: The Ontology is in OWL 2 DL

<table>
<thead>
<tr>
<th>Imports</th>
<th>number</th>
</tr>
</thead>
<tbody>
<tr>
<td>found no import</td>
<td></td>
</tr>
</tbody>
</table>

| Total Number of Classes          | 13     |
| Total Number of Datatype Properties | 8      |
| Total Number of Object Properties | 7      |
| Total Number of Annotations Properties | 1      |
| Total Number of Individuals      | 2      |
Technical Report

A-Z of

This is a complete alphabetical A-Z index of all terms, by class (categories or types) and by property.

Classes:
- PurchaseableItem
- Range
- Body
- Viewer
- Money
- Camera
- Large-Format
- Lens
- Window
- BodyWithNonAdjustableShutterSpeed
- Digital
- SLR

Properties:
- body
- shutter-speed
- viewFinder
- part
- lens
- cost
- compatibleWith
- min
- focal-length
- max
- f-stop
- aperture
- units
- currency
- size

More Details

PurchaseableItem

- Has Subclass: Body | Camera | Lens
- DataTypeProperties:
  - ObjectProperties: cost
  - COMMENT: "Example for an annotated owl class"

Range

- Has Subclass:
- DataTypeProperties: min | max | units
Visualization Report
# Versioning Plug-In – SVont

<table>
<thead>
<tr>
<th>Plug-In SVN Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>checkout</td>
<td>Check out working copy from project achieve</td>
</tr>
<tr>
<td>status</td>
<td>Status of the working copy</td>
</tr>
<tr>
<td>diff</td>
<td>Diff between two revisions or paths</td>
</tr>
<tr>
<td>commit</td>
<td>Commit working copy to repository</td>
</tr>
<tr>
<td>info</td>
<td>Info about repository and working copy</td>
</tr>
</tbody>
</table>

**Illustration:**

- **Iteration 1:** Version 1
- **Iteration 2:**
  - Version 1
  - Version 2
- **...**
- **Iteration n-1:**
  - Version 1
  - Version 2.x
  - Version 3
- **Iteration n:**
  - Version 4
  - Version 3.1
OntoMaven – Plug-In Goal „diff“

------------------------ DIFF INFORMATION ------------------------
Ontology File: ...\...\...\...\...\camera.owl

================== ACTUAL CHANGES ==============================
Axioms were added to the repository, or deleted from the working copy.
SubClassOf(<http://www.xfront.com/owl/ontologies/camera/#Money> owl:Thing)
Declaration(Class(<http://www.xfront.com/owl/ontologies/camera/#Money>))

================================

--------- MORE INFO ---------------
The above changes of the OWL classes are dependent on the following axiom.
currency <------ DataProperty (Domain)
cost <------ ObjectProperty (Range)

================================
SVont – Subversion for Ontologies

Classical SVN Clients

Ext. SVN Interface

Command Handler

ext. SVN commands

Metadata Rep.

Change Log

Change Detector

SVOnT Server

classical SVN commands

Class. SVN Interface

Ext. Precommit

SVN Workflow

Class. SVN Server

SVOnT Client

SVN Commands

Commit

Revision Rollback

Ontology Diff

Revision Calculator

Working Copy

SVOnT Server

Change Detector

Ontology Diff

Consist. Check

SVOnT Client

SVN Commands

Commit

Revision Rollback

Ontology Diff

Revision Calculator

Working Copy
SVOnt - Commit Workflow

SVOnt – Subversion for Ontologies
Test Plug-In – Test Cases for KBs

- Test Cases constrain the possible models and **approximate the intended models** of the knowledge base
  - **Queries** are used to test the rule base

- A test case is defined by $T := \{X, A, N\}$, where
  - $X \subseteq L$ assertion base (input data, e.g. facts)
  - $A \in L$ a formula denoting a test query
  - $N := +, -$ a positive or negative label

- Semantics

  \[
  M_0 \models_{TC} (X, A, +) \iff \forall m \in M_0 : m \in \sum (\text{Mod}(X), \text{R}) \Rightarrow m \in \text{Mod}(A)
  \]

  \[
  M_0 \models_{TC} (X, A, -) \iff \exists m \in M_0 : m \in \sum (\text{Mod}(X), \text{R}) \Rightarrow m \notin \text{Mod}(A)
  \]

- $\models_{TC}$ compatibility relation
- Mod association function between sets of formulas and sets of models
- $\Sigma$ model selection function

$A \notin C_R(X)$ for $T:=\{X, A, +\}$ and $A \in C_R(X)$ for $T:=\{X, A, -\}$
- $C_R(X)$ deductive closure of $X$. Decidable inference operator based on formal proofs
POM.xml Test Plug-In Configuration

<plugin>
  <groupId>de.csw.MvnOnt</groupId>
  <artifactId>MvnOwlTest</artifactId>
  <version>1.0-SNAPSHOT</version>
  <configuration>
    <owlfile>owl/1a.owl</owlfile>
  </configuration>
  <executions>
    <execution>
      <goals>
        <goal>owltest</goal>
      </goals>
    </execution>
  </executions>
</plugin>

<plugin>
  <groupId>de.csw.MvnOnt</groupId>
  <artifactId>MvnOwlEntailment</artifactId>
  <version>1.0-SNAPSHOT</version>
  <configuration>
    <premise_file>owl/1a.owl</premise_file>
    <conclusion_file>owl/1aconclusion.owl</conclusion_file>
  </configuration>
  <executions>
    <execution>
      <goals>
        <goal>owlentailment</goal>
      </goals>
    </execution>
  </executions>
</plugin>
Summary

- OntoMaven is not a full Ontology Editor
  - It automates and supports the life cycle management of ontology projects and ontology artifacts

- Declarative description in POM defining project conventions and APIs
  - Additional resource directories
  - Plug-in configuration
  - Artifact libraries
  - Repository and deployment configuration
  - ...

- Three types of repositories
  - Plain file system folder
  - Folder served by HTTP daemon
  - “Full” repository (with indexing, search, cache, …)
Thank you ...

Questions?

http://www.corporate-semantic-web.de/ontomaven.html
https://github.com/ag-csw/OntoMaven

AG Corporate Semantic Web, FU Berlin
paschke@inf.fu-berlin.de
http://www.inf.fu-berlin/groups/ag-csw/
http://www.corporate-semantic-web.de