



OMV / CTS2 Crosswalk

Outline

- **Common Terminology Services 2 (CTS2) - a brief introduction**
- **CTS2 and OMV – a crosswalk**

**OMV / CTS2
Crosswalk**

CTS2 – A BRIEF INTRODUCTION

Common Terminology Services 2

CTS₂

A standard for a shared semantic model and API for the query, interchange and update of terminological content.

Terminological content: code sets, value sets, lexicons, thesauri, classification systems, ontologies, ...

CTS2 Why?

Terminological Resources (Ontologies, classification systems, code sets, value sets...) are the “semantic backbone” of information exchange

Examples: ICD-9, ICD-10, MEDRA, Gene Ontology, SNOMED-CT, LOINC, UNSPSC, FMA, Agrovoc, Dublin Core, SKOS, RDF, OWL, ISO Language Codes, ISO Country Codes, ...

CTS2 Why?

... thousands of institution / application specific enumerations, code sets and value sets.

- Resources published in different formats...**
- ... using different grammars**
- ... with different update and release cycles...**

CTS2 Why?

Interoperability requires that information source and sink have the same set of shared “meaning”...

... especially as many of these resources become “logic based” (aka. Declarative Programming)

DL Foundations

Syntax

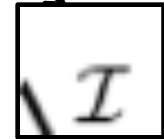
Input
 TOP
 BOTTOM
 NUMBER
 INTEGER
 STRING
 (and $C_1 \dots C_n$)
 (or $C_1 \dots C_n$)
 (not C)
 (all $R C$)
 (some R)

Mathematics

Syntax
 Abstract
 \top
 \perp
 $C_1 \sqcap \dots \sqcap C_n$
 $C_1 \sqcup \dots \sqcup C_n$
 $\neg C$
 $\forall R:C$
 $\exists R$

Interpretation

Extension
 Δ^I
 \emptyset
 the numbers
 the integers
 the strings
 $C_1^I \cap \dots \cap C_n^I$
 $C_1^I \cup \dots \cup C_n^I$
 $\Delta^I \setminus C^I$
 $\{d \in \Delta_a^I \mid R^I(d) \subseteq C^I\}$
 $\{d \in \Delta^I \mid R^I(d) \neq \emptyset\}$



Why It Matters

Labels, Definitions, Examples, Usage Notes, etc. are the entry point and the exit point from formal ontologies...

... but are only of value if we know where to find them.

Finding Definitions Today

RDF:

```
<rdf:Property rdf:about="http://www.w3.org/1999/02/22-rdf-syntax-ns#predicate">
  <rdfs:isDefinedBy rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#" />
  <rdfs:label>predicate</rdfs:label>
  <rdfs:comment>The predicate of the subject RDF statement.</rdfs:comment>
  <rdfs:domain rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Statement" />
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource" />
</rdf:Property>
```

BFO:

```
<owl:Class rdf:about="&snap;FiatObjectPart">
  <rdfs:subClassOf rdf:resource="&snap;MaterialEntity" />
  <owl:disjointWith rdf:resource="&snap;Object" />
  <owl:disjointWith rdf:resource="&snap;ObjectAggregate" />
  <rdfs:label rdf:datatype="&xsd:string">fiat object part</rdfs:label>
  <rdfs:comment rdf:datatype="&xsd:string">Definition: A material entity [s
  <rdfs:comment rdf:datatype="&xsd:string">Examples: upper and lower lobes
  <rdfs:comment rdf:datatype="&xsd:string">Synonyms: fiat substance part</r
</owl:Class>
```

Finding Definitions Today

OCRe:

```
<owl:ObjectProperty rdf:about="&ontologies;OCRe.owl#OCRe900040">  
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>  
  <rdfs:label rdf:datatype="&xsd:string">has anchor time</rdfs:label>  
  <statistics:definition rdf:datatype="&xsd:string">The reference time point</statistics:definition>  
  <rdfs:domain rdf:resource="&ontologies;OCRe.owl#OCRe400012"/>  
  <rdfs:range rdf:resource="&ontologies;OCRe.owl#OCRe400024"/>  
</owl:ObjectProperty>
```

NCIt: Note multiple definitions *and* provenance

```
<property>  
<name>DEFINITION</name>  
<value><![CDATA[<def-source>NCI-GLOSS</def-source><def-definition>Treatment using more th  
</property>  
<property>  
<name>DEFINITION</name>  
<value><![CDATA[<def-source>MSH2002_06_01</def-source><def-definition>Drug therapy with t  
</property>  
<property>  
<name>DEFINITION</name>  
<value><![CDATA[<def-source>CSP2002</def-source><def-definition>combination of drugs in t  
</property>
```

Finding Definitions Today

The list continues – especially when you include non-OWL ontologies (SNOMED-CT for example), and classification systems, thesauri, code/value pairs, etc.

CTS2 Goals

- **Specify a common model of what is common amongst these resources**
- **Include metadata about what the resources are for, who publishes them, how often they are released**
- **Create mechanisms for federation, distribution, incremental update and history**

CTS2

Goals (continued)

- **Provide a bridge between the emerging Semantic Web community (RDF, SKOS, OWL, SPARQL) and structured models of information**

**OMV / CTS2
Crosswalk**

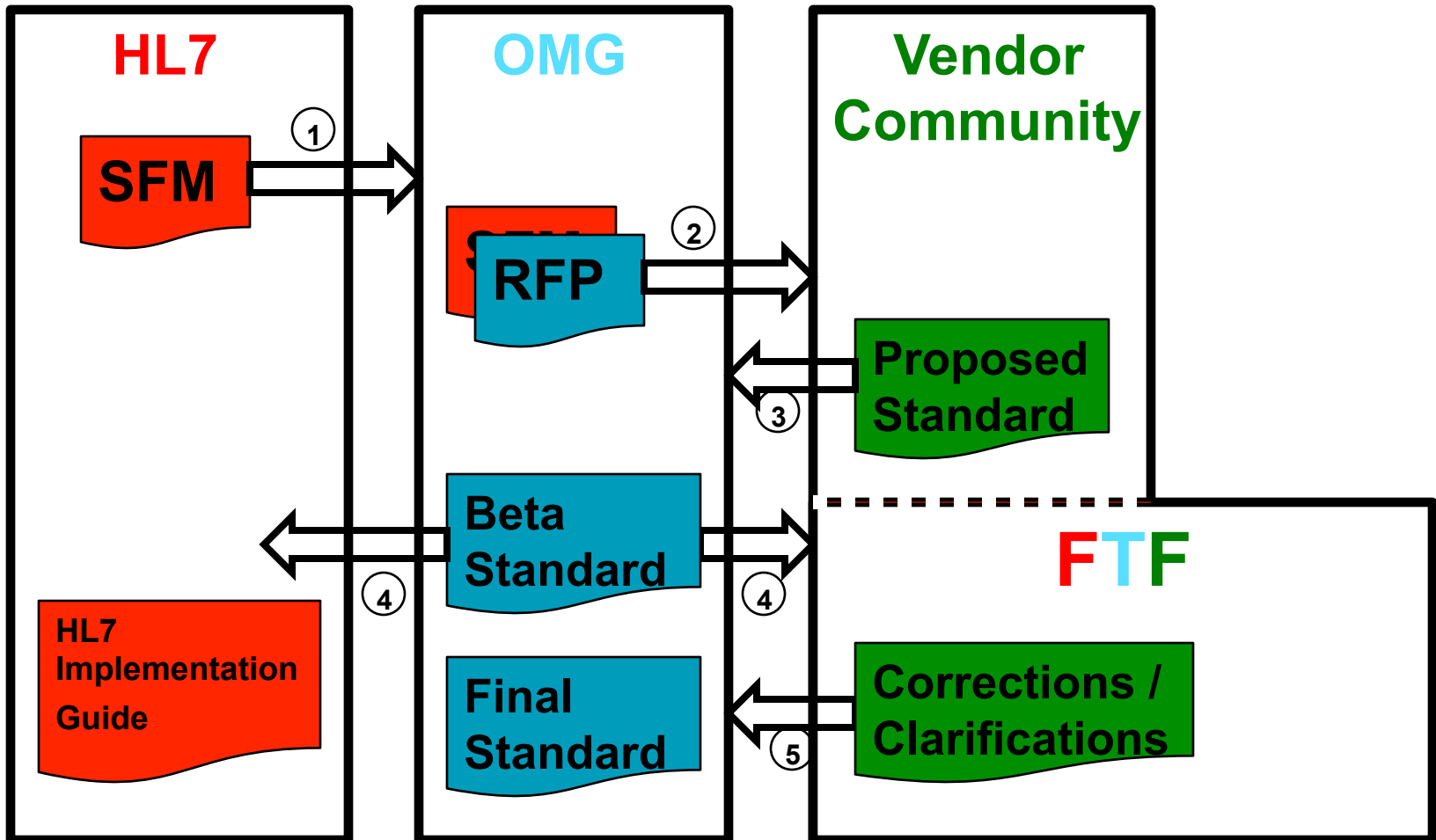
CTS2 PROCESS

CTS₂

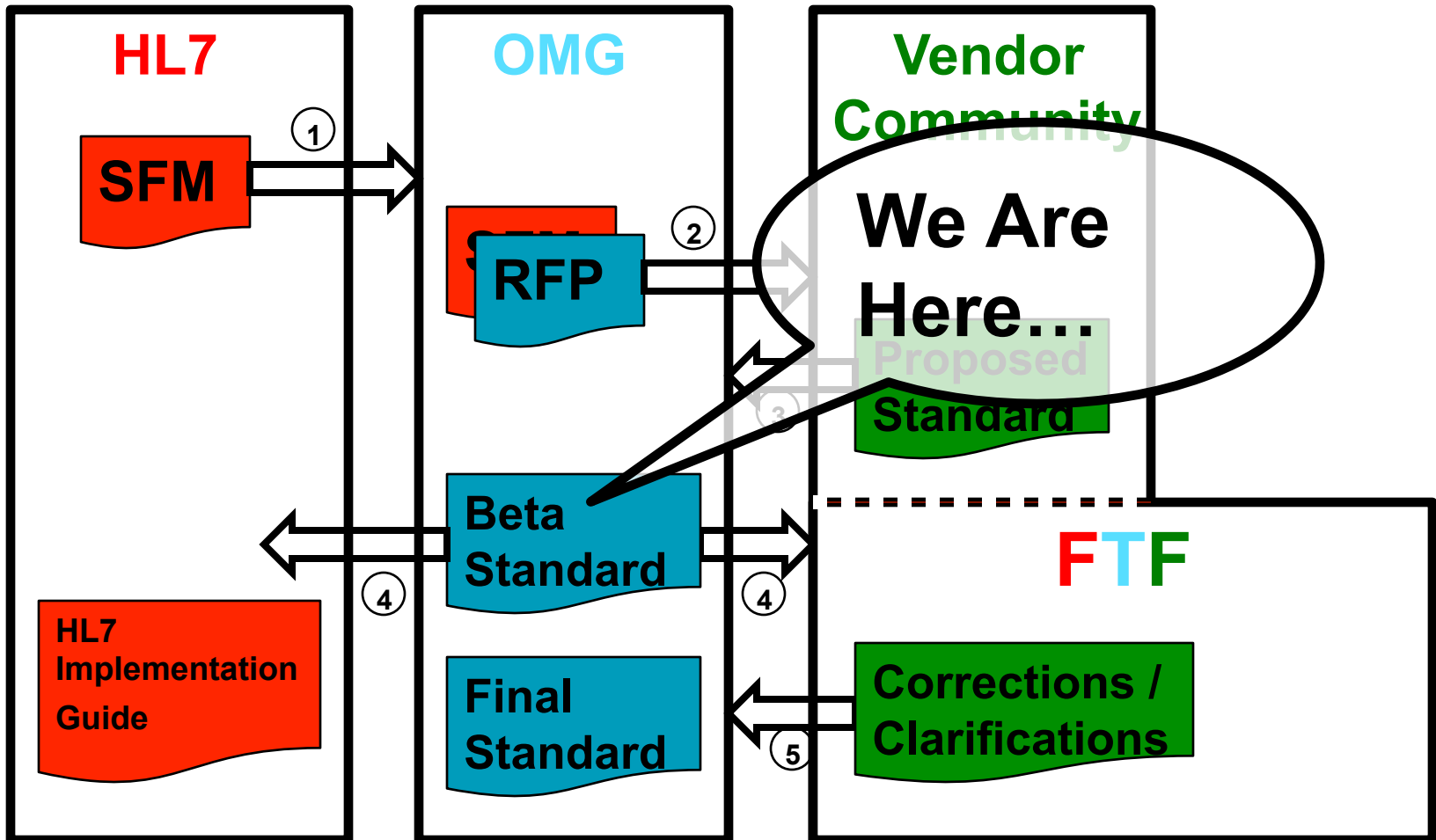
Developed through the Healthcare Services Specification Project (HSSP) - a collaboration between Health Level 7 (HL7) and the Object Management Group

- **HL7 provides the requirements as a Service Functional Model**
- **OMG develops the formal specification**
- **HL7 adopts and validates via an HL7 Implementation Guide**

Healthcare Services Specification Project (HSSP) Workflow



Healthcare Services Specification Project (HSSP) Workflow



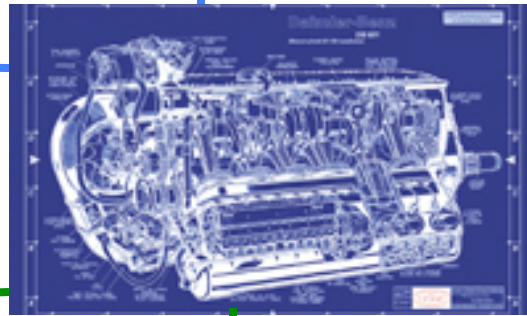
CTS2 Beta Standard

CTS2 is an *Application Programming Interface (API)* specification.

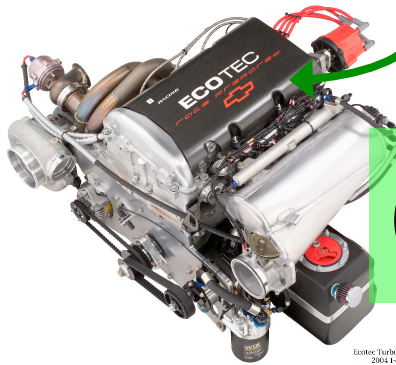
- **It defines the semantics, syntax and valid interactions that can occur**
- **CTS2 is not software - it is a “blueprint” for building and using software**
- **If everyone follows the blueprint (and the blueprint is sufficiently precise) then CTS2 clients and services can interoperate**

CTS2 Standard as a Blueprint

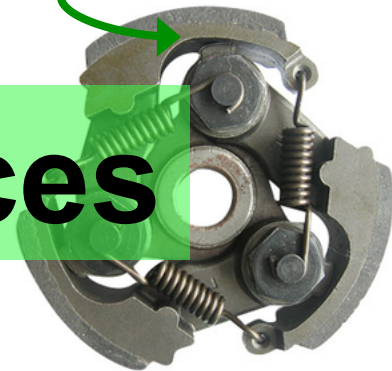
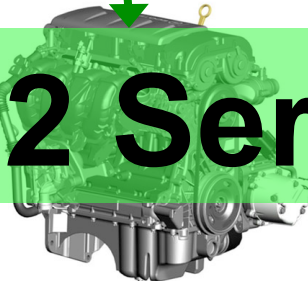
CTS2 Clients



CTS2 Services



Ecotec Turbo Race Engine
2004 4-cylinder
for Chevrolet Cavalier



Key Points

- **Based on Representational State Transfer Architectural Paradigm**
- **Heavily influenced by BioPortal and BioPortal REST API**
- **ORWG and OMV input used for model validation**
- **Modular Implementation – build/use only what you need**
 - **Resources**
 - **Functionality**
 - **Representation**

Key Points (continued)

- **Designed for distribution and federation (!)**
- **Generic – NOT healthcare specific**
- **Supports Semantic Web – RDF and OWL2**
- **Not intended to be constraining**
 - **Extensions are ok – in fact encouraged!**
 - **Purpose is not to say what *can* be done, but rather to say how common things can be done consistently**

**OMV / CTS2
Crosswalk**

CTS2 CONFORMANCE POINTS

CTS2 Conformance Philosophy

- **“Linear Value Proposition”** (as described by Charlie Meade) – easy things are easy and complexity is proportional to gain
- **Implement (or use) exactly what is needed**
 - **Resources**
 - **Functionality**
 - **Representation**

CTS₂ Resource profiles

- **Code System Catalog Entry**
- **Code System Version**
- **Entity Description**
- **Association**
- **Map Catalog Entry**
- **Map Version**
- **Value Set Catalog Entry**
- **Value Set Definition**
- **Concept Domain Catalog**
- **Concept Domain Binding**
- **Statement**

CTS2 Conformance Points Behavioral Perspective

- **Read – direct access**
- **Query – search and discovery**
- **Import/Export – external formats**
- **Update – incremental update**
- **History – change history**
- **Temporal – state of service at point in time**
- **Maintenance – construct incremental updates**

CTS2 Conformance Points Representational Perspective

- **XML**
 - **XML Schema**
 - **ISO 21090***
- **JSON**
- **RDF***
- **POJO***

* Not present in Beta 1.0 Specification

**OMV / CTS2
Crosswalk**

CTS2 / OMV CROSSWALK

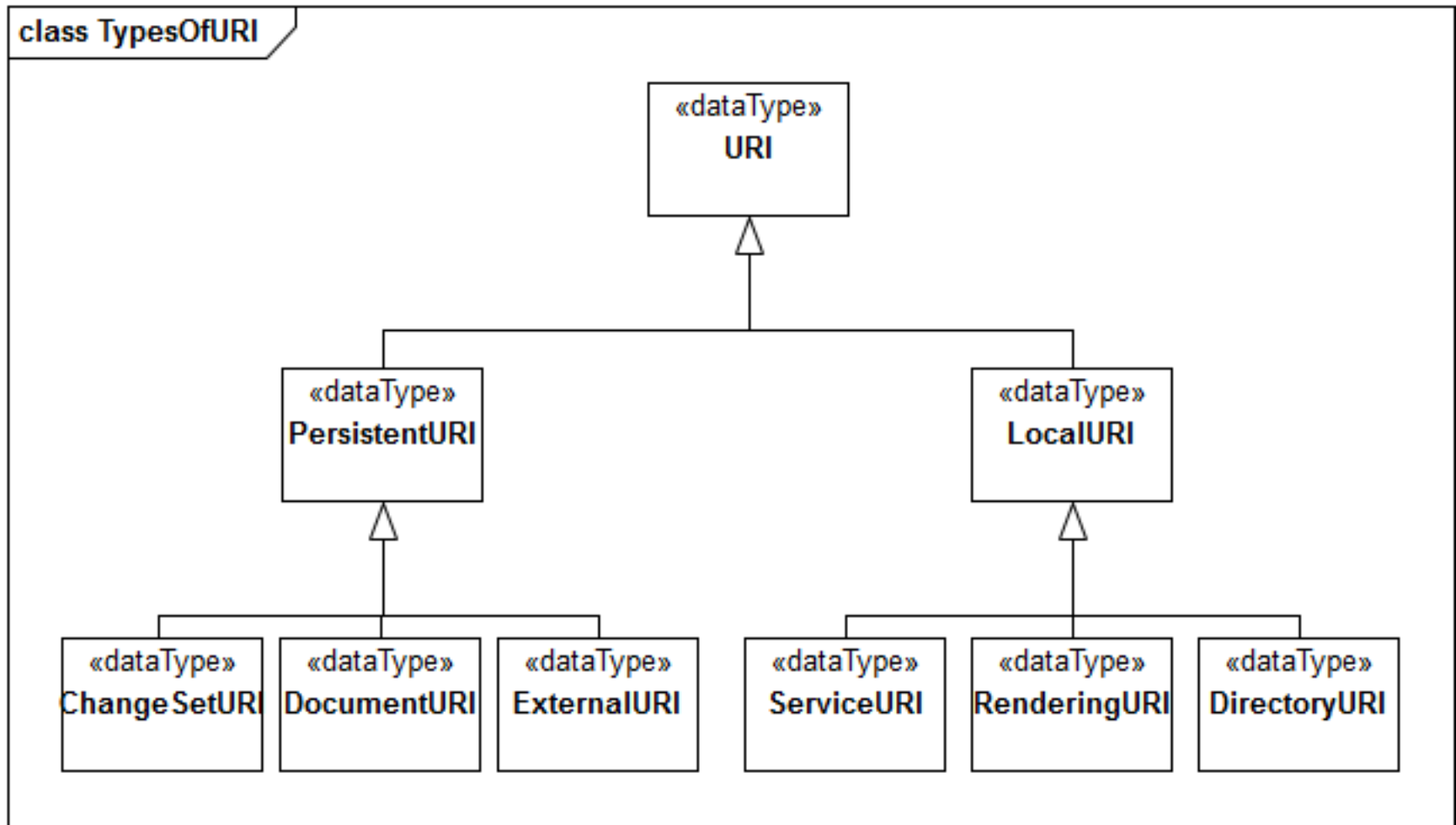
CTS2 / OMV Crosswalk Steps

- 1. A few CTS2 model details**
- 2. CTS2 view on resource / resource version**
- 3. Dive-down into the actual map**

**OMV / CTS2
Crosswalk**

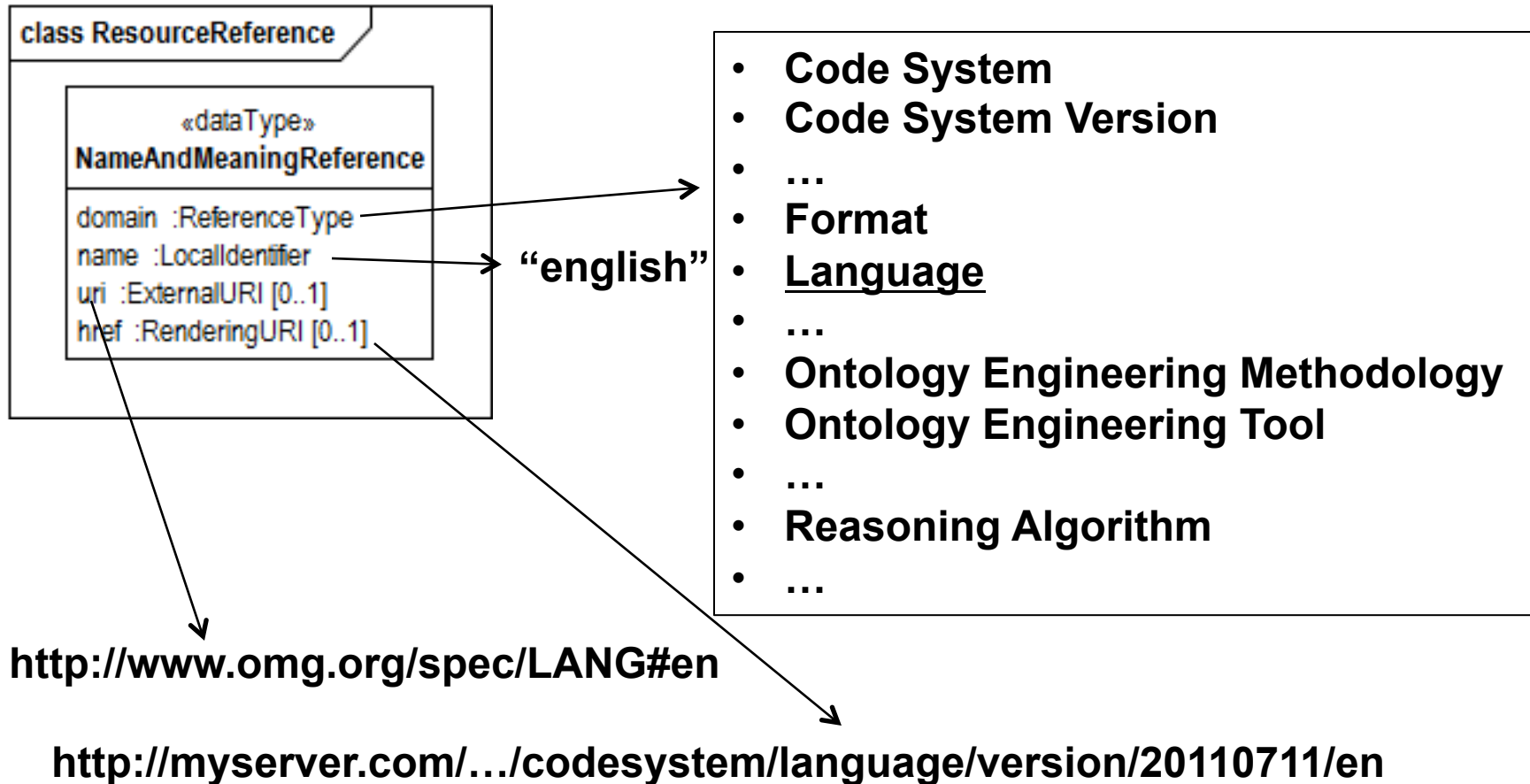
1. A FEW CTS2 MODEL DETAILS

Types of URI



NameAndMeaningReference

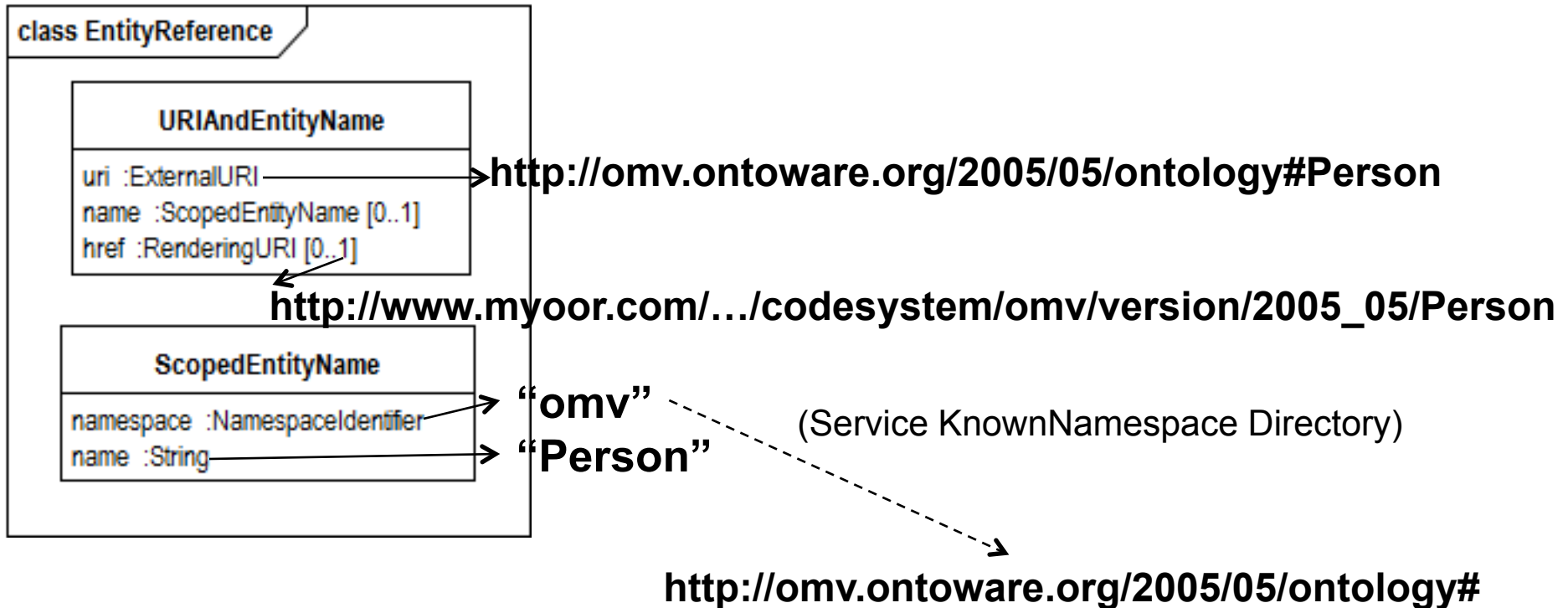
Associates a URI with a local “domain”



NameAndMeaningReference example

```
<core:resourceType uri=http://www.w3.org/2002/07/owl#Ontology>  
  <core:namespace>owl</core:namespace>  
  <core:name>Ontology</core:name>  
</core:resourceType>
```

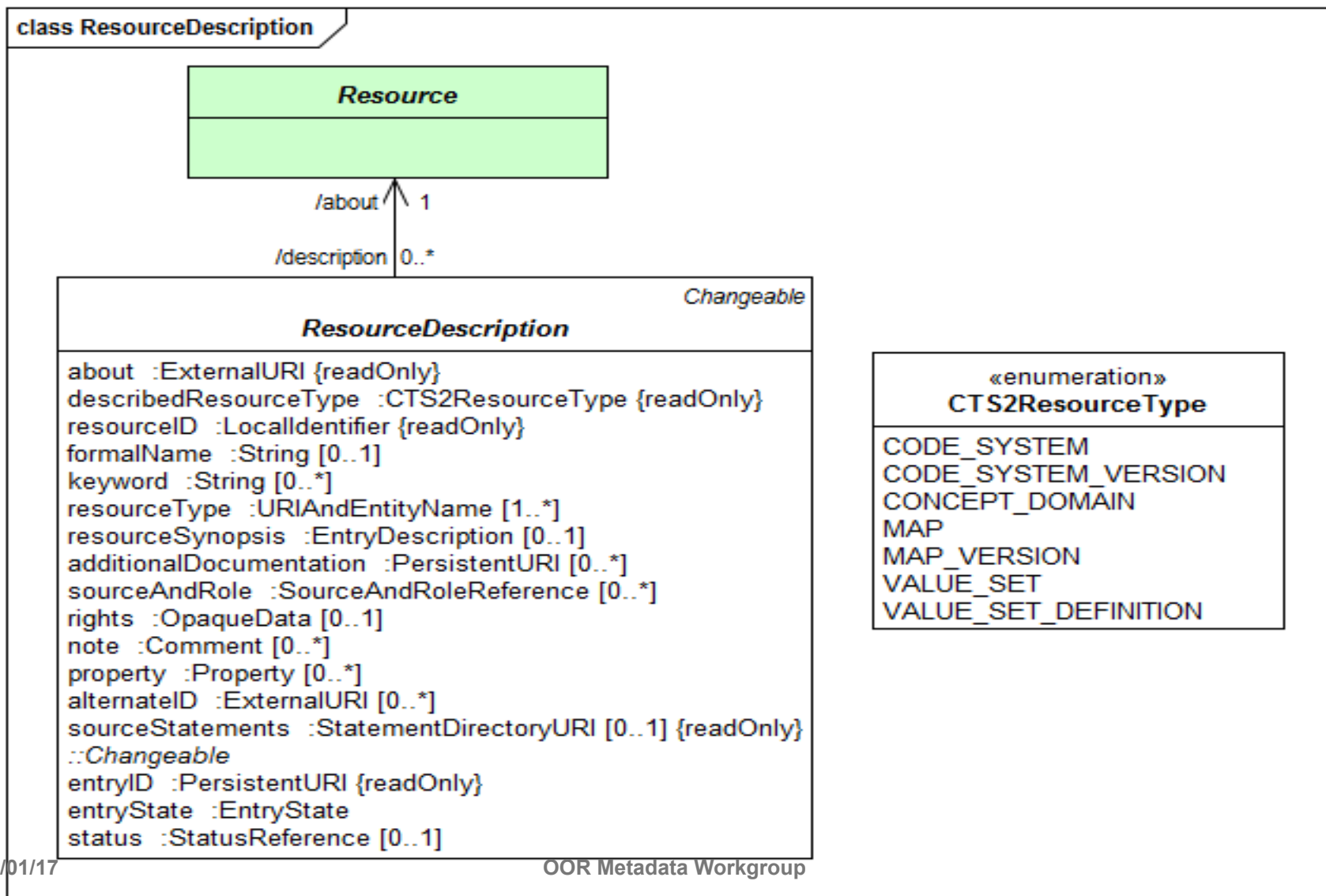
Referencing a class/property/ individual in an ontology



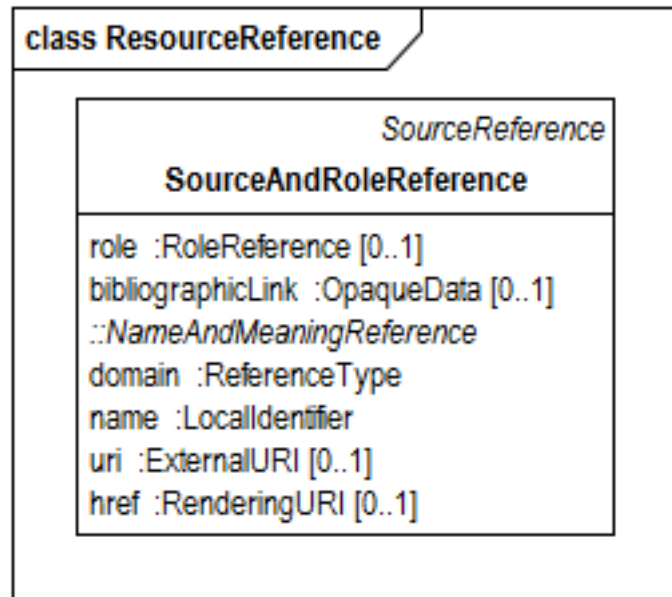
EntityReference Example

```
<core:predicate uri=  
http://omv.ontoware.org/2005/05/ontology/  
usedOntologyEngineeringTool href = "...">  
  <core:namespace>omv</core:namespace>  
  <core:name>usedOntologyEngineeringTool</core:name>  
</core:predicate>
```

Resource Description



SourceAndRoleReference



dc:creator
(for page / fragment / etc...)

SOURCE
(local name)
(uri or source)

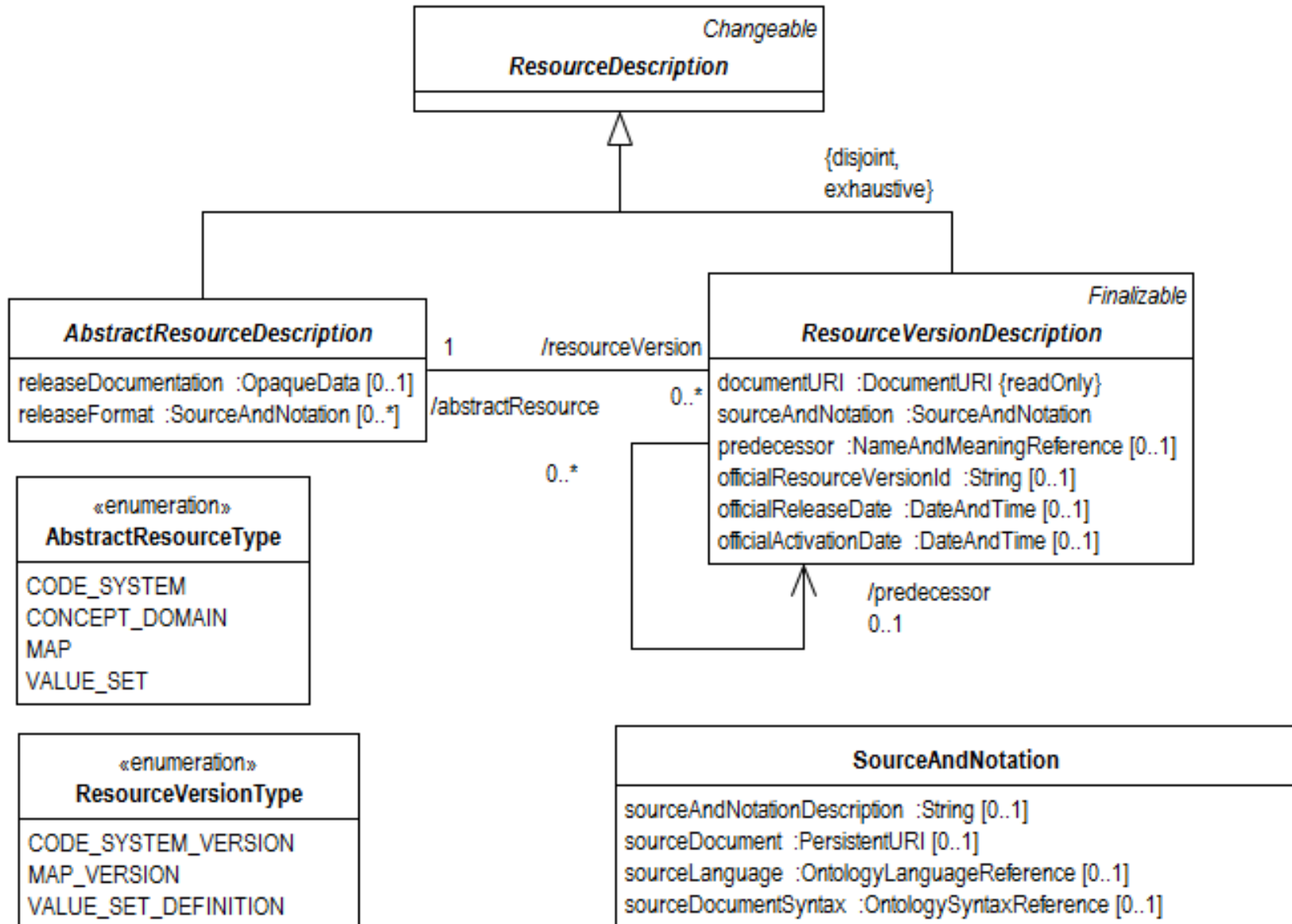
SourceAndRoleReference Example

```
<core:sourceAndRole>  
  <core:source uri="http://www.nlm.nih.gov">NLM</core:source>  
  <core:role uri="http://purl.org/dc/elements/1.1/creator">Creator</core:role>  
</core:sourceAndRole>
```

**OMV / CTS2
Crosswalk**

2. RESOURCE VS. RESOURCEVERSION

class ResourceDescriptionDetail





class CodeSystemCatalogEntry

*AbstractResourceDescription***CodeSystemCatalogEntry**

```

codeSystemName :CodeSystemName {readOnly}
codeSystemCategory :CodeSystemCategoryReference [0..1]
ontologyDomain :OntologyDomainReference [0..*]
ontologyType :OntologyTypeReference [0..1]
designedForOntologyTask :OntologyTaskReference [0..*]
hasOntologyLanguage :OntologyLanguageReference [0..1]
includes :CodeSystemReference [0..*]
versions :CodeSystemVersionCatalogEntryDirectoryURI [0..1] {readOnly}
currentVersion :CodeSystemVersionReference [0..1] {readOnly}
usedOntologyEngineeringTool :int [0..*]
::AbstractResourceDescription
releaseDocumentation :OpaqueData [0..1]
releaseFormat :SourceAndNotation [0..*]
::ResourceDescription
about :ExternalURI {readOnly}
describedResourceType :CTS2ResourceType {readOnly}
resourceID :LocalIdentifier {readOnly}
formalName :String [0..1]
keyword :String [0..*]
resourceType :URIAndEntityName [1..*]
resourceSynopsis :EntryDescription [0..1]
additionalDocumentation :PersistentURI [0..*]
sourceAndRole :SourceAndRoleReference [0..*]
rights :OpaqueData [0..1]
note :Comment [0..*]
property :Property [0..*]
alternateID :ExternalURI [0..*]
sourceStatements :StatementDirectoryURI [0..1] {readOnly}
::Changeable
entryID :PersistentURI {readOnly}
entryState :EntryState
status :StatusReference [0..1]

```

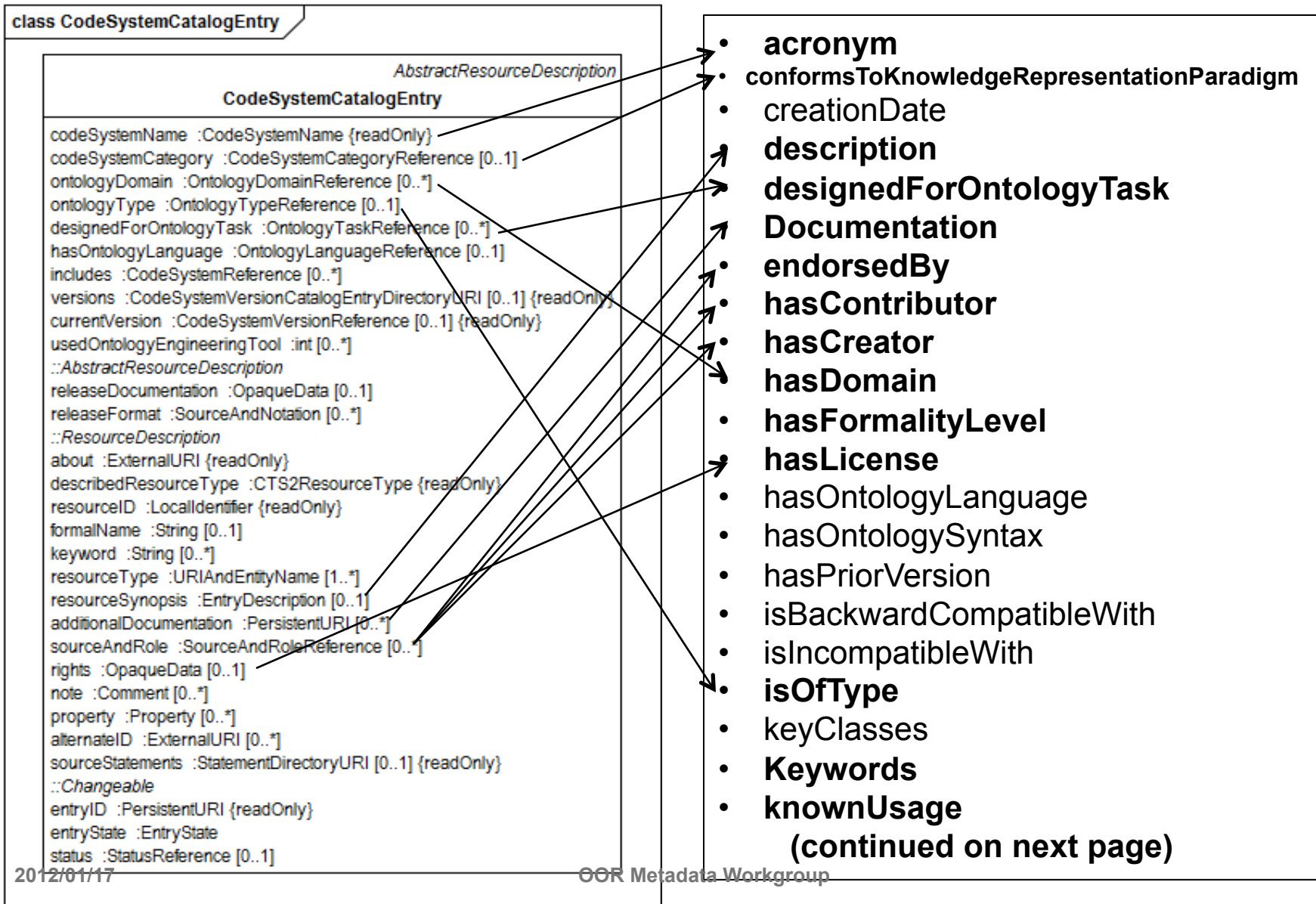
Why “CatalogEntry”



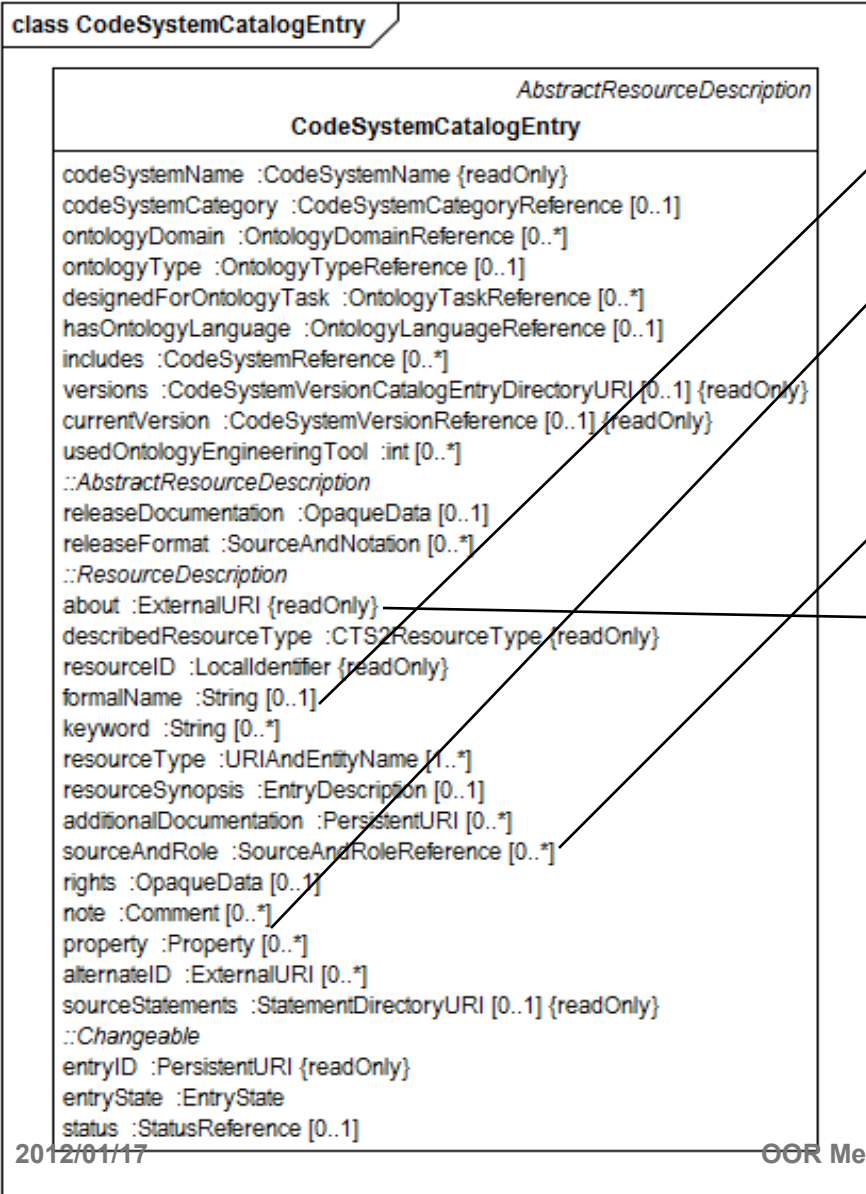
class CodeSystemCatalogEntry	
	<i>AbstractResourceDescription</i>
	CodeSystemCatalogEntry
codeSystemName	:CodeSystemName {readOnly}
codeSystemCategory	:CodeSystemCategoryReference [0..1]
ontologyDomain	:OntologyDomainReference [0..*]
ontologyType	:OntologyTypeReference [0..1]
designedForOntologyTask	:OntologyTaskReference [0..*]
hasOntologyLanguage	:OntologyLanguageReference [0..1]
includes	:CodeSystemReference [0..*]
versions	:CodeSystemVersionCatalogEntryDirectoryURI [0..1] {readOnly}
currentVersion	:CodeSystemVersionReference [0..1] {readOnly}
usedOntologyEngineeringTool	:int [0..*]
	:: <i>AbstractResourceDescription</i>
releaseDocumentation	:OpaqueData [0..1]
releaseFormat	:SourceAndNotation [0..*]
	:: <i>ResourceDescription</i>
about	:ExternalURI {readOnly}
describedResourceType	:CTS2ResourceType {readOnly}
resourceID	:LocalIdentifier {readOnly}
formalName	:String [0..1]
keyword	:String [0..*]
resourceType	:URIAndEntityName [1..*]
resourceSynopsis	:EntryDescription [0..1]
additionalDocumentation	:PersistentURI [0..*]
sourceAndRole	:SourceAndRoleReference [0..*]
rights	:OpaqueData [0..1]
note	:Comment [0..*]
property	:Property [0..*]
alternateID	:ExternalURI [0..*]
sourceStatements	:StatementDirectoryURI [0..1] {readOnly}
	:: <i>Changeable</i>
entryID	:PersistentURI {readOnly}
entryState	:EntryState
status	:StatusReference [0..1]

Ceci n'est pas une code system

Code System to OMV



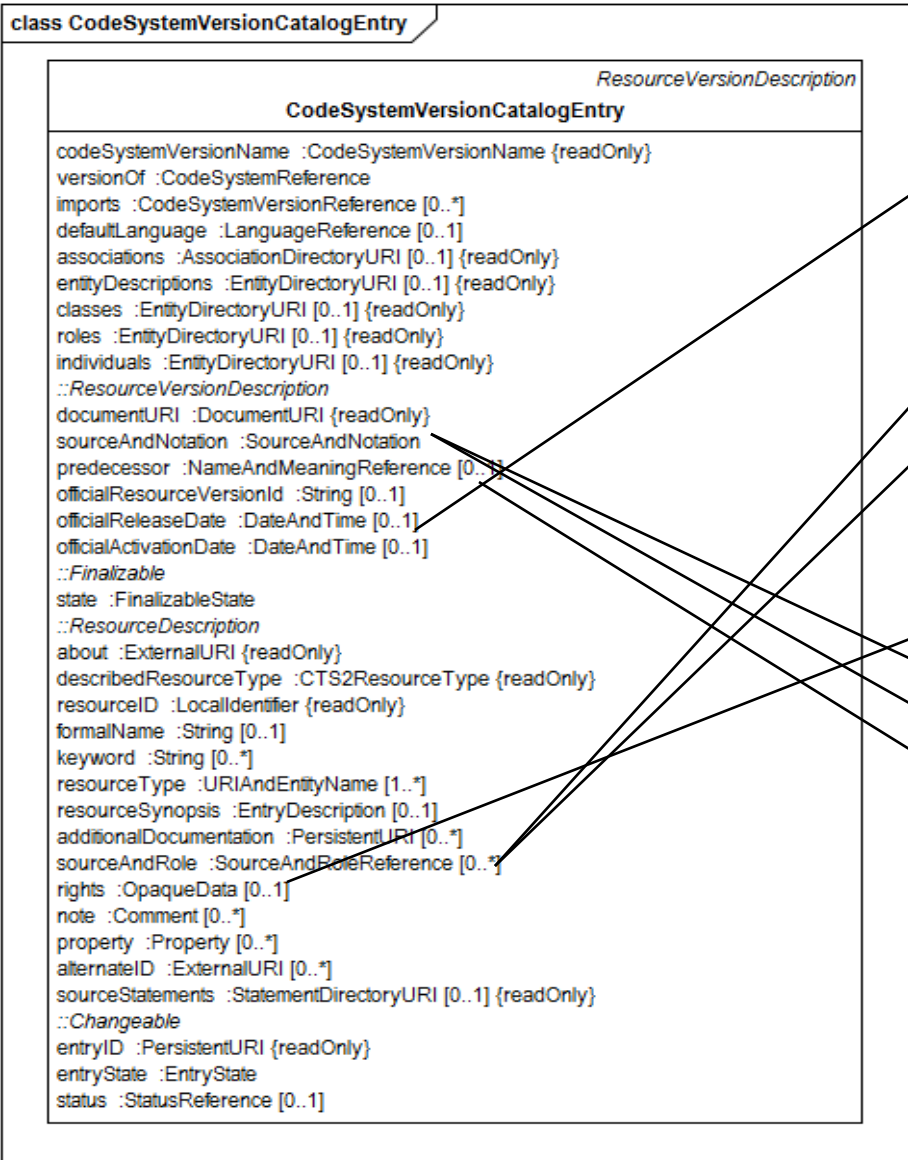
Code System to OMV (cont)



- (continued from previous page)
- **name**
 - naturalLanguage
 - **notes**
 - numberOfAxioms
 - numberOfClasses
 - numberOfIndividuals
 - numberOfProperties
 - **reference**
 - resourceLocator
 - status
 - **URI**
 - **usedOntologyEngineeringMethodology**
 - useImports
 - version

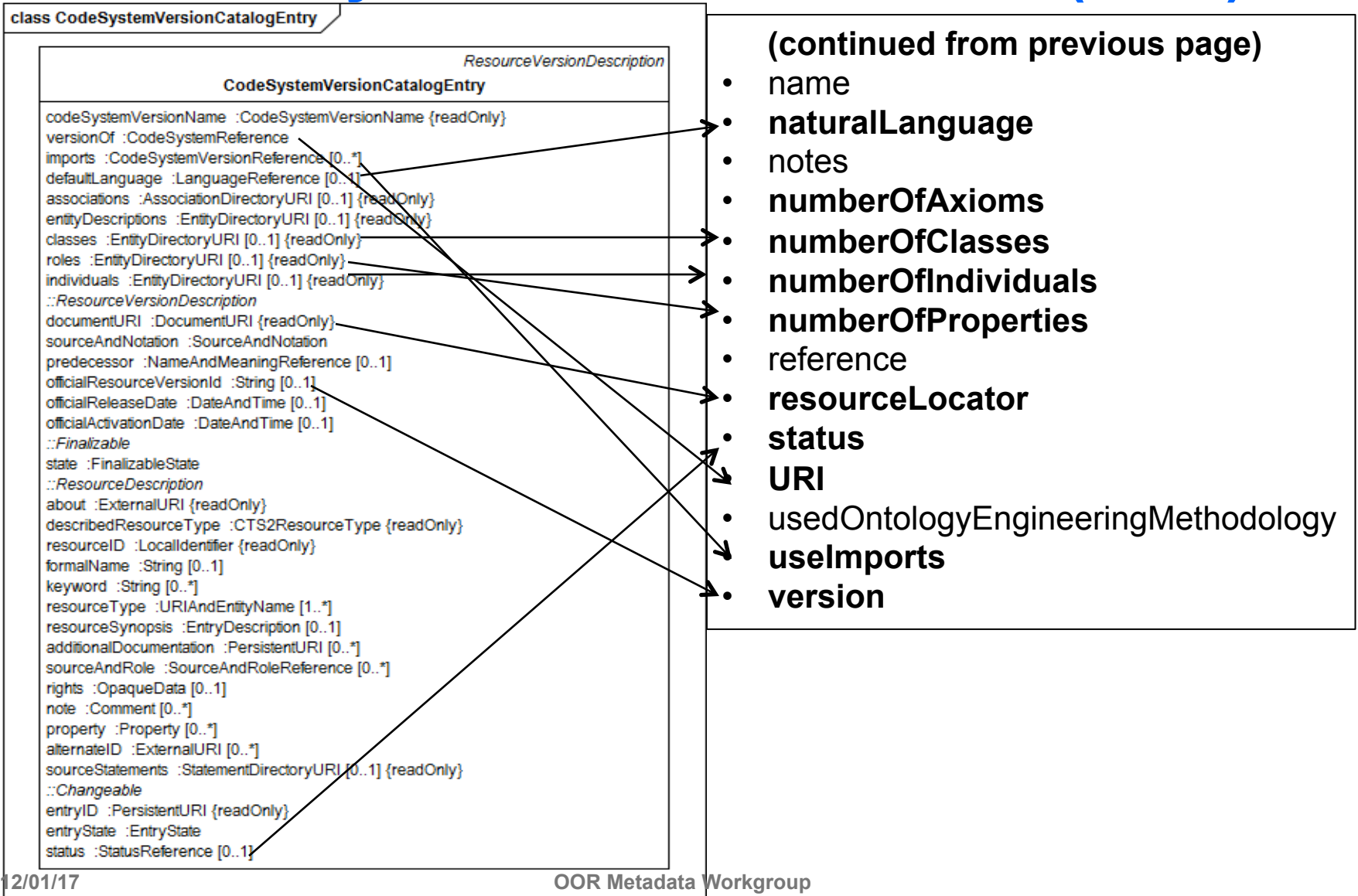


Code System Version to OMV



- acronym
- conformsToKnowledgeRepresentationParadigm
- **creationDate**
- description
- designedForOntologyTask
- documentation
- **endorsedBy**
- **hasContributor**
- hasCreator
- hasDomain
- hasFormalityLevel
- **hasLicense**
- **hasOntologyLanguage**
- **hasOntologySyntax**
- **hasPriorVersion**
- **isBackwardCompatibleWith**
- **isIncompatibleWith**
- isOfType
- keyClasses
- keywords
- knownUsage
- **(continued on next page)**

Code System Version to OMV (cont)



OMV and CTS2

Summary of Differences

- **Abstract vs. Resource Version**
 - **CTS2 – 1..***
 - **OMV - 1..1**
- **Naming**
 - **CTS2 – nouns only / pref to DC, SKOS etc.**
 - **OMV – mixture of nouns and verbs / OMV specific**
- **Model Scope**
 - **OMV – includes models of Party, Person Location, etc.**
 - **CTS2 – out of scope. Simple URI reference**

OMV and CTS2

Summary of Differences (cont)

- **“Value sets”**
 - **OMV** – model for **FormalityLevel**, **OntologyTask**, **OntologyType**, etc.
 - **CTS2** – referenced by **type/URI tuple (NameAndMeaning)**. Additional information available in *EntityDescription* part of service if needed.
- **Source and Role**
 - **OMV** – **hasCreator**, **hasContributor**, **endorsedBy**
 - **CTS2** – *SourceAndRole* – **URI for source and URI (typically drawn from DC) for role of source**

OMV and CTS2

Summary of Differences (cont)

- **Unmapped OMV fields**
 - **numberOfAxioms** – lacks consistent semantic interpretation and no examples present
 - **hasFormalityLevel** – CTS2/OMWG had difficult aligning w/ existing content. CTS2 is coarser – skos:conceptScheme or owl:Ontology
 - **isBackwardCompatibleWith, isIncompatibleWith** – CTS2 community found (almost) no examples and saw no significant application
 - **keyClasses** – rarely available, almost always == root classes
 - **knownUsage** – thought to be of value, but was considered to be too fluid / specific to be part of catalog. (May be reconsidered)
 - **usesOntologyEngineeringMethodology** – omitted for lack of reference value set and need of further discussion

OMV and CTS2

Summary of Differences (cont)

Note on unmapped fields:

CTS2 has “property” attribute (predicate / target) [0..*] that covers the “none of the above”

OMV and CTS2

Summary of Differences (cont)

- **‘Canonical RDF’**
 - **CTS2 will defer to DC / SKOS / OWL / FOAF tags when overlaps exist**

**OMV / CTS2
Crosswalk**

CURRENT STATE AND NEXT STEPS

Current State CTS2 Specification

- **CTS2 PIM / HTTP REST PSM adopted as OMG standard in June**
- **OMG FTF - Finalization Task Force Report due in April**
 - **Error correction and clarification (finish Z, much more documentation)**

Current State

CTS2 Implementation Guides

- **IHTSDO (SNOMED-CT) has formed a group to develop the SNOMED-CT CTS2 Implementation Guide**
 - **Target draft document Mar 2012**
- **HL7 CTS2 Implementation Guide**
- **Targeting RDF/OWL implementation guide middle of 2012**

Reference Links

- <http://informatics.mayo.edu/cts2/framework> - SDK and examples
- <http://informatics.mayo.edu/cts2> - overview site (old at the moment but will be updated shortly)
- <http://www.bioontology.org/wiki/index.php> -
 - CTS2 wrapper for BioPortal XML available end of Jan
 - CTS2 wrapper for BioPortal RDF coming first half 2012
- [http://informatics.mayo.edu/cts2/index.php?title=CTS2 and OMV](http://informatics.mayo.edu/cts2/index.php?title=CTS2%20and%20OMV) - summary of OMV crosswalk

Examples

RxNorm ORWG Example -

- <http://informatics.mayo.edu/exist/cts2/rest/codesystem/RxNorm>
- <http://informatics.mayo.edu/exist/cts2/rest/codesystem/RxNorm?format=json>
- [**http://informatics.mayo.edu/exist/cts2/rest/codesystem/RxNorm/version/RxNorm_10AB_110307F**](http://informatics.mayo.edu/exist/cts2/rest/codesystem/RxNorm/version/RxNorm_10AB_110307F)

(Note: may have to use “show source” because of embedded HTML)

Examples

BioPortal Wrapper:

- <http://informatics.mayo.edu/cts2/rest/codesystems>
- <http://informatics.mayo.edu/exist/cts2/rest/codesystem/NCIm>

eXist Implementation:

- <http://informatics.mayo.edu/exist/cts2/rest/codesystems>

BioPortal RDF Implementation (pre-alpha):

- <http://informatics.mayo.edu/cts2/services/bioportal-rdf/codesystems>