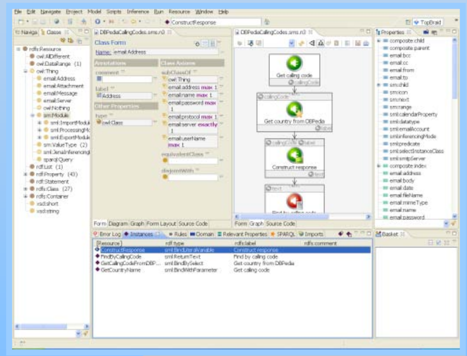
NExIOM, the NASA Constellation Program Ontologies

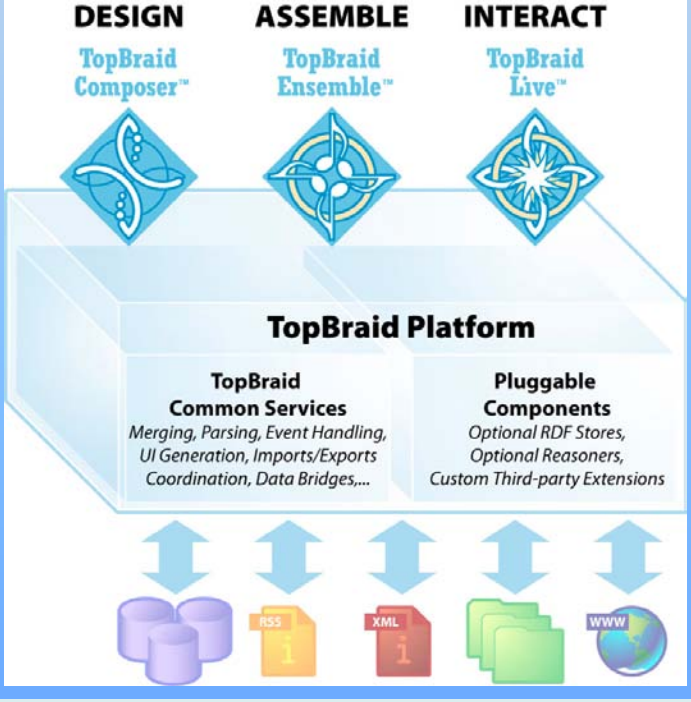
"How they are supporting NASA Constellation Program Data Architecture and its applications"

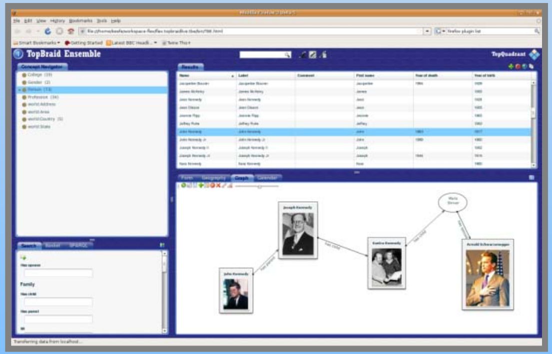
Ralph Hodgson, TopQuadrant and
NASA NExIOM Ontologies Lead

Introducing TopQuadrant

TopQuadrant is a Semantic Web Technologies Training, Consulting and Products Company. Formed in 2001, TQ was the first US company devoted to Semantic Web Technologies. TopBraid Suite is the company's product offering for RDF/OWL modeling environments, semantic platforms and rich end-user ontology-driven applications.

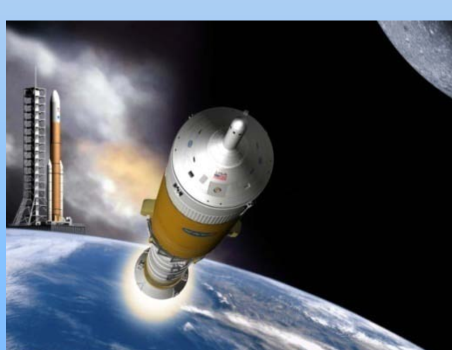






TopQuadrant has been working with NASA since 2002 on Ontologies for Aerospace Engineering

What is NExIOM?



NExIOM, the *NASA Exploration Initiatives Ontology Models* formalize the way machines (and people) refer to NASA Elements, their Scientific and Engineering disciplines, related work activities, and their interrelationships throughout the NASA Constellation Program.

Through the use of knowledge representations information is intelligible and actionable to machines, tools, and people. Information can be found, aggregated and reasoned over to generate products, enable interoperability between systems and tools, and inform decisions.

NExIOM consists of Models, a Semantic Infrastructure, and Services, integrated with operational tools and systems.

See http://ontolog.cim3.net/file/work/OKMDS/2008-03-20_Organizing-Science-for-Discovery-at-NASA/NASA-Constellation-Program-Ontologies--RalphHodgson_20080320.pdf

Product Data Exchange Challenges

Issue

- Application and data heterogeneity
- Ambiguous definitions
- Inconsistent (and sometimes conflicting) terminology
- Limited/No explicit relationships between data and tools
- N^2 integration challenge
- Insufficient Provenance

Impact

Translation efforts

Constant reformatting

Correction due to wrong/incomplete data

Time consuming manual effort

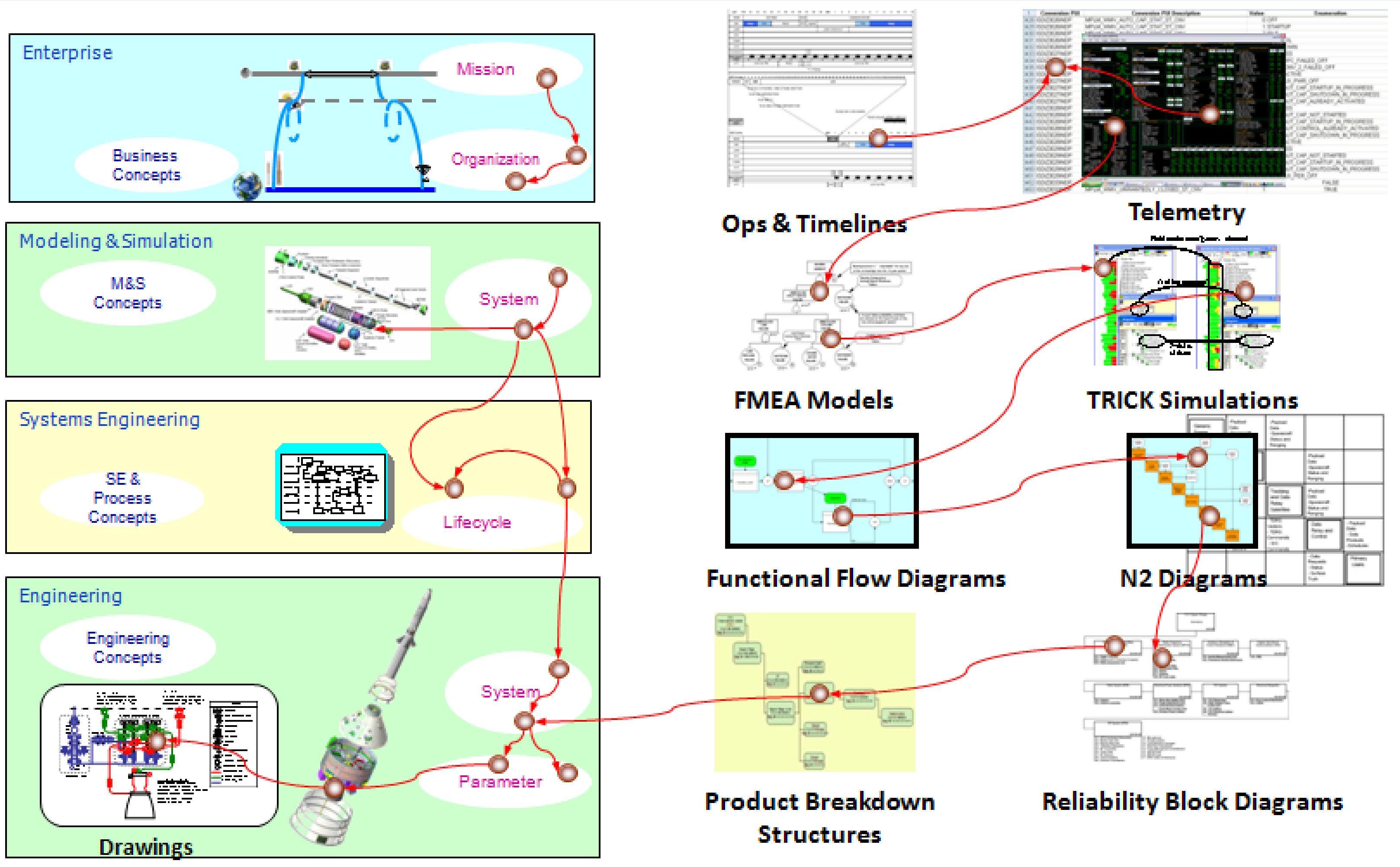
Outcome

- System Failures
- Rework
- Stressful workloads
- Reduced time for higher value work
- Lower confidence
- Additional effort checking
- Potential for cascading problems

NExIOM Goals for CxP

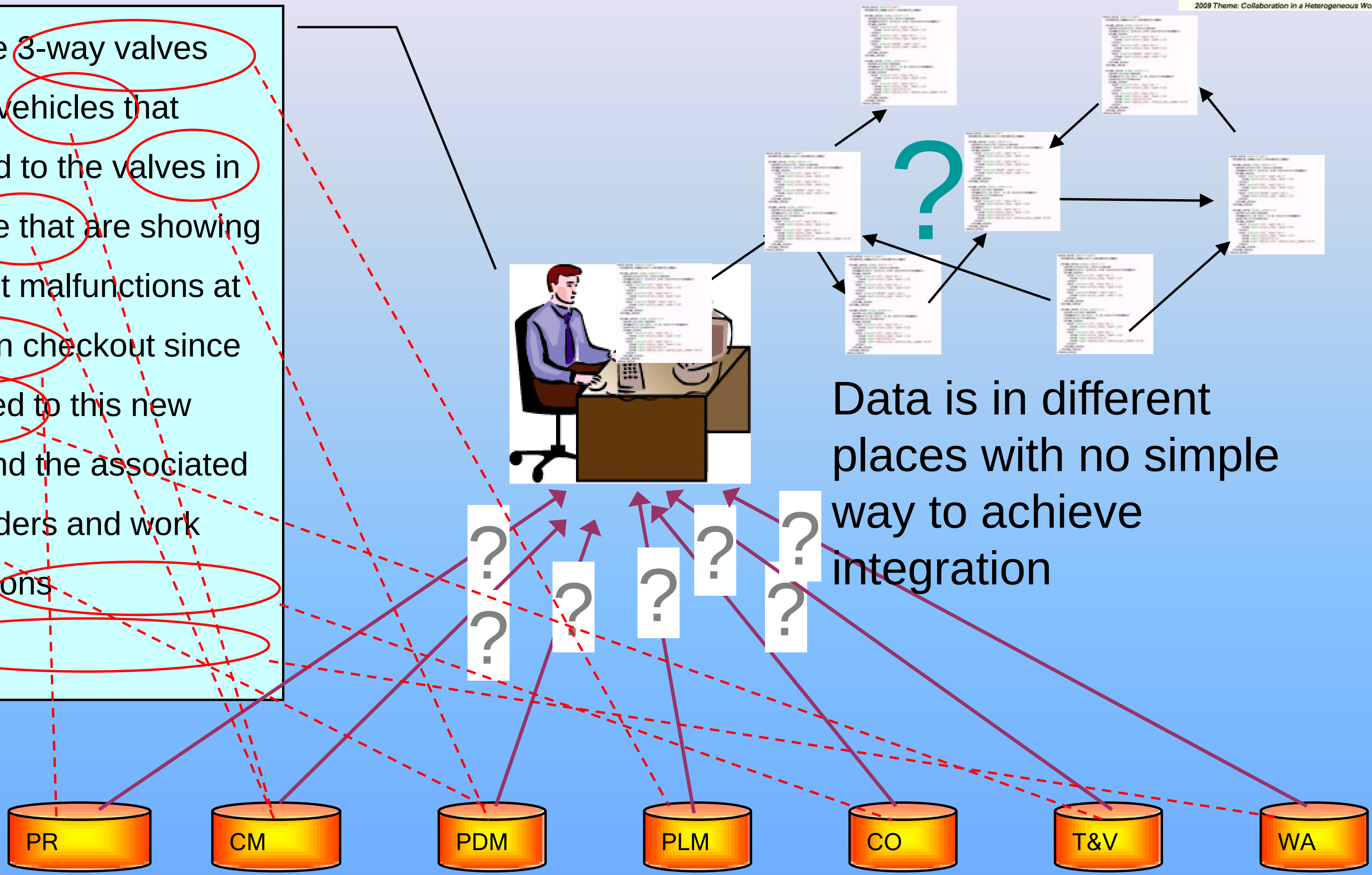
- **Constellation Program needs a uniform and consistent method for treatment of engineering data**
  - Specification of data and data structures
  - Processing/Use of data
  - Exchange of data
  - Discovery of data
  - Understanding Authority of data
  - Understanding Pedigree of data
  - Defining Relationships between data
  - Relating data to processes, organizations, software applications, hardware systems, etc.
- **This capability provides for general interoperability**
  - Not only for engineering modeling and simulation,
  - But also across CxP disciplines, domains, systems, processes, applications, DBs, etc.

Motivating Scenario #1: "Connect the dots" across Information Objects



Motivating Scenario #2: The Integration Challenge

Find all the 3-way valves across all vehicles that correspond to the valves in this vehicle that are showing intermittent malfunctions at this point in checkout since we changed to this new supplier and the associated change orders and work authorizations



Data is in different places with no simple way to achieve integration

Ontologies allow the meaning of data to be expressed so that data can be related across databases with different schemas

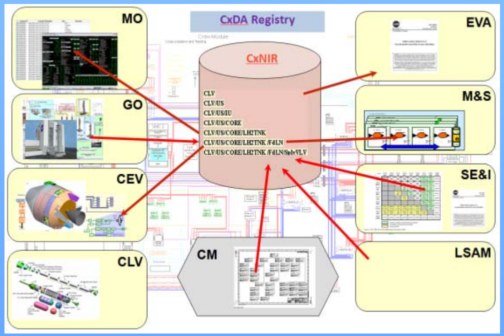
Motivating Scenario #3: The Terminology Challenge

NExIOM, the *NASA Exploration Initiatives Ontology Models* formalize the way machines (and people) refer to NASA Elements, their Scientific and Engineering disciplines, related work activities, and their interrelationships in the Enterprise

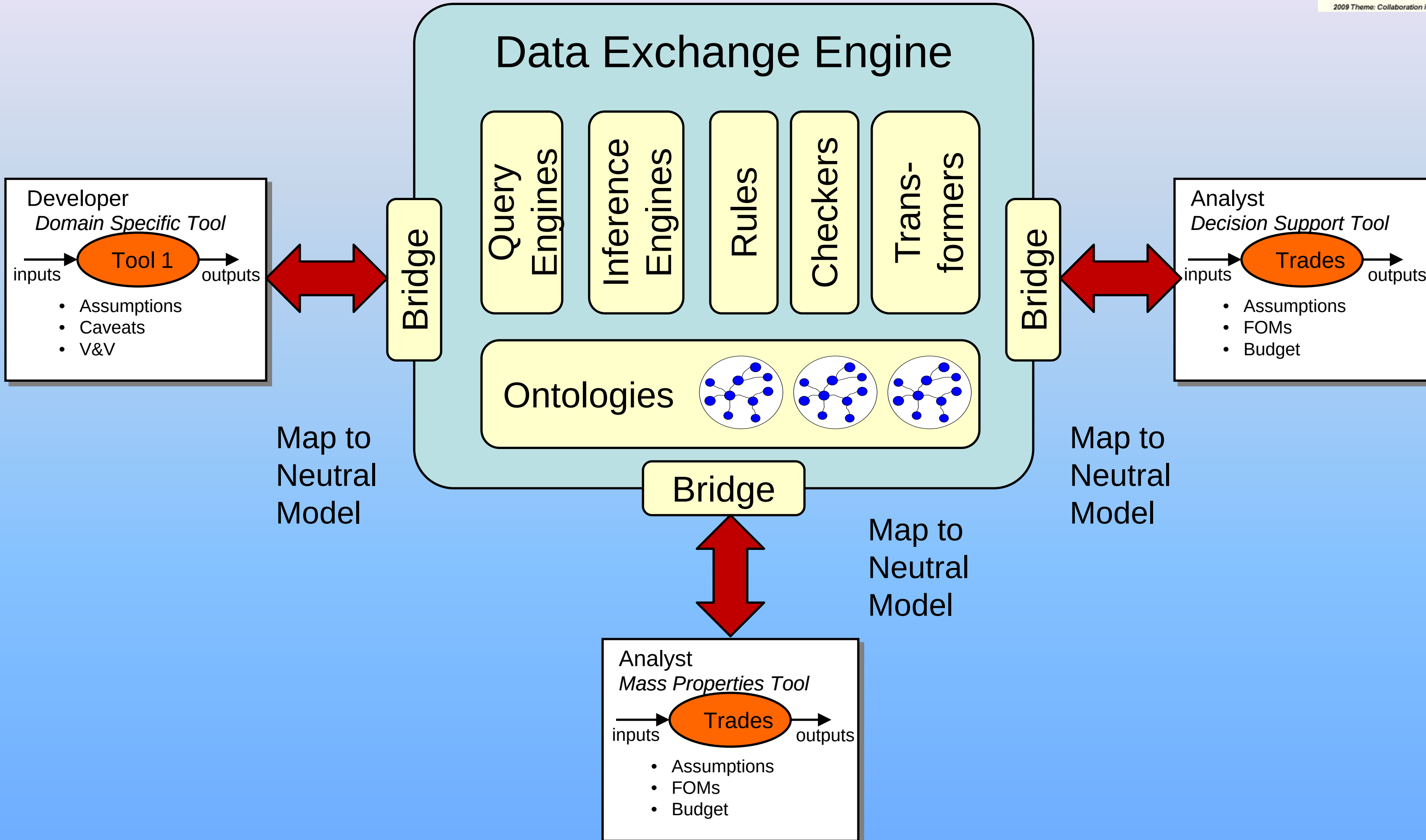
Are these the same valves?

Hardware Nomenclature: Flow Control Valve	=	Software Nomenclature: 3-Way Mix Valve	=	Telemetry/ Telecommand Nomenclature: Heat eXchanger Bypass valve

An Ontology-Based Registry defines the concepts and relationships of an area of knowledge, relating information in different contexts



Motivating Scenario #4: Data Exchange



Ontology-Based Data Integration and Translation – map to a common model using queries and rules – perform checks and transformations.

NExIOM Approach

- **Achieving NExIOM goals requires the following**
  - A standard method of defining and specifying data
    - ontologies
  - A standard method of describing data structures and data relationships
    - ontologies
  - A common terminology with consistent definitions
    - NExIOM Standard Vocabularies
  - A standard method of mapping one data element/set to another
    - mediation schemas in ontologies
  - A standard method of relating data to processes, aoftware applications, hardware systems, etc.
    - ontologies
  - A standard method of encoding (formatting) data
    - XML
- **Note: these are all aspects of a Data Model or Ontology**

Interoperability is about Semantics – where are the standards for that?

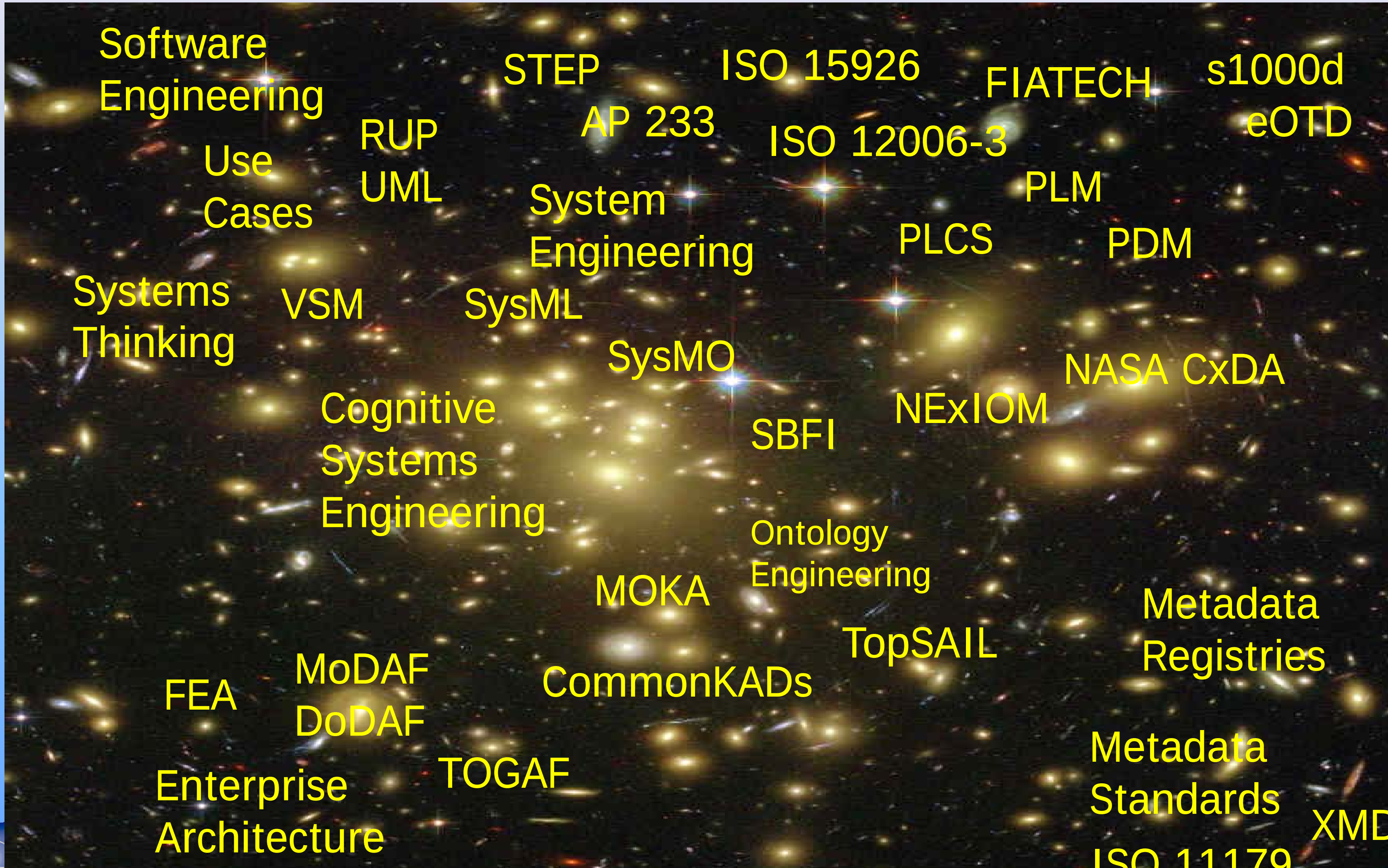
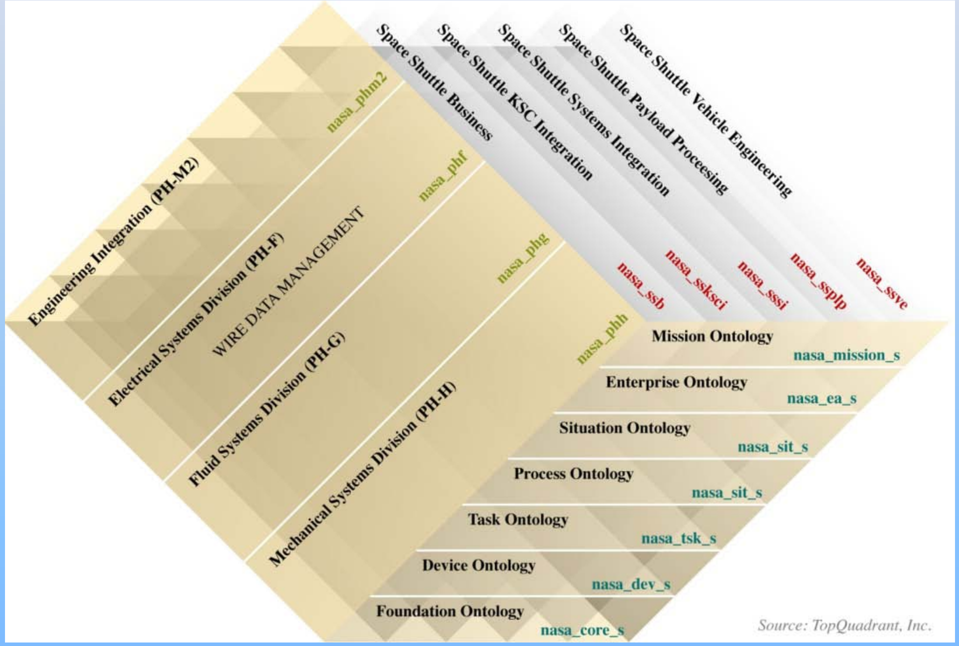


Image source: http://hubblesite.org/newscenter/archive/2003/01/ – Abell 1689 deep space image

NASA NExIOM Modular Ontologies

- ~120 Schema Ontologies
- 100's Datasets
- ~ 20 of Aggregation, Bridging, Mapping and Proxy Ontologies



Ontologies are partitioned according to domains, disciplines, organizations and levels of specificity. Named graphs are aggregated through configuration ontologies according to specific needs.

How Semantic Web Technologies support the NASA Constellation Program

- **Data Architecture**
  - Name and Identifier Rules
  - Data Types, Information Types and Structures
  - Document Generation
- **System of Registries**
  - Controlled Vocabularies for Units, Data Types, Quantities and Enumerations
  - Knowledge Capture
- **Telemetry and Command (C3I)**
  - Specifications of Metadata, Packet Definitions
  - Command and Parameter Registries
- **Co-existence of OWL and XML**
  - Schema and XML Generation: XML SchemaPlus
- **Tool Interoperability**
  - Tool Specifications and Parameter Interoperability
- **System Ontologies**
  - How does NExIOM relate to SysMO and SysML
- **Concluding Remarks**

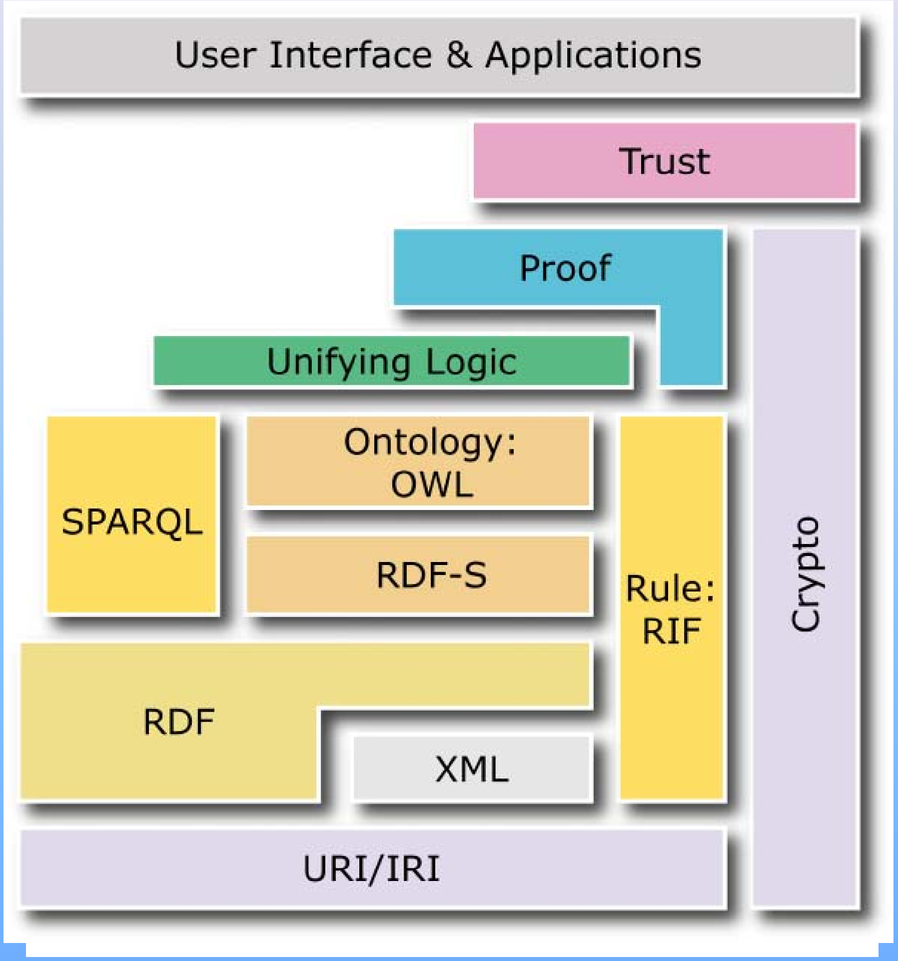
Semantic Web Technology Primer

Key Benefits of Semantic Technology

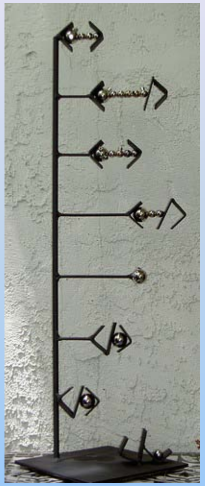
- **Information Integration**
  - Mappable terms to build consistent & extensible vocabularies.
  - Integrate models with both structured and unstructured data
- **Search and Analysis**
  - Semantic relationships between data enable powerful queries that leverage knowledge organized by people to deliver specific answers in a highly scalable fashion
  - Non-programmers can connect , search and analyze data
- **Application Longevity and Flexibility**
  - Future-proof applications (30, 50 100 years) by enabling knowledge workers to participate in model-based application development

OWL – think of it as XML++

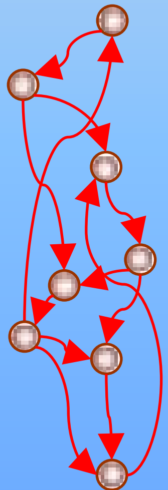
- OWL = Web Ontology Language
  - A language for describing a domain of interest
  - Classes of things, properties of things, relationships between things
  - A standard defined by the World-Wide Web Consortium (W3C)
- How does it relate to XML?
  - OWL can be serialized in XML and N3
  - OWL is built on the Resource Description Framework (RDF)
  - OWL constructs allow us to say things that XML Schema does not allow



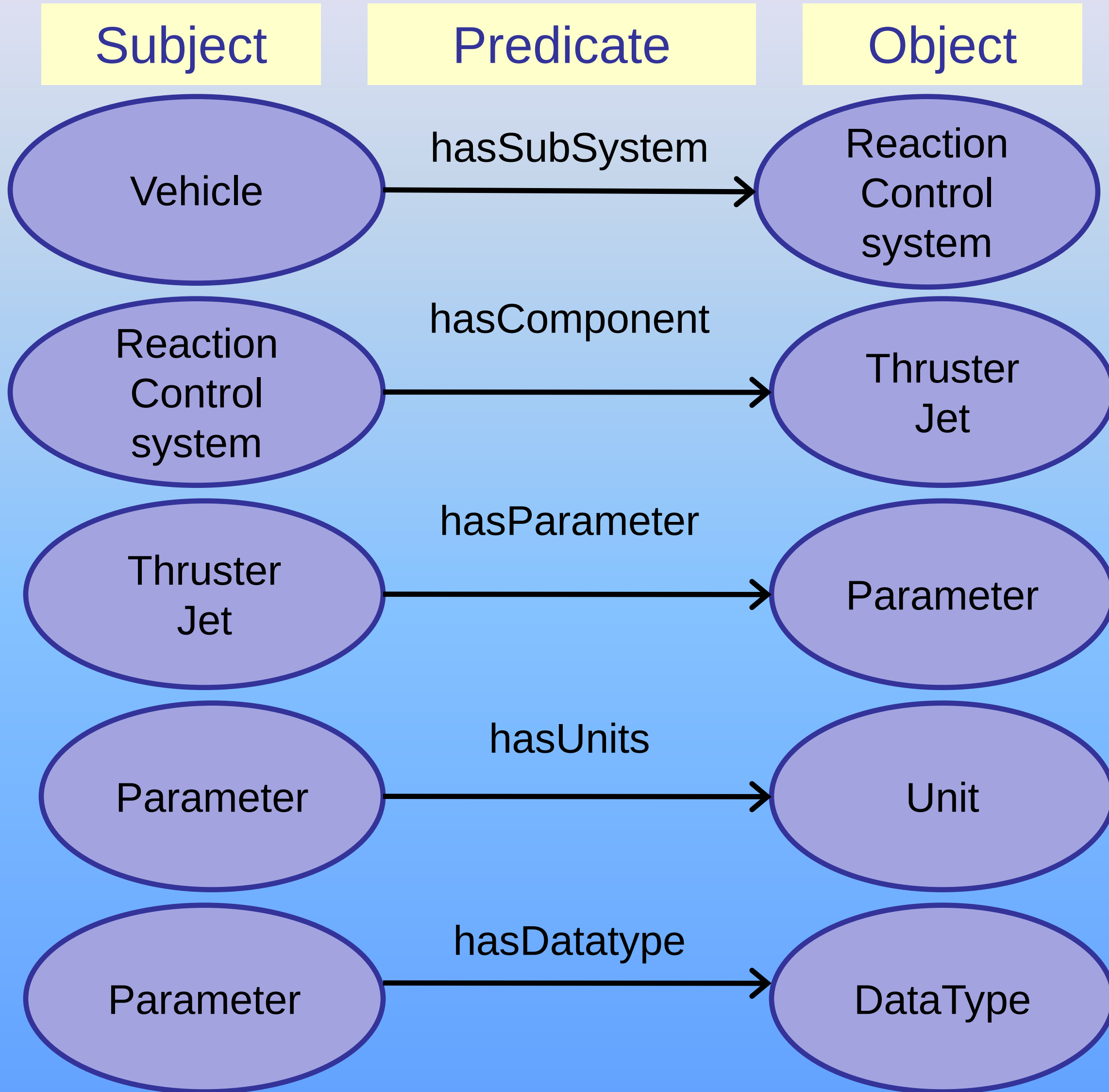
Why OWL - the Ontology Web Language?



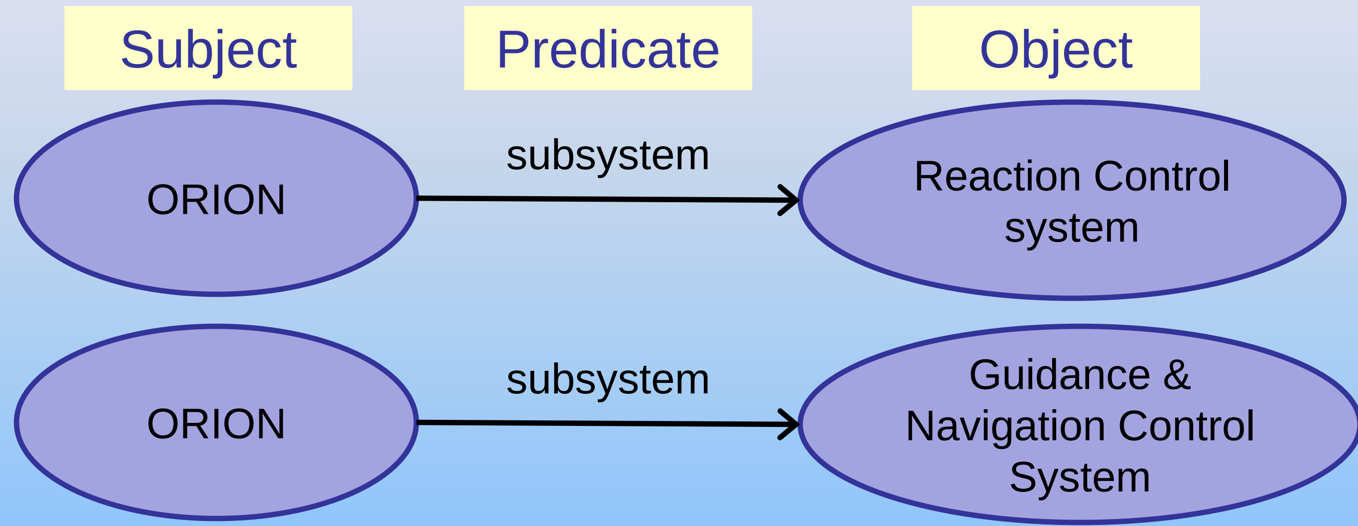
- **XML is document-based not model-based**
  - Hierarchies of Containers with weak support for relationships
  - Weak support for aggregation (combining documents)
  - Schema Limitiations
- **UML is Object-Based**
  - Restricted Type System
  - Weak on Relationships
  - Weak notion of identity
  - Metamodel (Schema) is in a different language
- **OWL is Set-Based**
  - Expressive Type System
  - Strong on Relationships
  - Strong notion of identity
  - Graphs not Trees
  - Metamodel is in the same language



Semantic Web Key Idea # 1 – "Think Triples": Subject Predicate Object



Semantic Web Key Idea # 2 – Identifiers not Names ("Everything has a URI")



cev:ORION	rdf:type	nasa:Vehicle;
cev:ORION	sys:subsystem	cx:RCS

\+

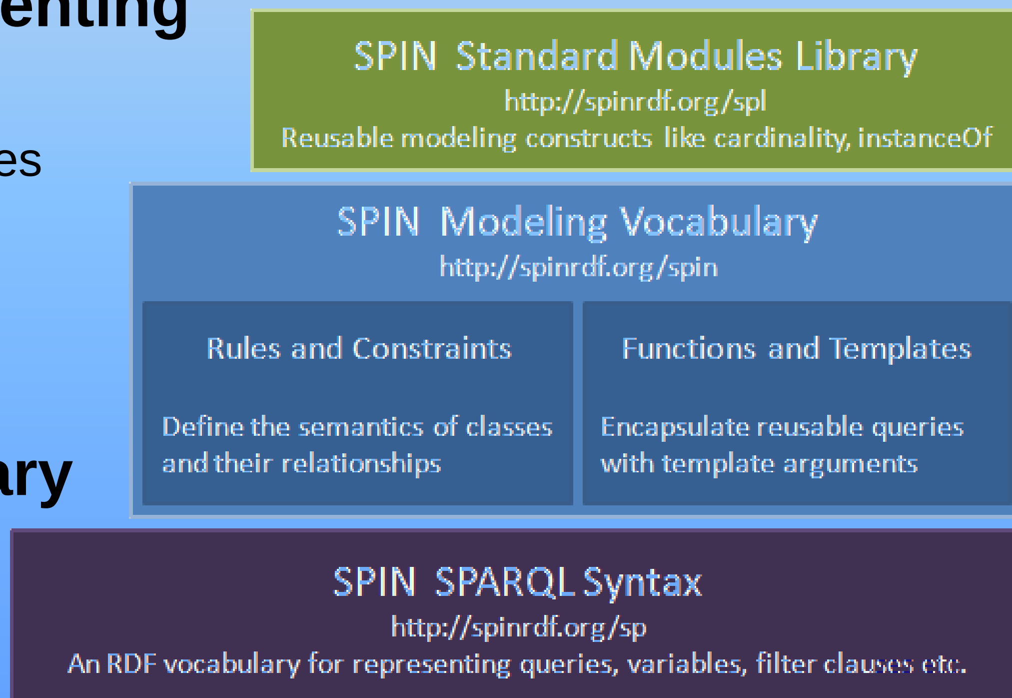
cev:ORION	rdf:type	nasa:Vehicle;
cev:ORION	sys:subsystem	cx:GNCS

→ **Statements in different models but same URIs means more information about the same things**

Key to Product Data Exchange is Acquisition, Interpretation and Transformation of Quality Data – This needs a Rules Language

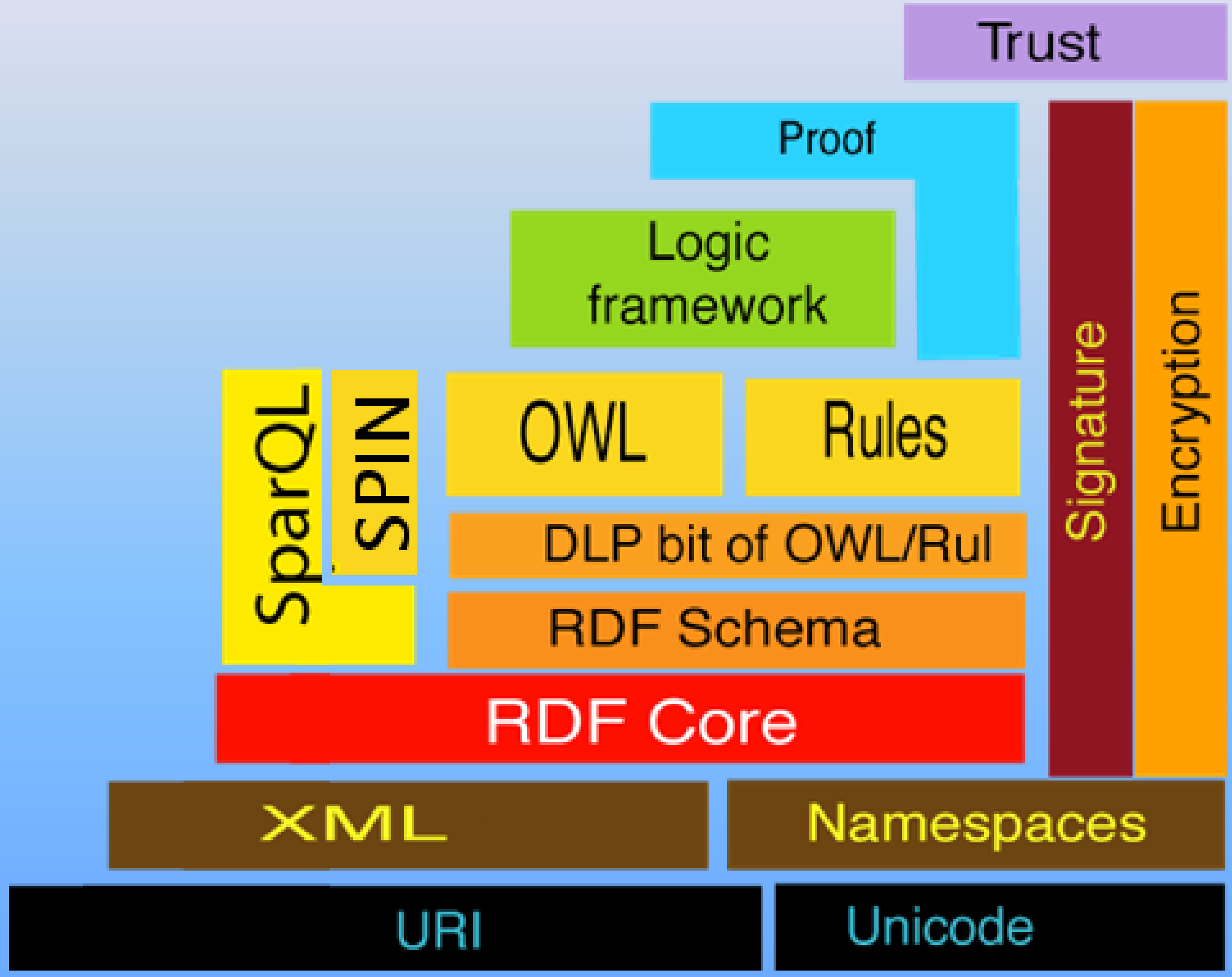
Introducing SPIN

- **SPIN - SPARQL Inferencing Notation**
  - define constraints and inference rules on Semantic Web models
  - http://spinrdf.org
- **Specification for representing SPARQL with RDF**
  - RDF syntax for SPARQL queries
- **Modeling vocabulary**
  - constraints, constructors, rules
  - templates, functions
- **Standard Modules Library**
  - small set of frequently needed SPARQL queries



Open source at http://spinrdf.org

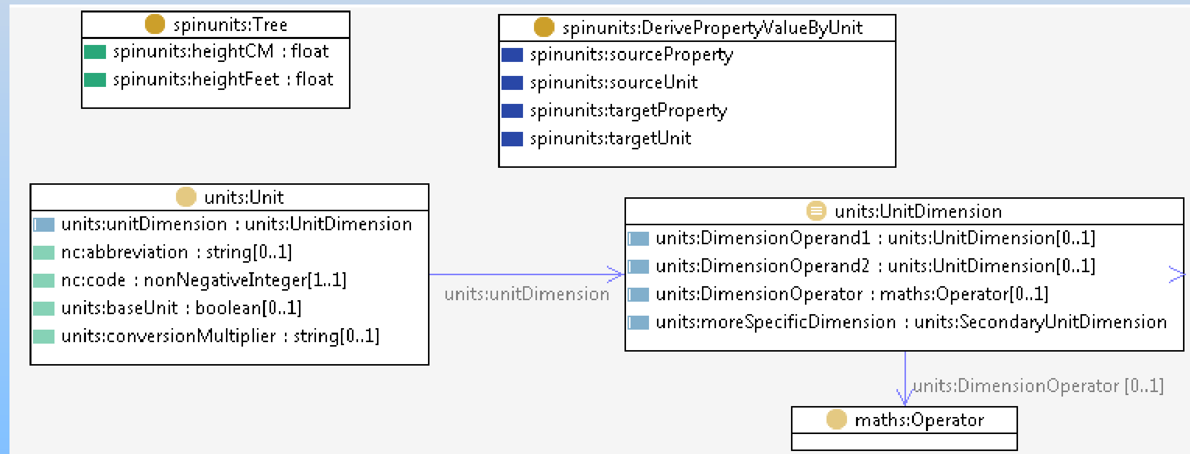
SPIN Position in the Semantic Web Layer Cake

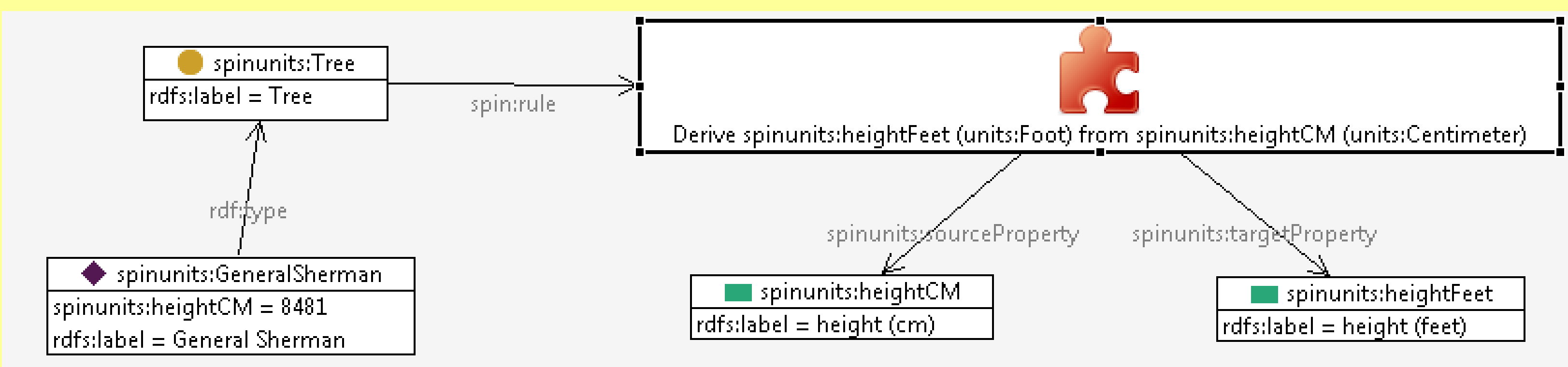


SPIN Example of Units Conversion: the height of “General Sherman”

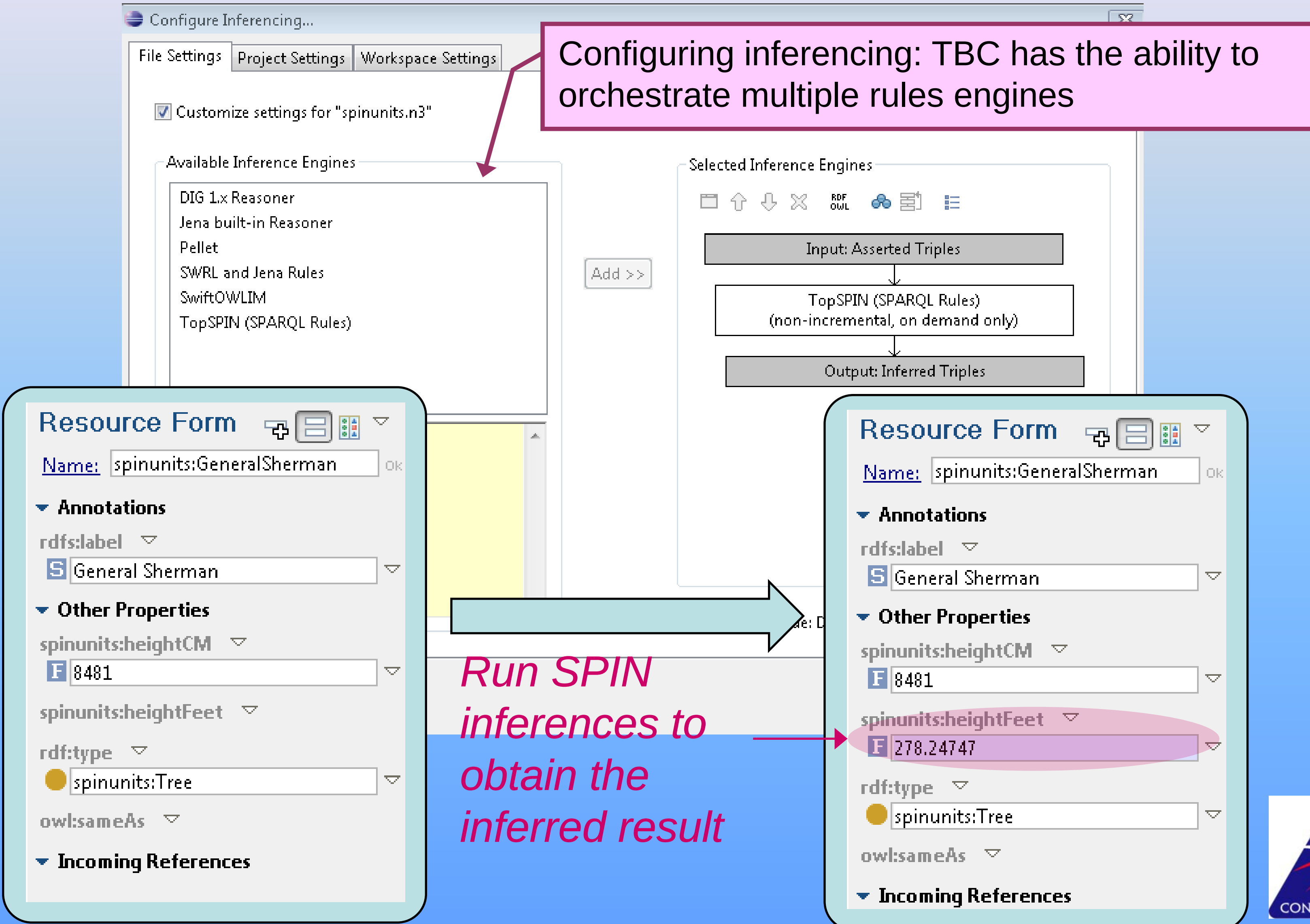
Using SPIN for rule-based Units Conversion.

The example invokes a rule that automatically converts the height of the General Sherman tree from Centimeters to Feet.





SPIN Example of Units Conversion: the height of “General Sherman”



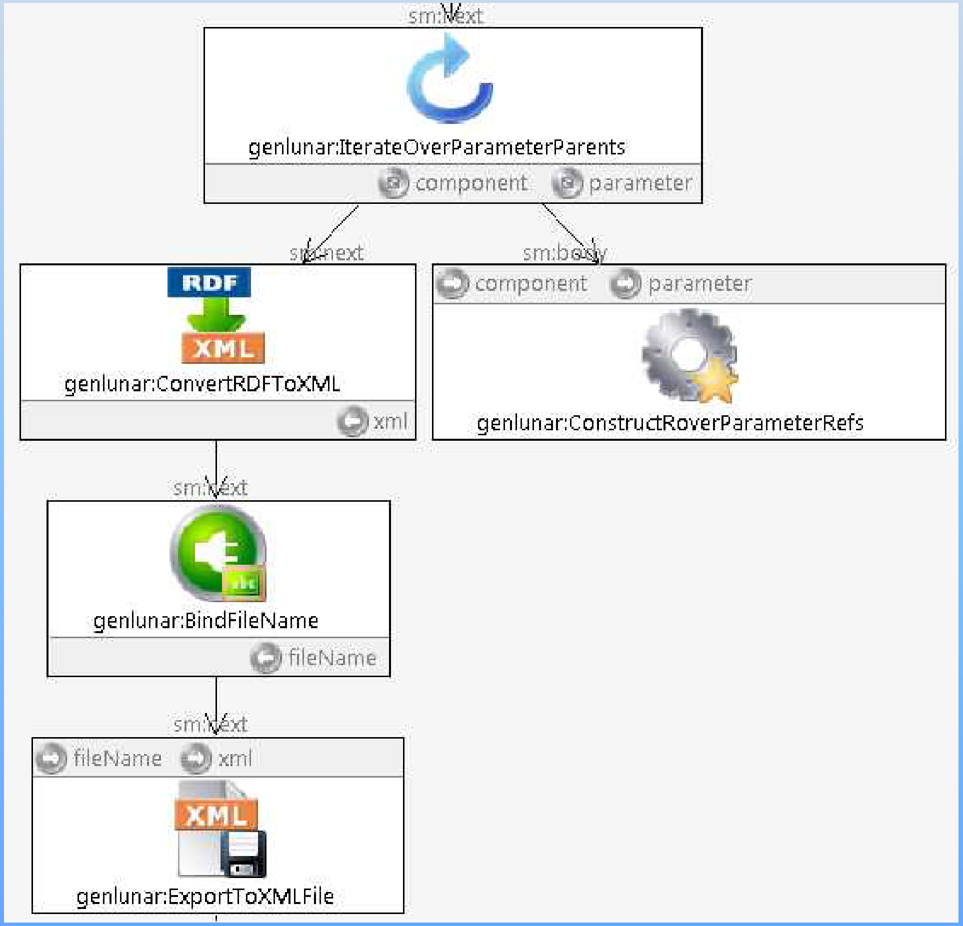
Transformation and Mediation Using Semantic Technology (1)

SPARQL is both a Query Language and a Rules Language

```
PREFIX lunar-rover:<https://nst.nasa.gov/esmd/cx/lunar-rover.owl#>
PREFIX system: <https://nst.nasa.gov/esmd/cx/system.owl#>
PREFIX assembly: <https://nst.nasa.gov/esmd/cx/assembly.owl#>
CONSTRUCT {
    ?assemblyType a owl:Class .
    ?assemblyType sxml:element "lunar-rover:subAssembly" .
    ?assemblyInst a ?assemblyType .
    ?chassis composite:child ?assemblyInst .
    ?assemblyInst genlunar:ref ?assemblyQName .
    ?assemblyInst genlunar:type ?assemblyActualTypeQName .
}
WHERE {
    ?chassis a lunar-rover:VehicleChassis .
    ?chassis assembly:subAssembly ?assembly .
    ?assembly pf:splitURI ( ?ns ?local ) .
    ?assembly a ?assemblyActualType .
    ?assemblyType tops:constructName( "genlunar:LunarRoverAssembly" ) .
    ?assemblyInst tops:constructName ( "genlunar:Assembly-{0}" ?local ) .
    ?assemblyQName tops:constructString ( "lunar-rover:{0}" ?local ) .
    ?assemblyActualTypeQName tops:qname ?assemblyActualType
}
```

Transformation and Mediation Using Semantic Technology (2)

SPARQLMotion Scripting Language is used for generation NExIOM-compliant XML from OWL Models

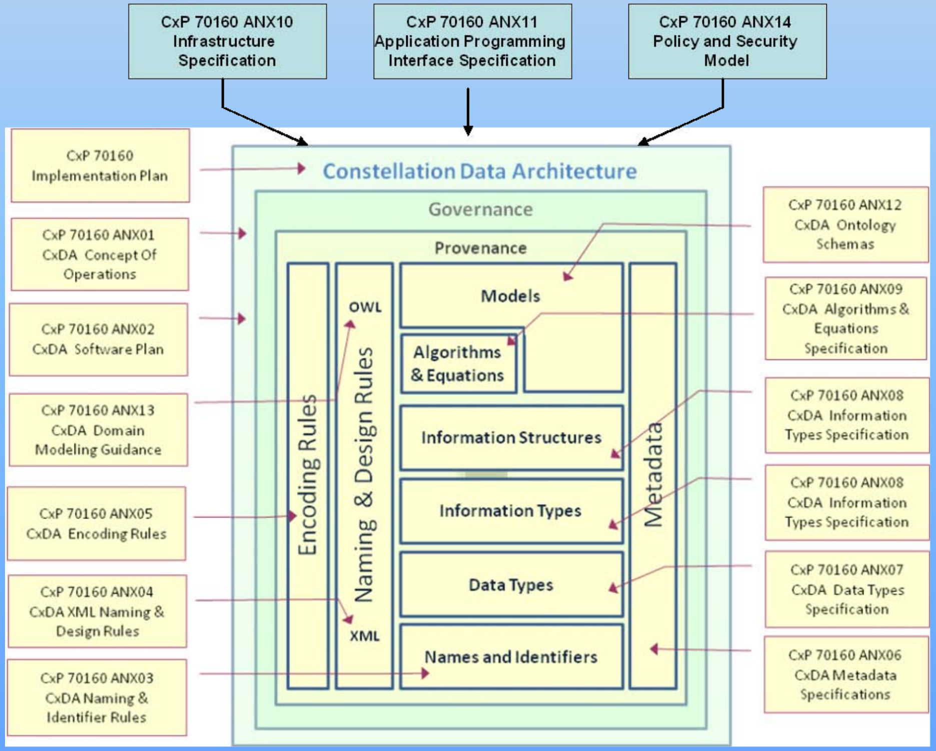


Note: the notation used in this diagram (from TopBraidComposer).

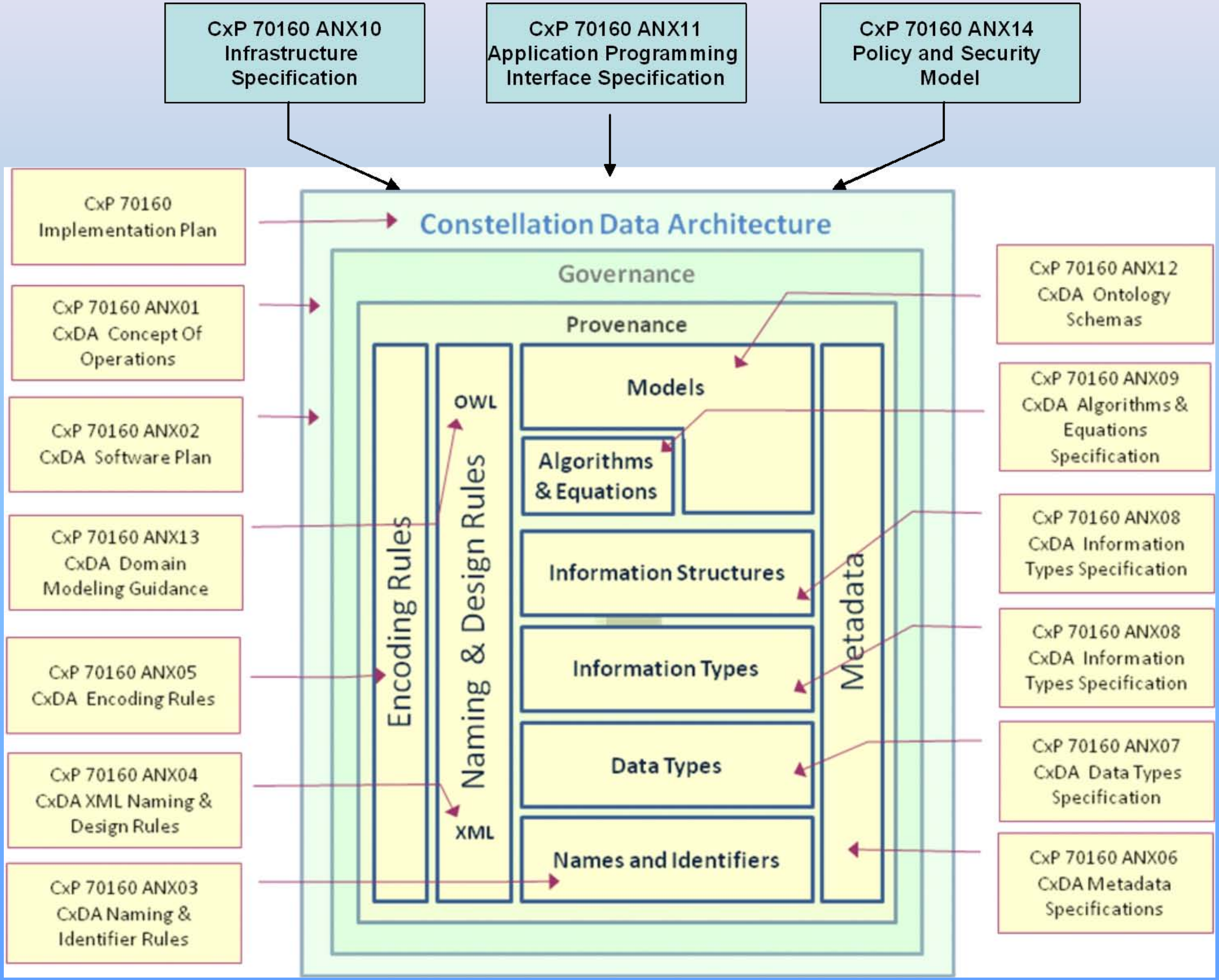
Key Motivations for Ontology-Driven Applications

- **Models need precise specification of their attributes and relationships**
  - Organized by Structure, Behavior, Function, Interaction (SBFI)
  - Standard Units, Datatypes, Enumerations, Physical Properties
  - Common Naming and Identifier Rules
- **Semantic Web Technologies provide precise semantics for modeling**
  - URIs enable modularity and composition
  - RDF models relationships
  - OWL provides semantics for class and type models
- **Semantic Web Technologies must co-exist with XML Technology**
  - “There is a lot of XML practices out-there”

Constellation Data Architecture - CxDA



NASA CxDA establishes a framework for names, identifiers, data and information types for consistency and interoperability

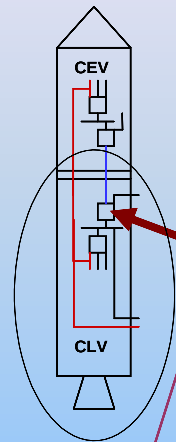


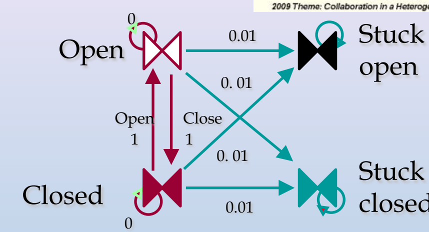
CxDA, based on the NExIOM Ontologies, needs to address multiple levels of the Constellation Systems

Requirement

To support making CxP data and systems operable for the long term, adapt to changes, usable in a variety of contexts, provide a method of connecting all data, requires a neutral data model

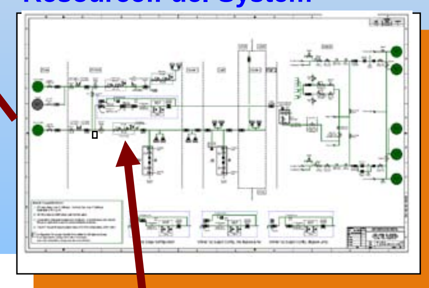
Constellation System: CLV

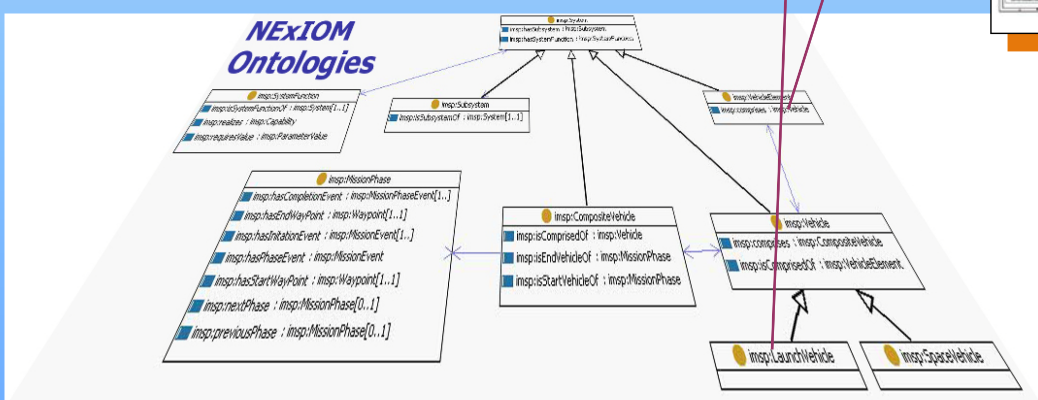




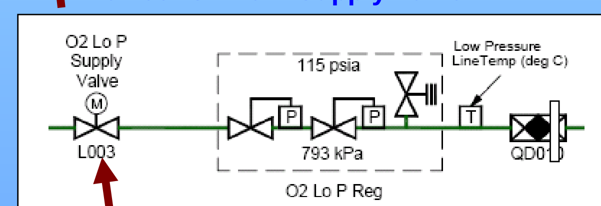
Sub-System: Propulsion

Resource:Fuel System





Device: O2 Lo P Supply Valve



Parameter: Valve State

Parameter	Type	Value	Units
Flow Rate	Real	258.75	l/sec
Valve State	ValvePosn	Open	None

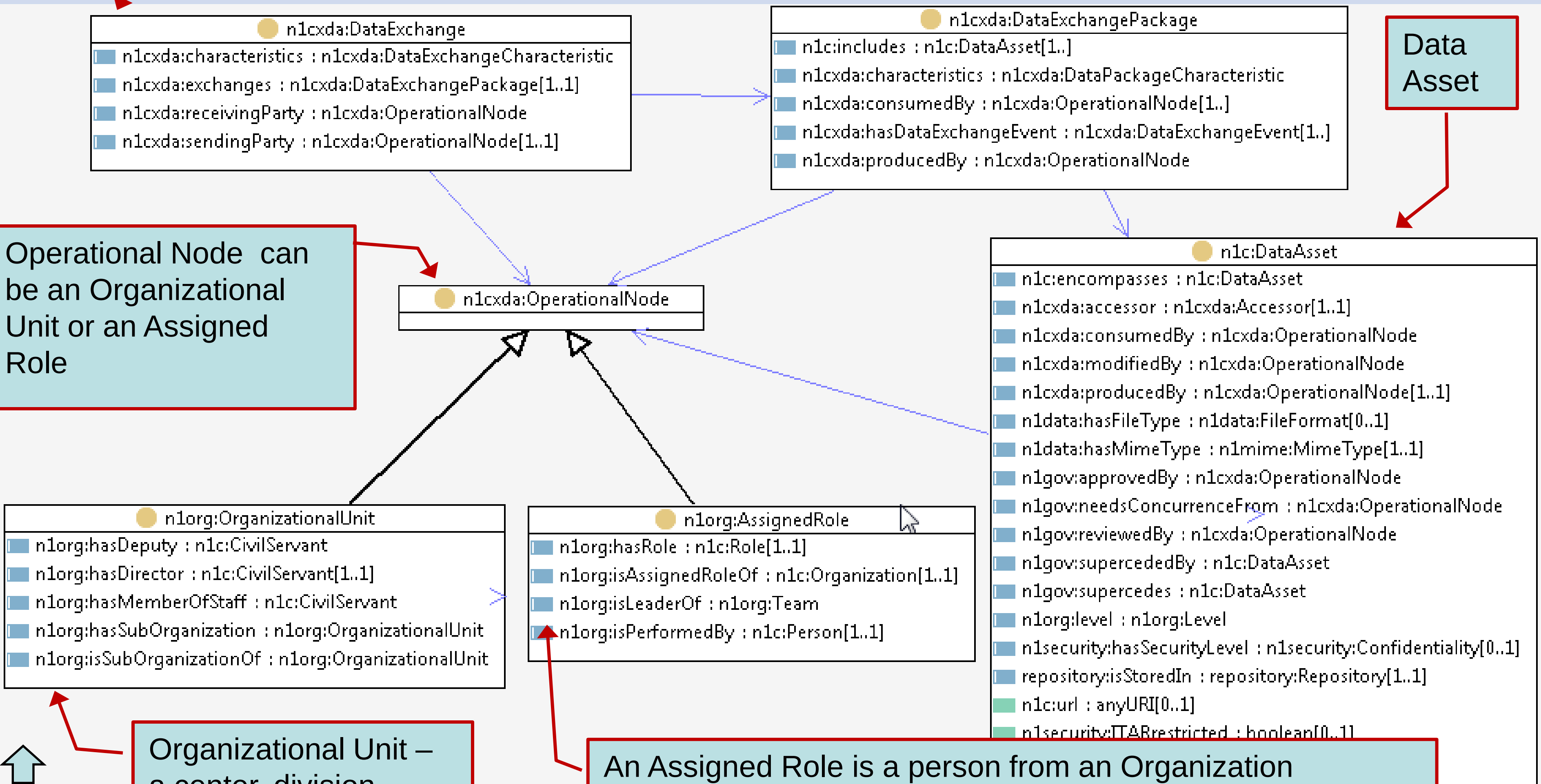
Data Exchanges occur between "DoDAF-like" operational nodes – Organizational Units and Assigned Roles

Data Exchange has characteristic properties and sending and receiving parties

Data Exchange Package has producers and consumers

Data Asset

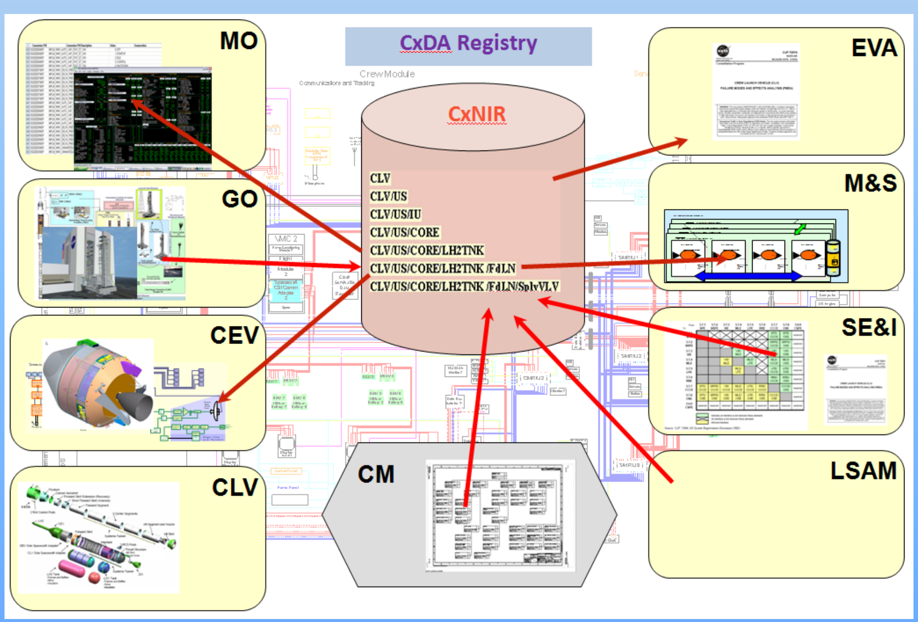
Operational Node can be an Organizational Unit or an Assigned Role



Organizational Unit – a center, division, office, etc.

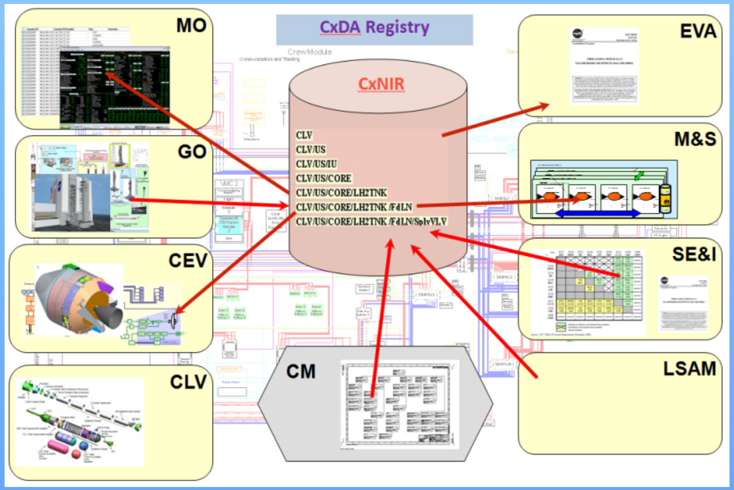
An Assigned Role is a person from an Organization performing a discipline (role)

System of Registries

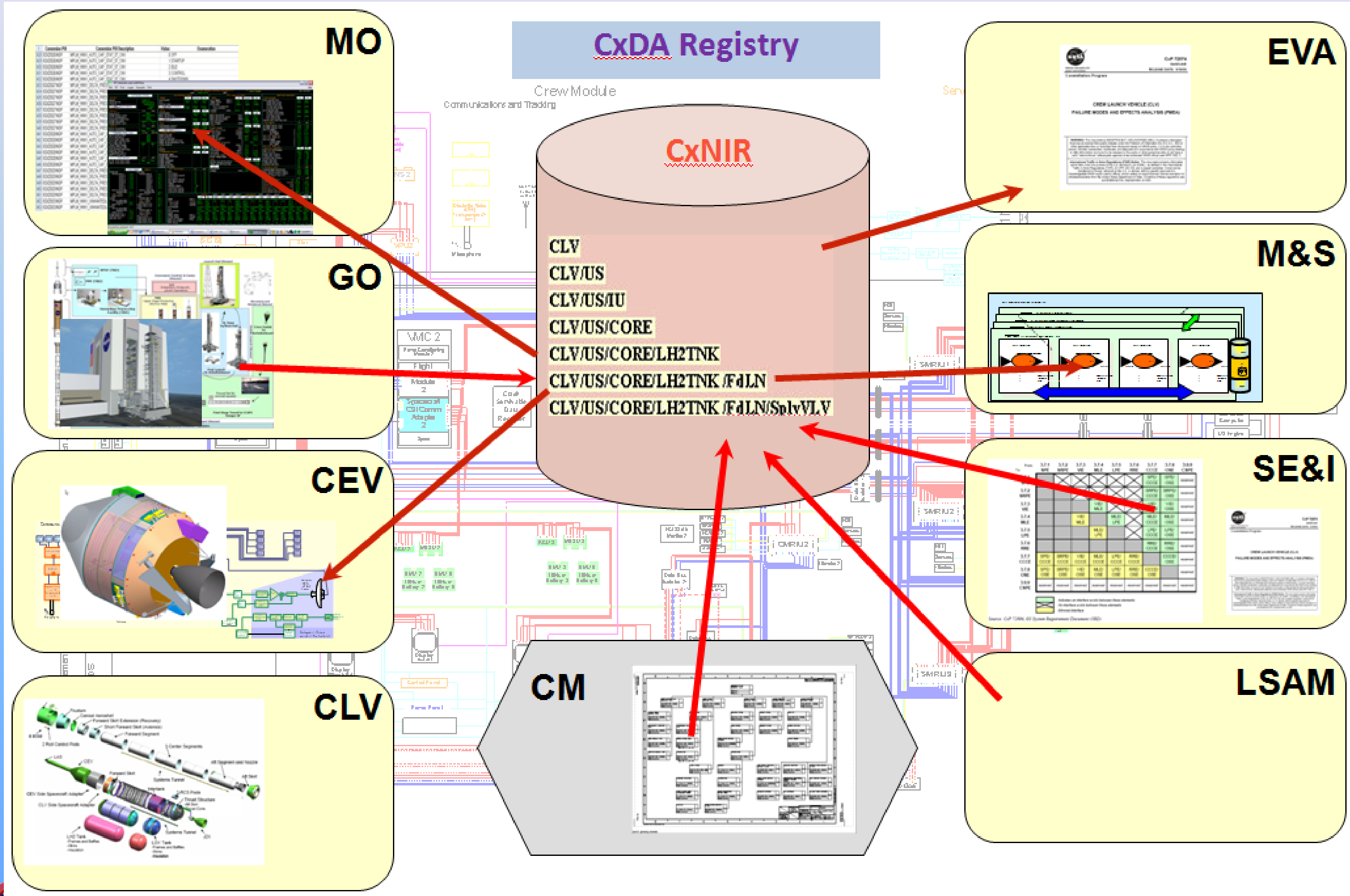


The NASA CxDA System of Registries (SoR)

- **Provide consistent definitions of data**
  - across time, between organizations, between processes.
- **Connect "silos" of information**
  - captured within applications or proprietary file formats, through the use of standardized data definitions
- **Support the exchange of information**
  - Using formats and protocols - XML and Web Services



CxDA System of Registries (SoR)



C3I - Telemetry and Commands

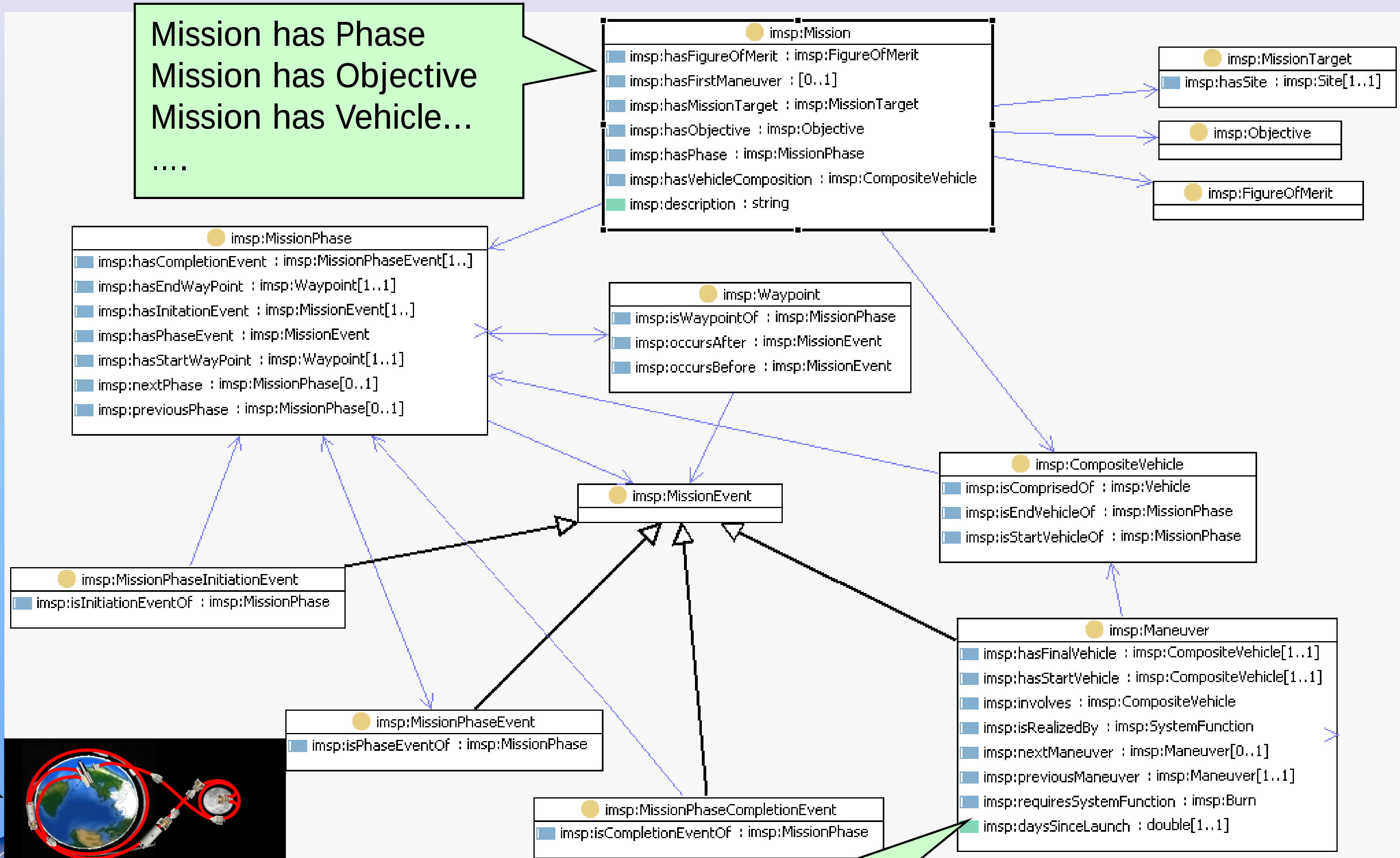
C3I Ontologies

- **Telemetry Parameters and Commands**
  - Packet Definitions
  - Command Definitions
- **Wider Context for Configuration and Re-Configuration**
  - Relating parameters to
    - System Properties
    - Measurement Specifications
    - Sensor Specifications
    - Calibration
    - Alarms
    - Criticality with respect to Mission Phases and Maneuvers
- **Communications**
  - Links
  - Association of Links with Telemetry

C3I Ontologies generate specifications and XML Schemas for Metadata and C3I Workproducts

NExIOM C3I Ontologies relate System Parameters and Commands to Mission Phases and Maneuvers

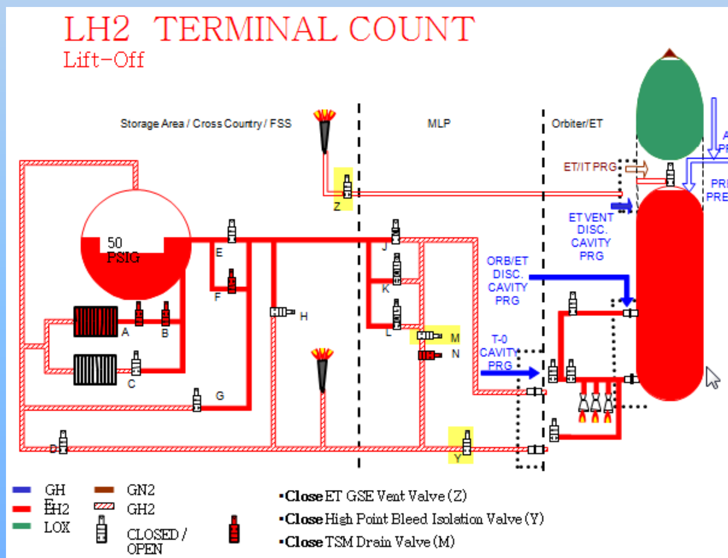
Mission has Phase
Mission has Objective
Mission has Vehicle…
….

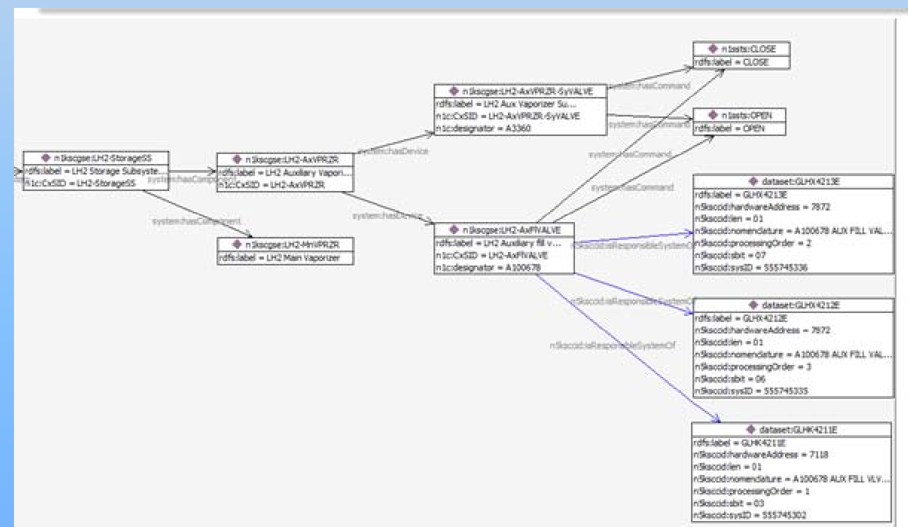


Maneuver requires Burn

KSC Launch Control System

Ontologies model and locate Devices in their Functional Hierarchies for checking out of launch sequence operations





Quantities and Units in the NExIOM Ontology

- **References**
  - "CODATA Recommended Values of the Fundamental Physical Constants: 2006" . Committee on Data for Science and Technology (CODATA).
  - "International System of Units (SI), 8th Edition". Bureau International des Poids et Mesures (BIPM).
  - "NIST Reference on Constants, Units, and Uncertainty". National Institute of Standards and Technology (NIST).
  - ESA Work on QUD - Quantities, Units, Dimensions

Motivation for the OWL ontology of physical quantities and units of measure is to satisfy the following requirements:

- **The ontology should support interoperability**
  - between disparate stakeholders using quantities and units
  - by providing controlled vocabularies and
  - through mutually agreed definitions of shared concepts.
- **The ontology should expose enough structure about the quantities and units**
  - to support conversion between commensurate units and
  - to perform dimensional analysis on the products and quotients of dimensional quantities.

The Units ontologies use a model based on dimensions and quantities. This model is being aligned with ESA QUD work.

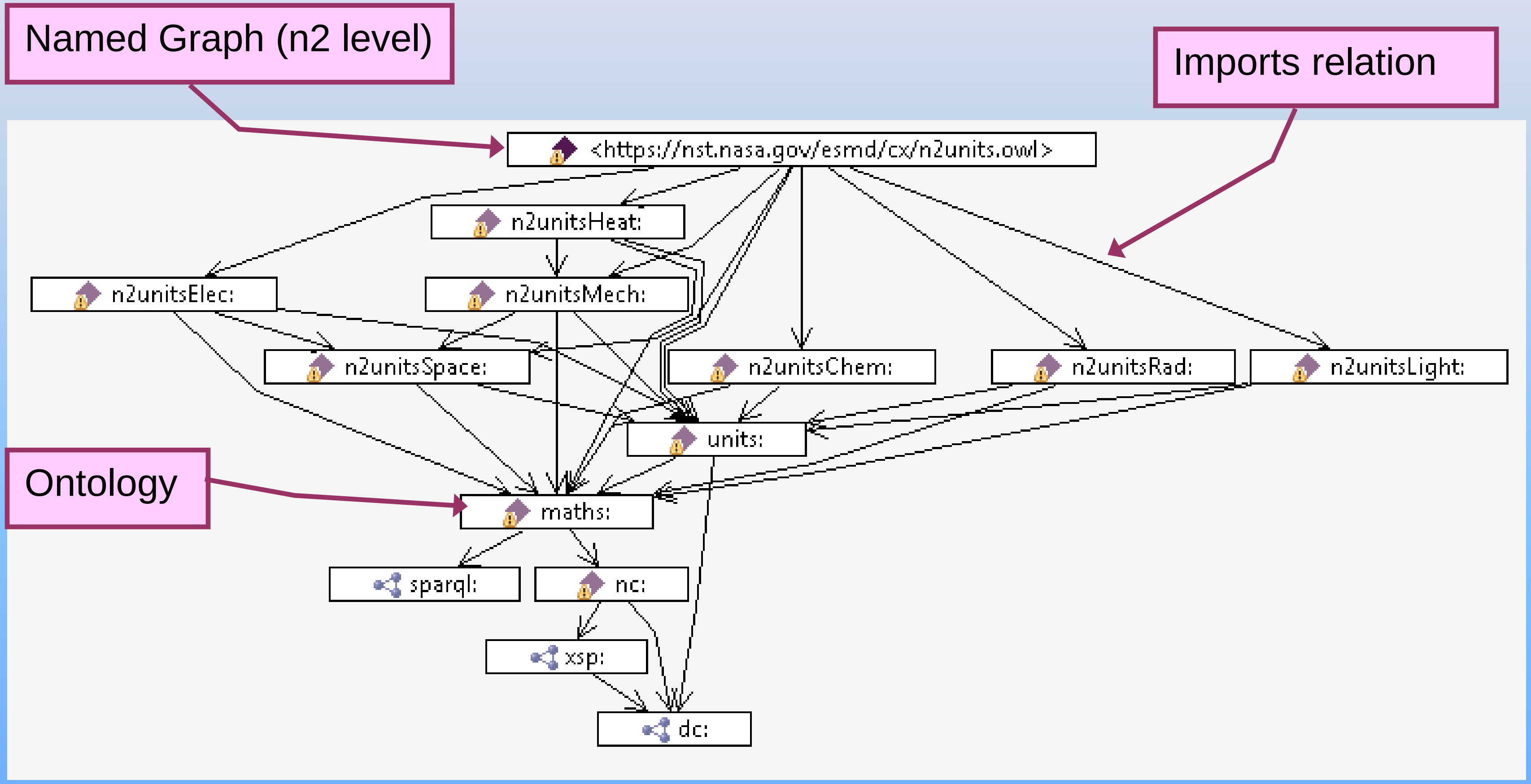
NExIOM Standard Vocabulary (NSV)

Basic physical quantities, forces & moments examples

Data-Name Identifier	Description	Definition	Symbol (Units)	Units
Potential	Potential	$\nabla\varphi = q$	L^2/T	SI
StreamFunction	Stream function (2-D)	$\nabla \times \psi = q$	L^2/T	SI
Density	Static density	(ρ)	M/L^3	SI
Pressure	Static pressure	(p)	$M/(LT^2)$	SI
Temperature	Static temperature	(T)	Θ	SI
EnergyInternal	Static internal energy per unit mass	(e)	L^2/T^2	SI
Enthalpy	Static enthalpy per unit mass	(h)	L^2/T^2	SI
Entropy	Entropy	(s)	$ML^2/(T^2\Theta)$	SI
EntropyApprox	Approximate entropy	$(s_{app} = p / \rho^{\gamma})$	$(L^{(3\gamma-1)})/((M^{(\gamma-1)}).T^2)$	SI
DensityStagnation	Stagnation density	(ρ_0)	M/L^3	SI
PressureStagnation	Stagnation pressure	(p_0)	$M/(LT^2)$	SI
TemperatureStagnation	Stagnation temperature	(T_0)	Θ	SI
EnergyStagnation	Stagnation energy per unit mass	(e_0)	L^2/T^2	SI
EnthalpyStagnation	Stagnation enthalpy per unit mass	(h_0)	L^2/T^2	SI
EnergyStagnationDensity	Stagnation energy per unit volume	(ρe_0)	$M/(LT^2)$	SI
VelocityX	x-component of velocity	$(u = q \cdot \boldsymbol{e}_x)$	L/T	SI
VelocityY	y-component of velocity	$(v = q . \boldsymbol{e}_y)$	L/T	SI
VelocityZ	z-component of velocity	$(w = q . \boldsymbol{e}_z)$	L/T	SI
VelocityR	Radial velocity component	$(q . \boldsymbol{e}_r)$	L/T	SI

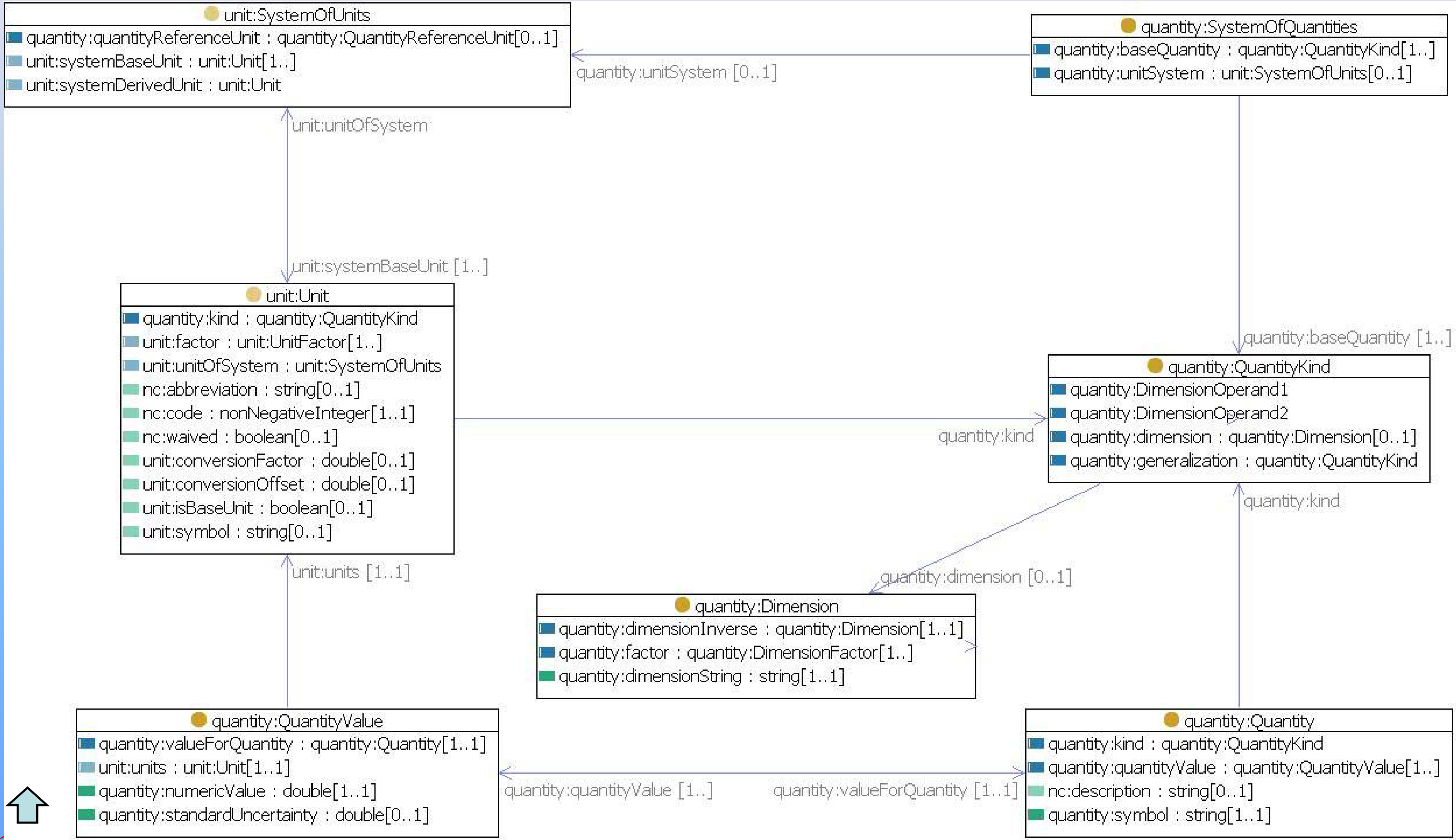
Data-Name Identifier	Description	Units
ForceX	$F_x = \boldsymbol{F} \cdot \boldsymbol{e}_x$	$\mathbf{ML/T^2}$
ForceY	$F_y = \boldsymbol{F} \cdot \boldsymbol{e}_y$	$\mathbf{ML/T^2}$
ForceZ	$F_z = \boldsymbol{F} \cdot \boldsymbol{e}_z$	$\mathbf{ML/T^2}$
ForceR	$F_r = \boldsymbol{F} \cdot \boldsymbol{e}_r$	$\mathbf{ML/T^2}$
ForceTheta	$F_\theta = \boldsymbol{F} \cdot \boldsymbol{e}_\theta$	$\mathbf{ML/T^2}$
ForcePhi	$F_\varphi = \boldsymbol{F} \cdot \boldsymbol{e}_\varphi$	$\mathbf{ML/T^2}$
Lift	L or L'	$\mathbf{ML/T^2}$
Drag	D or D'	$\mathbf{ML/T^2}$
MomentX	$M_x = \boldsymbol{M} \cdot \boldsymbol{e}_x$	$\mathbf{ML^2/T}$
MomentY	$M_y = \boldsymbol{M} \cdot \boldsymbol{e}_y$	$\mathbf{ML^2/T}$
MomentZ	$M_z = \boldsymbol{M} \cdot \boldsymbol{e}_z$	$\mathbf{ML^2/T}$
MomentR	$M_r = \boldsymbol{M} \cdot \boldsymbol{e}_r$	$\mathbf{ML^2/T}$
MomentTheta	$M_\theta = \boldsymbol{M} \cdot \boldsymbol{e}_\theta$	$\mathbf{ML^2/T}$
MomentPhi	$M_\varphi = \boldsymbol{M} \cdot \boldsymbol{e}_\varphi$	$\mathbf{ML^2/T}$
MomentXi	$M_\xi = \boldsymbol{M} \cdot \boldsymbol{e}_\xi$	$\mathbf{ML^2/T}$
MomentEta	$M_\eta = \boldsymbol{M} \cdot \boldsymbol{e}_\eta$	$\mathbf{ML^2/T}$
MomentZeta	$M_\zeta = \boldsymbol{r} \cdot \boldsymbol{e}_\zeta$	$\mathbf{ML^2/T}$
Moment_CenterX	$x_0 = \boldsymbol{r}_0 \cdot \boldsymbol{e}_x$	**L**
Moment_CenterY	$y_0 = \boldsymbol{r}_0 \cdot \boldsymbol{e}_y$	**L**
Moment_CenterZ	$z_0 = \boldsymbol{r}_0 \cdot \boldsymbol{e}_z$	**L**

Units Ontology Architecture



The Units ontologies use a model based on dimensions and quantities. This model is being aligned with ESA QUD work.

Diagram of Quantity Classes

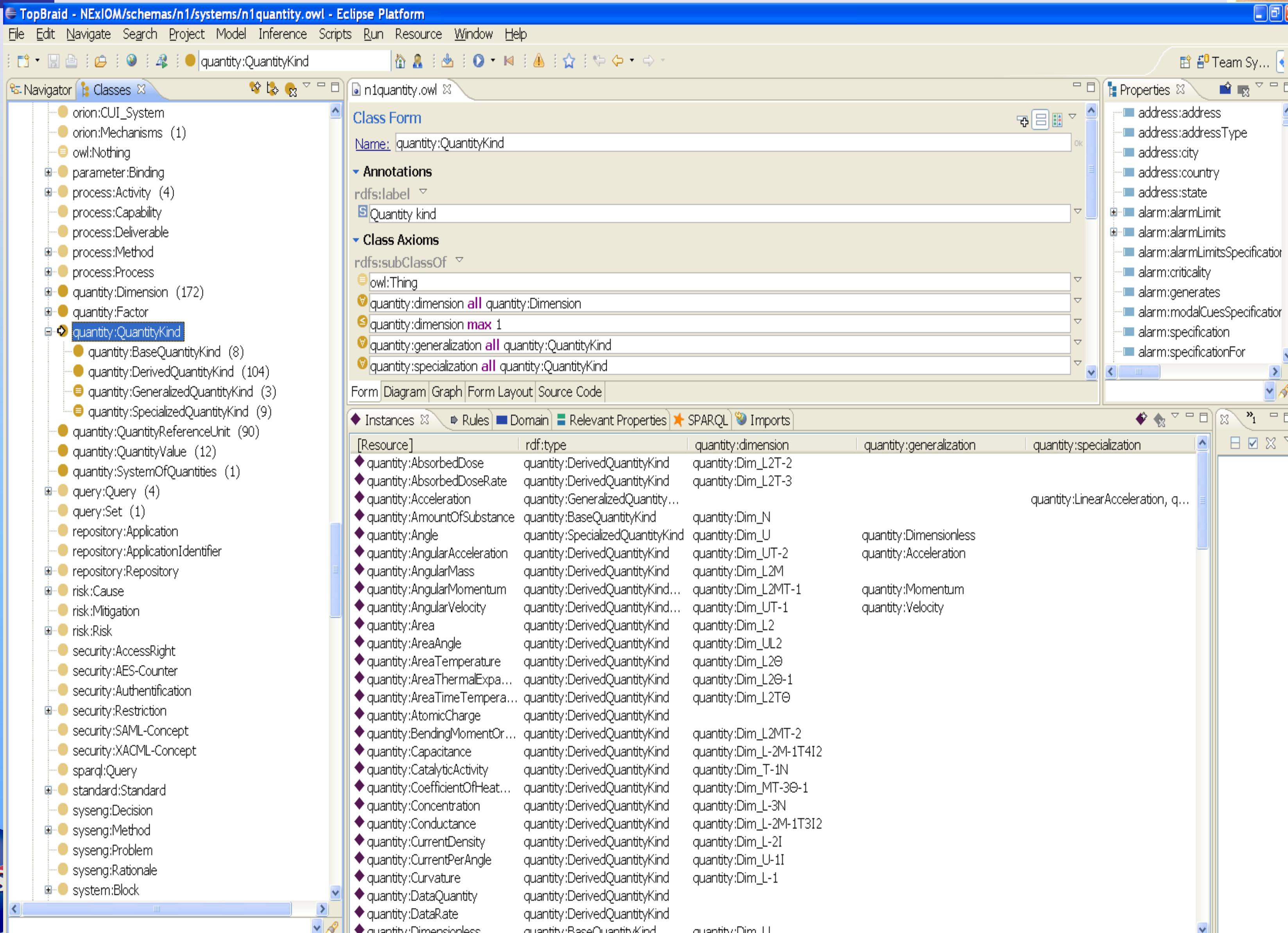


Kinds of Physical Quantities

- A *Quantity Kind* characterizes the physical nature or type of a measured quantity . E.g. length, mass, time, force, power, energy, etc.
- Typically, a small set of quantity kinds is chosen to be the *Base Quantity Kinds*. Other quantity kinds are defined in terms of the base set using physical laws or definitions. The latter are called *Derived Quantity Kinds*.
- A *System of Quantities* is a specification, typically developed and maintained by an authoritative source, that establishes:
  - Choice of the base quantity kinds for the system;
  - The formulas expressing each derived quantity kind in the system in terms of the base quantity kinds:
    - Force = Mass * Acceleration
    - Velocity = Length / Time
    - Electric Charge = Electric Current * Time
  - Example: International System of Quantities is the system of quantities used with the International System of Units. The ISQ is defined in ISO/IEC80000.

Quantity Kinds and their Dimensions (II)

PDE 2009
The 11th NASA-ESA Workshop on Product Data Exchange
Kent Space Center, 29 April - 1 May 2009



Dimensions and Dimensional Analysis

- **Dimensions are used to characterize quantities in terms of their dependence on a chosen set of base quantity kinds. The dimension of each base quantity kind is represented by its dimension symbol:**
  - SI Dimensions: Length (L), Mass (M), Time (T), Current (I), Temperature (Θ), Amount of Substance (N), Luminous Intensity (J).
- **The dimension of any quantity can be expressed as a product of the base dimension symbols raised to a rational power. For example, velocity can be expressed as length divided by time:**
  - V = L/T = L^1T^(-1)
  - Thus, velocity has the dimensions L^1T^(-1).
- **Dimensional Analysis:**
  - Only quantities with the same dimensions may be compared, equated, added, or subtracted.*
  - Quantities of any dimension can be multiplied or divided. The dimensionality of the resultant is determined by analyzing the product or quotient of the operands.**

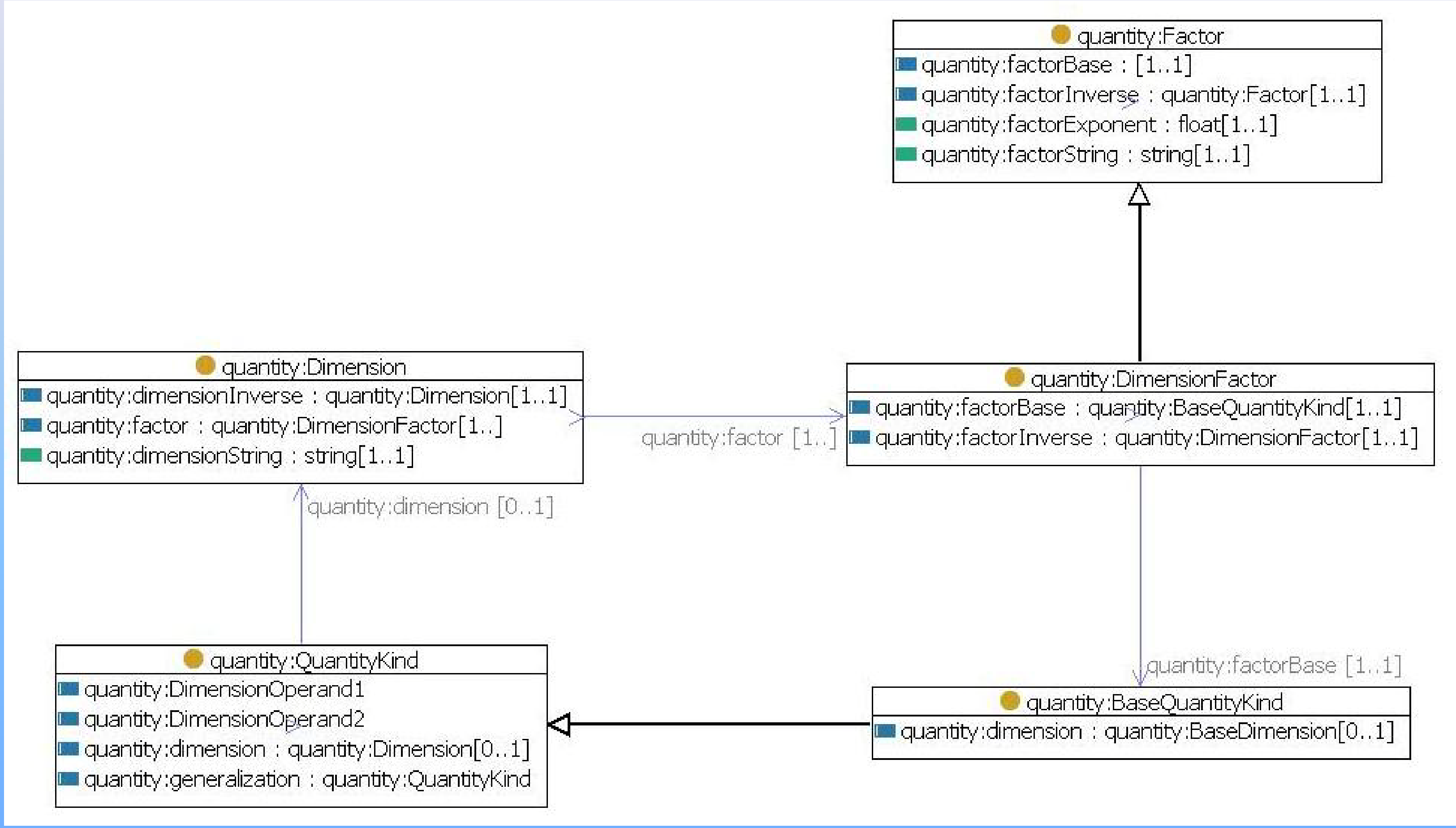
Units of Measure

- **A unit of measure establishes a reference scale for a quantity's dimension.**
- **System of Units is a choice of base units and derived units, together with their multiples and submultiples, defined in accordance with given rules, for a given system of quantities.**
  - Base units: Units corresponding to the base quantities in a system of quantities.
    - SI Base Units: Metre (Length), Kilogram (Mass), Second (Time), Ampere (Electric Current), Kelvin (Thermodynamic Temperature), Mole (Amount of Substance), Candela (Luminous Intensity)
  - Derived units: Units corresponding to the derived quantities in a system of quantities.
  - Coherent units: When coherent units are used, equations between the numerical values of quantities take exactly the same form as the equations between their corresponding quantity kinds. Thus if only units from a coherent set are used, conversion factors between units are never required.

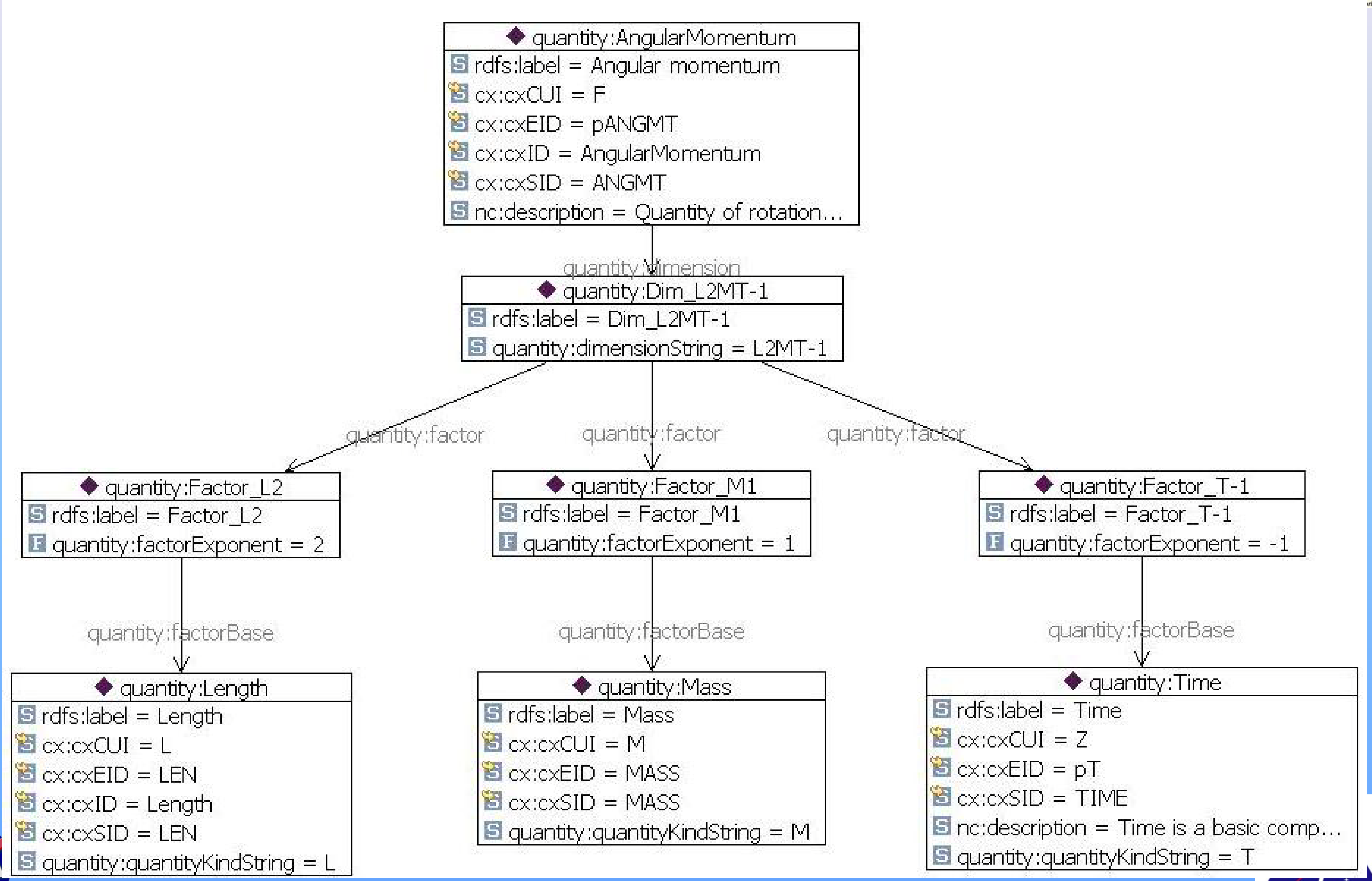
Formal Model of Quantities and Units in OWL

- **Quantities, Units and Values: These are the main classes for describing quantities and their values.**
  - quantity:Quantity
  - quantity:QuantityValue
  - unit:Unit
- **Quantity Structure: These are the main classes that characterize the physical properties of quantities and determine the commensurability between quantities (Dimensional analysis).**
  - quantity:QuantityKind
  - quantity:Dimension
- **Systems: These are the main classes used to describe existing agreements and standards establishing systems of quantities and units.**
  - quantity:SystemOfQuantities
  - unit:SystemOfUnits

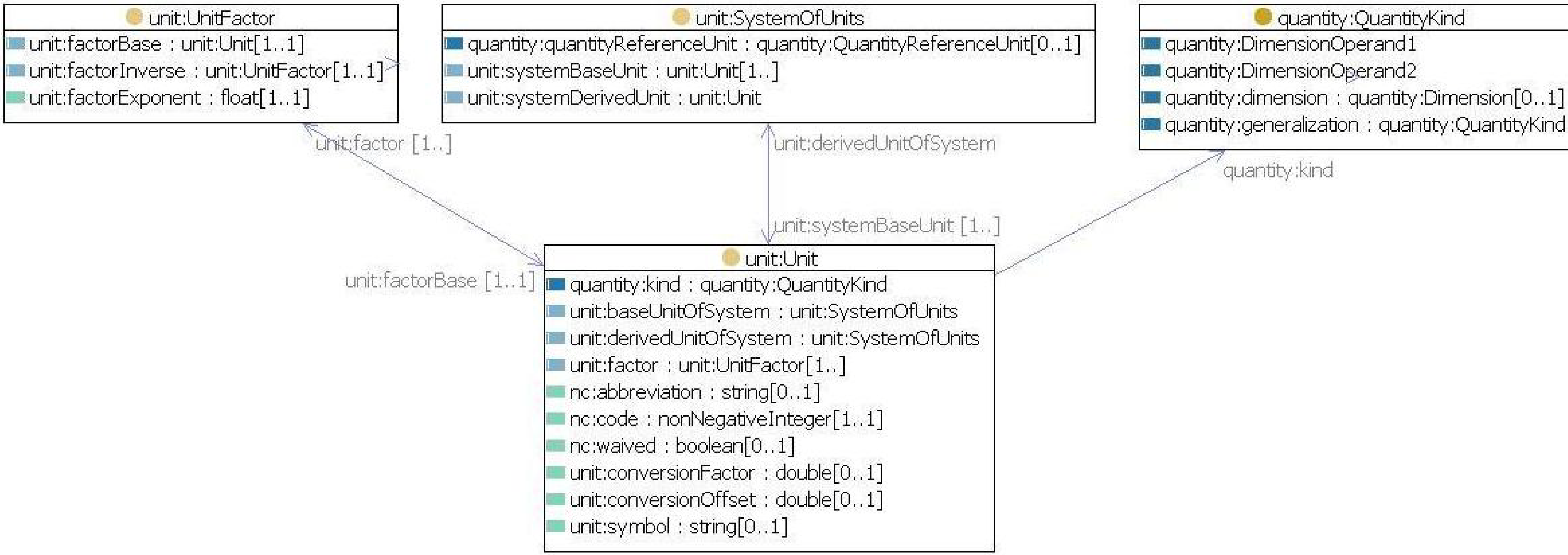
Expressing Dimensions as Products of Other Dimensions



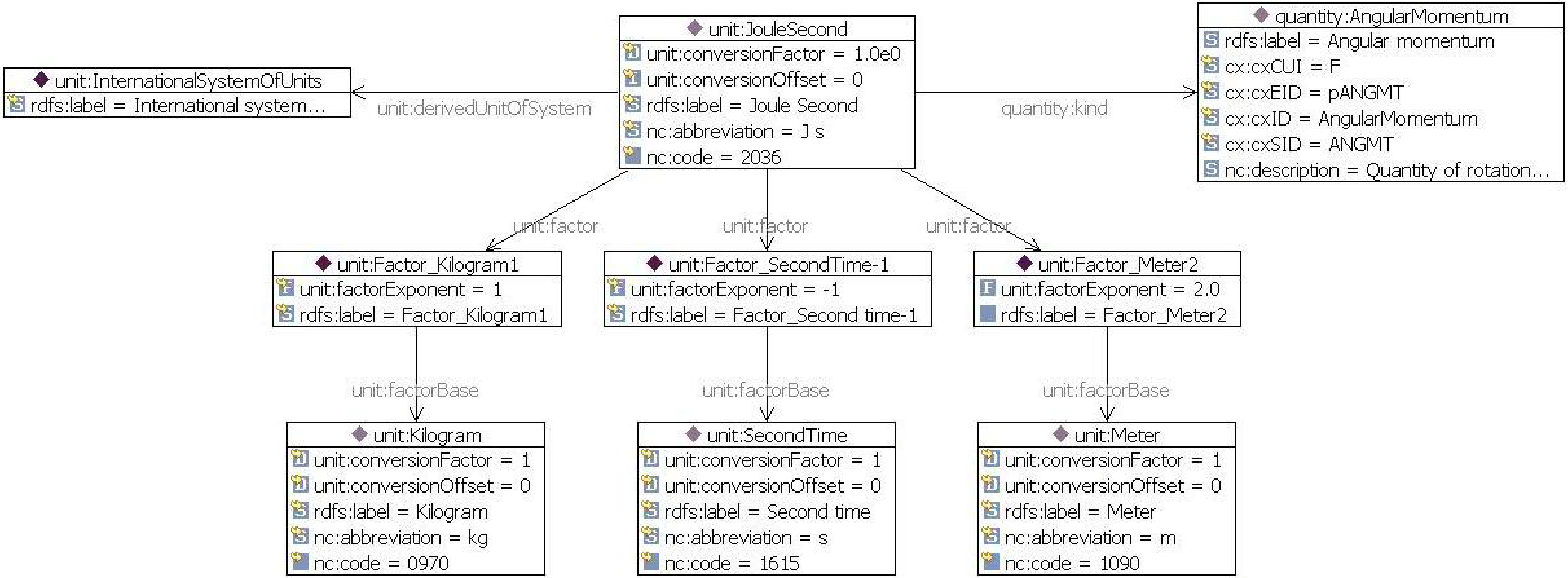
Example of Dimension Factors: Angular Momentum



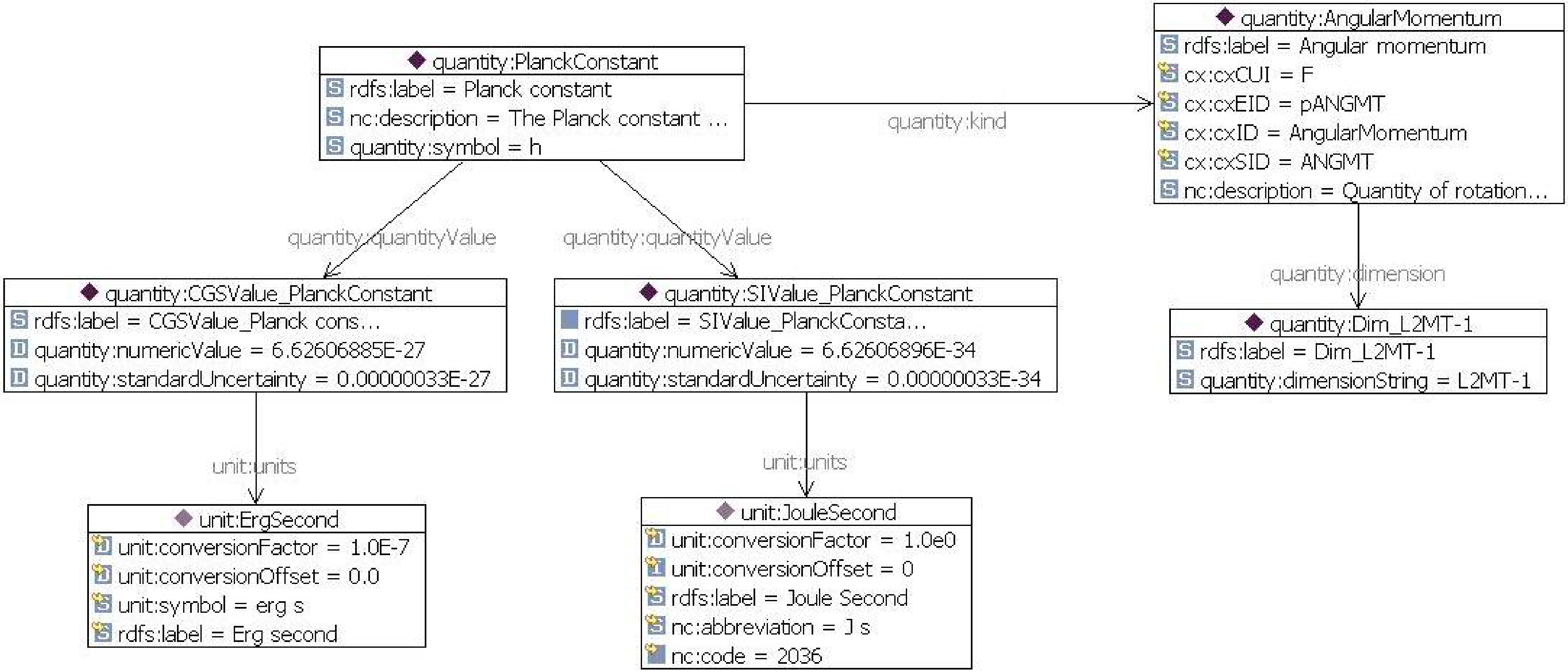
Expressing Units as Products of Base Units in a System of Units



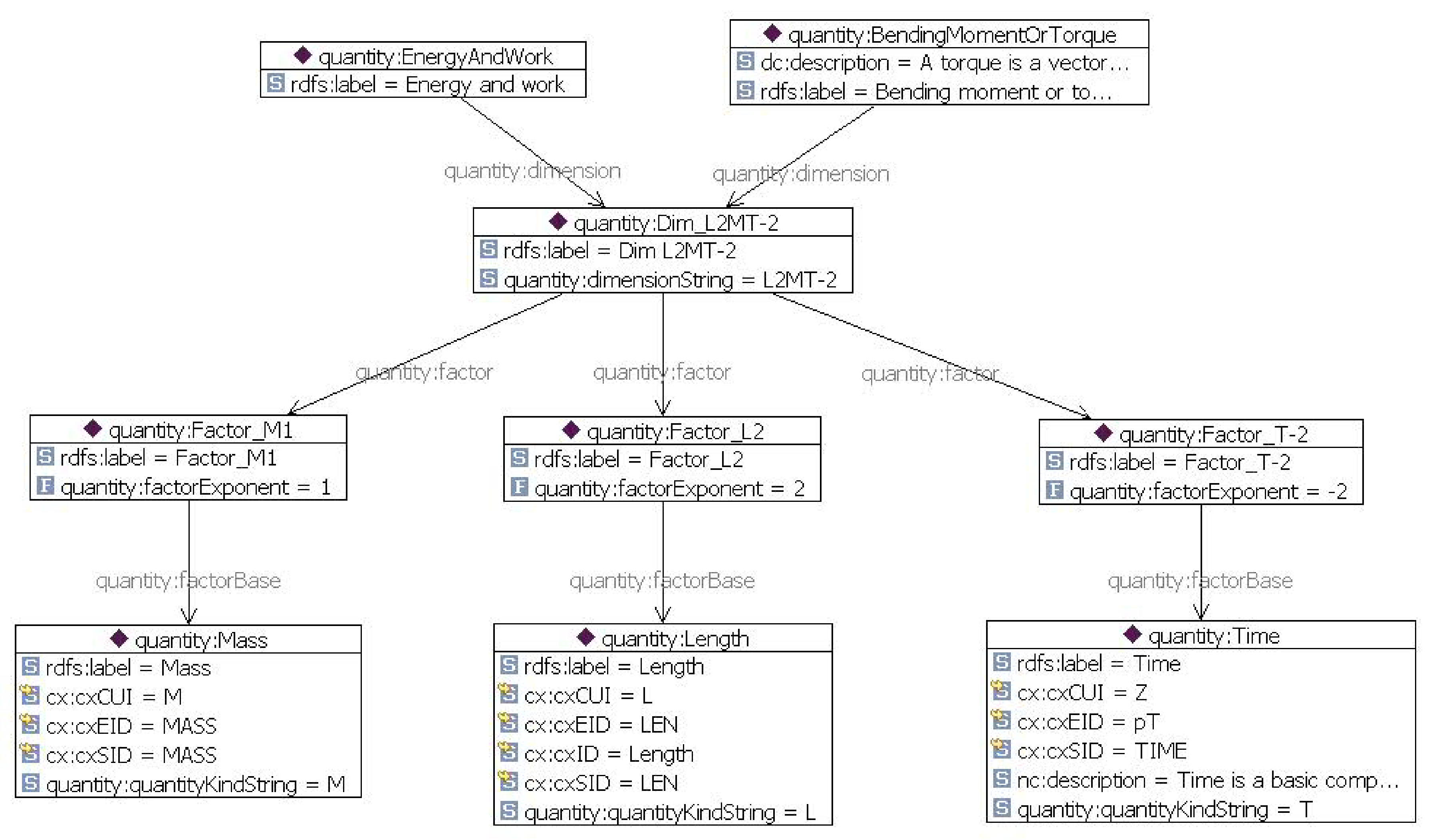
Example of Unit Factors: JouleSeconds



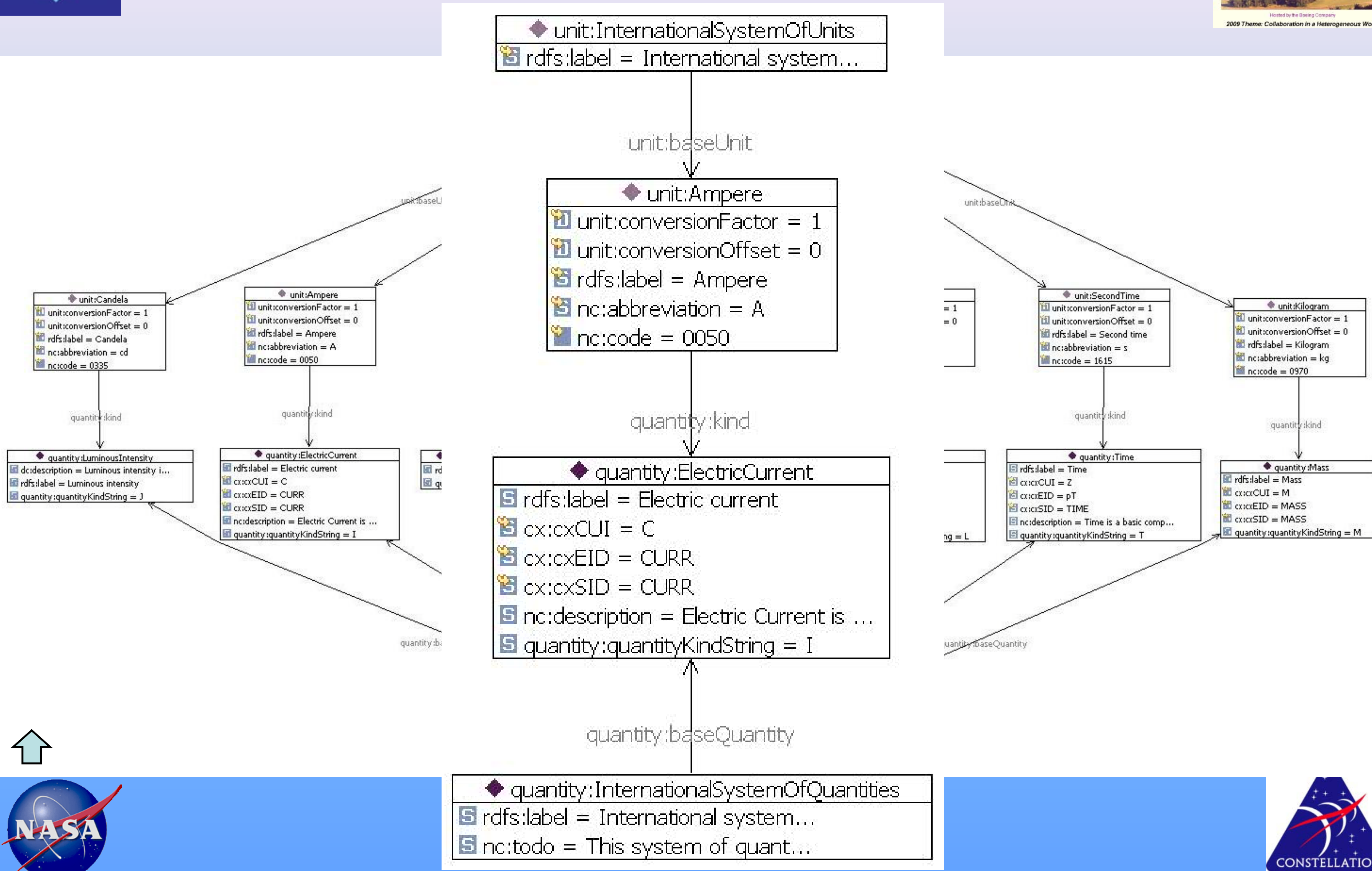
Example Quantity: Planck's Constant



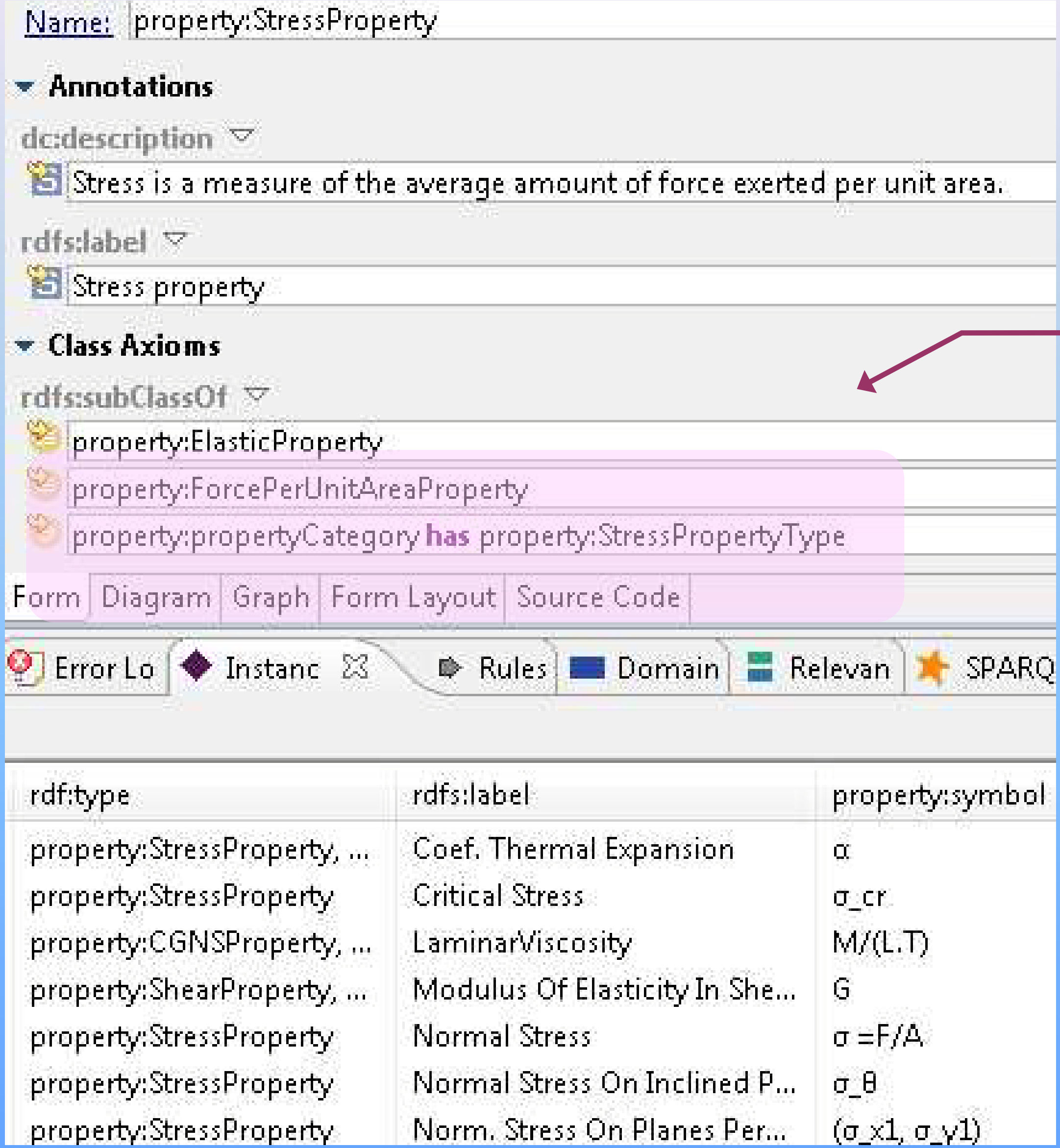
Shared Dimensions: Torque and Energy



SI Base Quantities and Units



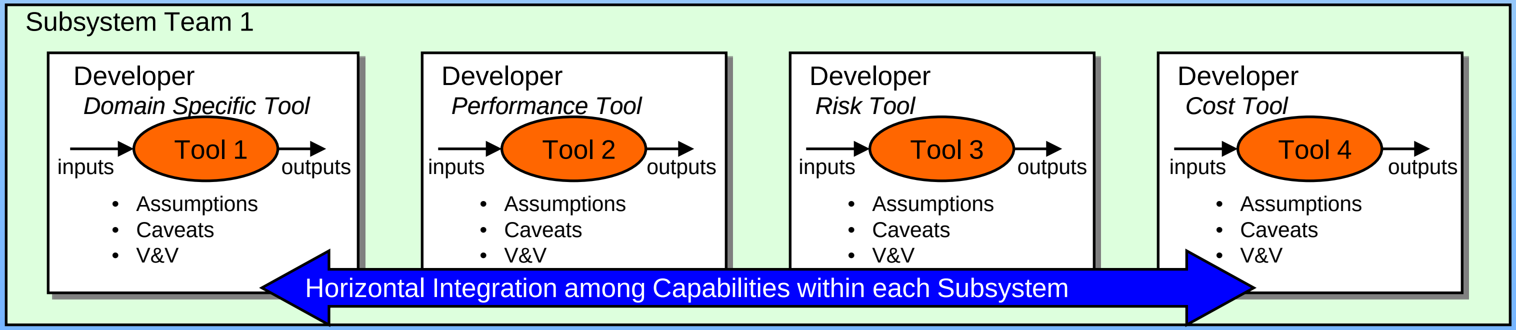
Properties can be of multiple types



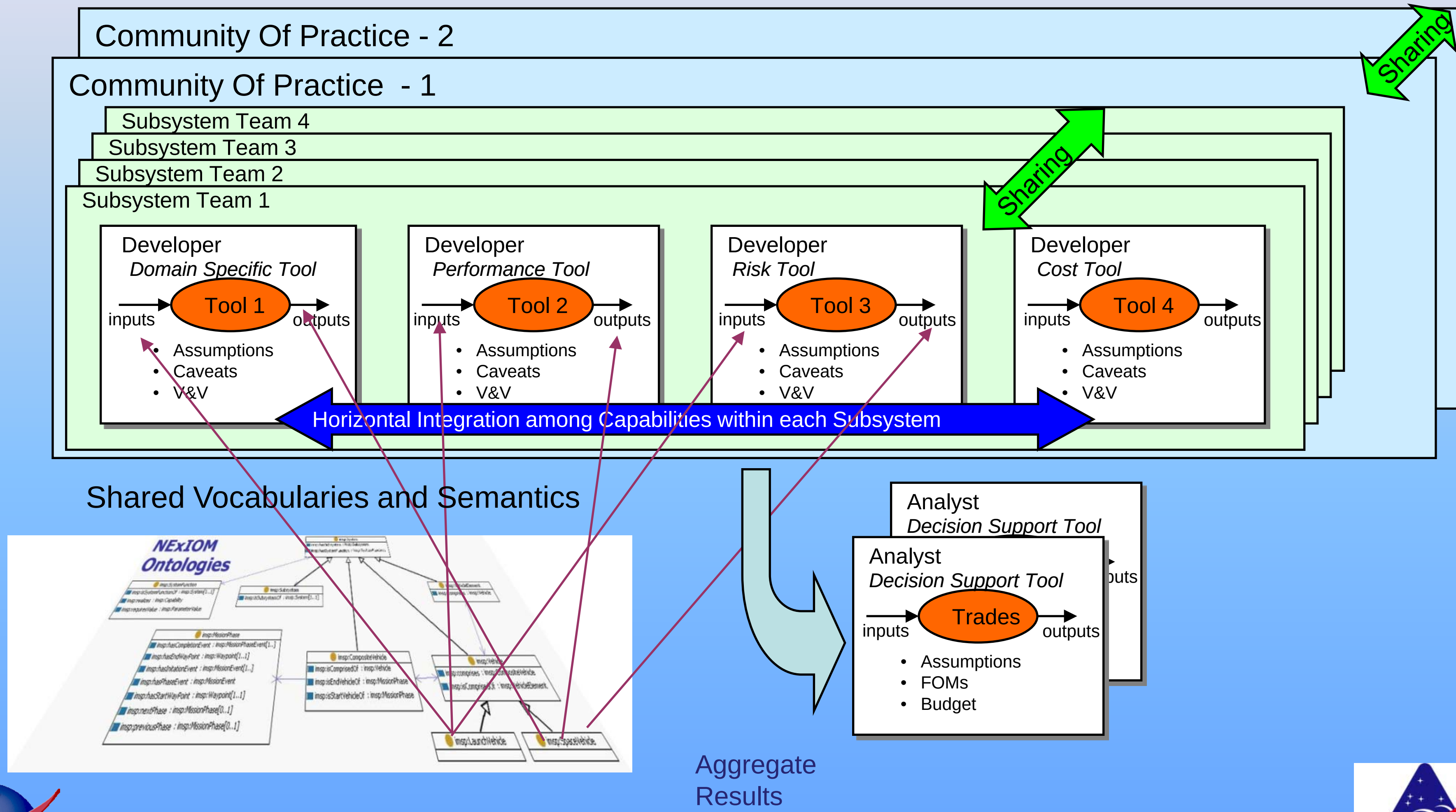
Emphasis of multiple types

NExIOM Standard Vocabularies

Tool Interoperability



Tool interoperatility through OWL-compliant XML schemas and controlled vocabularies



OWL Model of a Tool

nasa:Tool

- data:pedigree : tool:ToolPedigree[0..1]
- gov:applicableDocument : nasa:InformationAsset
- gov:hasOwner : nasa:Organization[1..1]
- gov:informativeReference : nasa:InformationAsset
- nasa:comparableTo : nasa:Tool
- nasa:documentation : tool:ToolDocumentation
- nasa:hasApplicabilityTo : technology:Discipline or nasa:System
- nasa:hasAssumption : nasa:Assumption
- nasa:hasConstraint : nasa:Constraint
- nasa:isUsedBy : org:AssignedRole or nasa:CommunityOfPractice
- nasa:pointOfContact : org:AssignedRole or person:PointOfContact
- security:hasSecurityLevel : security:Confidentiality[0..1]
- tool:author : nasa:Person[1..]
- tool:hasAccuracy : tool:AccuracyUncertaintyMeasure
- tool:hasDevelopmentParty : nasa:Party[1..1]
- tool:hasDevelopmentPointOfContact : person:PointOfContact
- tool:hasVerificationAndValidationPlan : infoasset:VerificationSpecification[0..1]
- tool:isBasedOn : nasa:Model
- tool:isGroundedIn : tool:Paradigm
- tool:isInvokedBy : tool:ToolInvocation
- tool:method : tool:ComputationalMethod
- nasa:comments : string
- nasa:description : string[1..1]
- nasa:purpose : string[1..1]
- nasa:version : string[1..1]
- security:ITARRestricted : boolean[1..1]
- security:proprietary : boolean[1..1]

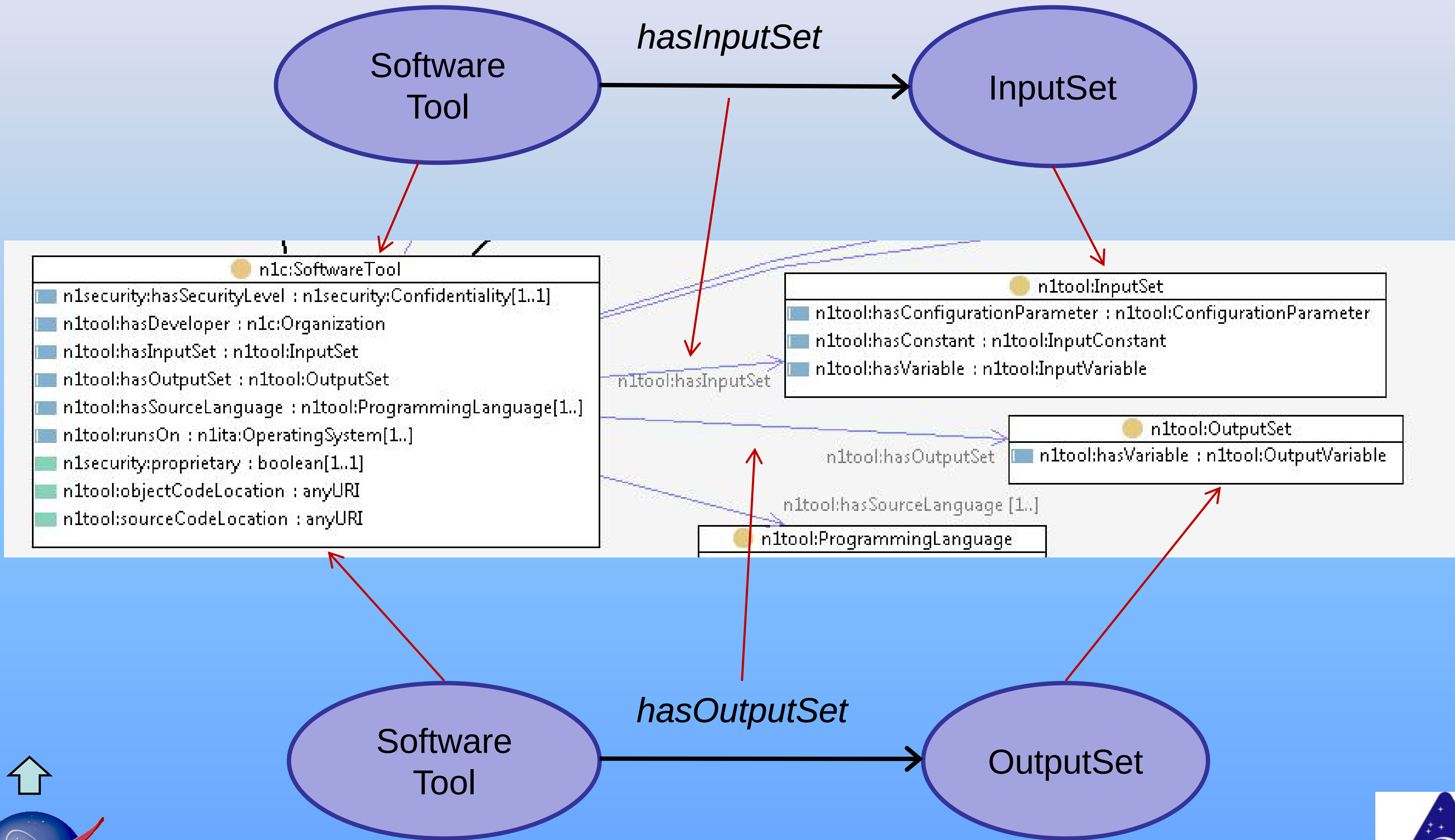
The class 'TOOL'

Object property 'isUsedBy' makes connection to 'AssignedRole' and/or 'CommunityOfPractices'

Object property for the tool's computational model

Data property attribute for 'ITAR Restriction'

A Software Tool has a Set of Inputs and a Set of Outputs



NExIOM Exchange Example: XML Model for a Software Tool and its Variables

The use of explicit IDs and REFs ensures reuse and makes the models more manageable

```
<tool:InputSet id="MyTOOL_IPSET">
  <tool:hasVariable IDREF="IV_MyTOOL_TITLE"/>
  <tool:hasVariable IDREF="IV_MyTOOL_IDCAL"/>
  <tool:hasVariable IDREF="IV_MyTOOL_REST"/>
  <tool:hasVariable IDREF="IV_MyTOOL_RESH"/>
  <tool:hasVariable IDREF="IV_MyTOOL_RADTMP"/>
  ….
  ….
   <tool:hasVariable IDREF="IV_MyTOOL_IABL"/>
 </tool:InputSet>
```

hasInputSet

has Variable

Tool

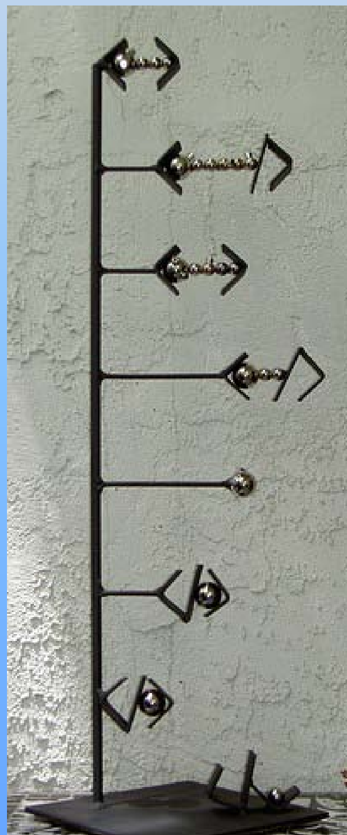
```
<tool:SoftwareTool id="MyTOOL">
   <nc:fullname> MyTOOL</nc:fullname>
   <nc:description> … </nc:description>
   <tool:hasInputSet IDREF="MyTOOL_IPSET"/>
 </tool:SoftwareTool>
```

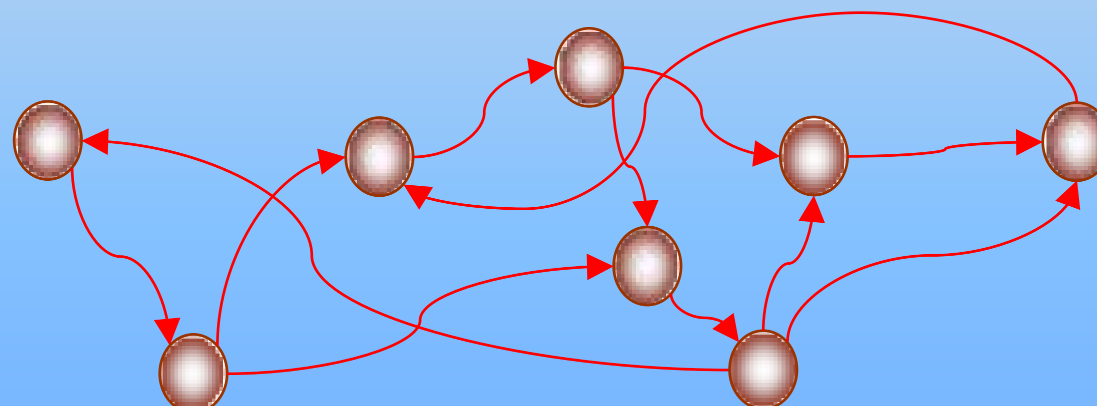
```
<tool:InputVariable   id="IV_MyTOOL_RADTMP"
   tool:represents="Param_MyTOOL_RADTMP"
   units:hasUnits="units:R">
   <nc:description>
```

has Units

```
<units:Unit nc:ID="units:Ampere" nc:abbreviation="A" nc:code="0050"
       rdf:type="units:Current,units:SiBase" rdfs:label="Ampere"/>
<units:Unit nc:ID="units:AmpereHour" nc:abbreviation="A-hr" nc:code="0055"
       rdf:type="units:Derived,units:ElectricCharge,units:NotUsedWithSi"
       rdfs:label="Ampere hour"/>
<units:Unit nc:ID="units:AmperePerDegree" nc:abbreviation="A/deg" nc:code="2280"
       rdf:type="units:CurrentPerAngle,units:IntermediateDerived,units:NonSi"
rdfs:label="Ampere per degree"/>
<units:Unit nc:ID="units:AmperePerMeter" nc:abbreviation="A/m" nc:code="0060"
       rdf:type="units:Derived,units:MagneticFieldStrength" rdfs:label="Ampere per meter"/>
```

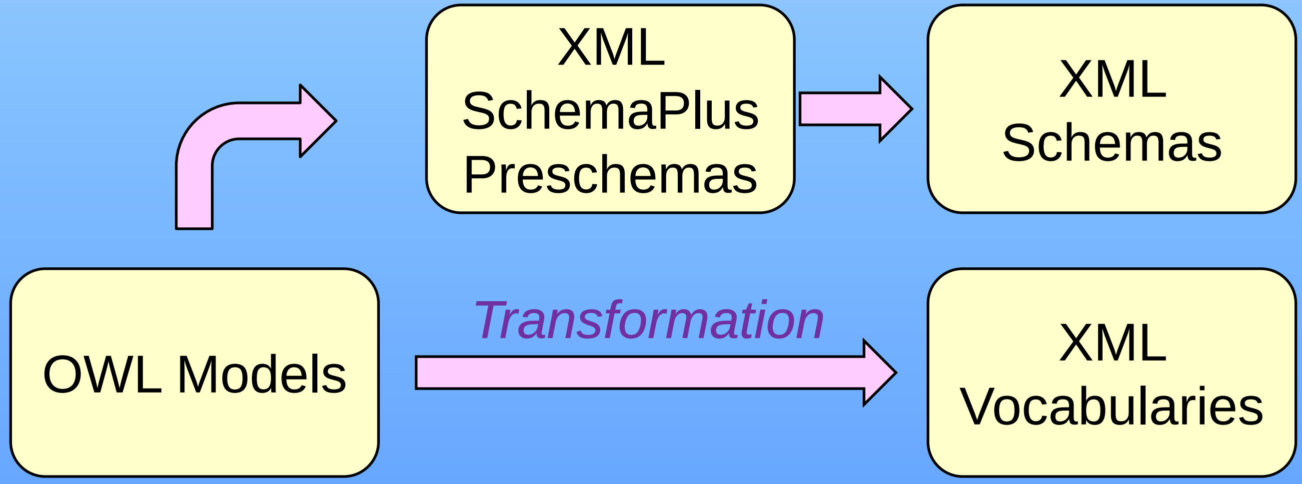
Co-existence between OWL and XML





XML SchemaPlus - semantic interoperability between the XML and OWL worlds

- **Problem: Ambiguous Semantics in XML**
  - XML documents are tree structures of nested elements
    - meaning of the nesting is typically not made explicit
  - XML attributes are commonly text strings with no semantics
    - meaning of the attribute is not explicit
- **Solution: NExIOM compliant XML**
  - XML SchemaPlus: the bridge between ontologies and XML
  - QNAMES for controlled vocabularies



coming soon: www.xspl.us

XML SchemaPlus – a language for specifying XML Document Structure

```
<?xml version="1.0" encoding="UTF-8"?>
<SchemaPlus xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="SchemaPlus.xsd">
   <RootElement name="TrainingExample"/>
   <ScalarElement name="SimulationTitle"></ScalarElement>
   <ReferenceElement name="Unit"></ReferenceElement>
   <NestedElement
      name="scenario“ type="ScenarioType">
   </NestedElement>
   <ObjectType name="ScenarioType">
      <Attribute name=“provenance”></Attribute>
      <CollectionElement
         name="simulationConfigurationParameter"
         type="SimulationConfigurationParameterType">
      </CollectionElement>
   </ObjectType>
</SchemaPlus>
```

Retaining OWL Semantics

Example of SchemaPlus: Sensor Specification

Sensor is a ‘NestedElement at the Root level

```
<NestedElement name="Sensor" type="system:SensorType"></NestedElement>
  <ObjectType name="SensorType" namespace="system" baseType="system:DeviceType">
    <Attribute ref="data:bitsPerSample" type="xs:int"></Attribute>
    <Attribute ref="data:bitsPerSecond" type="xs:int"></Attribute>
    <Attribute ref="data:samplePerSecond"></Attribute>
    <Attribute ref="device:accuracy"></Attribute>
    <Attribute ref="device:frequencyResponse"></Attribute>
    <Attribute ref="device:lowerRange"></Attribute>
    <Attribute ref="device:resolution"></Attribute>
    <Attribute ref="device:upperRange"></Attribute>
    <Attribute ref="system:io_device" type="xs:string"></Attribute>
    <Attribute ref="system:sharedSensor" type="xs:boolean" use="required"></Attribute>
    <Attribute ref="system:tolerancePressureMax" type="xs:float" use="required"></Attribute>
    <Attribute ref="system:tolerancePressureMin" type="xs:float" use="required"></Attribute>
    <Attribute ref="system:toleranceTemperatureMax" type="xs:float” use="required"></Attribute>
    <Attribute ref="system:toleranceTemperatureMin" type="xs:float"  use="required"></Attribute>
    <Attribute ref="system:toleranceVibrationMax" type="xs:float" use="required"></Attribute>
    <ReferenceElement name="measures" minOccurs="1" maxOccurs="1"
       type="system:ParameterType"></ReferenceElement>
  </ObjectType>
```

Specification of an Attribute with semantics from the model

Specification of a Reference Element

Example: XSD for Sensor Specification

```
<xs:complexType xmlns="system" name="SensorType">
  <xs:annotation>
    <xs:documentation>An Object Type</xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="system:DeviceType">
      <xs:sequence>
        <xs:element name="measures" minOccurs="1" maxOccurs="1">
          <xs:annotation>
            <xs:documentation>Reference Element. Attribute nc:ref is used to point to the referenced value, which must be of type
                system:ParameterType</xs:documentation>
          </xs:annotation>
          <xs:complexType>
            <xs:annotation><xs:documentation>A Reference Element Type</xs:documentation>
            </xs:annotation>
            <xs:attributeGroup ref="nc:W3C-AttributeGroup"/>
            <xs:attributeGroup ref="nc:NC-AttributeGroup"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attribute ref="data:bitsPerSample">
        <xs:annotation>
          <xs:documentation>An Attribute. <xs:annotation>
              <xs:documentation>Value of this attribute should be of type xs:int</xs:documentation>
            </xs:annotation>
          </xs:documentation>
        </xs:annotation>
      </xs:attribute>
      <xs:attribute ref="data:bitsPerSecond">
        <xs:annotation>
          <xs:documentation>An Attribute. <xs:annotation>
              <xs:documentation>Value of this attribute should be of type xs:int</xs:documentation>
            </xs:annotation>
          </xs:documentation>
        </xs:annotation>
      </xs:attribute>
…
```

The Sensor becomes an XSD Complex Type

Sensor extends Device

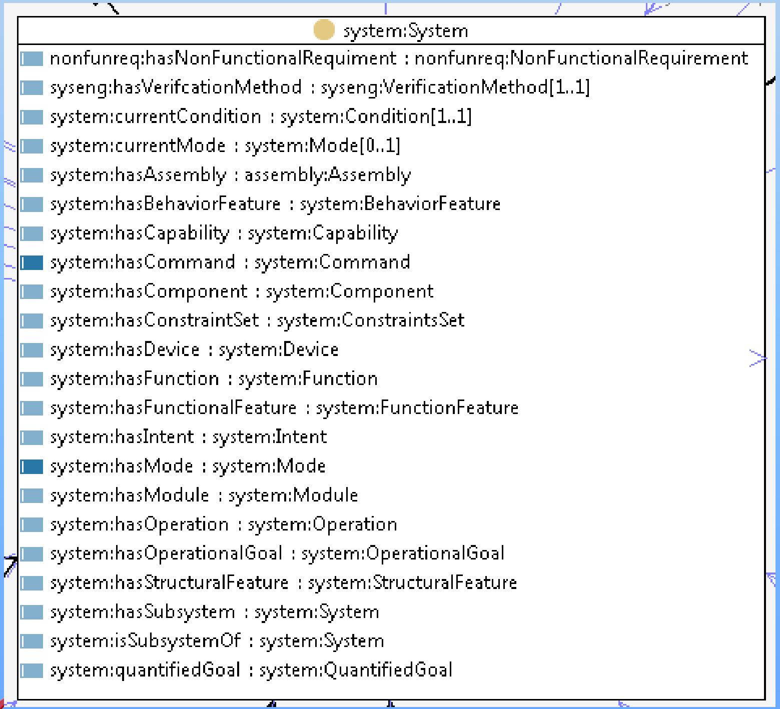
Sensor 'measures' Parameter

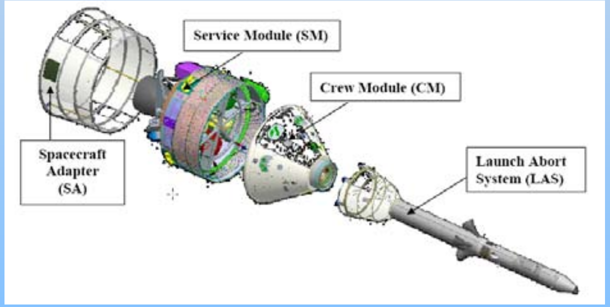
Attribute definition

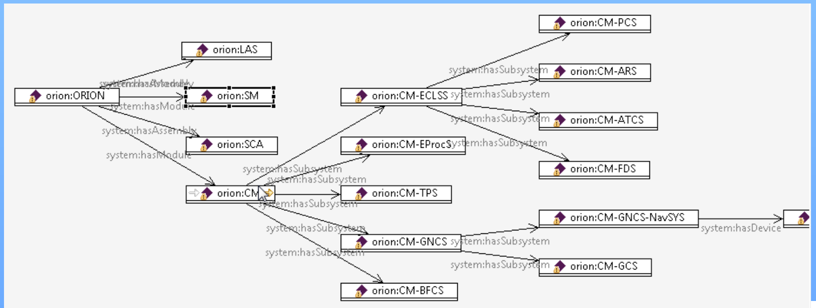
System Ontologies

A System is modeled using "SBFI" formalism

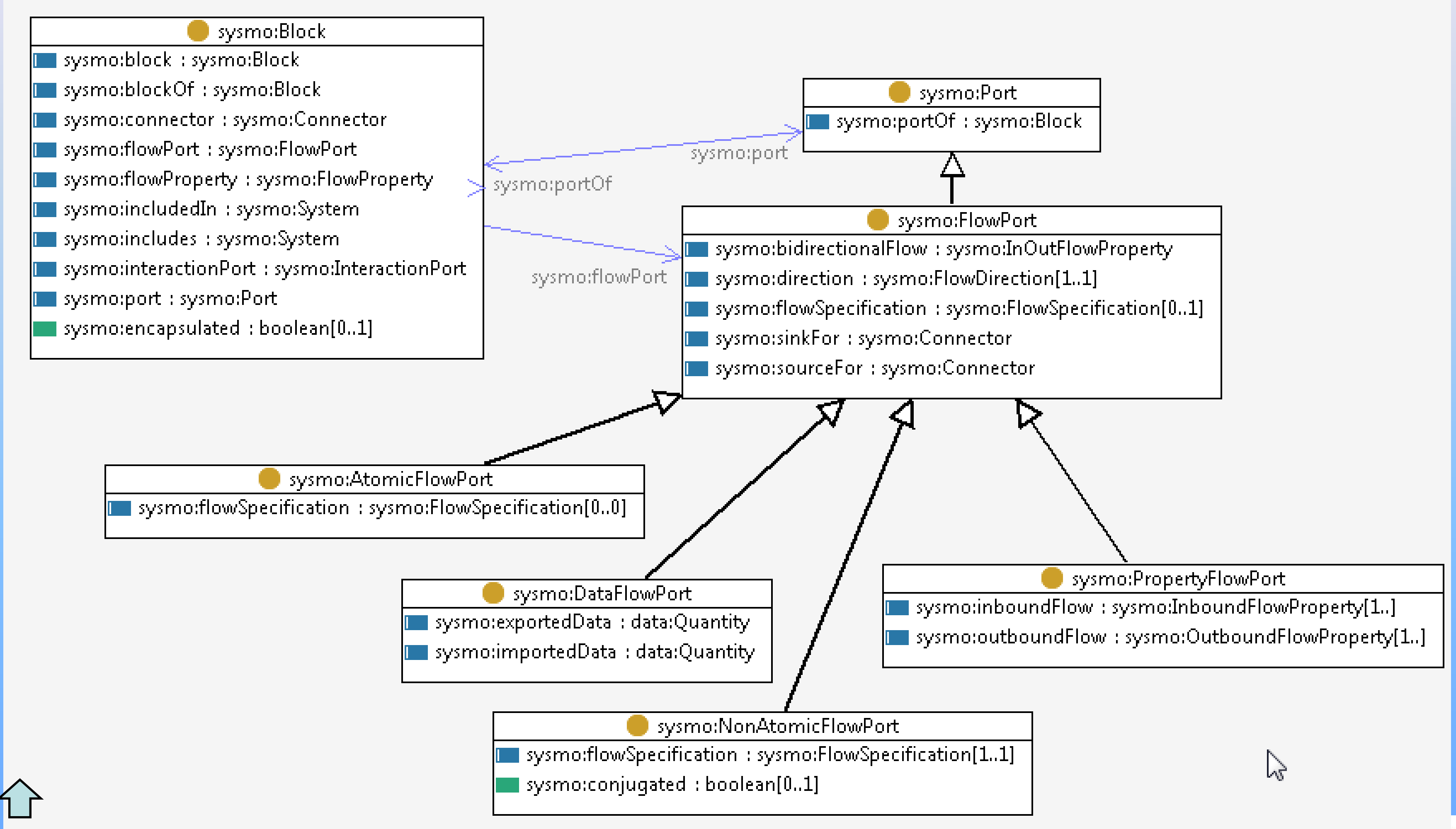
The NASA System Ontology extends SBF with other SE modeling constructs. Some SysML concepts are modeled directly others are modeled more expressively than what is possible in UML.



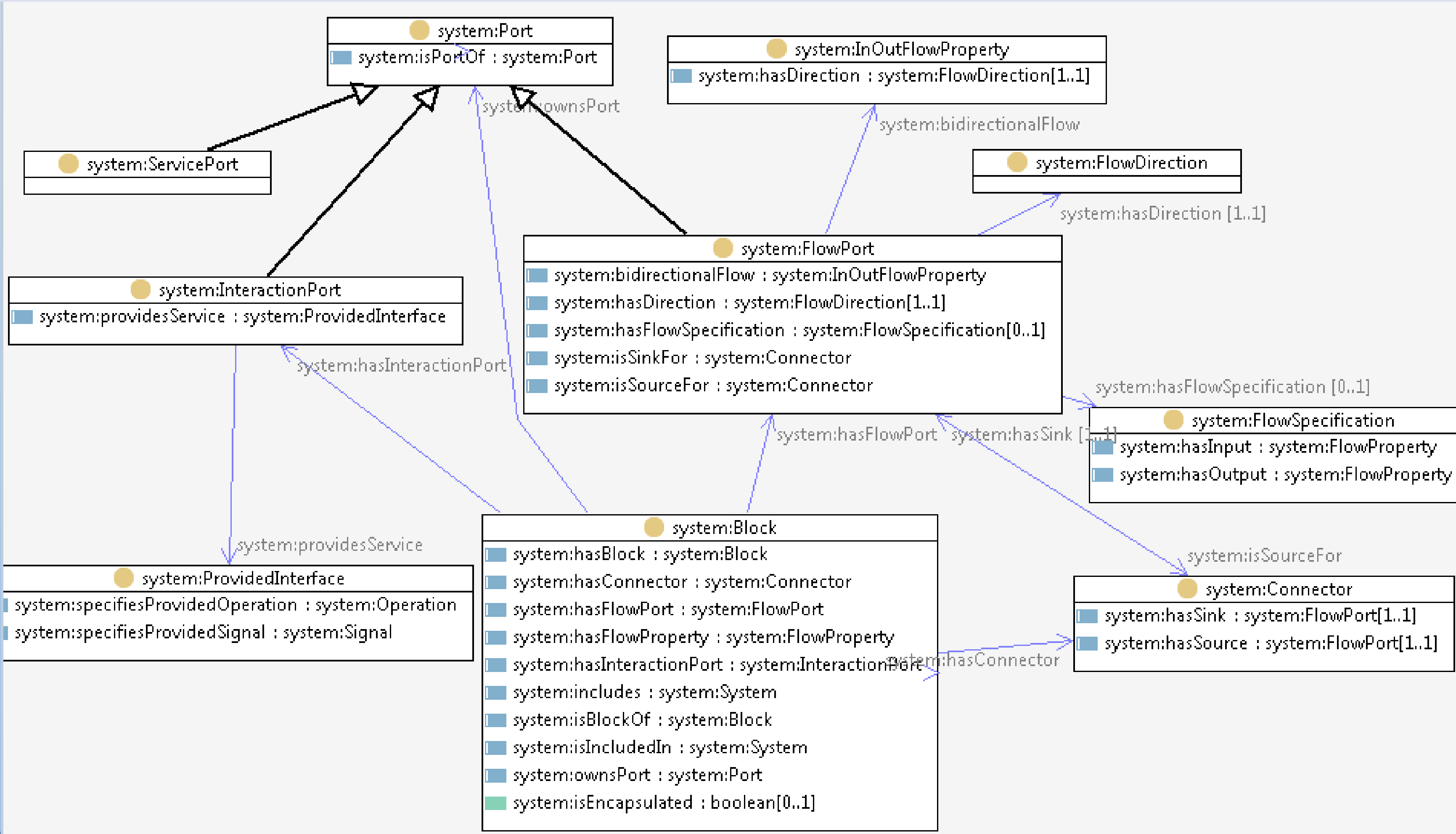




SysMO Ontology Example: Blocks and Ports



SysMO Example: SysML FlowPort

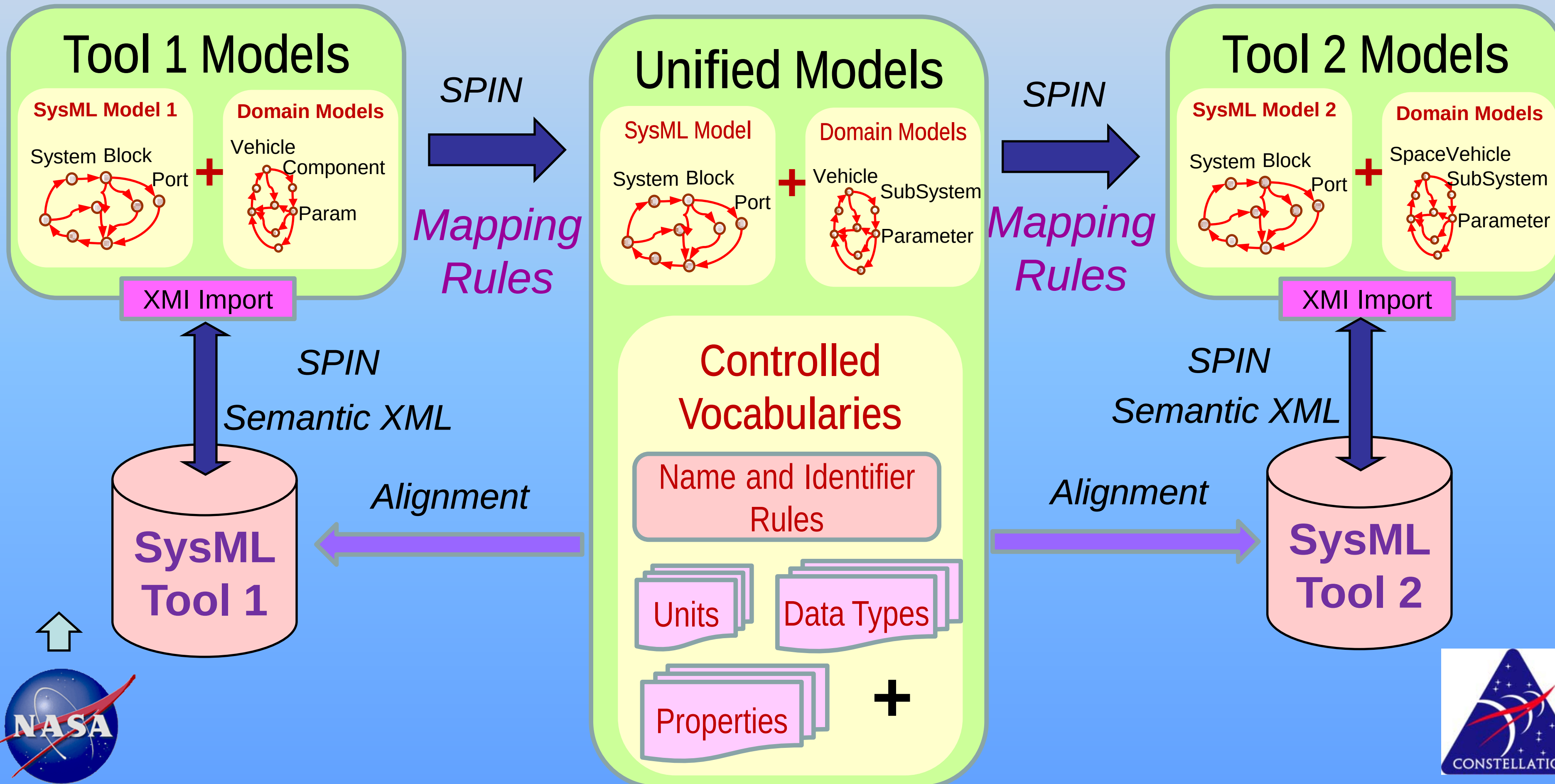


FlowPort in N3 Format

```
system:FlowPort
    a       owl:Class ;
    rdfs:label "Flow port"^^xsd:string ;
    rdfs:subClassOf system:Port ;
    rdfs:subClassOf
        [ a       owl:Restriction ;
          dc:description "Indicates the direction in which an Atomic
    FlowPort relays its items. If the direction is set to in then the items
    are relayed from an external connector via the FlowPort into the
    FlowPort's owner (or one of its Parts). If the direction is set to out,
    then the items are relayed from the FlowPort's owner, via the
    FlowPort, through an external connector attached to the FlowPort,
    and if the direction is set to inout then items can flow both ways. By
    default, the value is inout." ;
          owl:allValuesFrom system:FlowDirection ;
          owl:onProperty system:hasDirection
        ] ;
    rdfs:subClassOf
        [ a       owl:Restriction ;
          owl:allValuesFrom system:FlowSpecification ;
          owl:onProperty system:hasFlowSpecification
        ] ;
    rdfs:subClassOf
        [ a       owl:Restriction ;
          owl:allValuesFrom system:InOutFlowProperty ;
          owl:onProperty system:bidirectionalFlow
        ] ;
 rdfs:subClassOf
        [ a       owl:Restriction ;
          owl:allValuesFrom system:Connector ;
          owl:onProperty system:isSourceFor
        ] ;
rdfs:subClassOf
        [ a       owl:Restriction ;
          owl:maxCardinality "1"^^xsd:int ;
          owl:onProperty system:hasFlowSpecification
        ] ;
rdfs:subClassOf
        [ a       owl:Restriction ;
          owl:allValuesFrom system:Connector ;
          owl:onProperty system:isSinkFor
        ] ;
    rdfs:subClassOf
        [ a       owl:Restriction ;
          owl:cardinality "1"^^xsd:int ;
          owl:onProperty system:hasDirection
        ] ;
    dc:description "Flow Ports are interaction points
    through which input and/or output of items such as
    data, material or some property such as torque,
    pressure or energy may flow. This enables the owning
    block to declare which items it may exchange with its
    environment and what are the interaction points
    through which the exchange is made. A FlowPort
    specifies the input and output items that may flow
    between a Block and its environment. FlowPorts are
    interaction points through which data, material or
    energy  'can' enter or leave the owning Block. The
    specification of what can flow is achieved by typing
    the FlowPort with a specification of things that flow. In
    general, flow ports are intended to be used for
    asynchronous, broadcast, or send and forget
    interactions." .
```

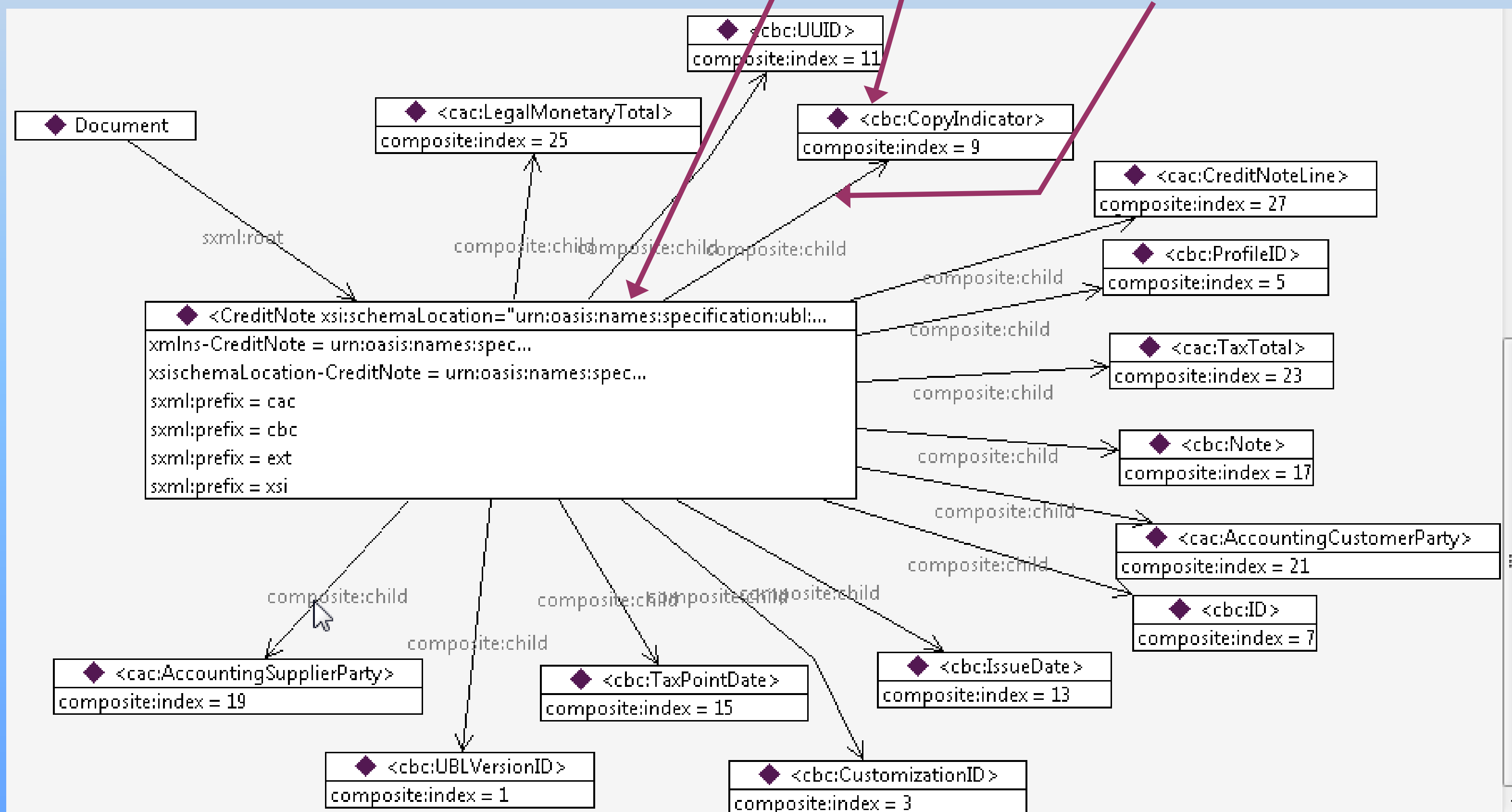
Capability Case: Ontology-Based Data Exchange (between SysML Tools)

Each tool's model is converted to triples using SPIN. Triples can be related through a unified SysMO Model. Data exchange and other operations are then possible.



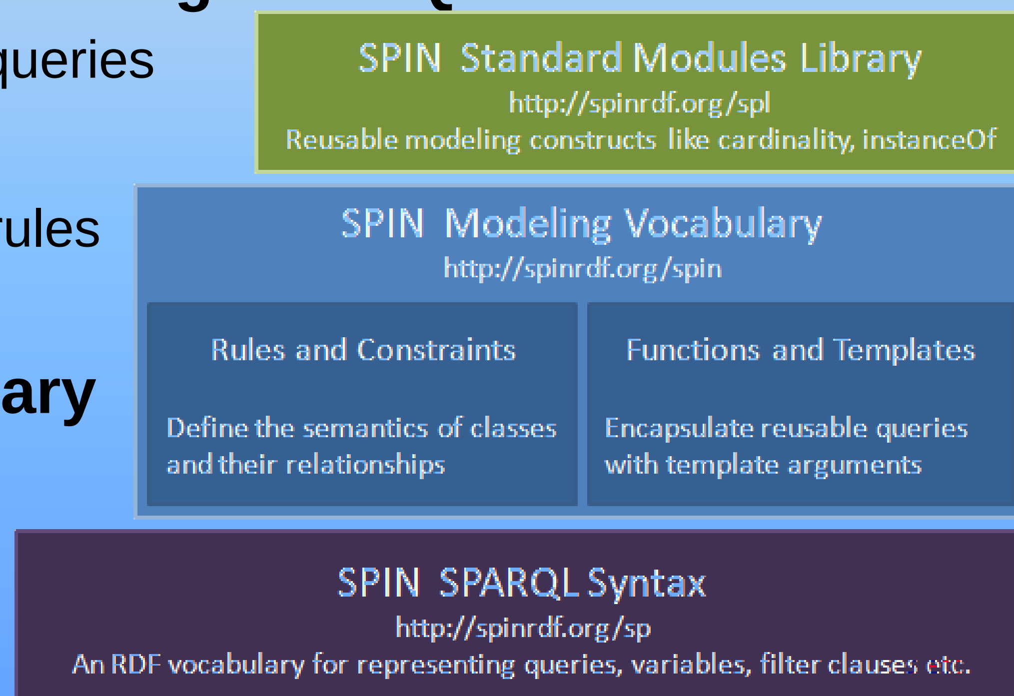
Semantic XML – representing XML in OWL

- **Composite Pattern**
  - Elements contain Elements
  - Elements have attributes



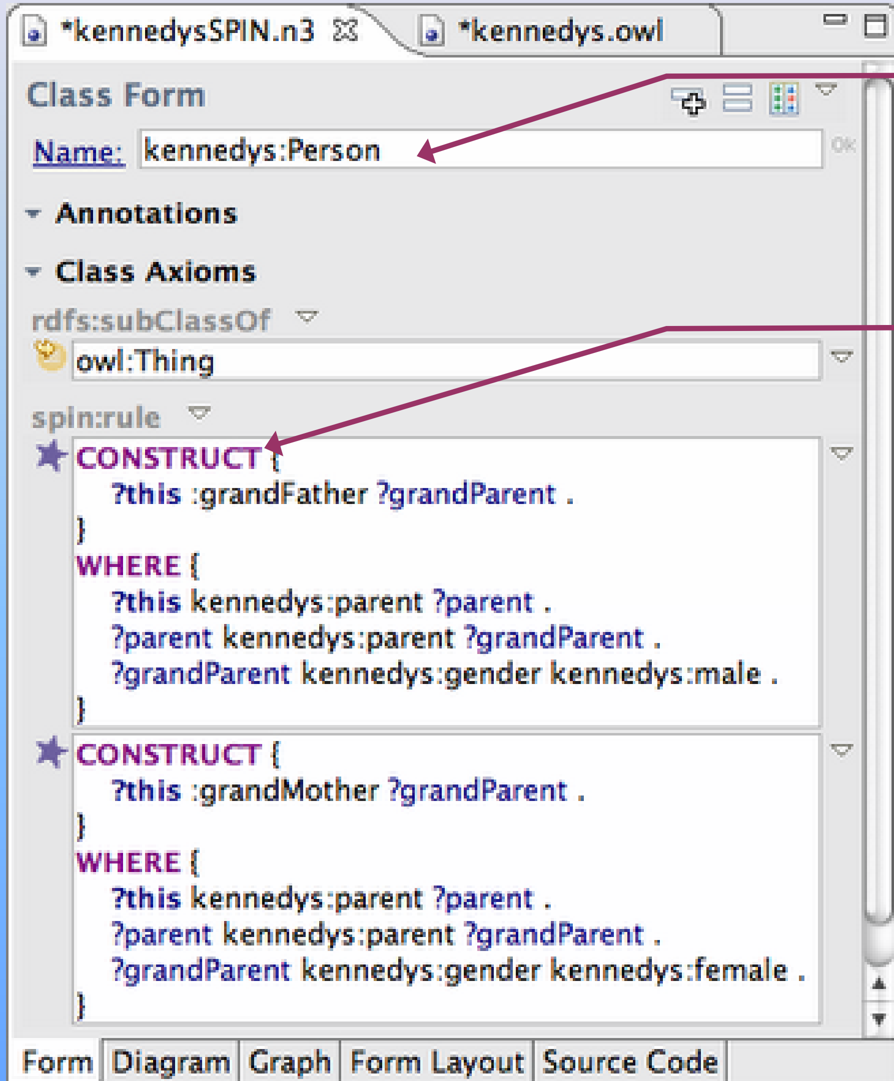
SPIN – using SPARQL as a Rules Language

- **SPIN – SPARQL Inferencing Notation**
  - A Rules Language based on SPARQL
  - define constraints and inference rules on Semantic Web models
  - http://spinrdf.org
- **Specification for representing SPARQL with RDF**
  - RDF syntax for SPARQL queries
- **Modeling vocabulary**
  - constraints, constructors, rules
  - templates, functions
- **Standard Modules Library**
  - small set of frequently needed SPARQL queries



more at www.spinRDF.org

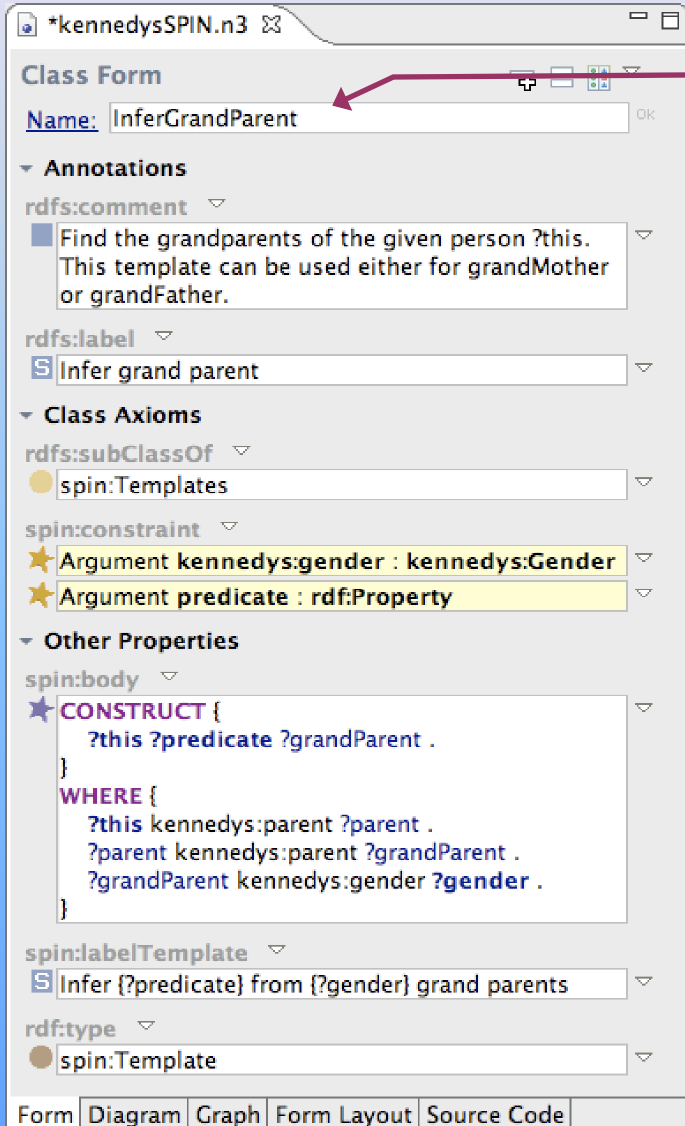
SPIN Example: Kennedy Family Rules – Using SPARQL as a Rules Language



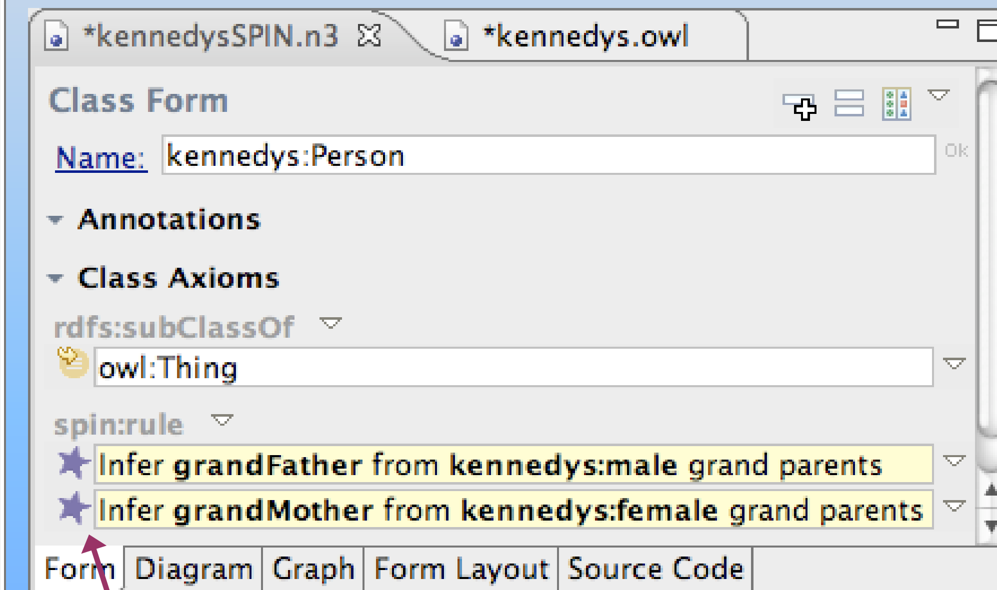
Person Class

SPIN Rule using SPARQL Construct

SPIN Example: Kennedy Family Rules (Template Version)

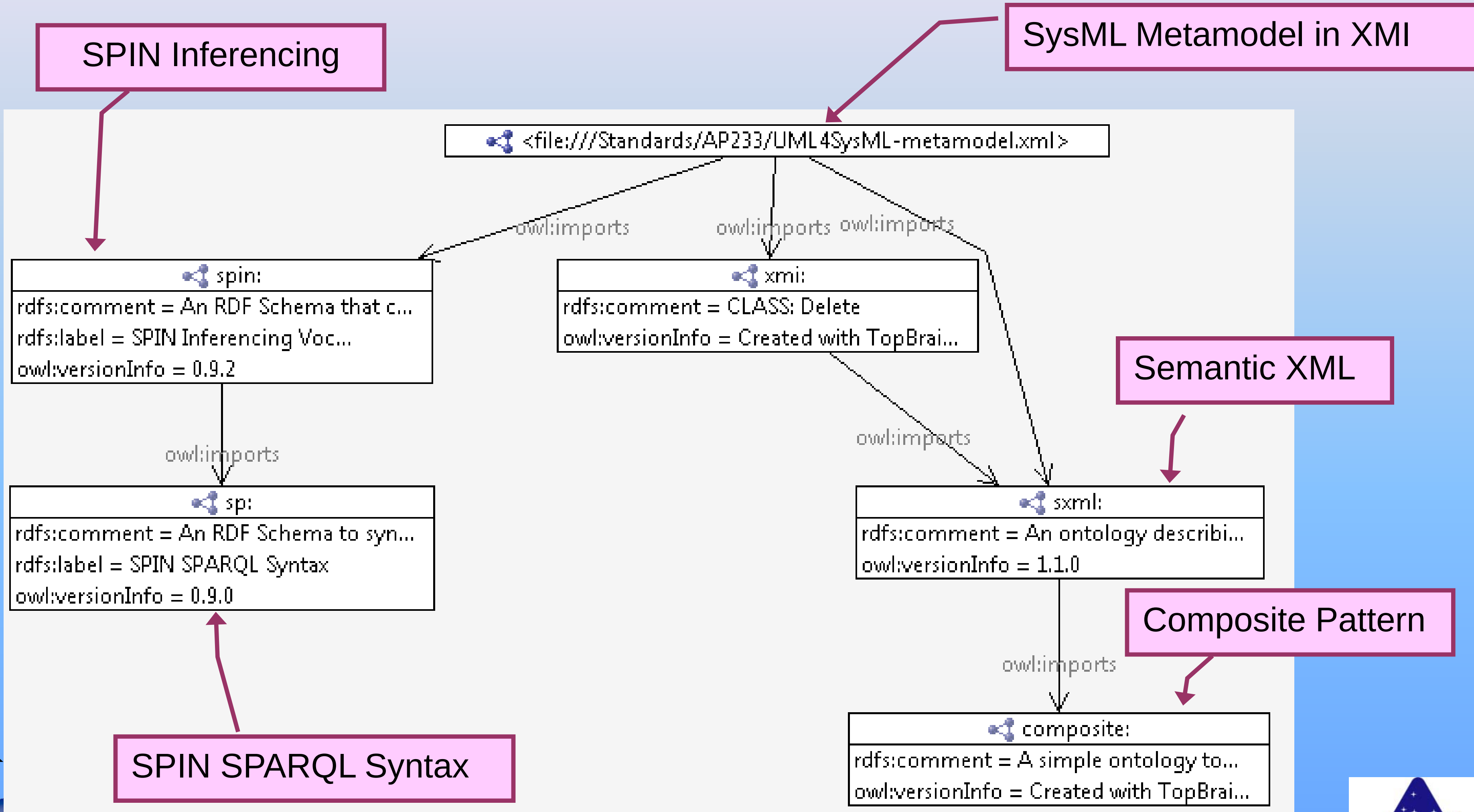


Template for Rule

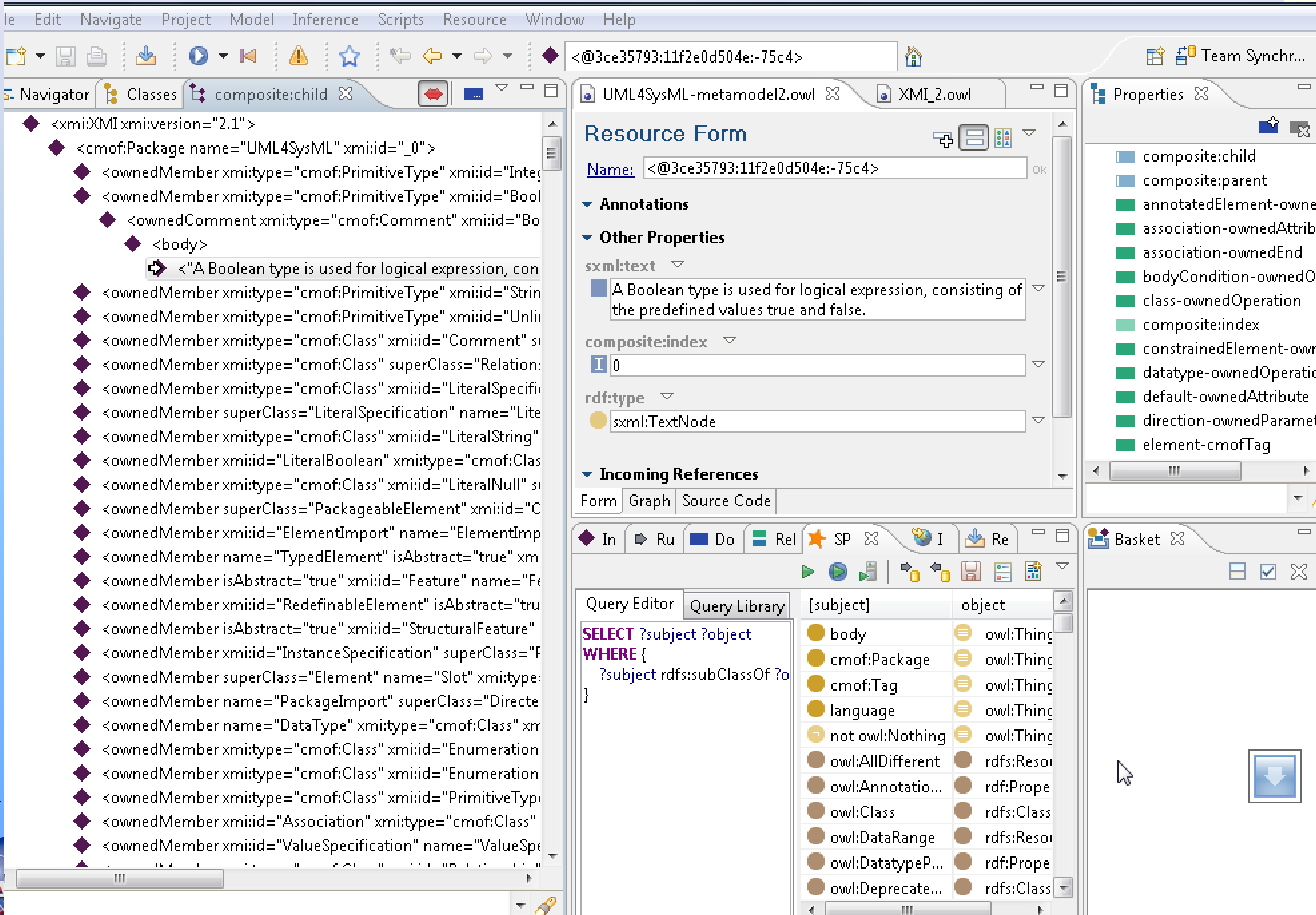


Class invokes rules by passing arguments to templates

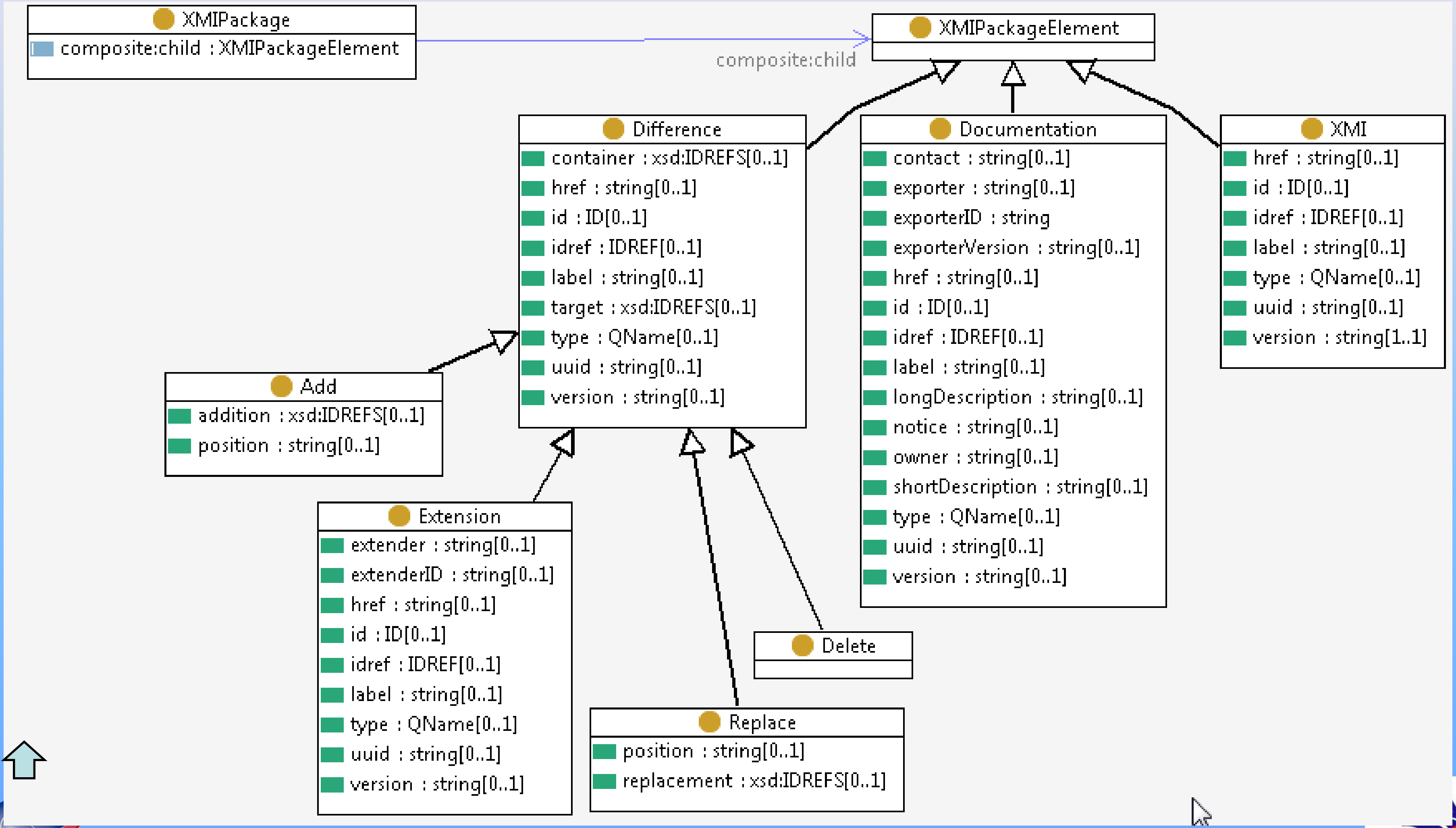
Ontology Architecture for SysMO Model Creation



Semantic XML translation of SysML using TopBraid Composer



Importing XMI.XSD file into TopBraid Composer provided the start for XMI Ontology



Concluding Remarks

- **Ontologies can and should be used for inferencing and specification to enable vendor-independent product data exchange**
  - OWL can be used as a vendor-neutral specification language
  - OWL is more expressive than XML, UML and ER models
- **OWL + Rules (SPIN) provides expressive support for data acquisition, interpretation, transformation and verification**
  - Data Exchange Engines should have ontologies of industry standards
- **OWL can interoperate with XML technologies through the use of (NASA) XML SchemaPlus and controlled vocabularies**
  - Controlled Vocabularies are key to PDE
- **Product Data Exchange makes no sense if you don't have Data Quality**
  - requires compliance to Naming and Identifier Rules (NASA NIR)
  - needs ontologies for vocabulary management and translations

Thank You

Ralph Hodgson

E-mail: [rhodgson at topquadrant.com](), [Ralph.Hodgson at nasa.gov]()

[http://del.icio.us/ralphtq]()

[http://twitter.com/ralphtq]()