NIEM-UML Webinar Series

Model Driven Information Sharing With

NIEM-UML

Agenda

- Introductions
- 10 Minute very high-level introduction
- Creating an IEPD with NIEM-UML (Demo)
- NIEM-UML Concepts in Detail
- Vendor Support
- Q&A

Webinar Materials
http://www.niem-uml.org/webinar/materials/

Introductions

Cory Casanave [cory-c at modeldriven dot com]

CEO, Model Driven Solutions

Actionable Architectures & Agile Solutions

Information Federation, SOA, EA and MDA Development

Open Source Supporting a Model Driven Approach (ModelDriven.org)

Object Management Group

Board of Directors

Standards work: UML, SoaML, BPMN, EDOC, AESIG, SIMF, Etc.

GovDTF: Chair – Open Government Workgroup

W3C

Government Linked Data (GLD) Workgroup

Government

NIEM: Co-chair NIEM-UML PIM Submission Team

CIO Council/DAS: Open Government Vocabularies Workgroup

Various government enterprise, information, SOA and systems

cory (dash) c at modeldriven dot com

NIEM-UML Players

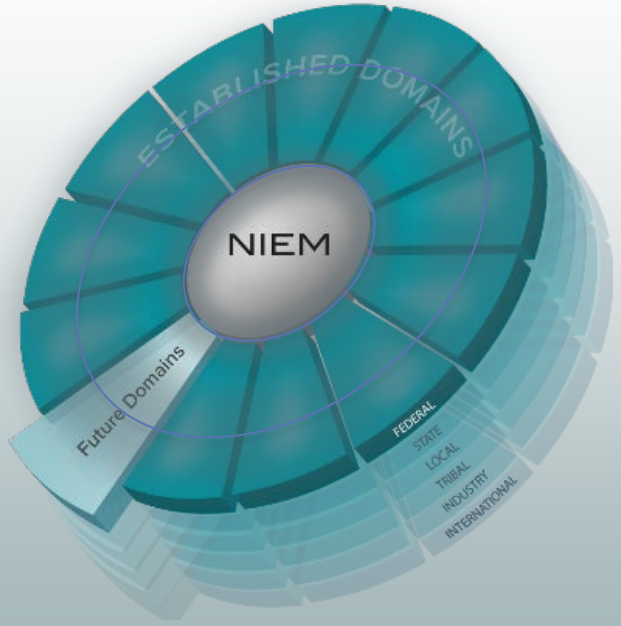
- The National Information Exchange Model (NIEM) is a well established U.S. sponsored program and community for standardized information exchange. The NIEM Program Management Office is organized under the Department of Homeland Security, the sponsor of NIEM-UML.
- The Object Management Group (OMG) is a leading industry consensus organization specifying many of the popular modeling and middleware standards.
- The Unified Modeling Language (UML®) is an established standard for modeling from the OMG.
- NIEM-UML is a profile of UML for modeling NIEM at a logical and XML specific level that is in the final stages of adoption within the OMG.
- Cameo NIEM-UML is the NIEM-UML plugin from for Magicdraw UML, used as the tooling in this demo

NIEM-UML Status

- The final NIEM-UML specification was recommended for adoption by the Object Management Group Government Domain Task Force (http://gov.omg.org/) and architecture board June 2012.
- The specification still has to go through additional OMG adoption processes and approval by the OMG Board of Directors, but adoption is quite certain.
- Implementations of the draft specification are already available and pilot projects have started– the PMO is seeking additional pilot projects.
- The specification and associated resources is available at: http://www.niem-uml.org

The Foundation of NIEM

Common Language
(Data Model Lifecycle)

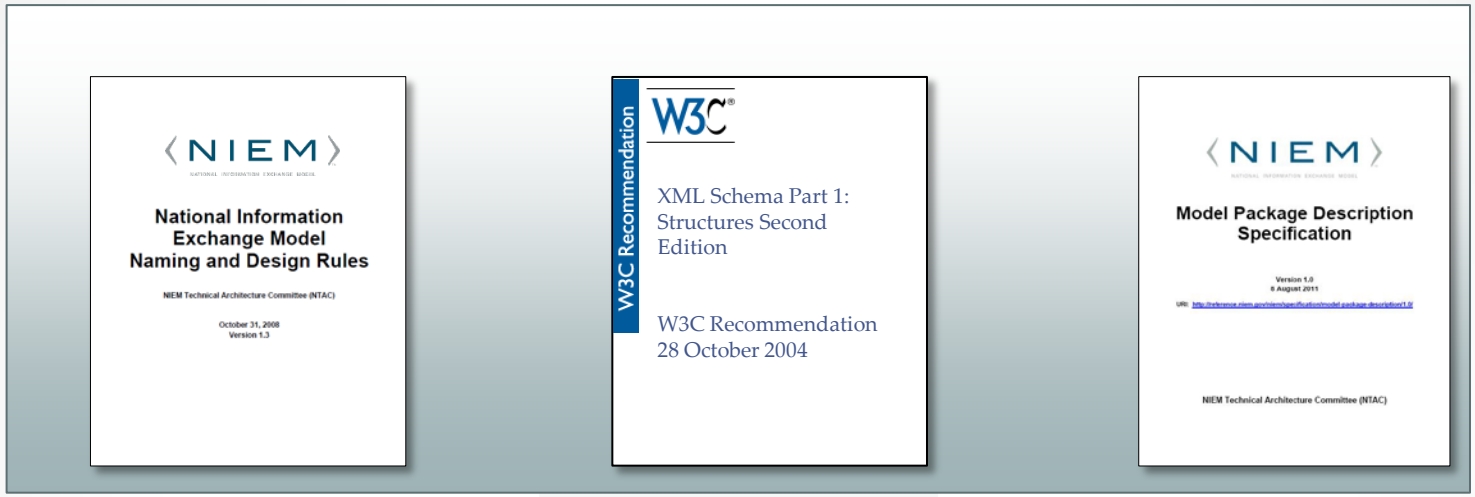


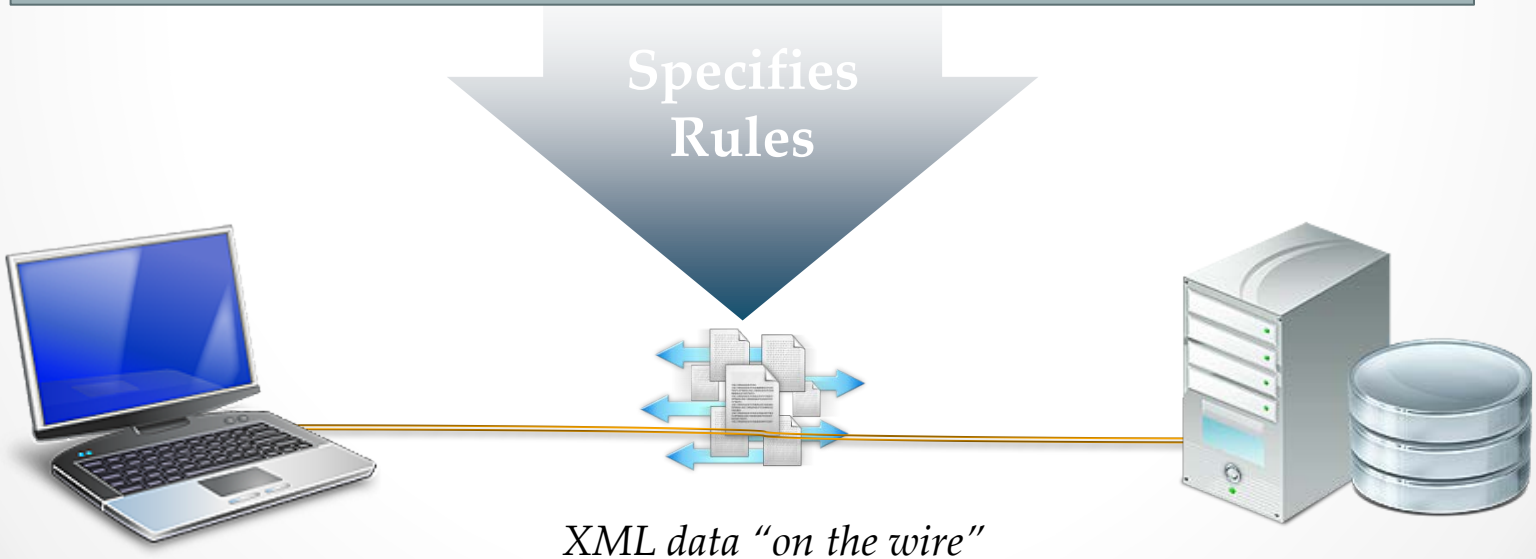
Built and governed by the business users at Federal, State, Local, Tribal and Private Sectors

Repeatable, Reusable Process
(Exchange Specification Lifecycle)

People & Organizations Sharing Information

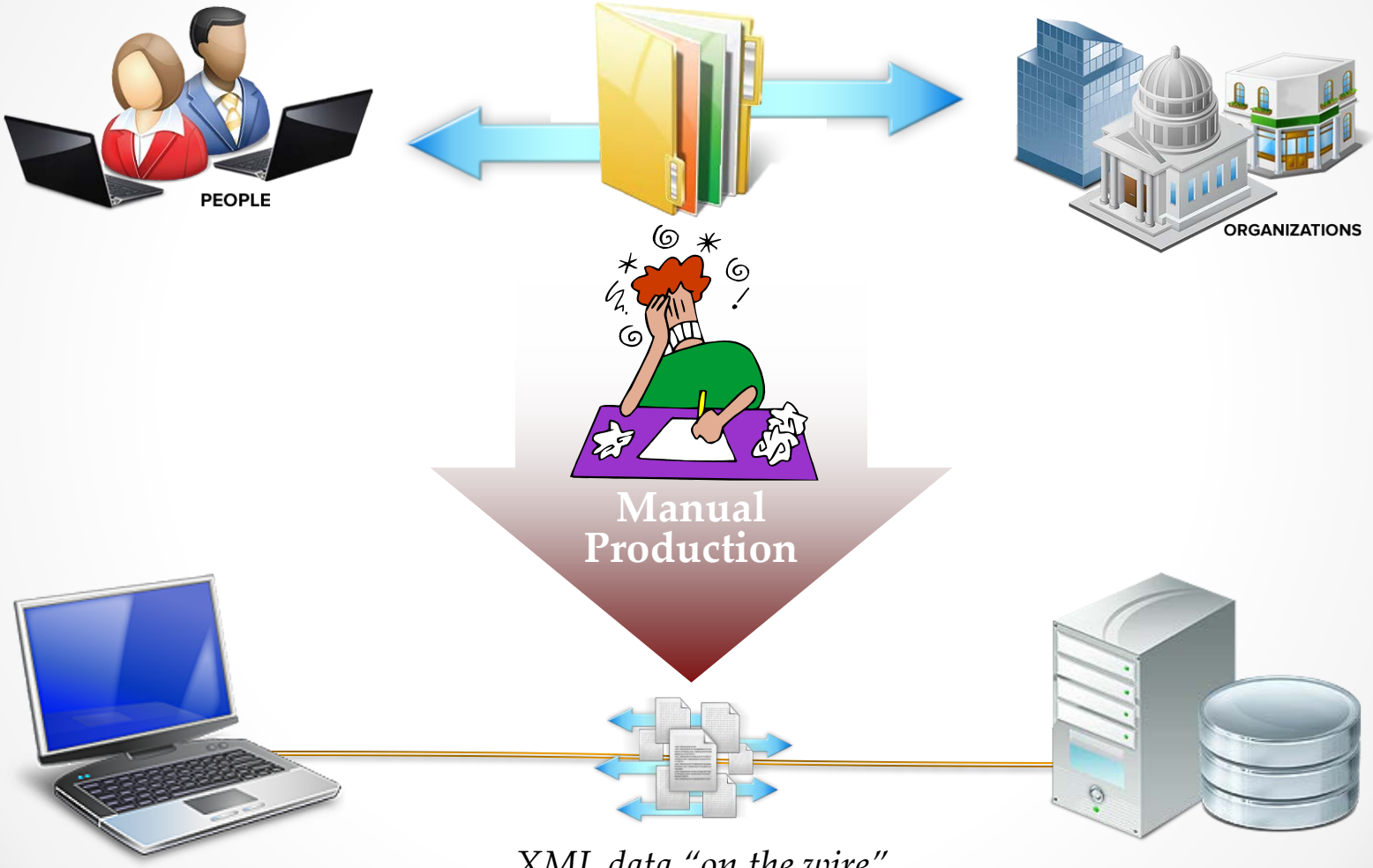
Information Exchange Based On NIEM



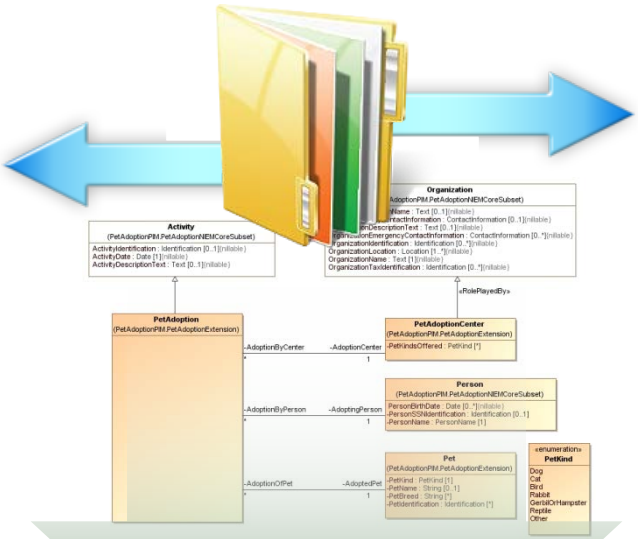


XML data "on the wire"

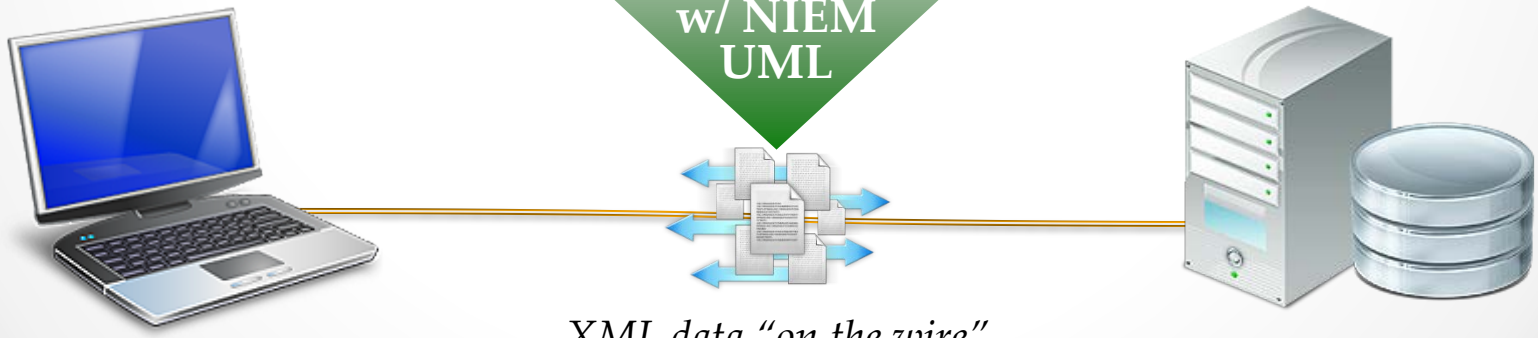
Information sharing bits & bytes
XML Data-In-Motion "on the wire"



Information sharing based on NIEM-UML

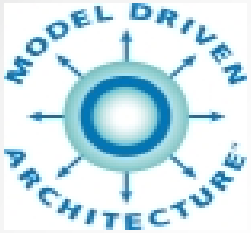


Automatic Generation w/ NIEM UML



XML data "on the wire"

Using Model Driven Architecture for NIEM



Using Model Driven Architecture (MDA) for NIEM Information Exchange has multiple advantages

- Models are easier for both business and technical stakeholders to understand
- MDA helps reduce the time and cost to develop and maintain information sharing solutions
- NIEM Naming, design and packaging rules are automatically applied and validated
- Processes, services and information can be part of a coherent system and system of systems architecture across the full life-cycle of solutions
- Multiple technologies can be supported using different MDA generation patterns, such as JSON or the Semantic Web

OMG-MDA standards you probably already know about

For example, a few include:

Unified Modeling Language
UML®, a standardized modeling language

Business Process Modeling Notation
BPMN™ , allows businesses understanding of their internal business procedures

Common Warehouse Metamodel
CWM™, the integration of the last two data warehousing initiatives

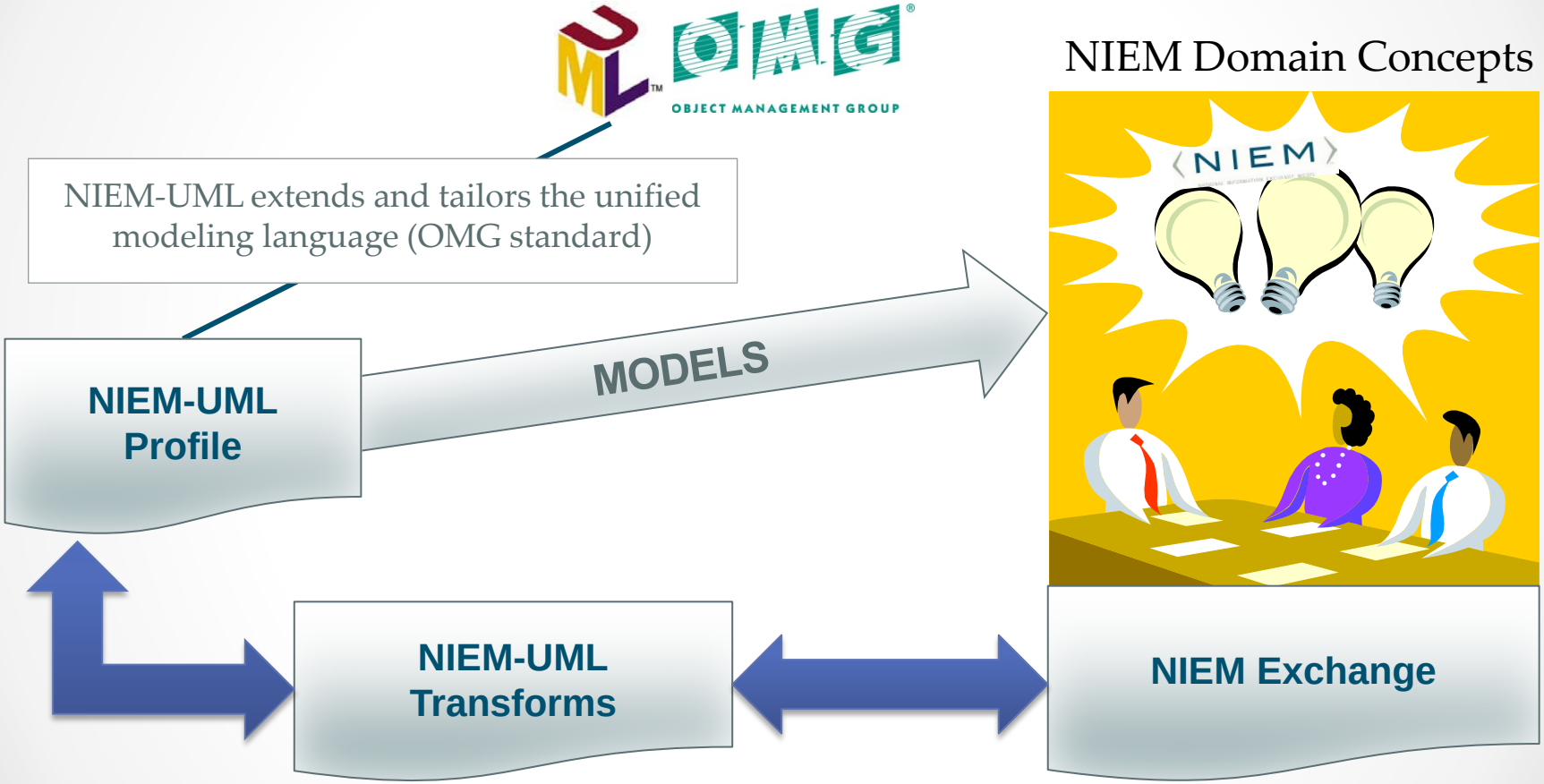
Meta-Object Facility
MOF™, a repository standard

XML Metadata Interchange
XMI®, a XML-UML standard

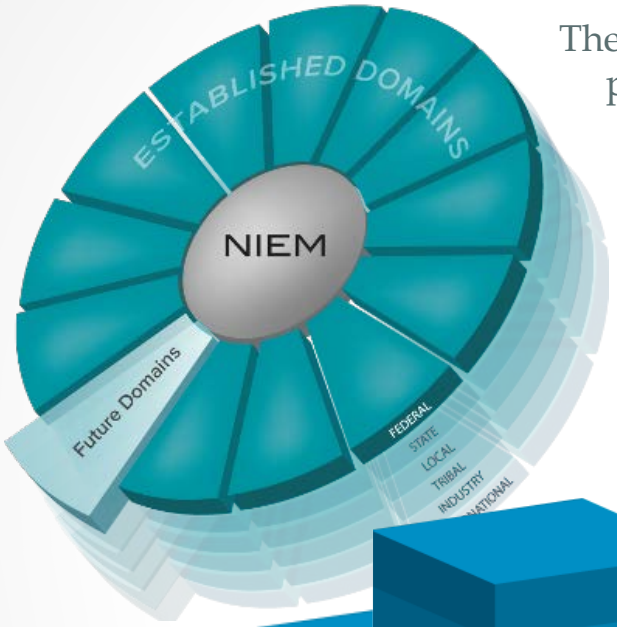
SoaML
Service Oriented Modeling Language

NIEM-UML becomes part of this family of standards

From UML to NIEM Exchanges

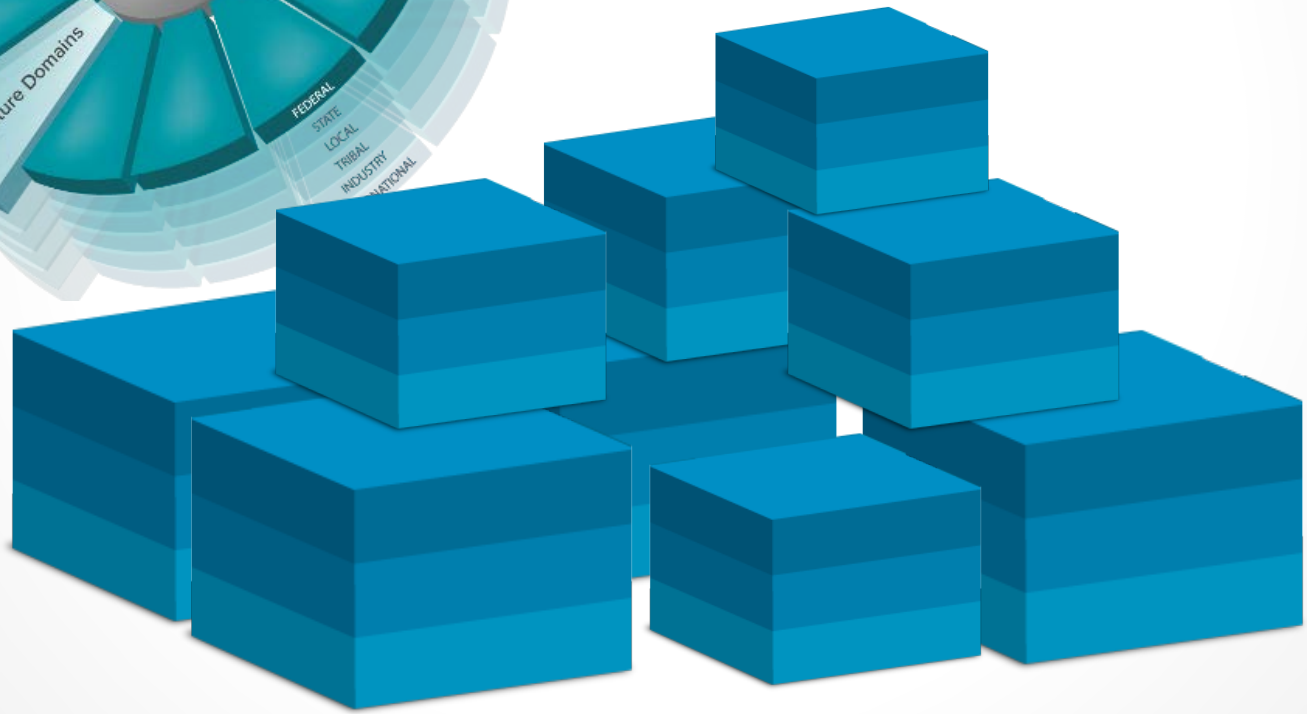


Reuse is the key to NIEM

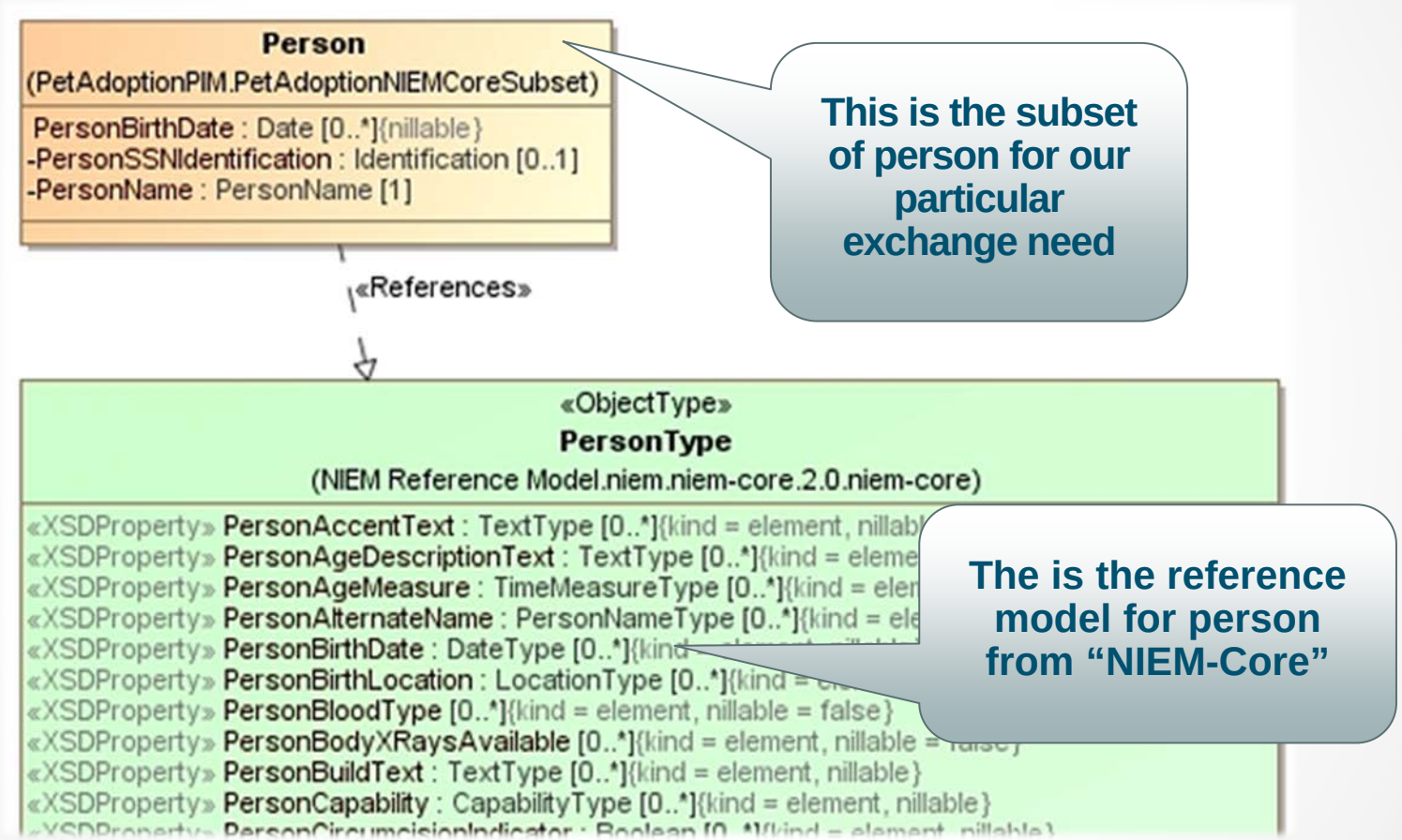


The NIEM "Core" and domain reference vocabularies provide the basis for the reuse that is central to NIEM.

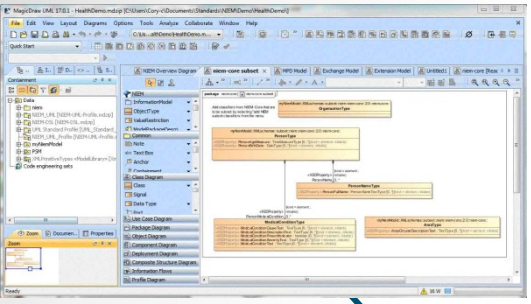
These vocabularies are now available as NIEM-UML models



Modeling reuse via subsetting



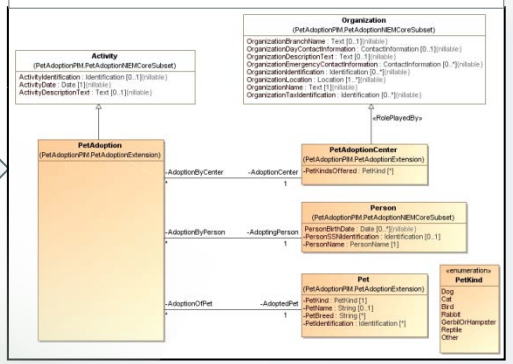
Using NIEM-UML to Model Information Exchanges

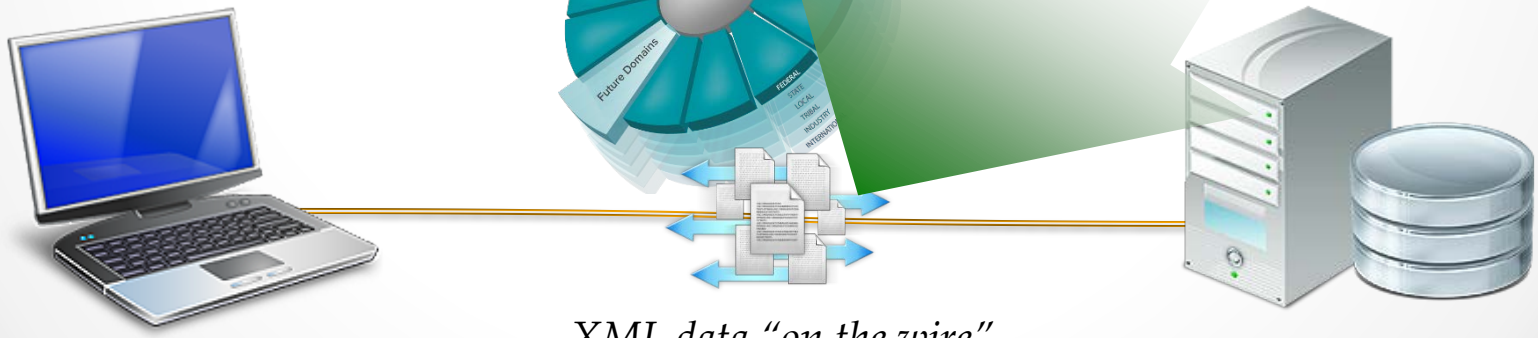


NIEM-UML Profile

Design and Model

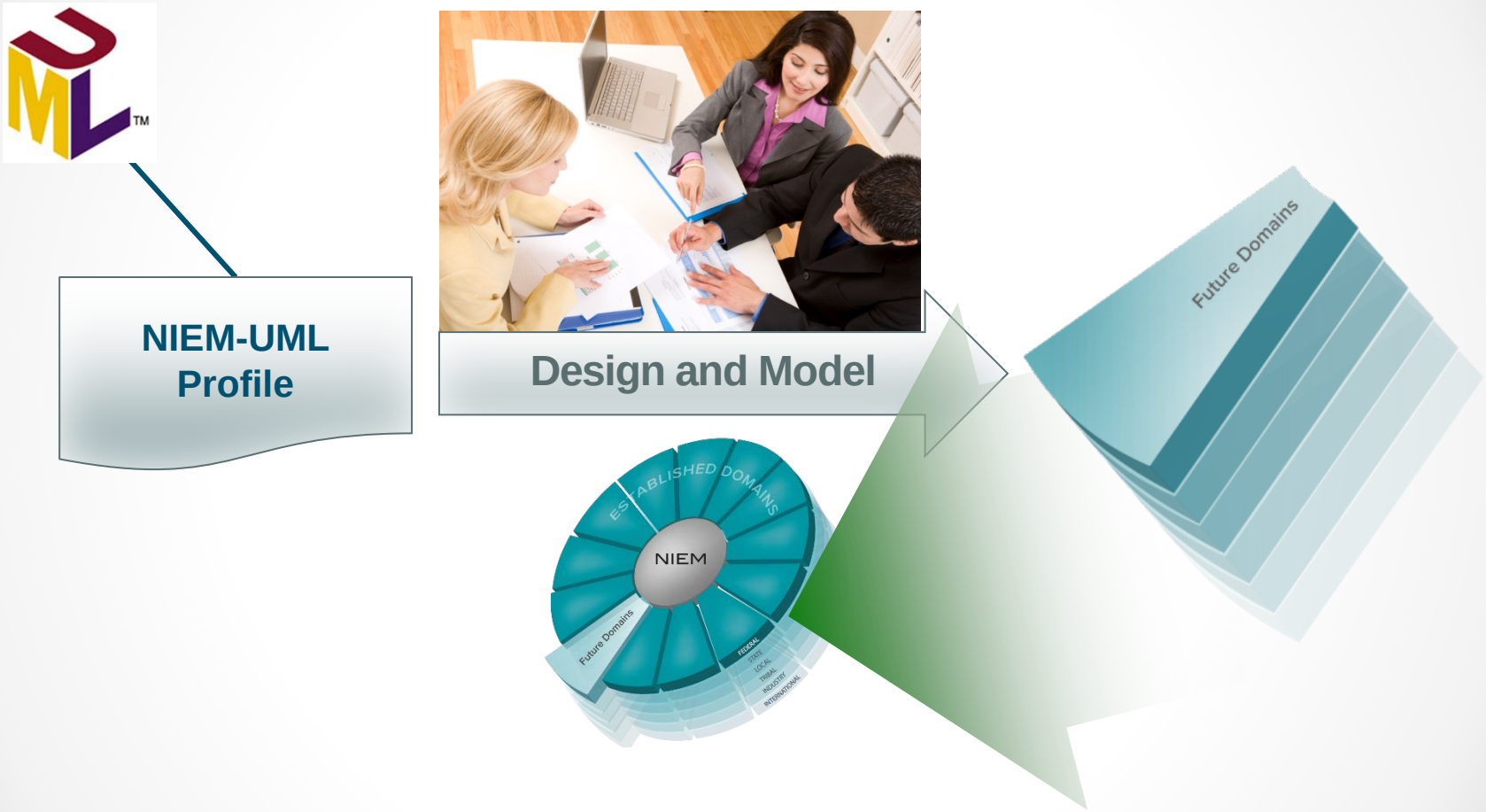
NIEM-UML Information Exchange Model





XML data "on the wire"

Using NIEM-UML to define new NIEM domains



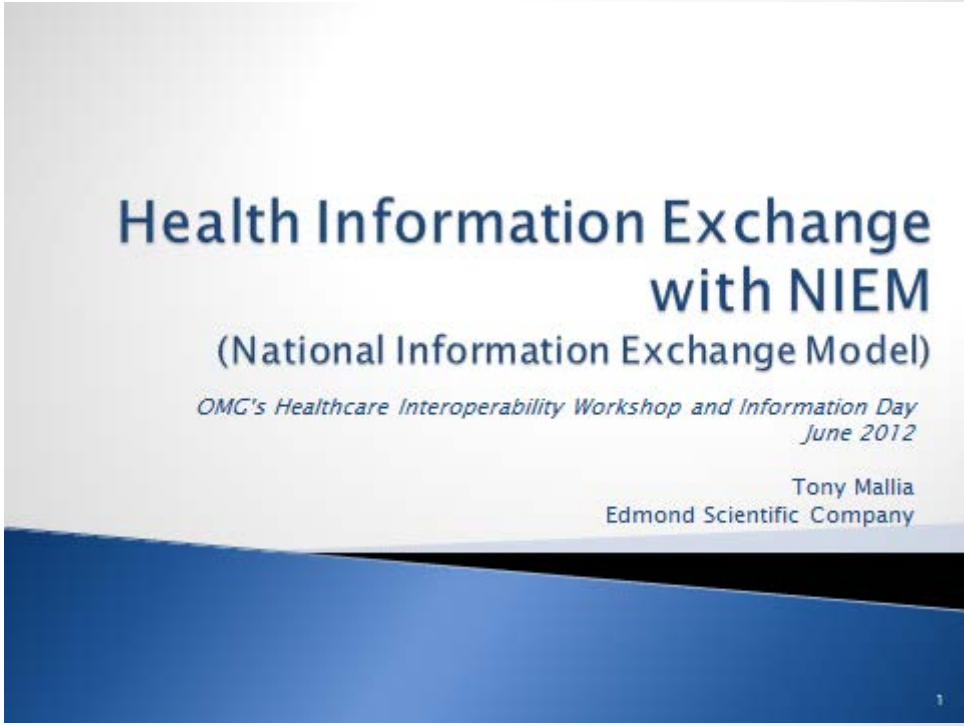
NIEM-UML Summary

- NIEM-UML is a new NIEM specification that provides for modeling NIEM in UML and producing or reverse engineering information exchange technical specifications
- This reduces the time, cost and learning curve of information exchange using NIEM
- MDA also provides for other aspects of the information sharing solution, such as: business processes, SOA services and back-end system integration
- Since NIEM-UML generates 100% NIEM conformant technical specifications, NIEM-UML Architects and Developers don't need to worry about as much about the technology details
- NIEM-UML can be extended to support other technologies, such as JSON and the semantic web
- NIEM-UML is in the final stages of the standards process, tools are available now and more are being built

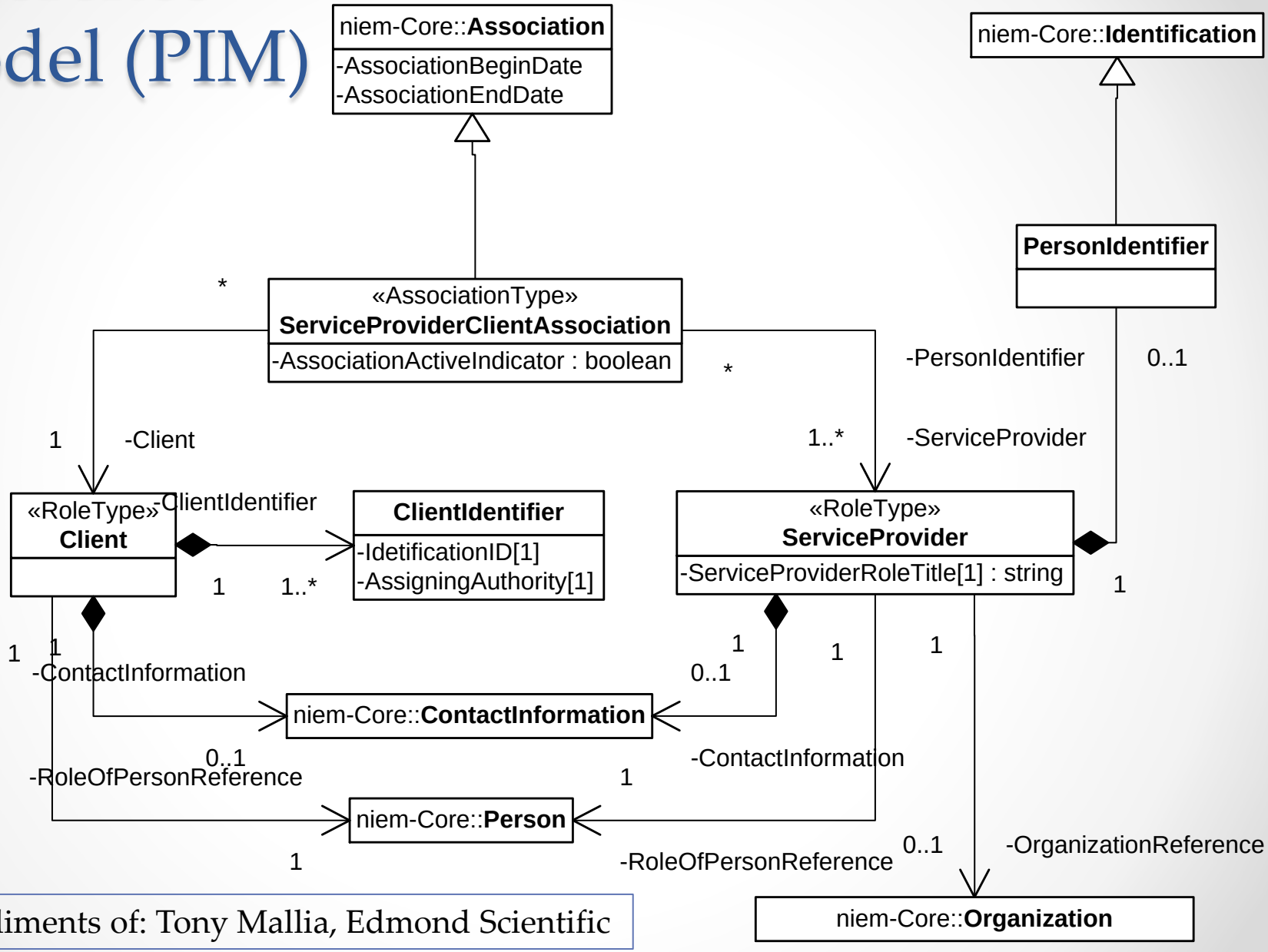
Quick Demo – Creating an IEPD

Context

- Health Care Information Exchange
- Based on design done by Tony Mallia
- Presented to OMG's Healthcare Interoperability Workshop
- Subject is exchange of healthcare medical condition involving a client (patient) and Healthcare Service Provider

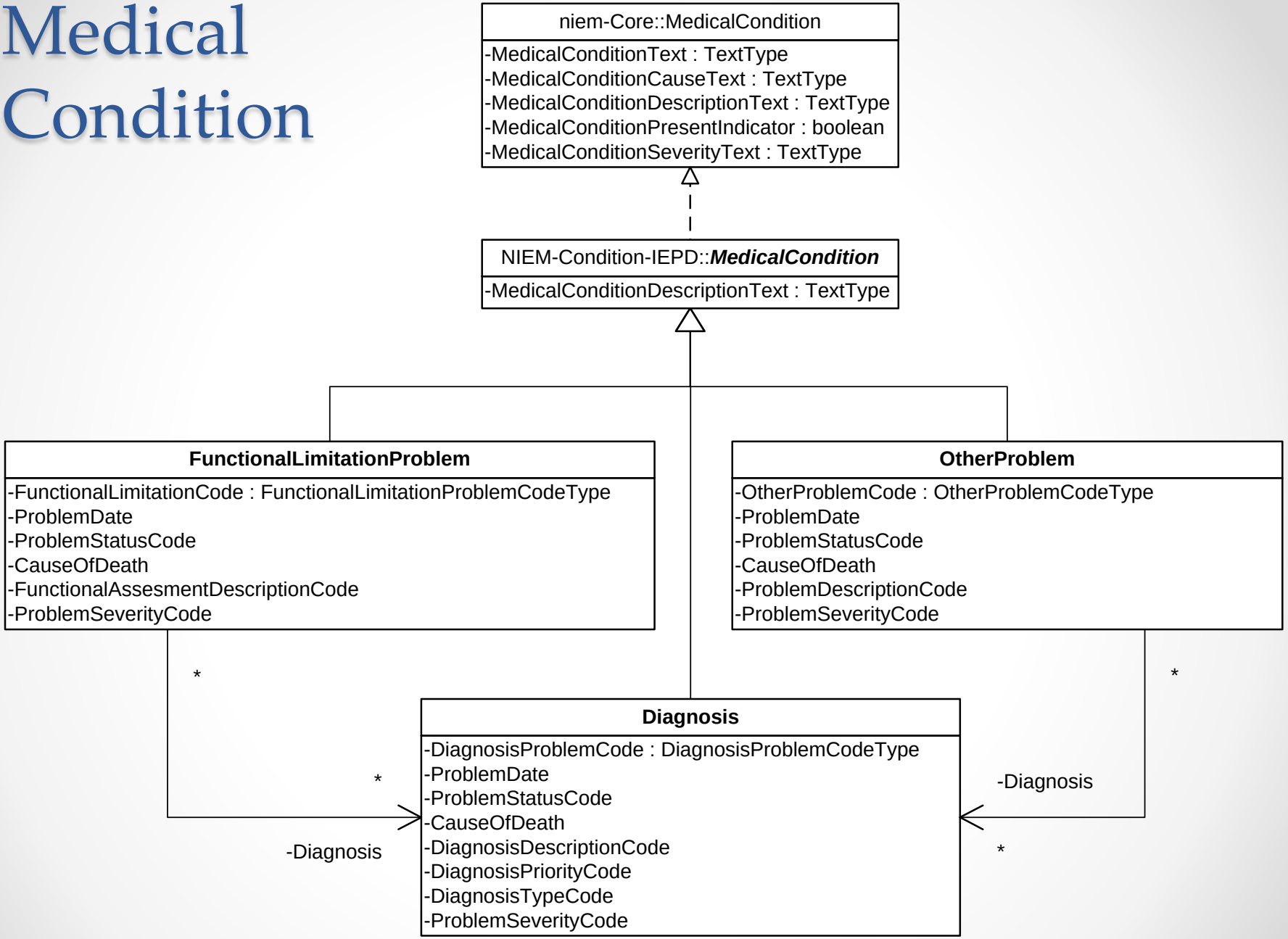


Reference Model (PIM)



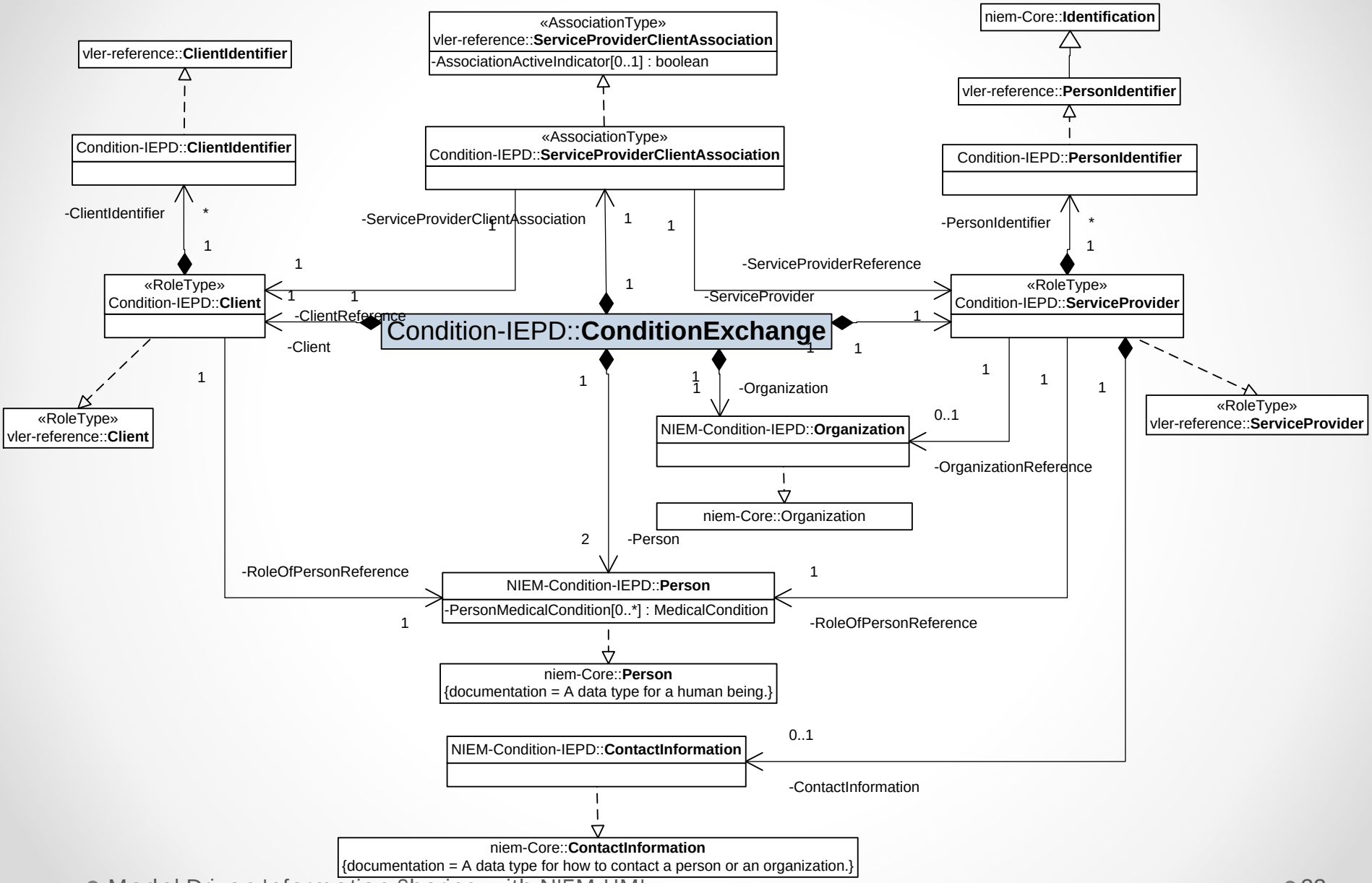
Compliments of: Tony Mallia, Edmond Scientific

Medical Condition



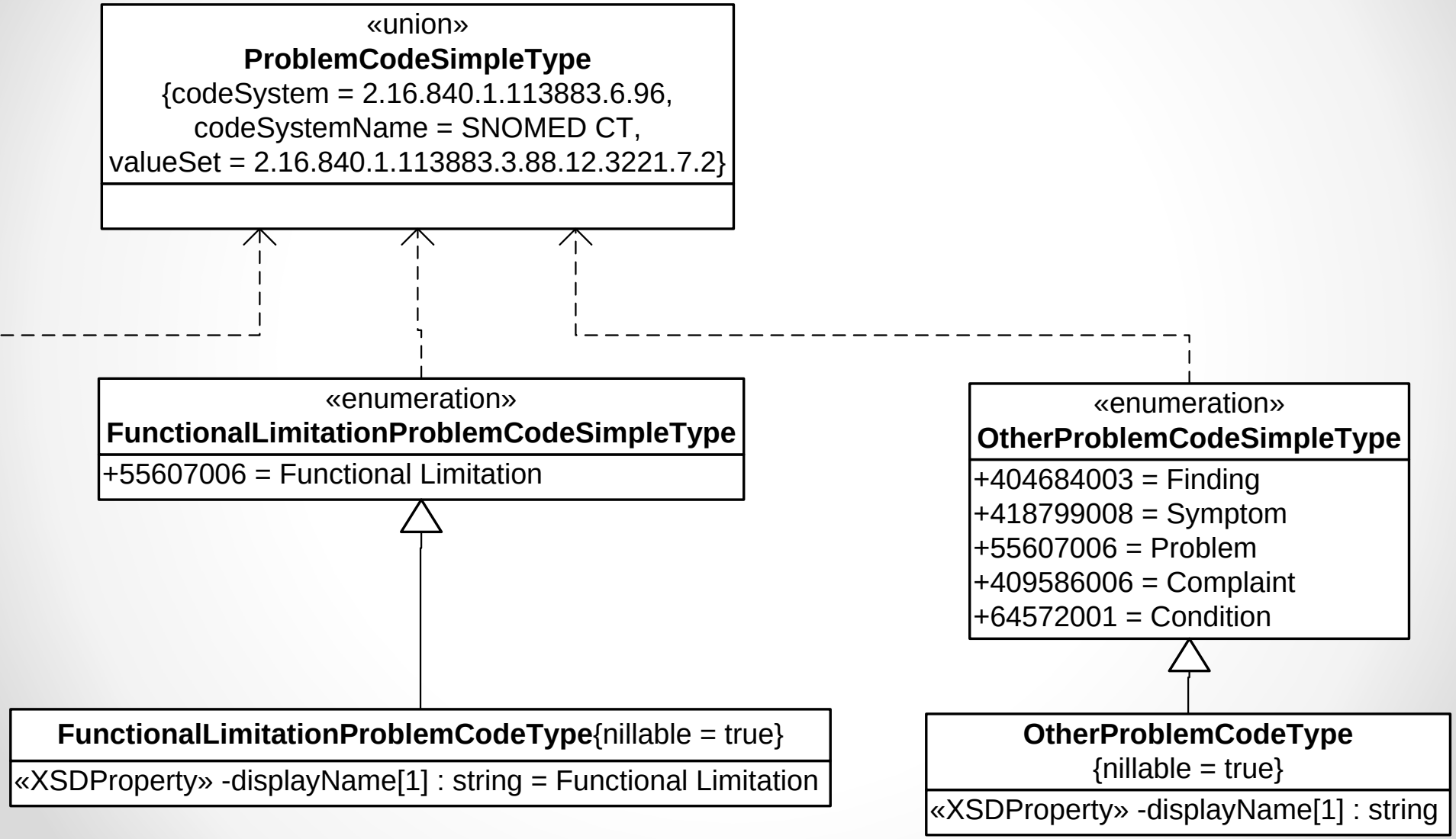
Compliments of: Tony Mallia, Edmond Scientific

Exchange Model



Compliments of: Tony Mallia, Edmond Scientific

Terminology (NIEM Code Lists)



Compliments of: Tony Mallia, Edmond Scientific

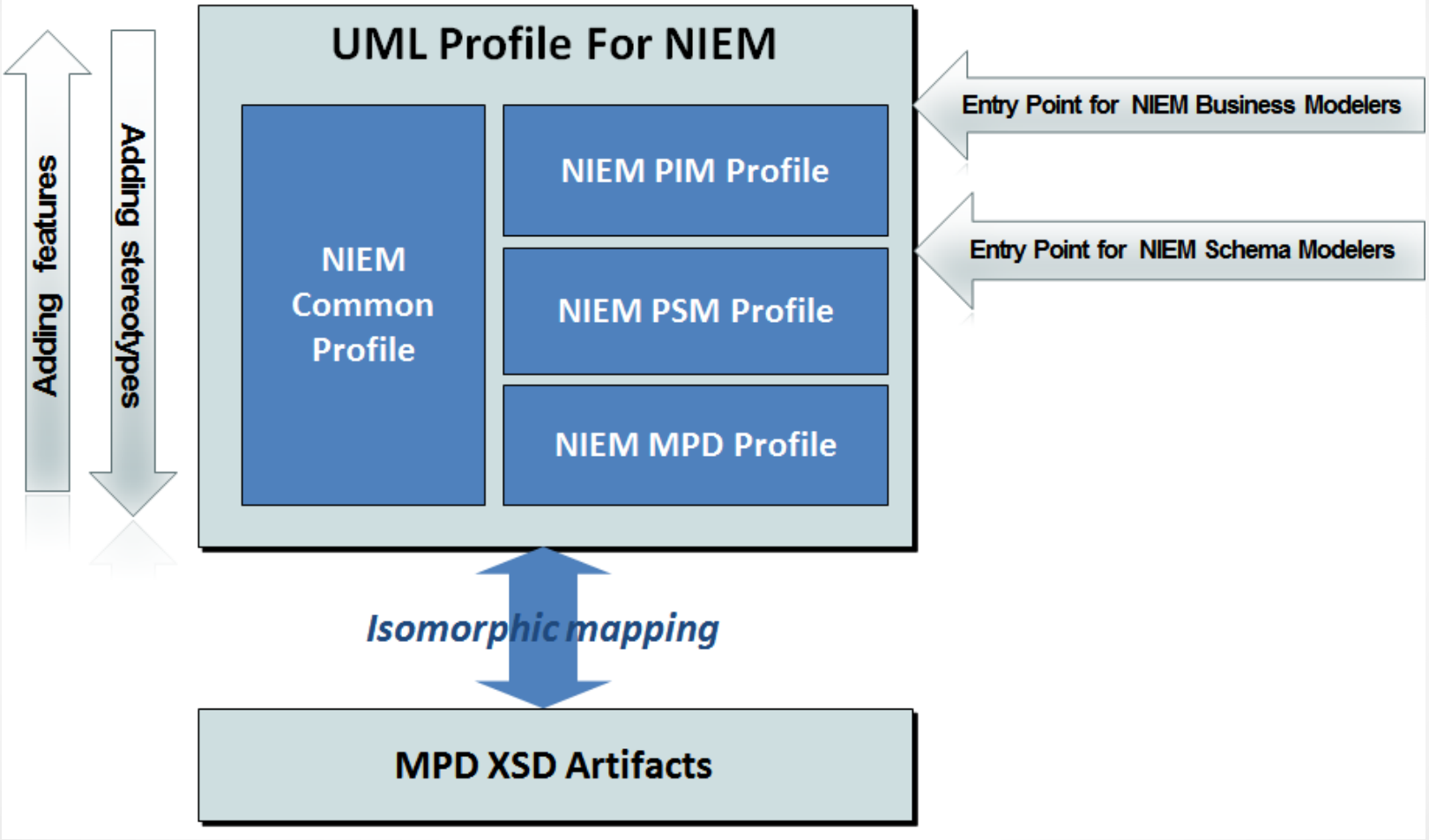
Lets Build (*most of*) This In NIEM-UML

Switching to UML tool

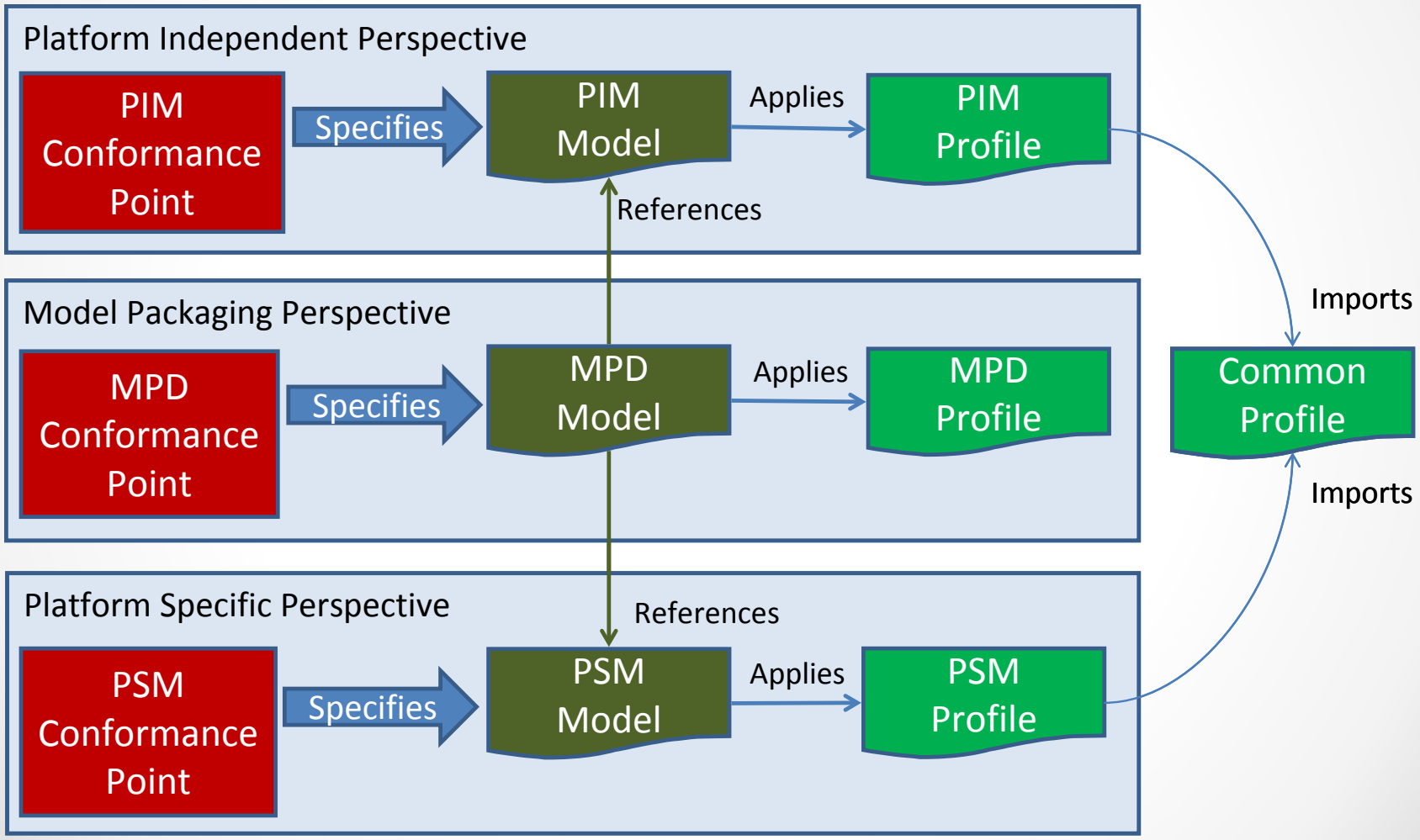
NIEM-UML Specification

NIEM Conformance

Support for PIM & PSM Perspectives

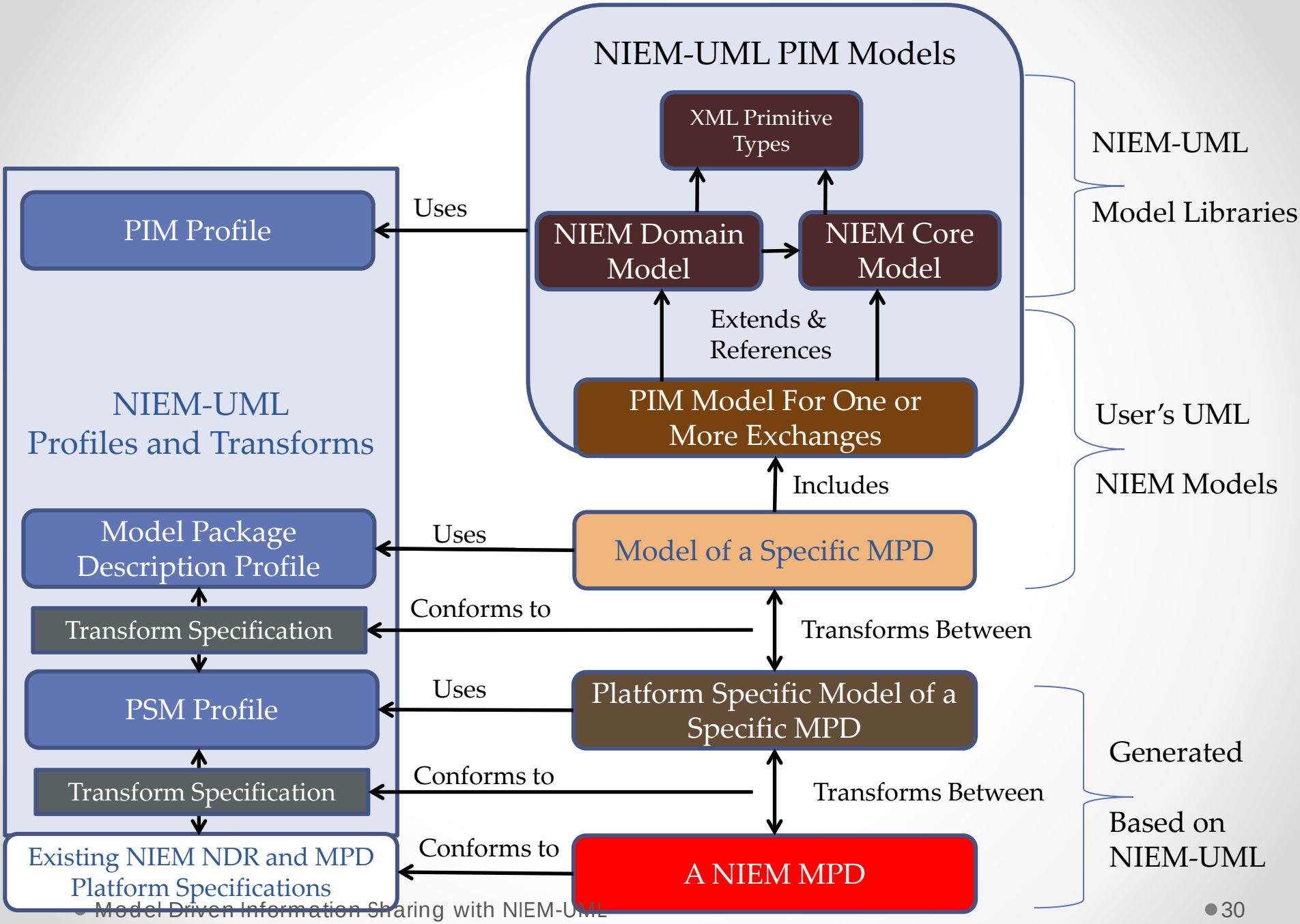


NIEM-UML Layered Architecture



What is the NIEM PIM Profile?

- A simplified subset of the Unified Modeling Language (UML)
- A set of UML constructs and stereotypes
  - Extends UML to represent NIEM **business concepts**
  - Business concepts are augmented with NIEM-Platform mapping information
  - Enforces NIEM rules by leveraging OCL – *a valid NIEM-UML model will produce a valid MPD*
- Representations correspond to commonly used UML patterns with well defined mapping to NIEM platform
- Provides a generalized information modeling environment not specific to NIEM schema
- Supports mapping to and from the NIEM platform, supporting and enforcing the NDR and MPD
  - E.g. name prefix and suffixes are added as specified by NIEM rules


NIEM-UML PIM Models
XML Primitive Types
NIEM Domain Model
NIEM Core Model
Extends & References
PIM Model For One or More Exchanges
NIEM-UML
Model Libraries
User's UML
NIEM Models
PIM Profile
Uses
NIEM-UML Profiles and Transforms
Includes
Model Package Description Profile
Uses
Model of a Specific MPD
Conforms to
Transform Specification
Transforms Between
PSM Profile
Uses
Platform Specific Model of a Specific MPD
Conforms to
Transform Specification
Transforms Between
Existing NIEM NDR and MPD Platform Specifications
Conforms to
A NIEM MPD
Generated
Based on NIEM-UML

NIEM-UML Platform Independent Model (PIM) By Example

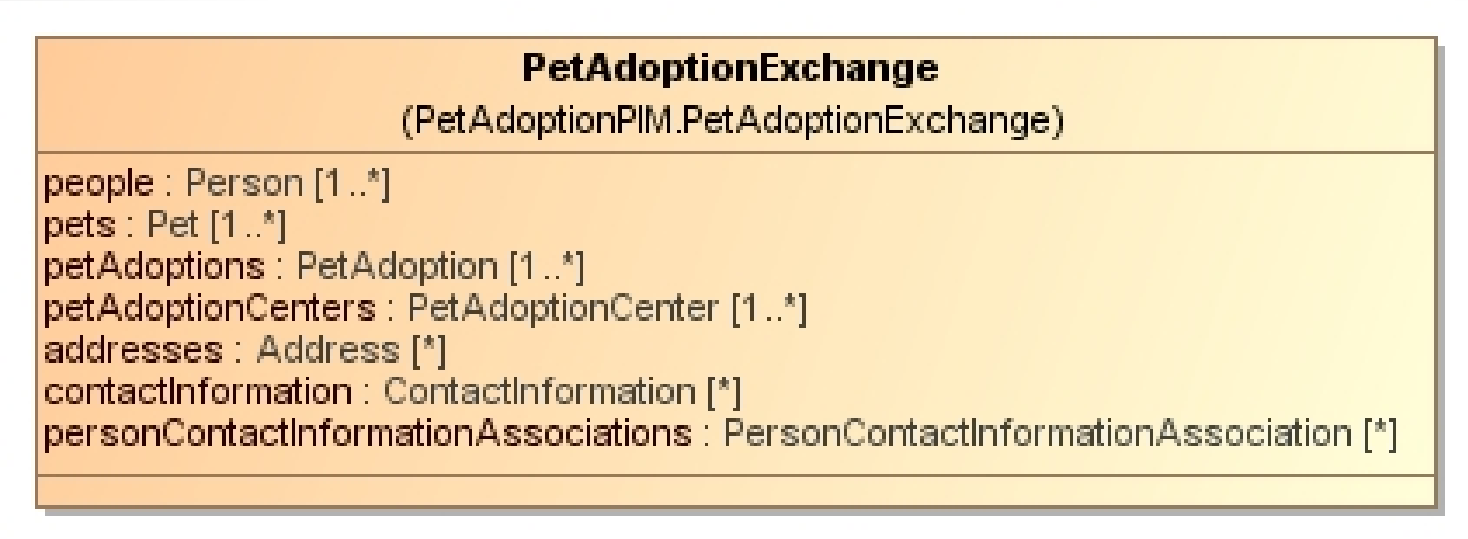
Pet Adoption Example

Data Exchange of adoptions by pet rescue centers

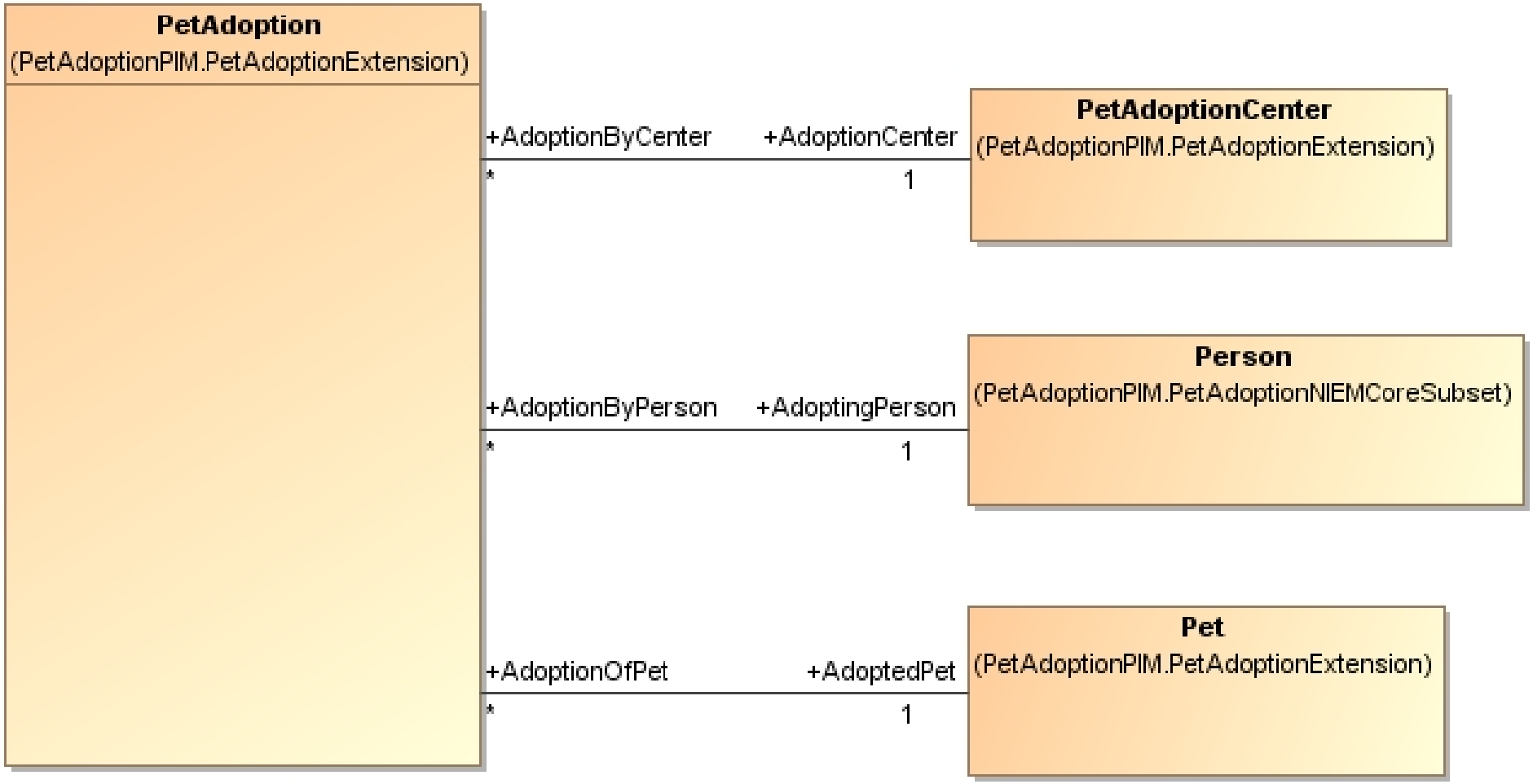
This is a high-level example, intended to provide a general idea of what a PIM looks like and what it provides.

Information to Exchange

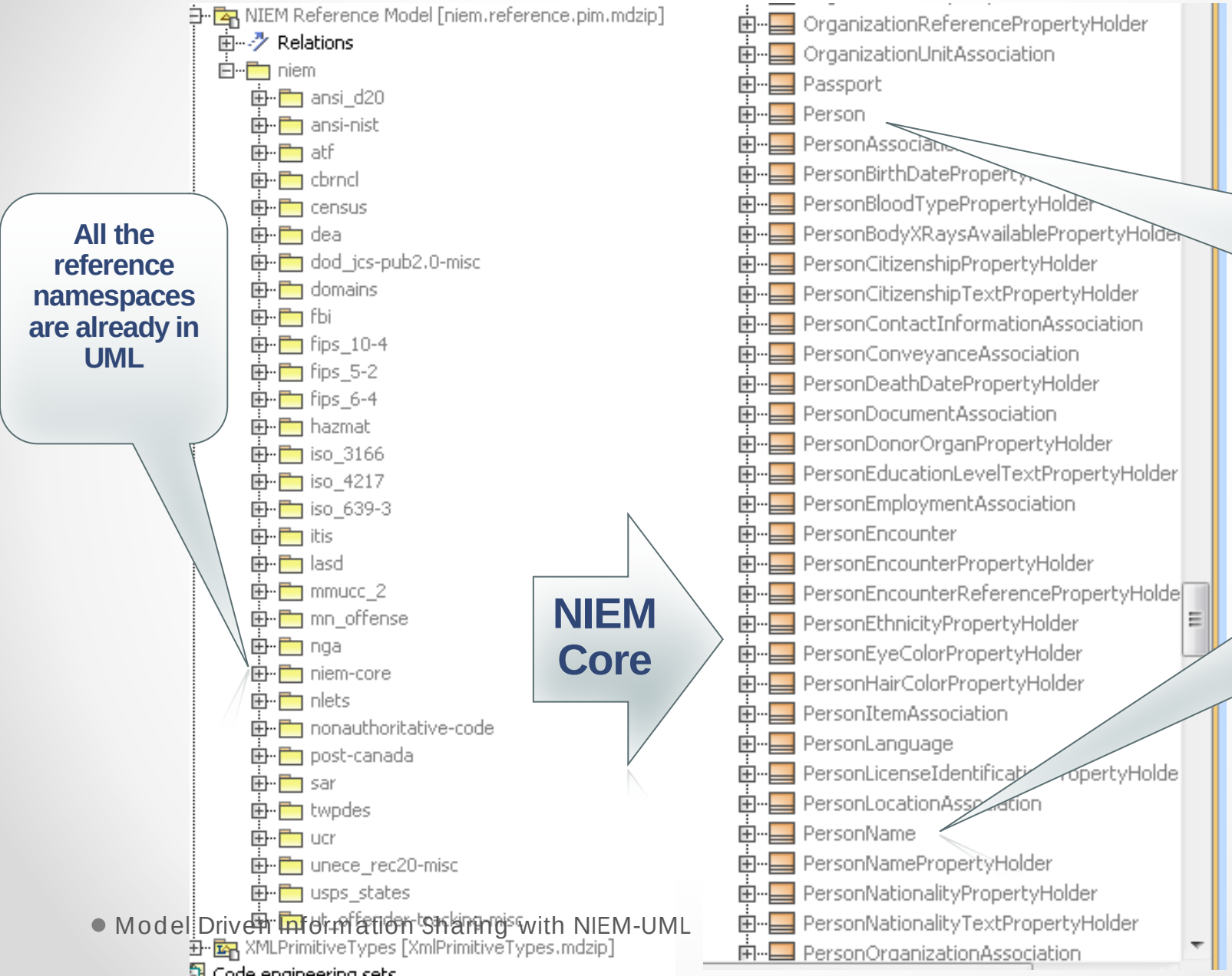
- Pet Adoptions
- Pets (Being adopted)
- People (Adopting)
- Pet Adoption Centers (Facilitating Adoptions)
- Addresses (Of people and adoption centers)
- Contact information (For people and adoption centers)
- Associations for contact information related to people



High-level information Model



This is NIEM - Reuse!



Model Reuse of NIEM Core

«References»

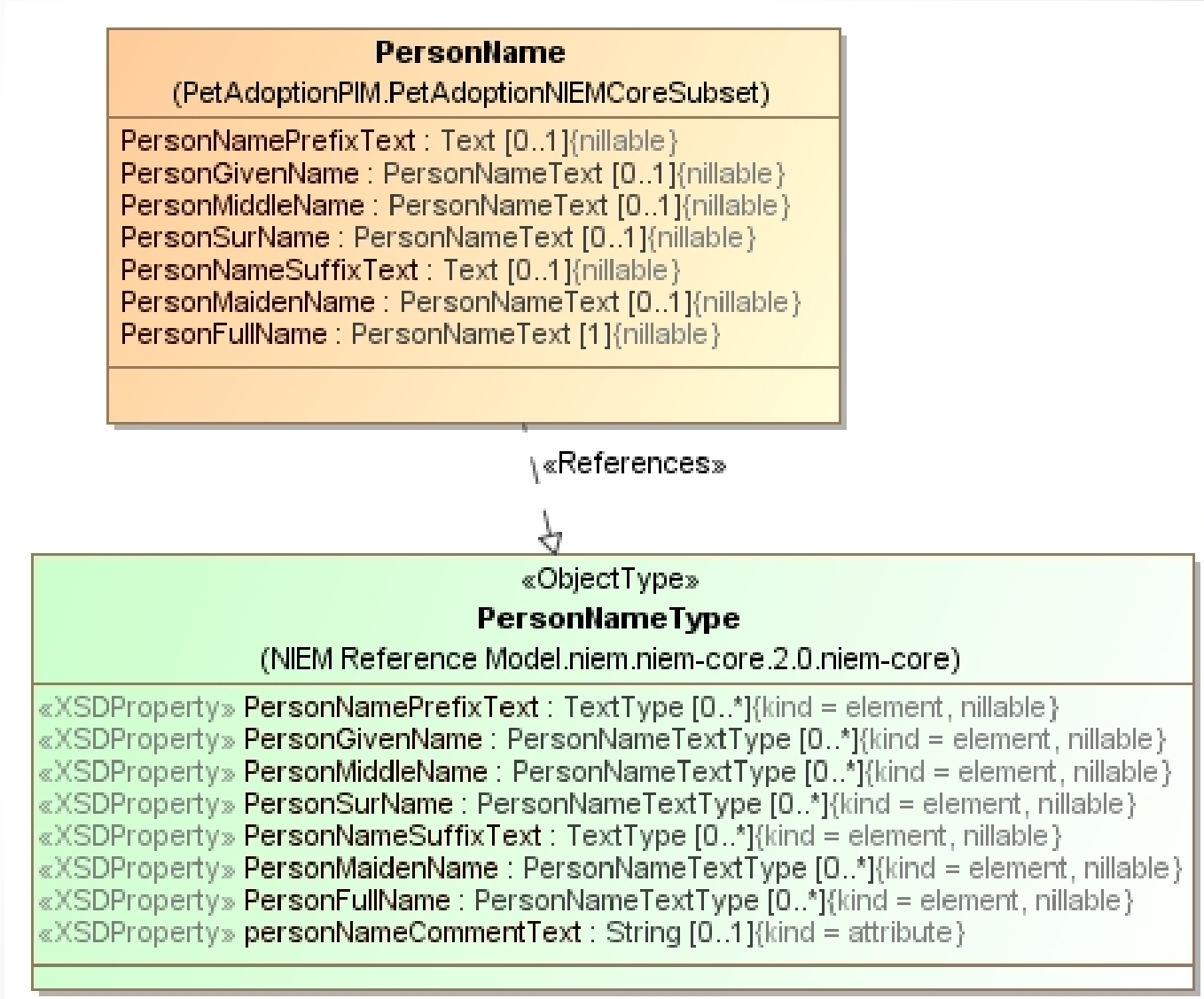
Create subsets of these in a subset namespace package – reference the reference classes

«ObjectType»
PersonType
(NIEM Reference Model.niem.niem-core.2.0.niem-core)

«XSDProperty» PersonAccentText : TextType [0..*]{kind = element, nillable}
«XSDProperty» PersonAgeDescriptionText : TextType [0..*]{kind = element, nillable}
«XSDProperty» PersonAgeMeasure : TimeMeasureType [0..*]{kind = element
«XSDProperty» PersonAlternateName : PersonNameType [0..*]{kind =
«XSDProperty» PersonBirthDate : DateType [0..*]{kind = element, nillable}
«XSDProperty» PersonBirthLocation : LocationType [0..*]{kind = element, nillable}
«XSDProperty» PersonBloodType [0..*]{kind = element, nillable = false}
«XSDProperty» PersonBodyXRaysAvailable [0..*]{kind = element, nillable = false}
«XSDProperty» PersonBuildText : TextType [0..*]{kind = element, nillable}
«XSDProperty» PersonCapability : CapabilityType [0..*]{kind = element, nillable}
«XSDProperty» PersonCircumcisionIndicator : Boolean [0..*]{kind = element, nillable}
«XSDProperty» PersonCitizenship [0..*]{kind = element, nillable = false}
«XSDProperty» PersonClothing : ClothingType [0..*]{kind = element, nillable}
«XSDProperty» PersonComplexionText : TextType [0..*]{kind = element, nillable}
«XSDProperty» PersonDeathDate : DateType [0..*]{kind = element, nillable}
«XSDProperty» PersonDependentQuantity : QuantityType [0..*]{kind = element, nillable}
«XSDProperty» PersonDescriptionText : TextType [0..*]{kind = element, nillable}
«XSDProperty» PersonDigitalImage : ImageType [0..*]{kind = element, nillable}

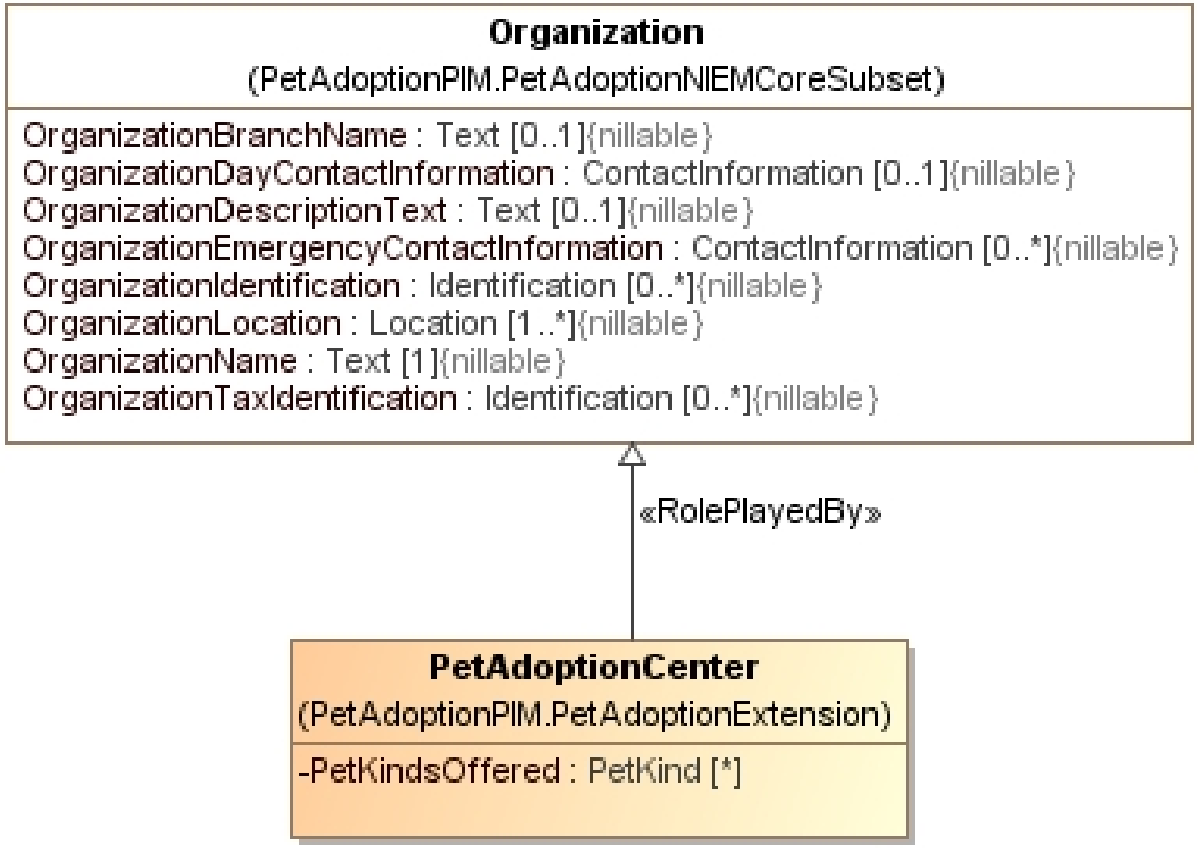
Find what you want from the reference namespaces – can copy/paste

Repeat as required



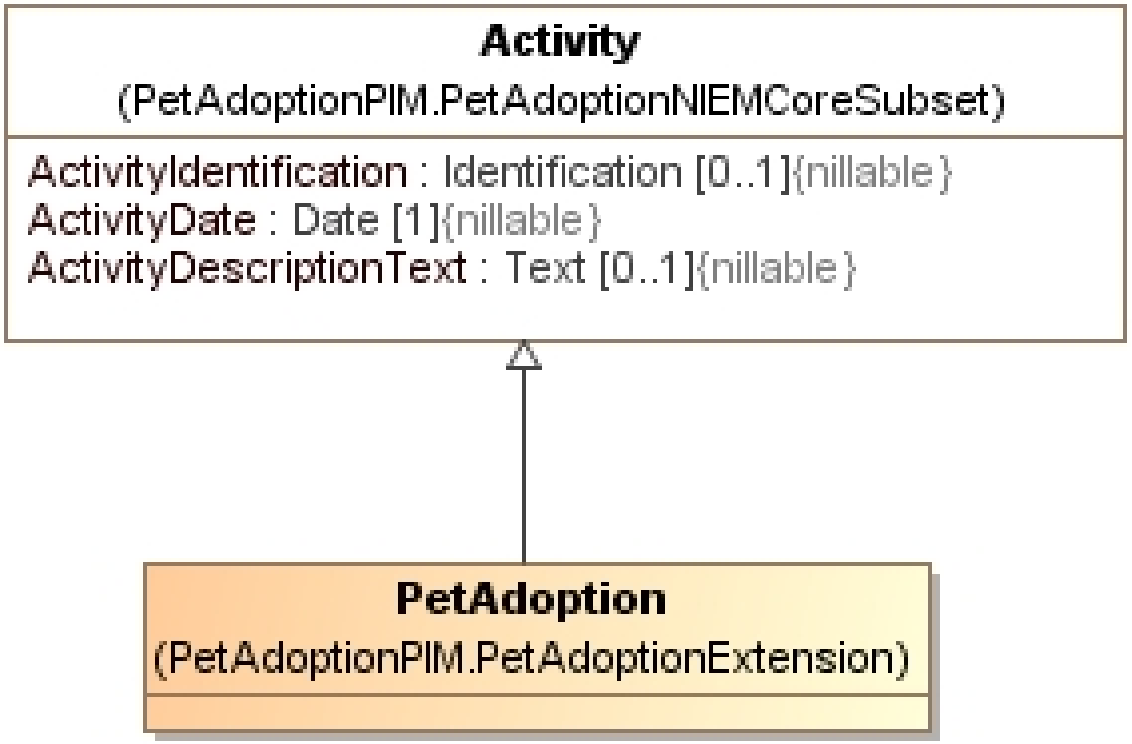
Roles of organizations

- What is an “Adoption Center”?
- It is a kind of organization
- But perhaps more properly a “role” an organization plays, as they could play other roles as well
- This is one representation of NIEM roles
- In the NIEM PSM, this becomes a property prefixed by “RoleOf”



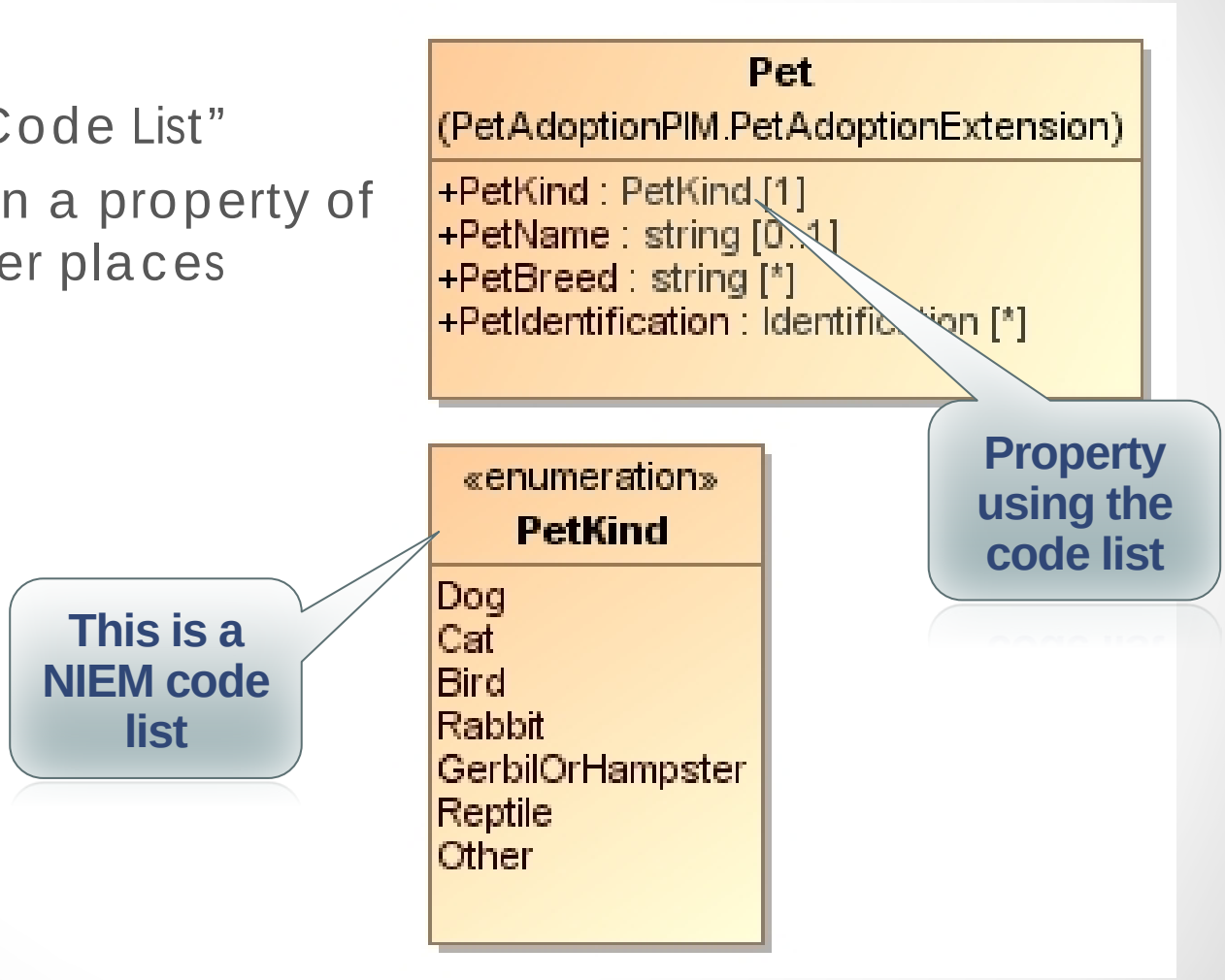
Adoptions as a NIEM Activity

- An adoption is a kind of activity
- We can reuse this from NIEM-Core as well

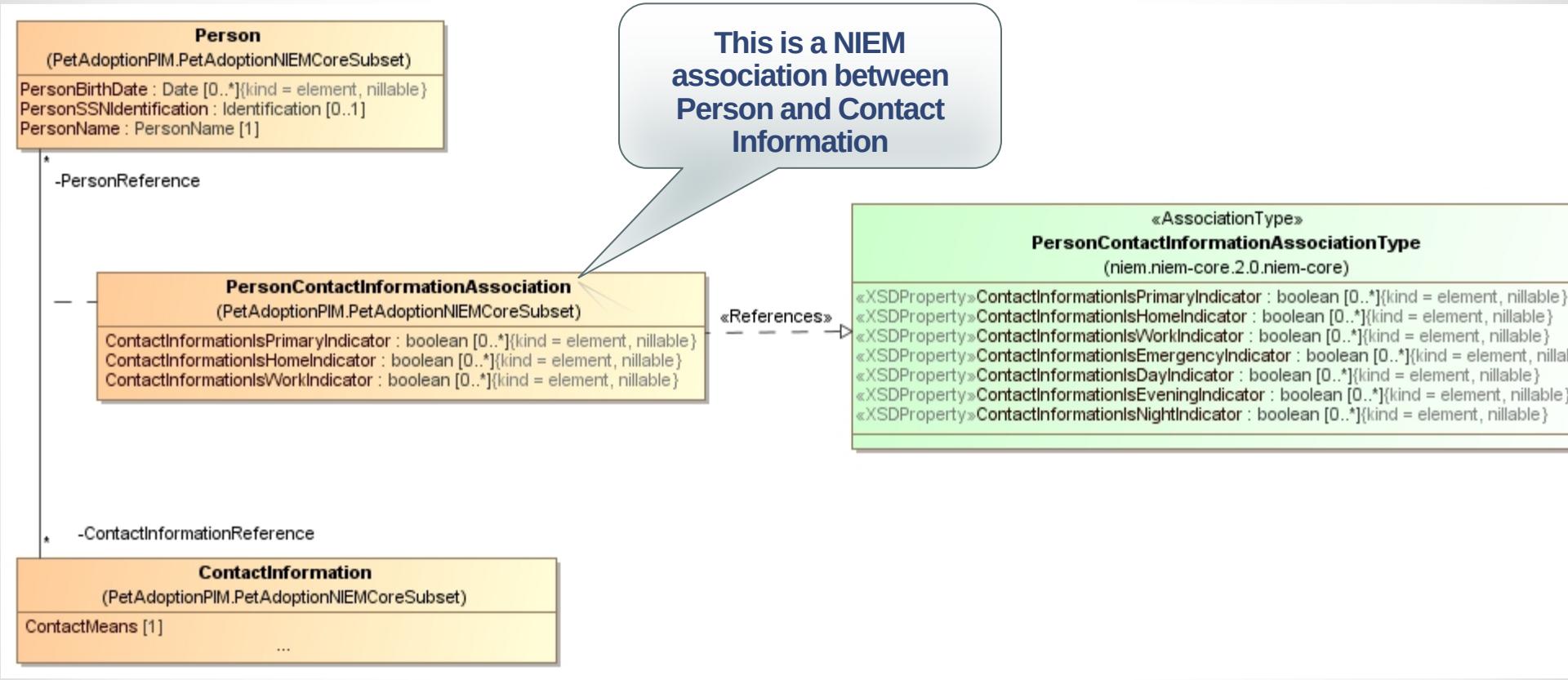


What Kinds Of Pets Are Adopted?

- PetKind is a NIEM "Code List"
- This can be a used in a property of a pet as well as other places

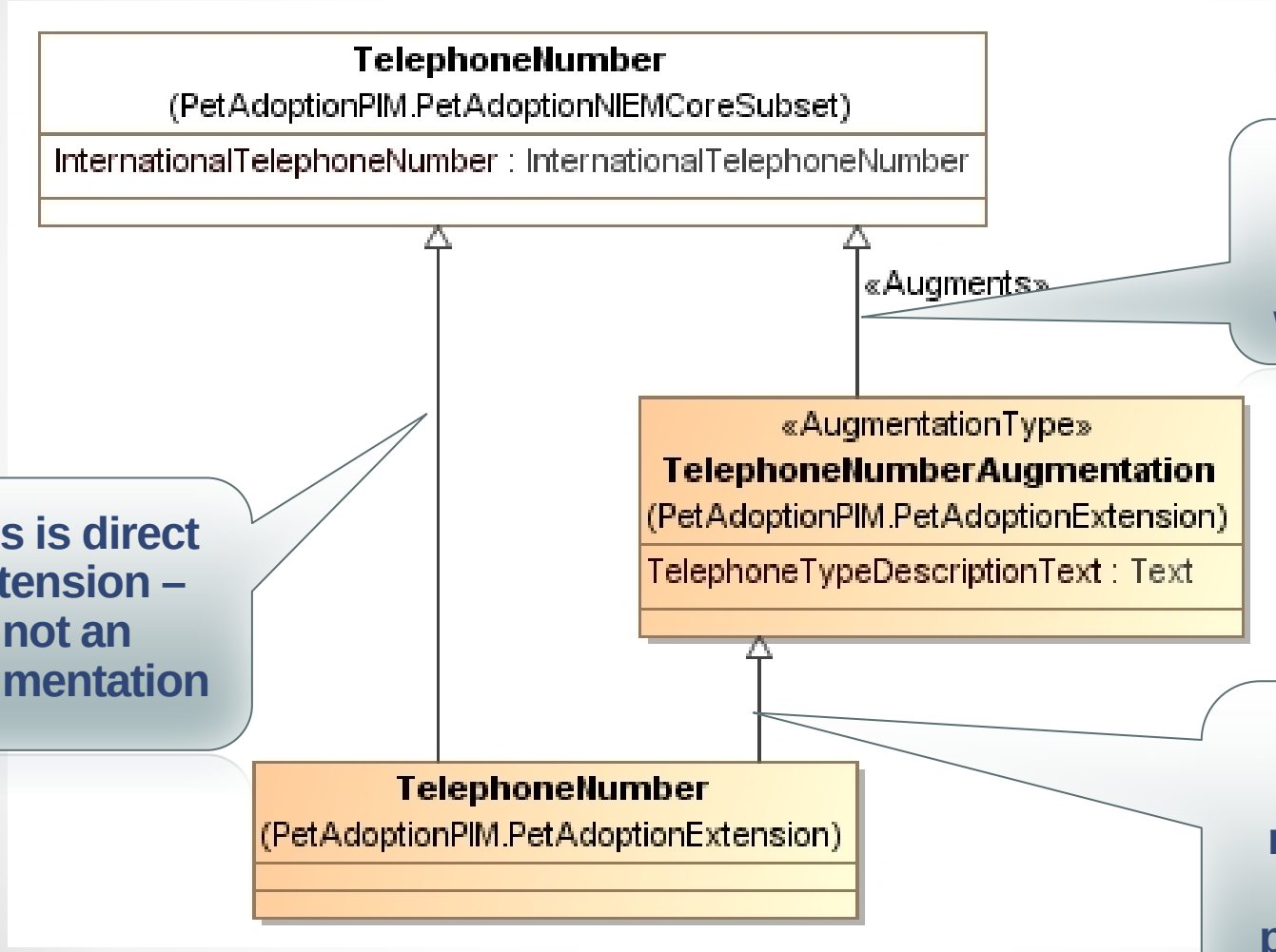


NIEM Associations



Associations Connect Objects – in this case people and contact information

Augmentations – Phone Number ++

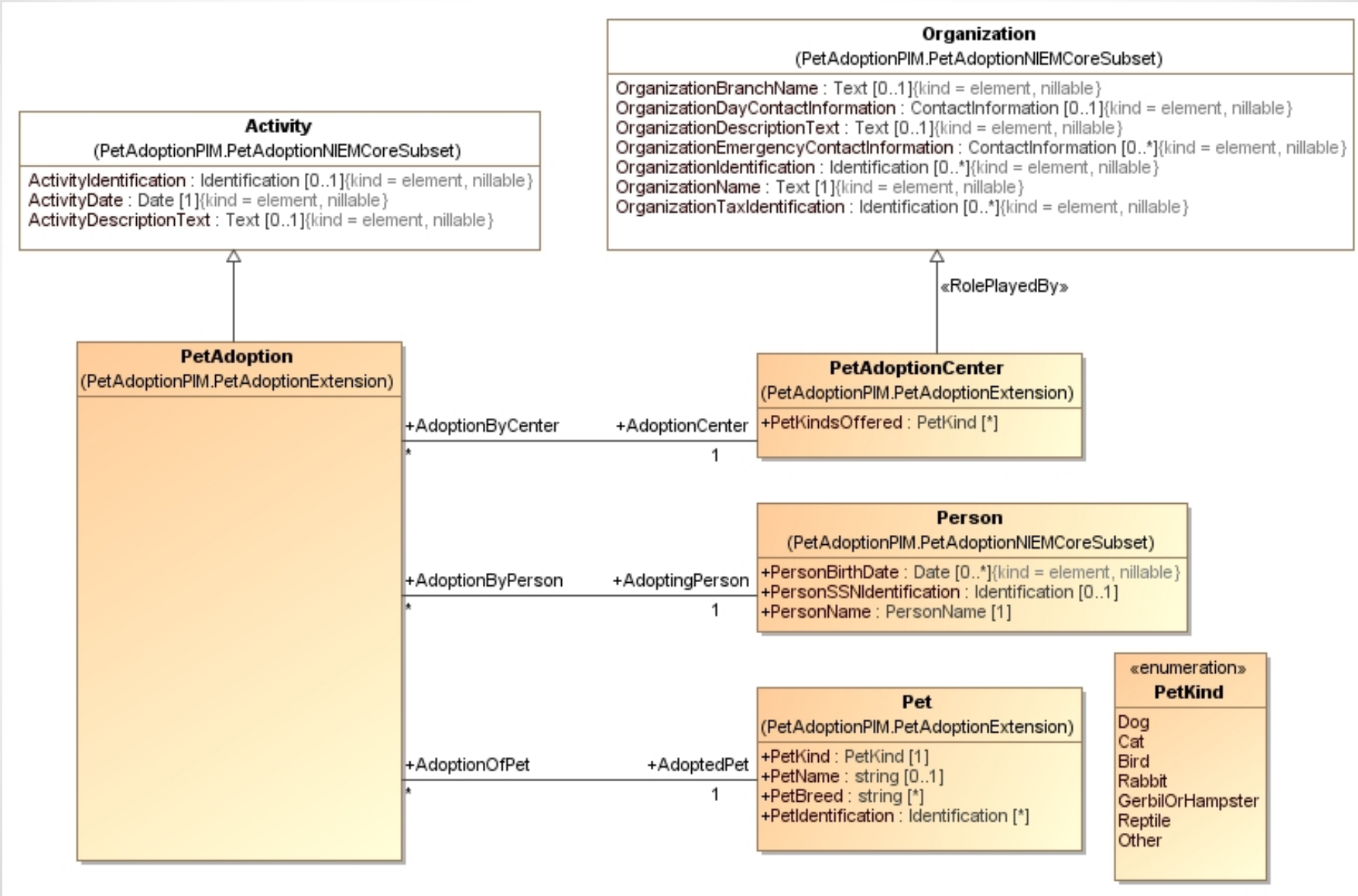


Optionally, an augmentation can be restricted to what it "applies to"

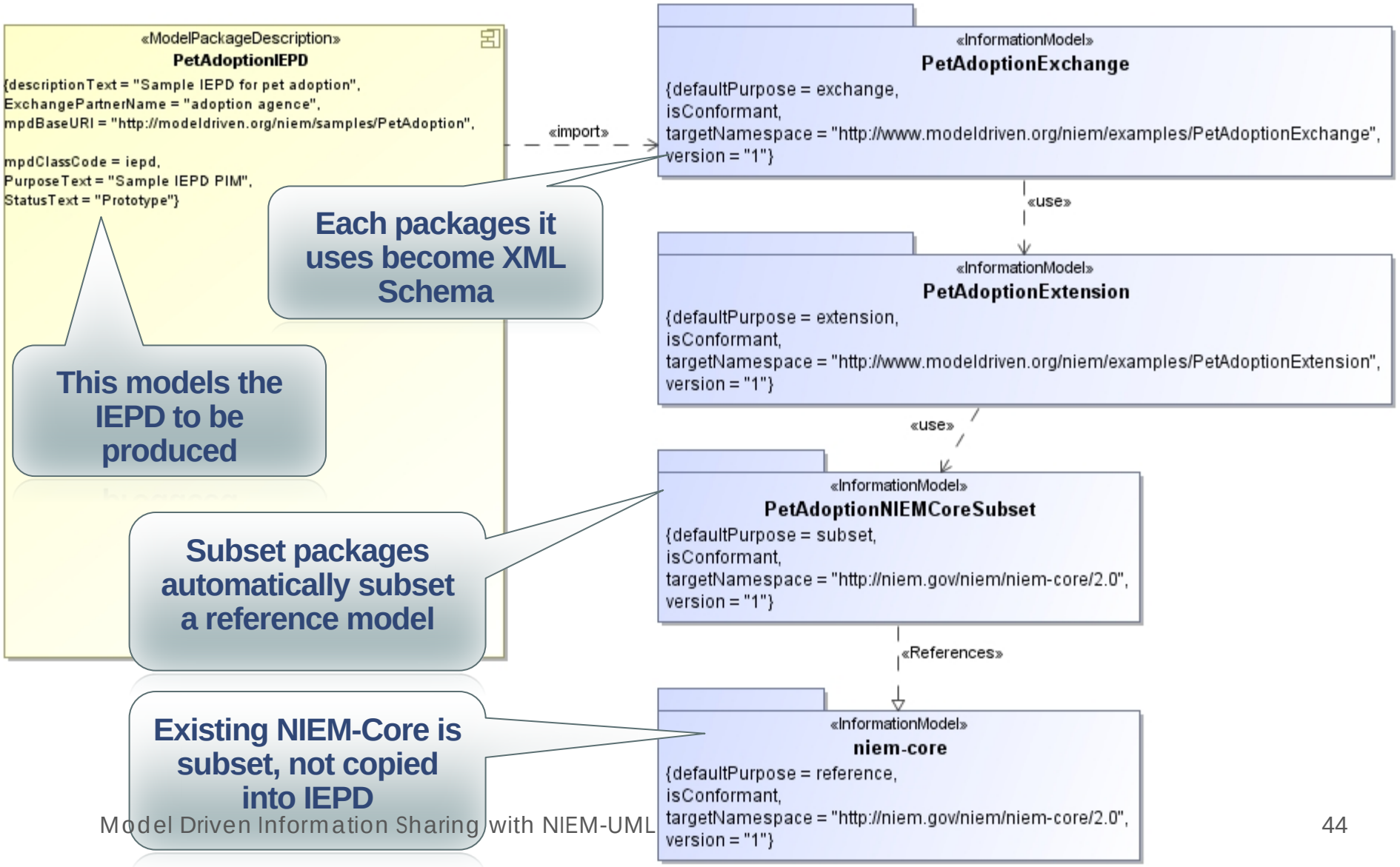
This is direct extension – not an augmentation

Inheriting an augmentation results in a NIEM augmentation property, not XSD extension

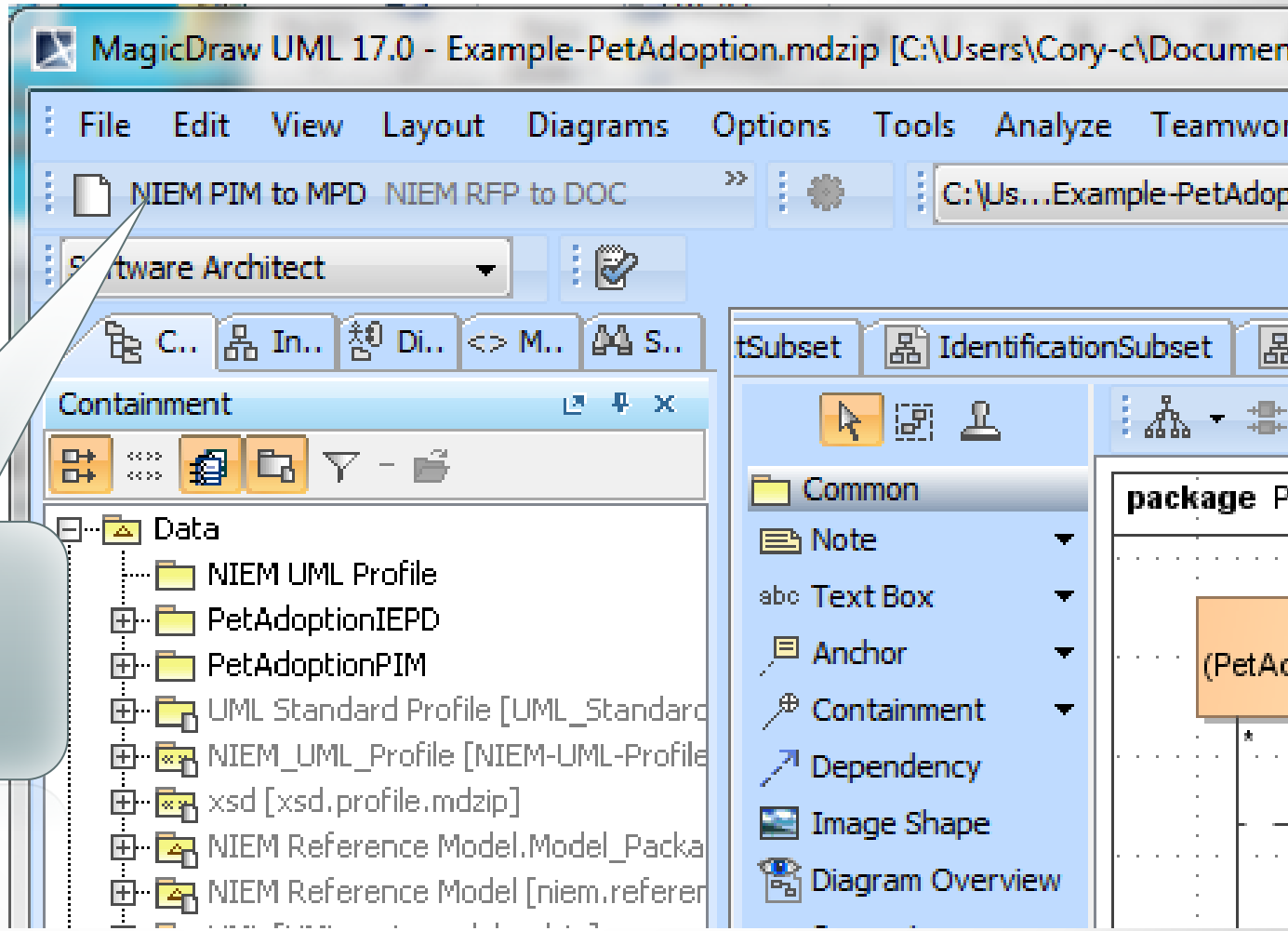
Completed High-Level Model



Adding the IEPD Metadata

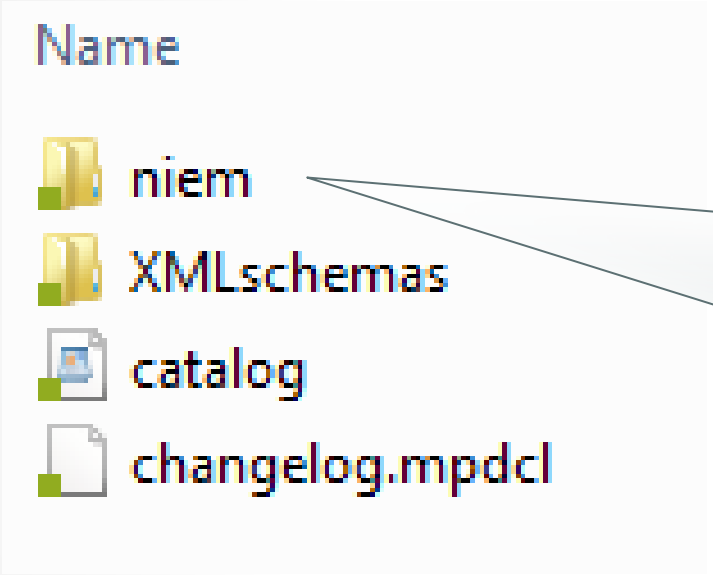


Create the IEPD from the model



You then tell your UML and/or MDA tool to make the IEPD

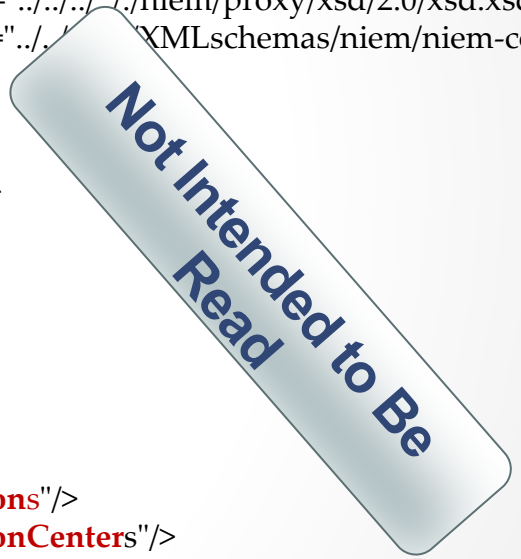
All the IEPD artifacts are then created by the MDA Automation



Given a model that satisfies the NIEM-UML profile a valid and complete IEPD is guaranteed to come out.

MDA Automation Also creates NIEM Conformant XML Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:Q1="http://www.modeldriven.org/niem/examples/PetAdoptionExtension" xmlns:i="http://niem.gov/niem/ap
 <xsd:import namespace="http://niem.gov/niem/appinfo/2.1" schemaLocation="../../../../../niem/appinfo/2.1/appinfo.xsd"/>
 <xsd:import namespace="http://niem.gov/niem/structures/2.0" schemaLocation="../../../../../niem/structures/2.0/structures.xsd"/>
 <xsd:import namespace="http://www.modeldriven.org/niem/examples/PetAdoptionExtension" schemaLocation="../../../../../XMI
 <xsd:import namespace="http://www.modeldriven.org/niem/examples/PetAdoptionExchange" schemaLocation="../../../../../XMI
 <xsd:import namespace="http://niem.gov/niem/appinfo/2.0" schemaLocation="../../../../../niem/appinfo/2.0/appinfo.xsd"/>
 <xsd:import namespace="http://niem.gov/niem/proxy/xsd/2.0" schemaLocation="../../../../../niem/proxy/xsd/2.0/xsd.xsd"/>
 <xsd:import namespace="http://niem.gov/niem/niem-core/2.0" schemaLocation="../../../../XMLschemas/niem/niem-core/2.0/nie
 <xsd:complexType abstract="false" name="PetAdoptionExchangeType">
  <xsd:annotation>
   <xsd:appinfo>
    <i:Base i:name="Object" i:namespace="http://niem.gov/niem/structures/2.0"/>
   </xsd:appinfo>
  </xsd:annotation>
  <xsd:complexContent>
   <xsd:extension base="s:ComplexObjectType">
    <xsd:sequence>
     <xsd:element maxOccurs="unbounded" minOccurs="1" ref="tns:People"/>
     <xsd:element maxOccurs="unbounded" minOccurs="1" ref="tns:Pets"/>
     <xsd:element maxOccurs="unbounded" minOccurs="1" ref="tns:PetAdoptions"/>
     <xsd:element maxOccurs="unbounded" minOccurs="1" ref="tns:PetAdoptionCenters"/>
     <xsd:element maxOccurs="unbounded" minOccurs="0" ref="tns:Addresses"/>
     <xsd:element maxOccurs="unbounded" minOccurs="0" ref="tns:ContactInformation"/>
     <xsd:element maxOccurs="unbounded" minOccurs="0" ref="tns:PersonContactInformationAssociations"/>
    </xsd:sequence>
   </xsd:extension>
  </xsd:complexContent>
 </xsd:complexType>
 <xsd:element abstract="false" name="People" nillable="false" type="nc:PersonType"/>
 <xsd:element abstract="false" name="Pets" nillable="false" type="Q1:PetType"/>
```



As part of this process

- The model is fully validated with "OCL Constraints" for NIEM Rules
- The produced PSM is also validated
- Many NIEM rules are taken care of automatically in the transformation rules such as Naming and Global elements
- The resulting IEPD is either valid or any problems noted (how being tool dependent)
- There are still a few subjective NDR Rules that can't be tested by the automation

How much is there to learn?

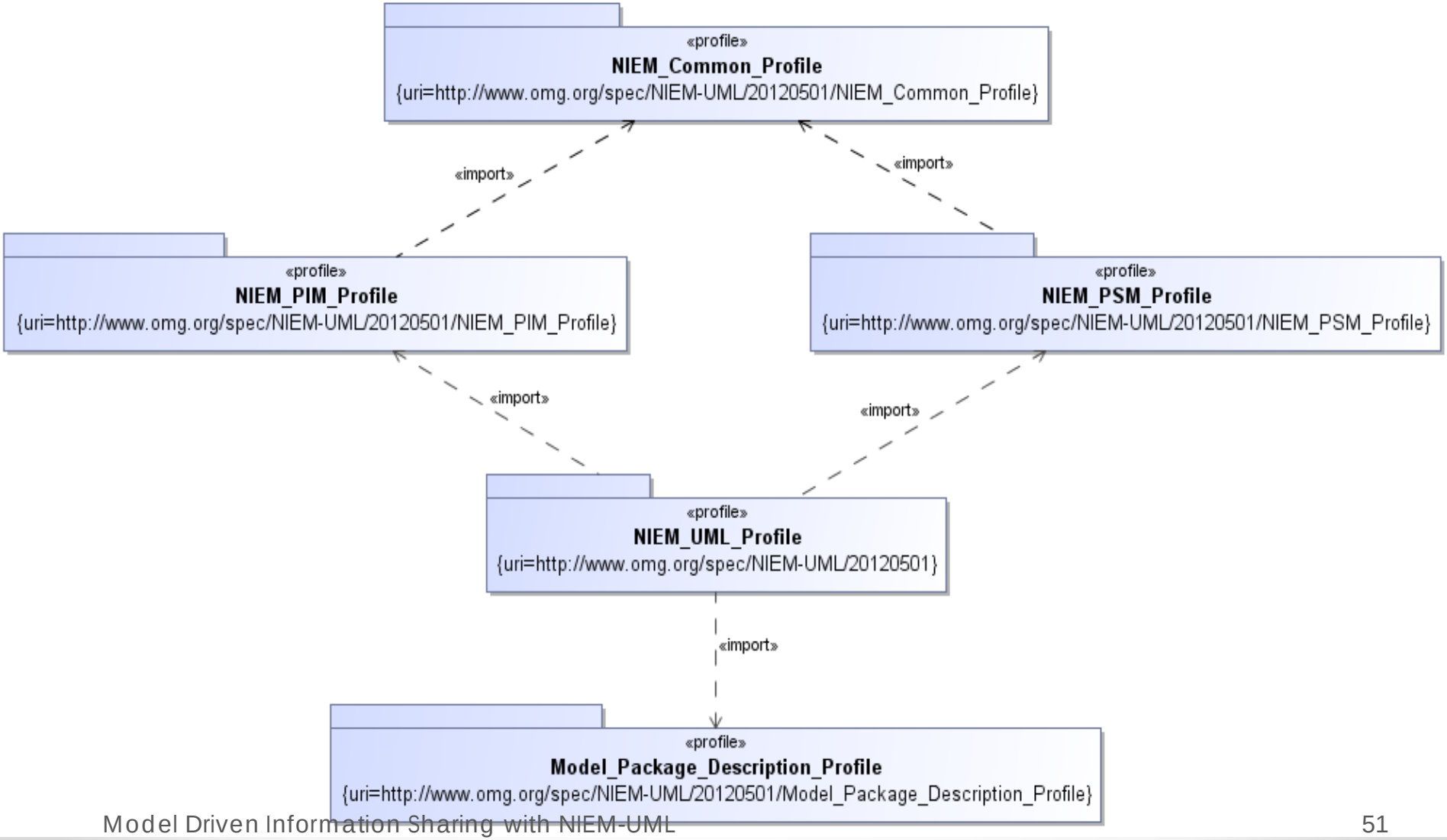
- NIEM Logical Concepts
  - Not the XSD and NDR Details
- The PIM and Common Profile
- The Model Package Description Profile
- A UML Tool (Class diagram subset)

Note: A Forester report estimated that 71% of software development teams already use UML

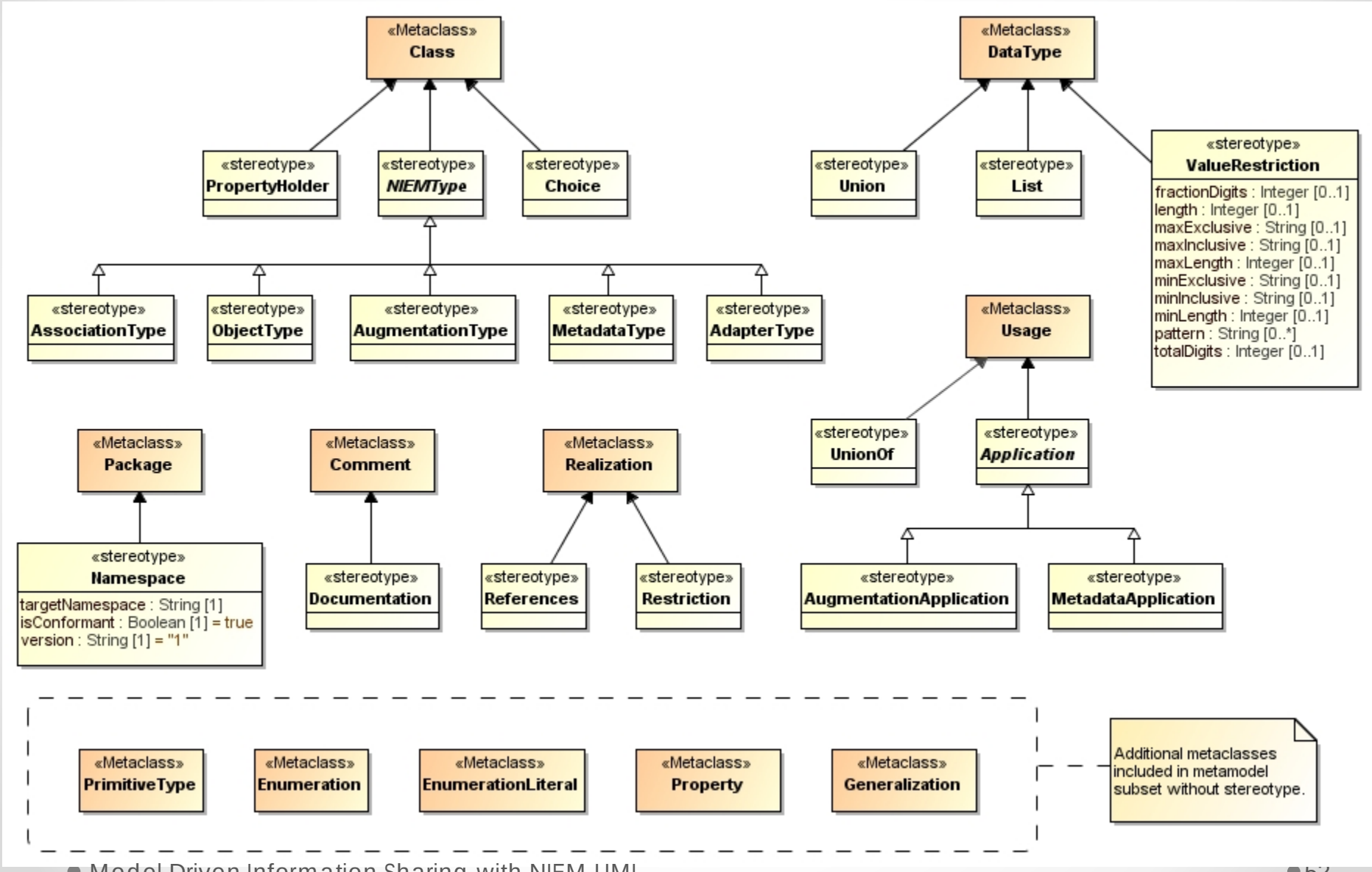
NIEM-UML Profiles

Getting Down to Detail

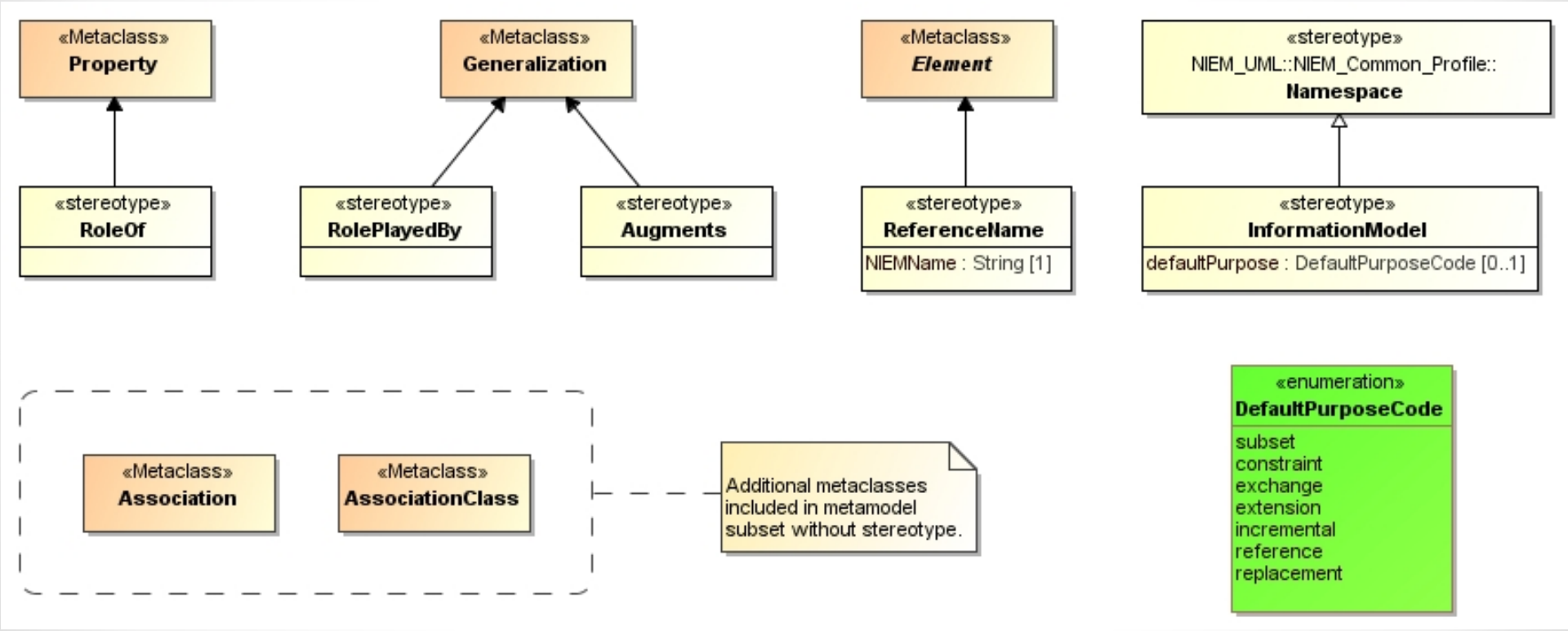
NIEM-UML Profile Structure



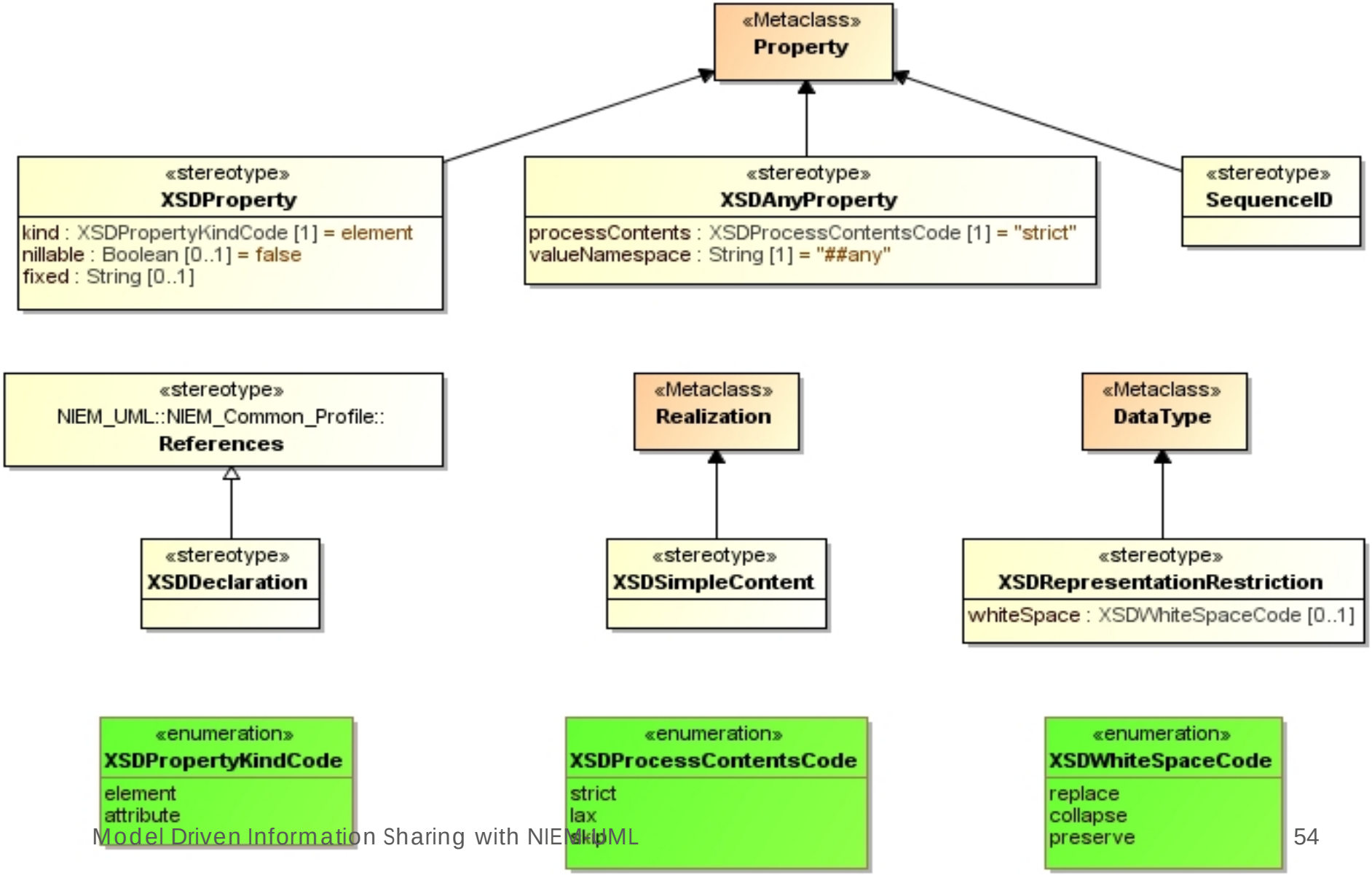
Common Profile



Platform Independent Profile



Platform Specific Profile

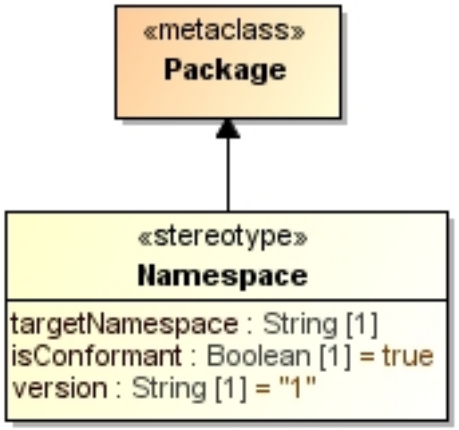


Namespaces

Namespace (Package)

- Namespace represents a namespace, which is implemented in XML Schema as a "schema" schema component.
- Namespace includes the following attributes: isConformant, namespace, and version.

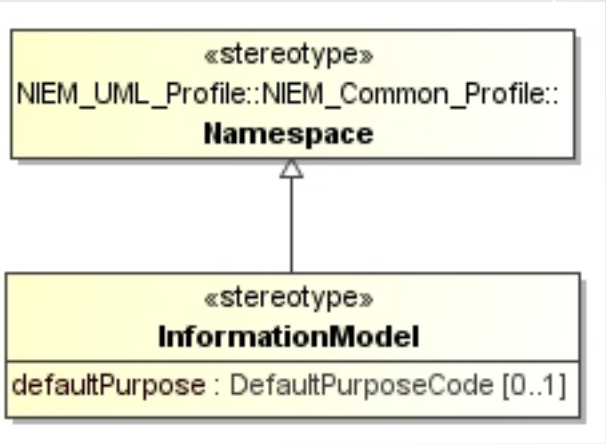
Example

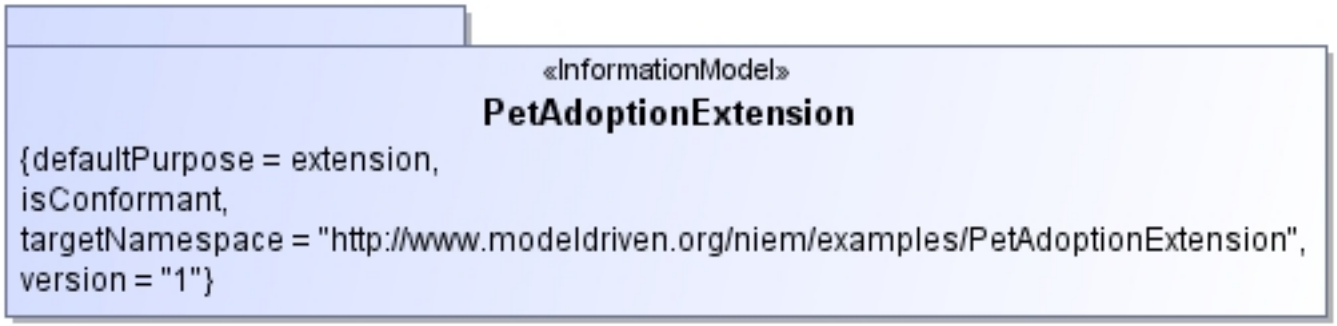


Information Models (PIM Only)

InformationModel (Package)

- Extends Namespace for the PIM
- Adds "default purpose"

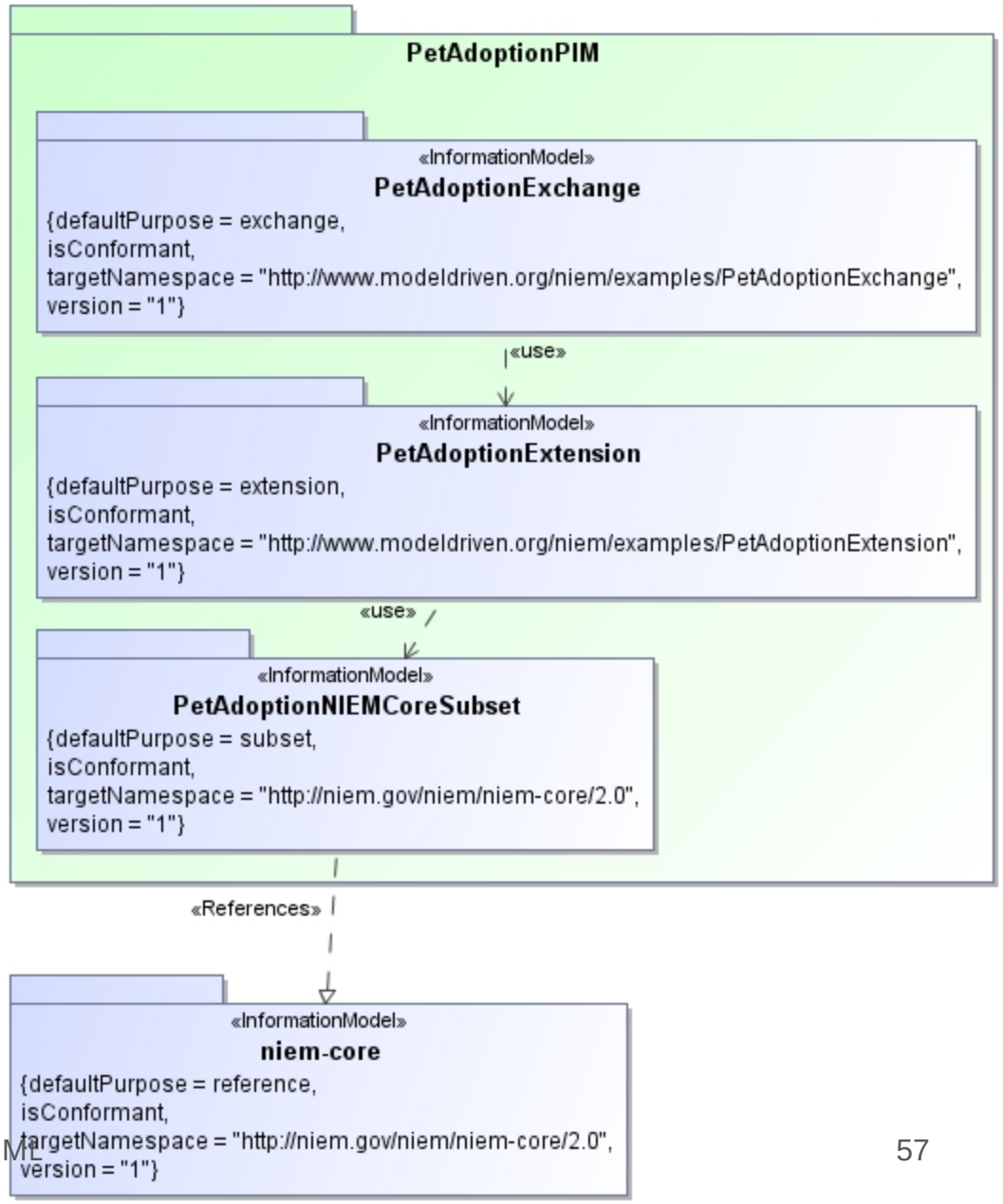




Relations between namespaces (PIM Only)

InformationModel (Package)

- Packages can “use” other packages
- Packages can <<Reference>> and subset reference packages
- Supports MPD/IEPD packaging



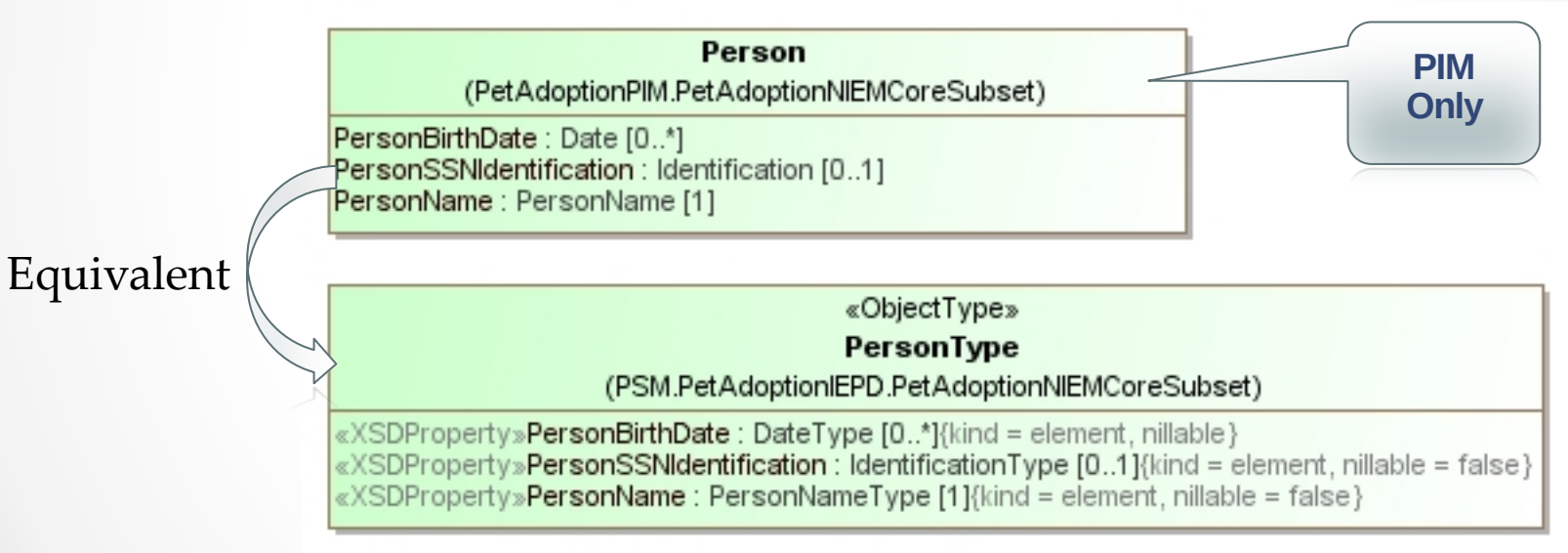
Representation of Complex Types

NIEM Complex Type	Representation in the NIEM-PIM
Object Type	Class – no stereotype is required, Object Type is the default.
Role Type	Use of <<RoleOf>> association and or <<RolePlayedBy>> generalization referencing the complex signifies that type is a role.
Association Type	<<AssociationType>> stereotype applied to the complex type or a UML association class.
Metadata Type	<<MetadataType>> stereotype applied to the complex type.
Augmentation Type	<<Augmentation>> stereotype applied to the complex type.
Adapter Type	<<AdapterType>> stereotype applied to the complex type. The initial version of the PIM does not include adapter types, these will be added in the final specification.

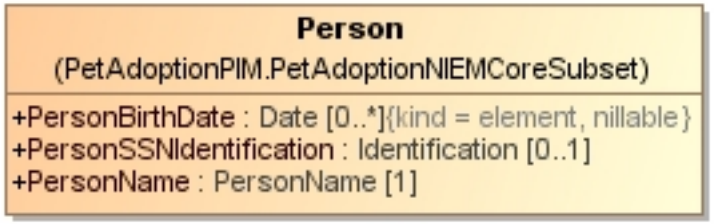
NIEM Object Types

NIEM Object Types

An object type is represented as a UML class, no stereotype is required. (PIM Only). In a PSM an object type must be stereotyped an <<ObjectType>>



Your basic "thing" in XML




```
<xsd:complexType name="PersonType">
<xsd:annotation>
<xsd:appinfo>
<i:Base i:name="Object" i:namespace="http://niem.gov/niem/structures/2.0"/>
</xsd:appinfo>
<xsd:documentation>A data type for a human being.</xsd:documentation>
</xsd:annotation>
<xsd:complexContent>
<xsd:extension base="s:ComplexObjectType">
<xsd:sequence>
<xsd:element maxOccurs="1" minOccurs="1" ref="nc:PersonBirthDate"/>
<xsd:element maxOccurs="1" minOccurs="1" ref="nc:PersonName"/>
<xsd:element maxOccurs="1" minOccurs="1" ref="nc:PersonSSNIdentification"/>
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
```

Elements are used in XSD data structures

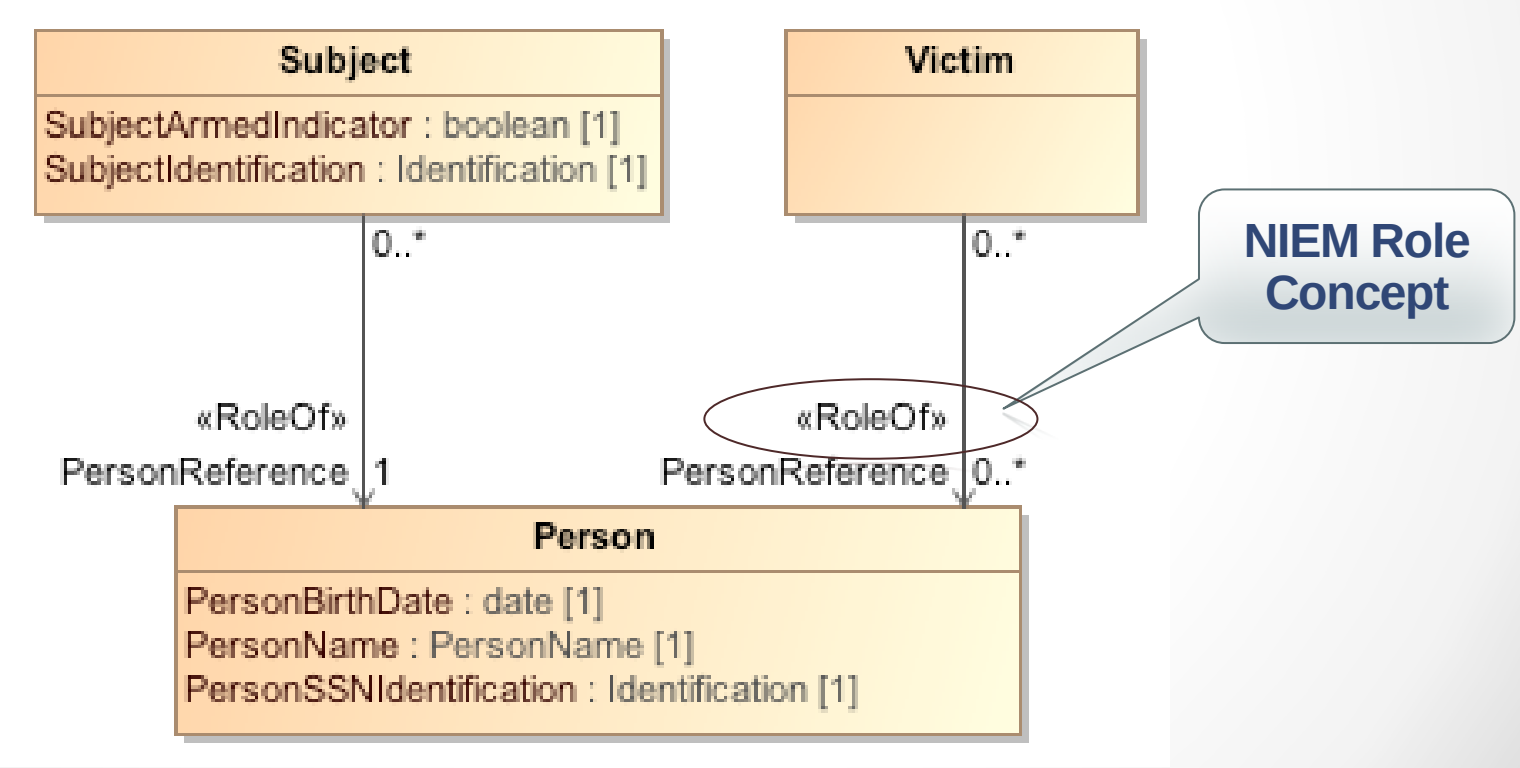
Every element becomes global for reuse (

```
<xsd:element name="PersonBirthDate" nillable="false" type="nc:DateType">
<xsd:annotation>
<xsd:documentation>A date a person was born.</xsd:documentation>
</xsd:annotation>
</xsd:element>
```

NIEM Roles

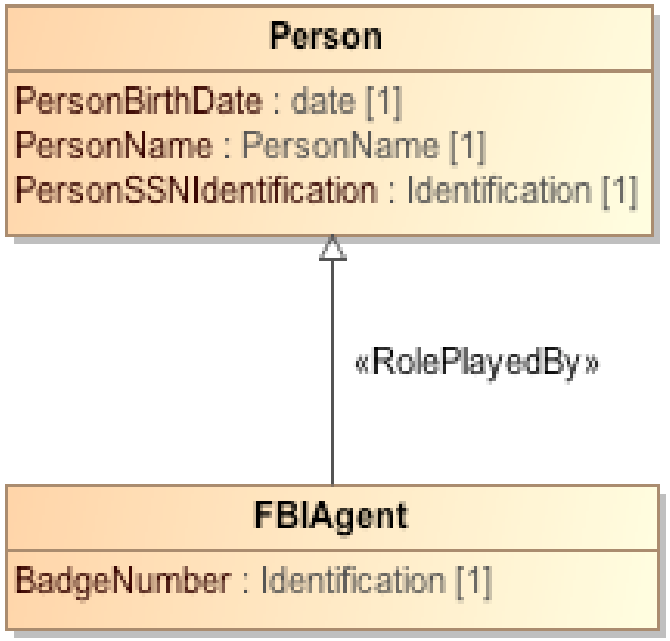
NIEM Roles

UML also has the capability to represent roles in their simpler form as UML association ends (The names on the ends of lines in a class diagram) or properties. To represent roles that are complex types a class or data type is used.



NIEM Roles

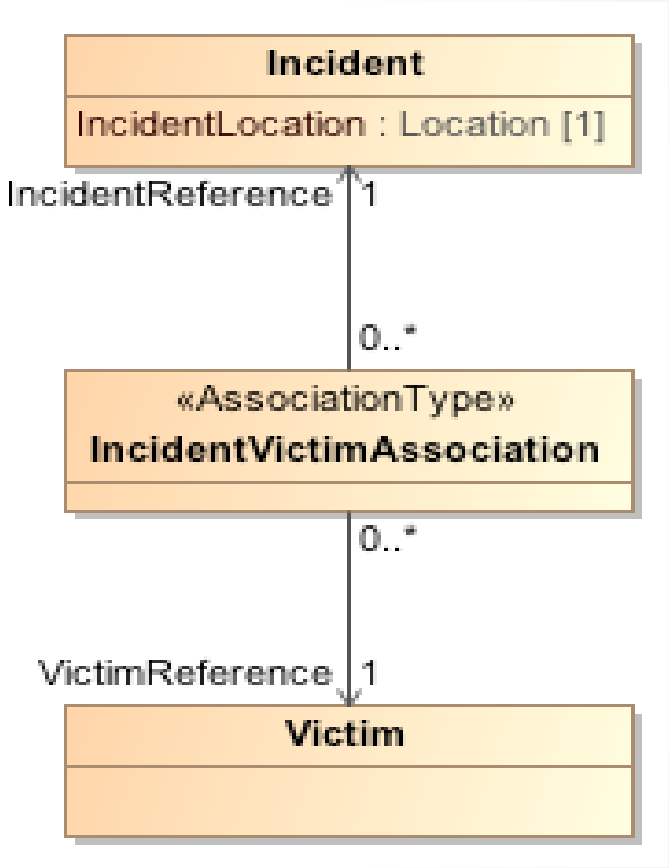
RolePlayedBy
Where a role represents an optional extension to an object type a <<RolePlayedBy>> stereotype of generalization may be used – maps to a NIEM RoleOf property.



NIEM Associations

NIEM Associations

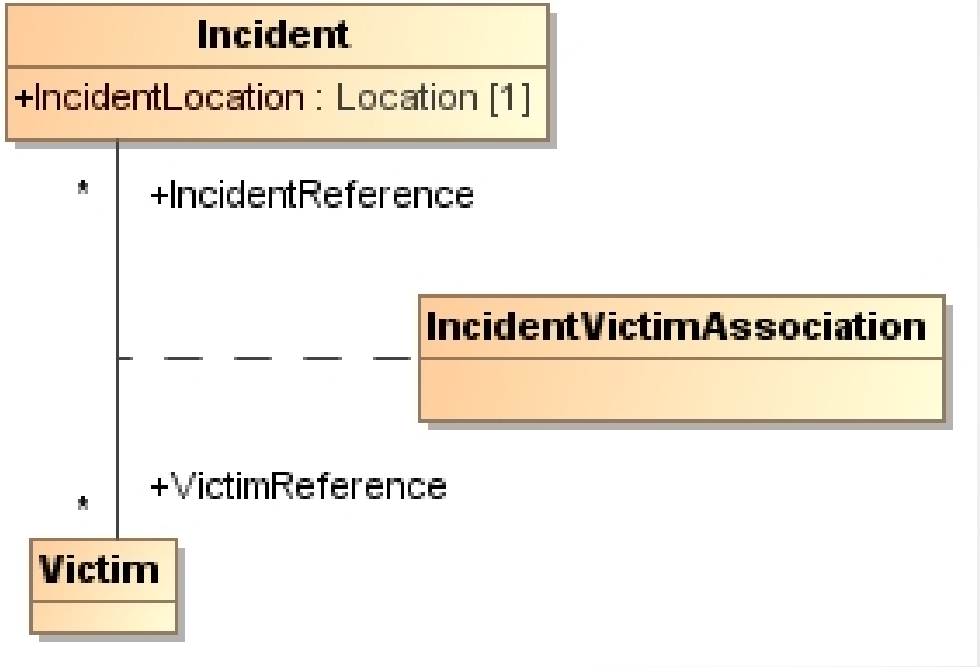
A UML Class stereotyped as an <<AssociationType>> represents a NIEM association using the rules of complex types. Each end of the NIEM association is represented as an independent UML association (an association line in a class diagram). The end is named on the related object side of the UML association and the cardinality of this relation will be the number of such objects that can participate in each association, this cardinality is usually one.



NIEM & UML Associations

NIEM Associations (PIM Only)

Alternative:
As UML includes a first-class concept of association classes, A NIEM association may also be represented as a UML association class (Line with a class attached by a dotted line), optionally having the <<AssociationType>> stereotype.



NIEM Metadata

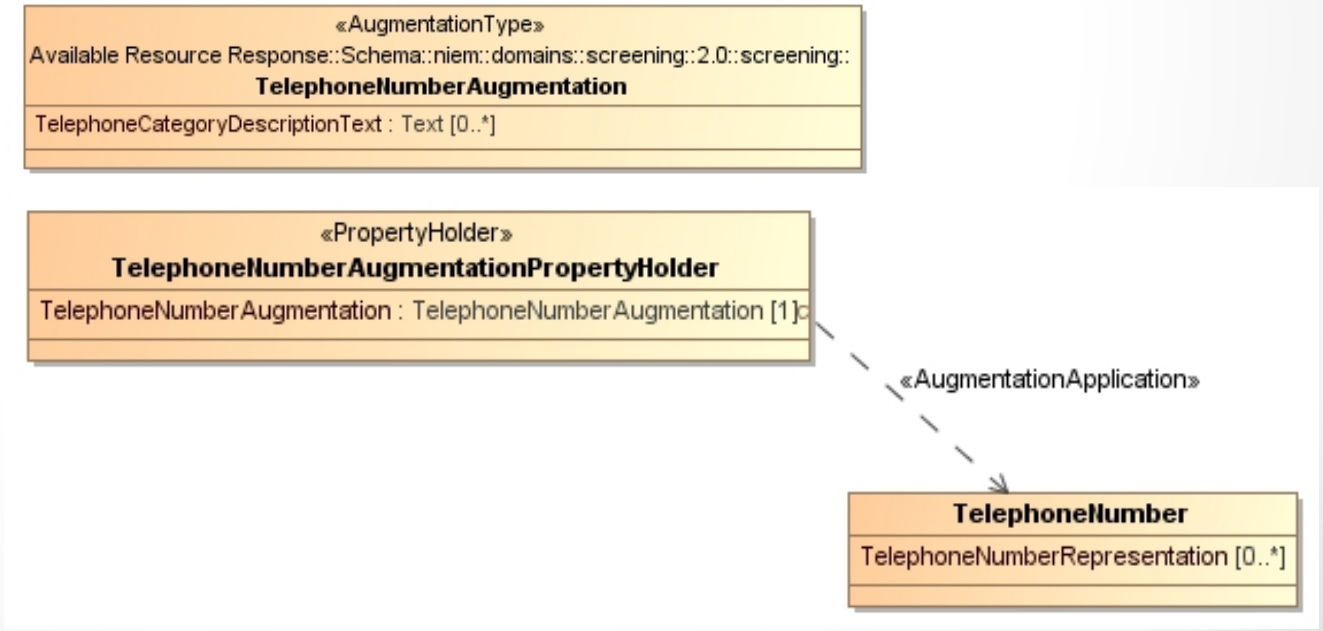
NIEM Metadata

A Metadata type is represented as a UML class with the <<MetaDataType>> Stereotype. A Metadata type may have a <<MetadataApplication>> dependency which restricts the class of objects the metadata may be applied to. Metadata without a <<MetadataApplication>> may be applied to any NIEM object.

NIEM Augmentations

NIEM Augmentations

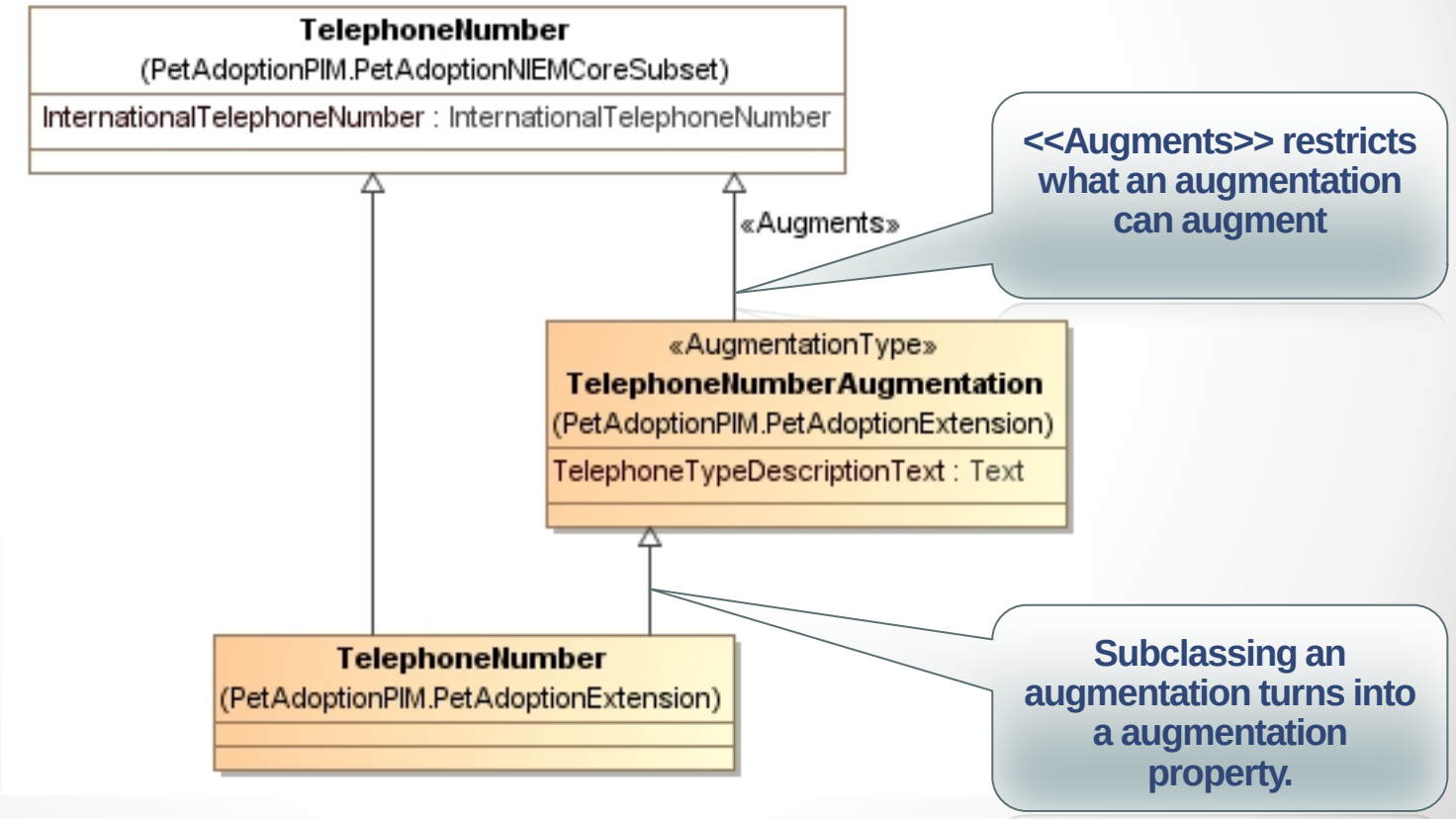
An Augmentation type is represented as a UML class with the <<AugmentationType>> Stereotype. A property typed by an augmentation type may have an <<AugmentationApplication>> dependency which restricts the class of objects that may contain a property typed by an augmentation (this is sometimes called the properties "domain"). Properties without an <<AugmentationApplication>> may be properties of any NIEM object.



NIEM Augmentations

NIEM Augmentations (PIM Only)

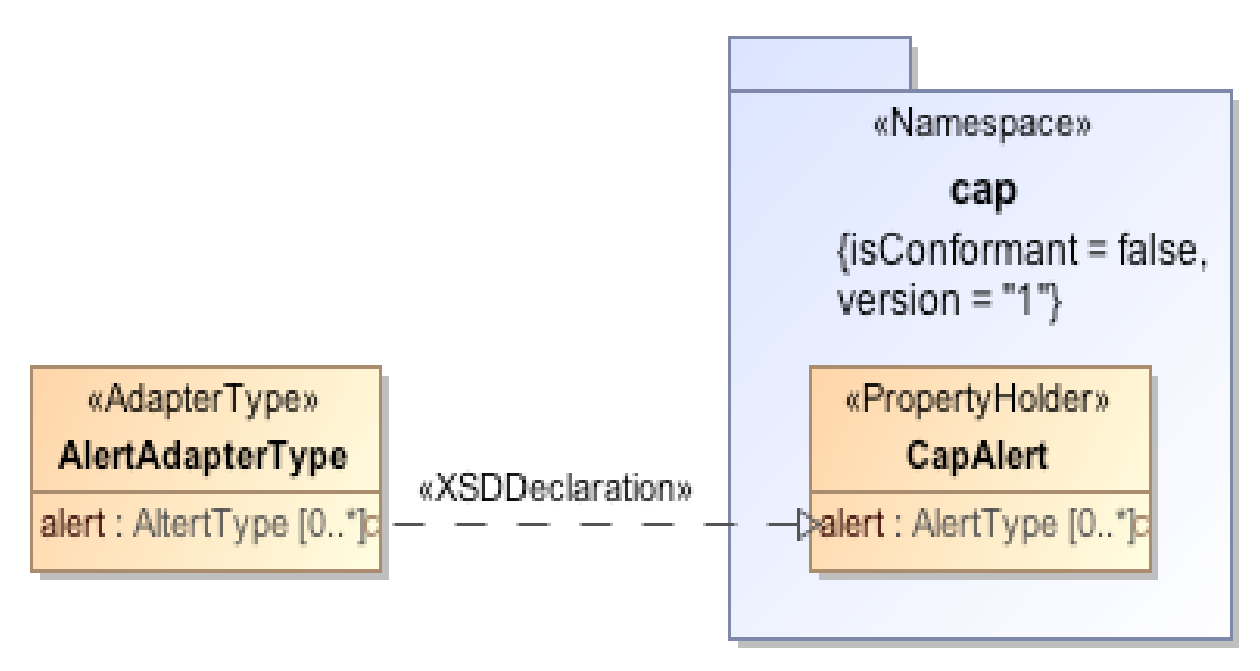
A generalization can be marked to <<Augments>> an <<AugmentationType>>. Inheriting an augmentation creates an augmentation property.



Adapter Types

Adapter Types

An *adapter type* is a NIEM object type that adapts external models for use within NIEM. An adapter type creates a new class of object that embodies a single concept composed of external elements. [NIEM-NDR 7.7]

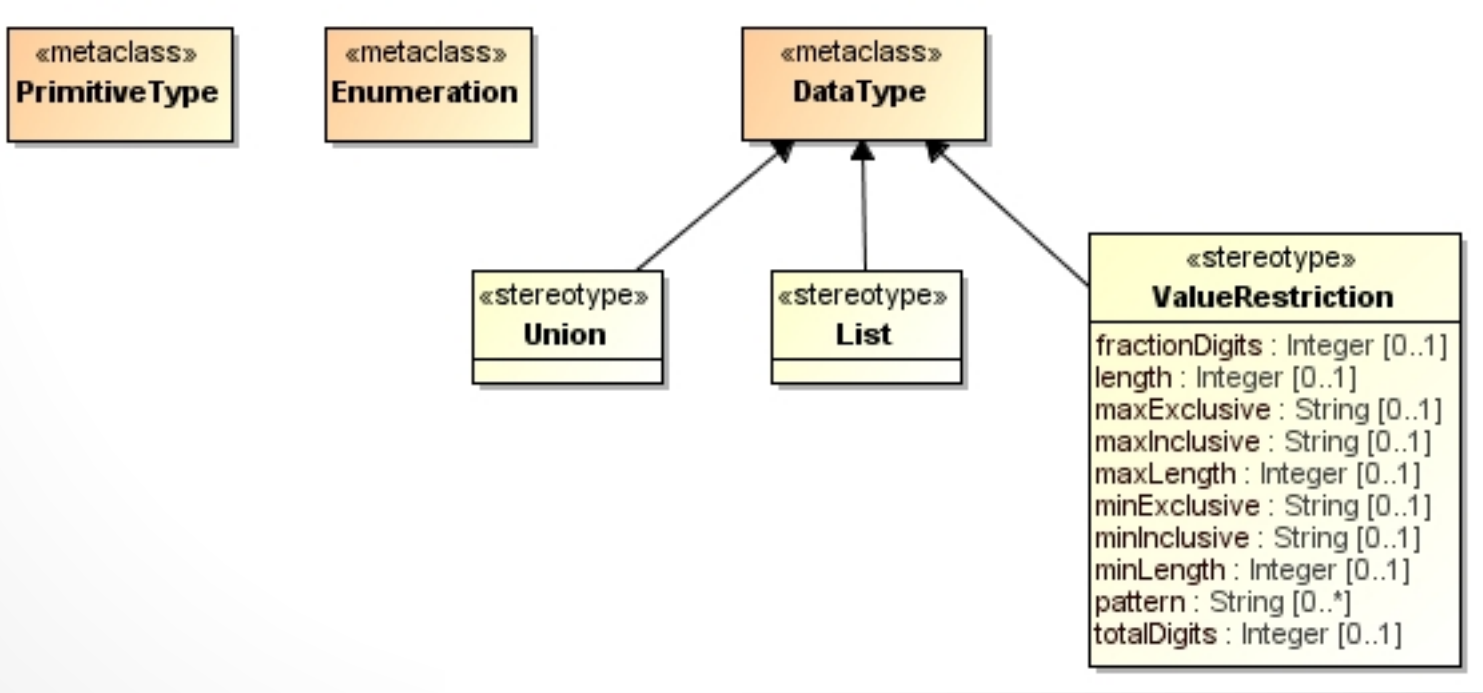


Modeling Simple Types

DataType and Related Elements

DataType (UML)

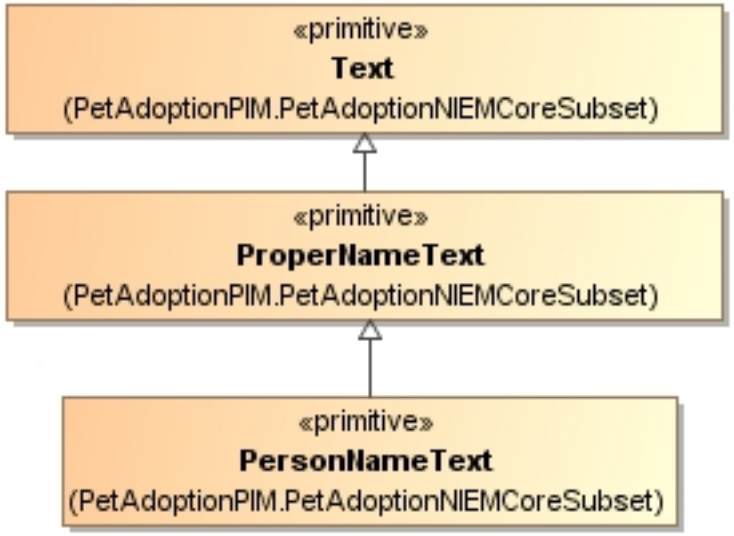
- DataTypes represents a simple type, which is implemented in XML Schema as a simple type definition component.
- PrimitiveType and Enumeration are kinds of DataTypes
- ValueRestriction, Union, and List stereotypes may be applied to DataType
- Realization or Generalization may relate DataTypes



DataType and Related Elements

PrimitiveType (UML)

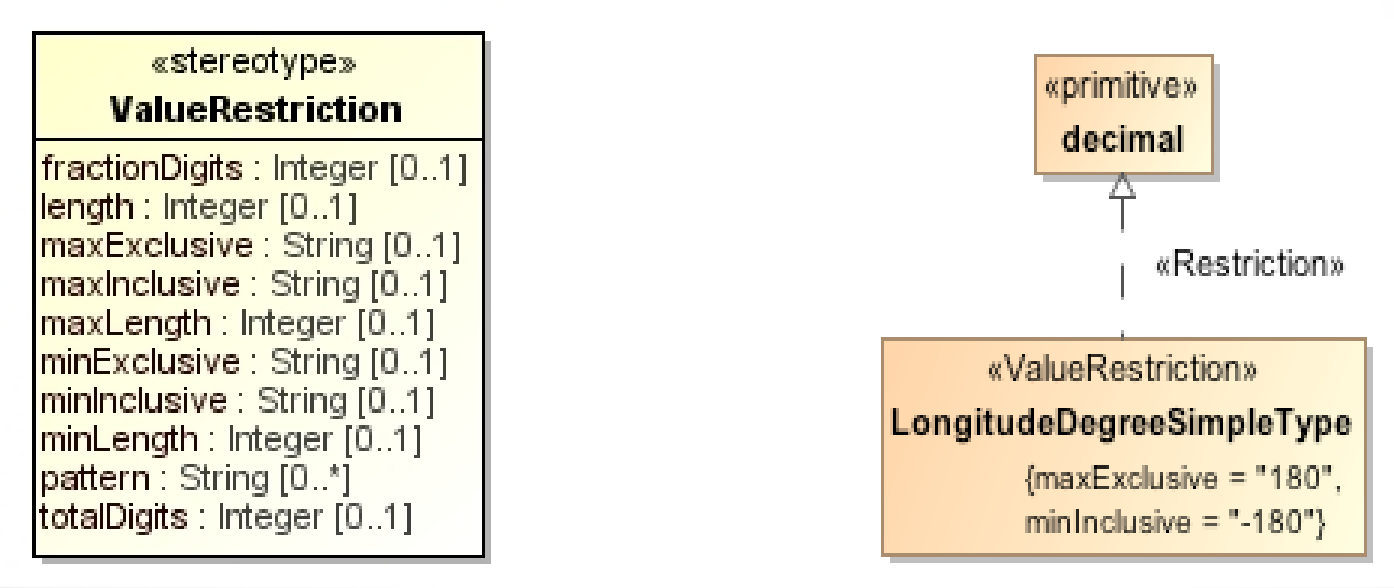
- Represents XML types and subtypes of them.



DataType and Related Elements

ValueRestriction (DataType)

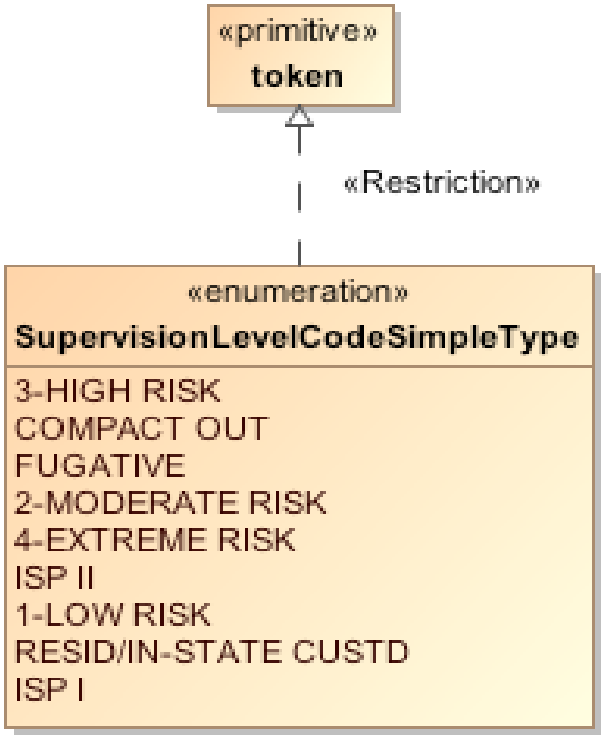
- ValueRestriction represents the facets of a simple type, which are implemented in XML Schema as the facets property of a simple type definition component.



NIEM CODE Lists

NIEM Code Types

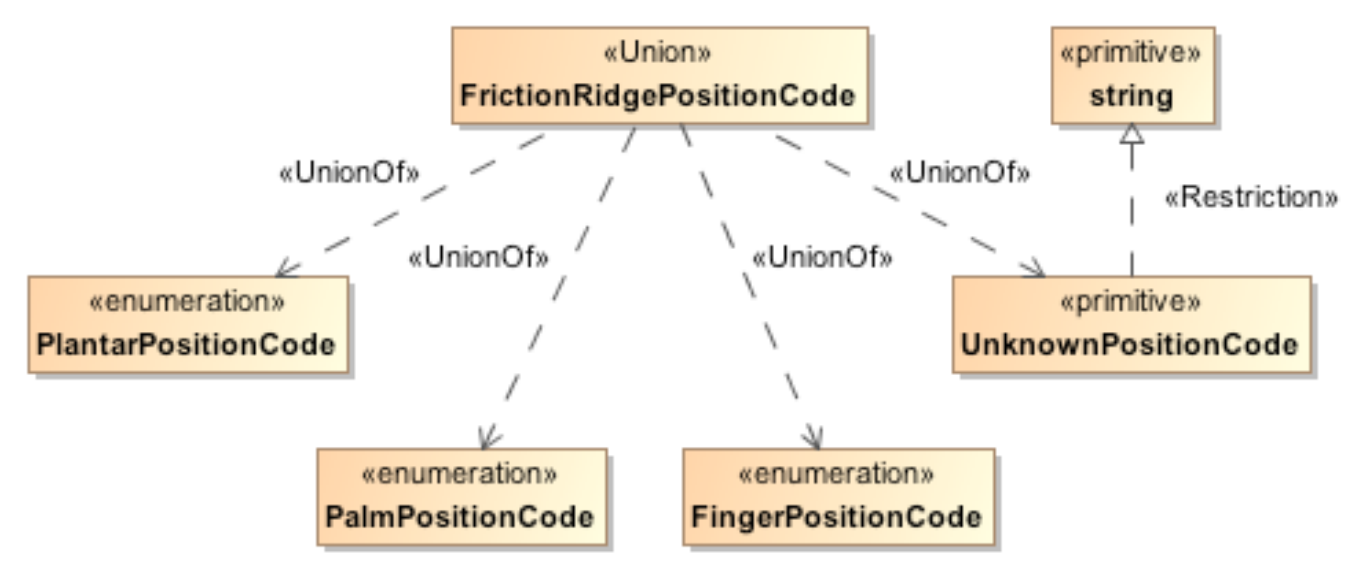
Code types are represented as UML enumerations. Each code value is one value of the enumeration.
An enumeration may <<Realize>> another DataType to indicate restriction.



DataType and Related Stereotypes

Union (DataType)

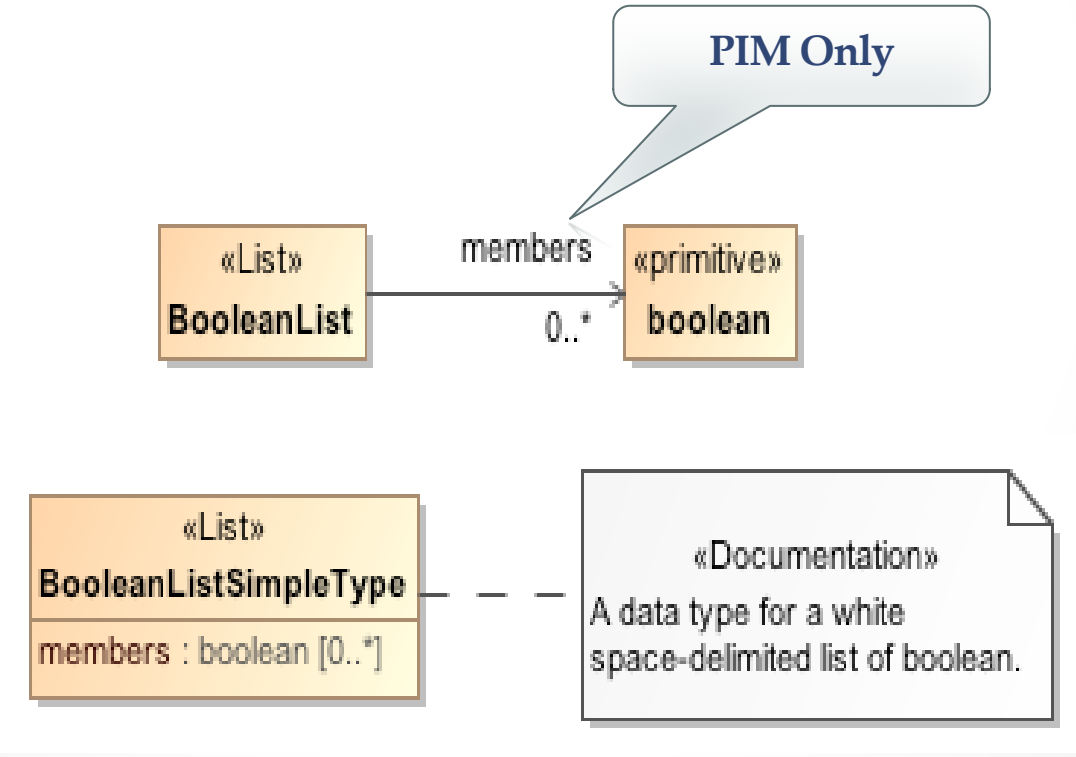
- Union represents a union simple type, represented in XML Schema as a simple type definition component for which the variety property is “union”. A dependency marked as <<UnionOf>> references each member of the union.



DataType and Related Stereotypes

List (DataType)

- List represents a list simple type, represented in XML Schema as a simple type definition component for which the variety property is “list”.
- A single property with an arbitrary name indicates the type of the list

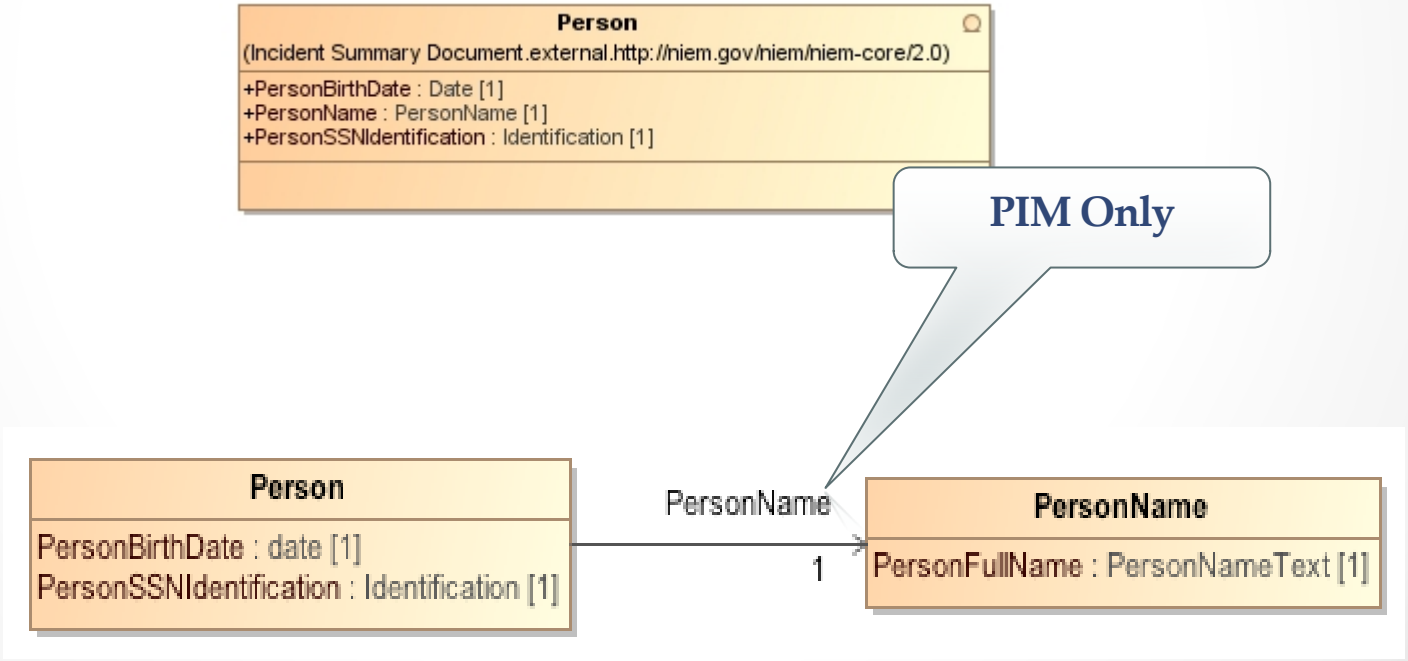


Modeling NIEM Properties

NIEM Properties

NIEM Properties

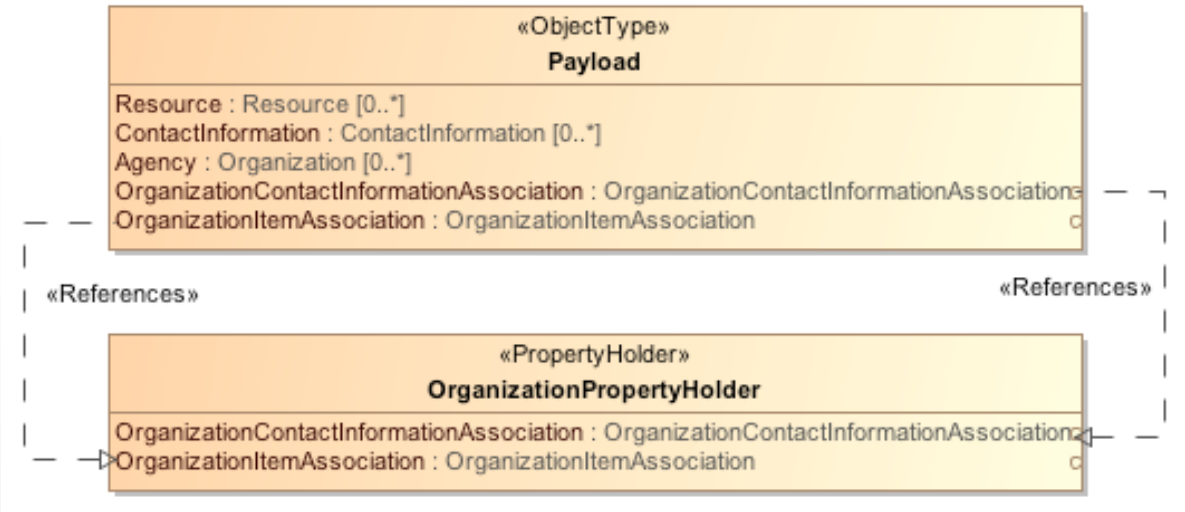
Non-reference properties:
Properties are represented as properties of UML classes or as ends of associations. Information from the UML property or association end definition includes the name, type and cardinality. Associations are only used in a PIM.
.



NIEM Property Reuse

NIEM Property Reuse and Subset Schema

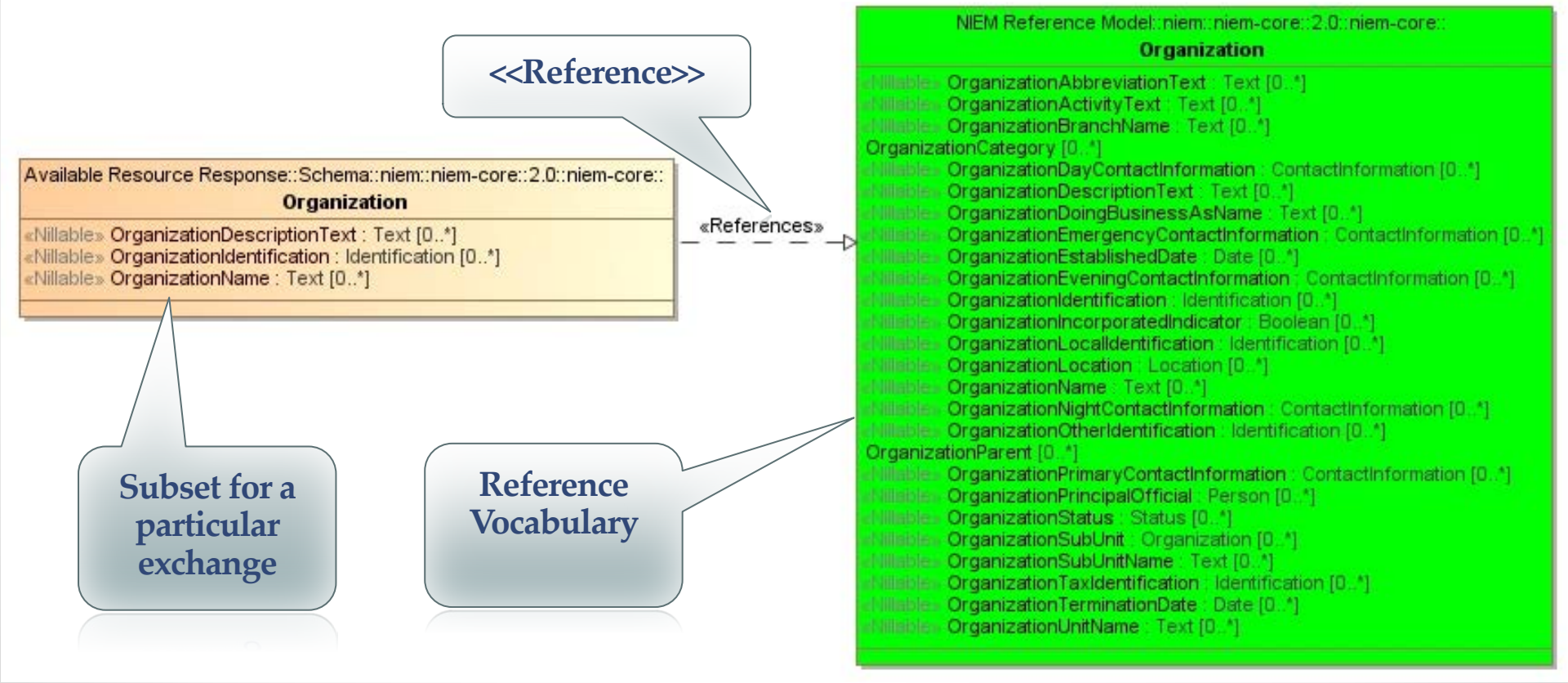
UML has no notion of properties independent of any class and the normal way to handle this in UML is to define classes, perhaps abstract, that are inherited. To be consistent with UML all properties are defined within a class (or data type). The <<References>> stereotype of realization is used to import properties from one class to another (perhaps in another name-space) to provide for the property reuse that is a principle of NIEM. The defining class can be complex type, an abstract type or a <<PropertyHolder>>. Property holders are a NIEM-PIM Stereotype specificity to hold properties not owned by a class in the namespace (top level properties).



Subsetting a Reference Vocabulary (PIM Only)

NIEM Property Reuse and Subset Schema

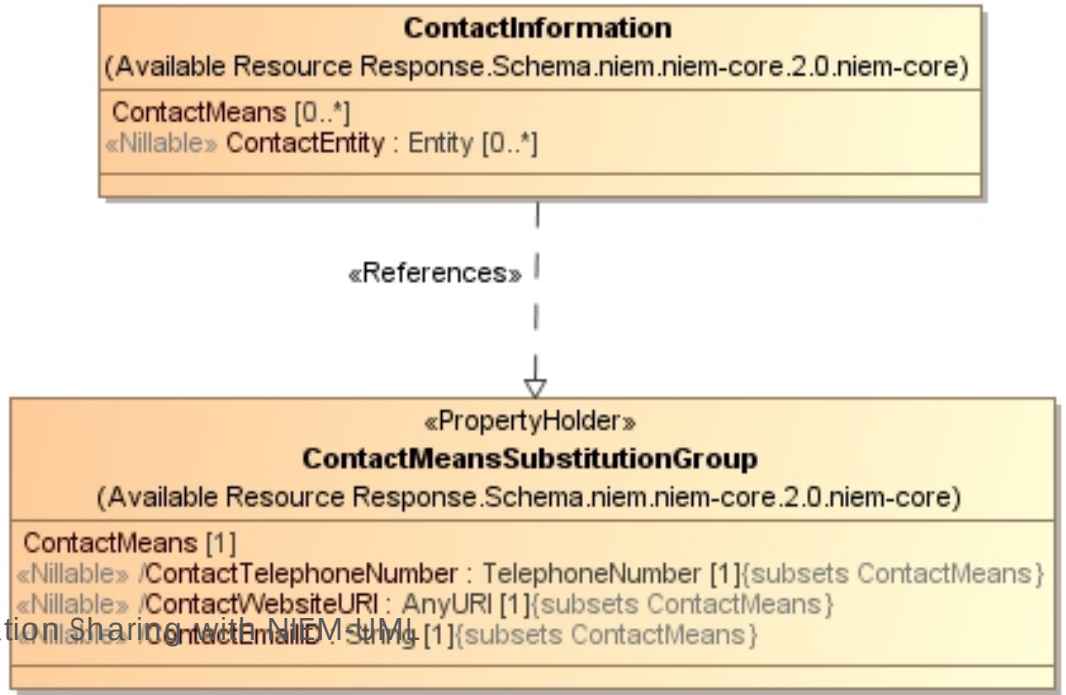
Classes and packages may subset NIEM reference vocabularies.



NIEM Substitution Groups

NIEM Properties

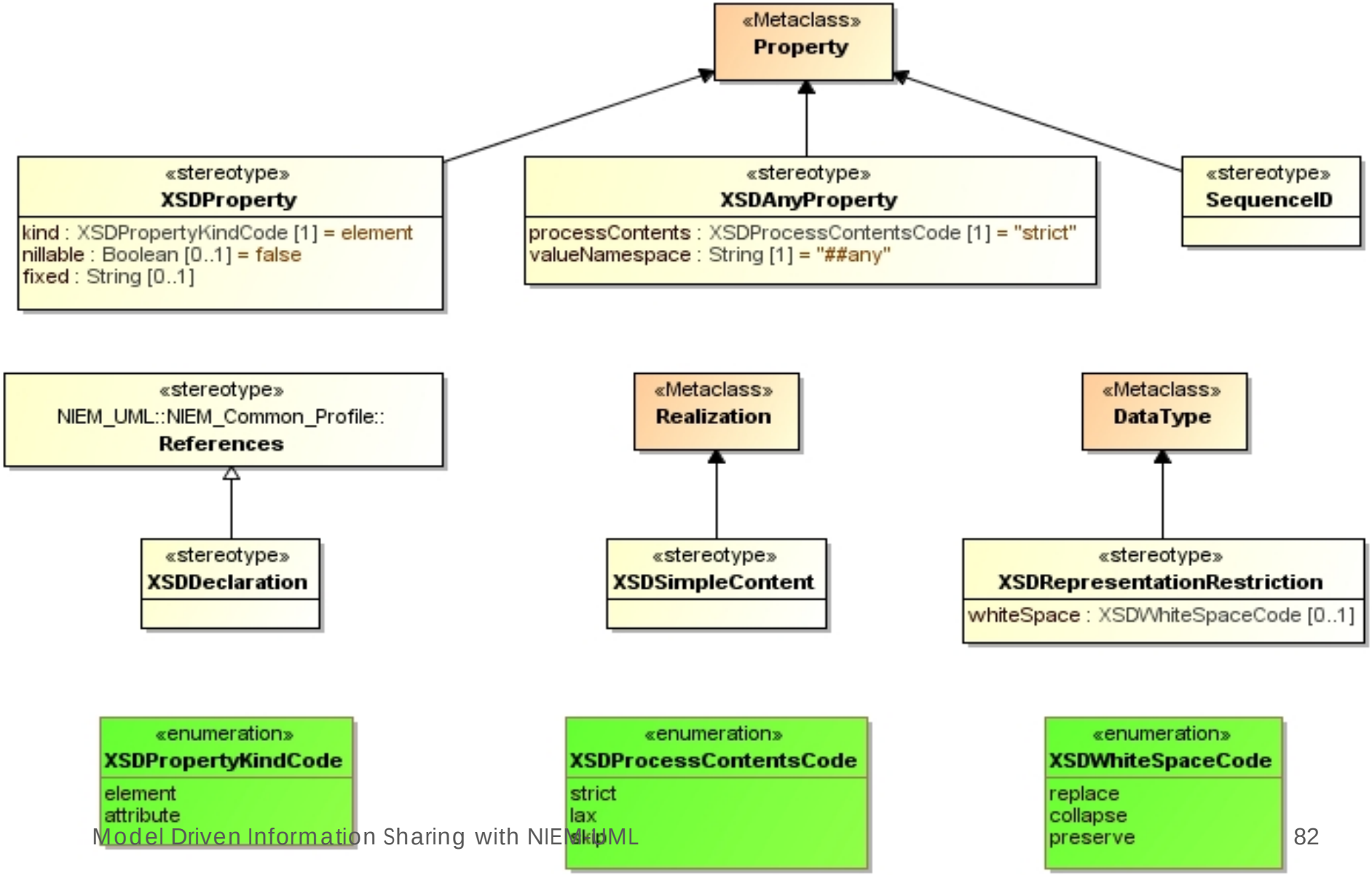
A substitution group is represented by UML property subsetting. A property that subsets another will be substitutable for the base property. All subset properties within a name space are normally grouped together into a single class with the name of the base property combined with the suffix "SubstitutionGroup" (Current implementation is generating "PropertyHolder"). Substitution groups are also declared as a <<PropertyHolder>> since the containing class is not consequential, it is simple a holder for the group of substitutable properties.



Platform Specific Profile

• • •

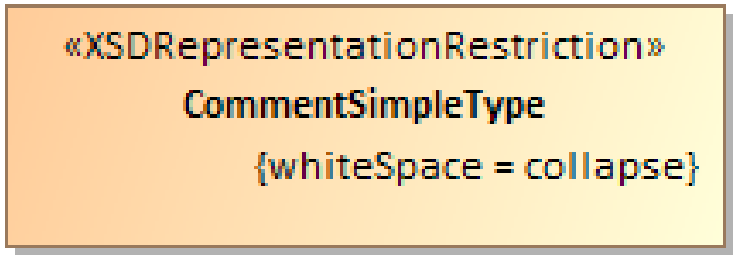
Platform Specific Profile



XSD Representation Restriction (DataType)

Indicates that the facets property of the XML Schema simple type definition includes a whiteSpace component.

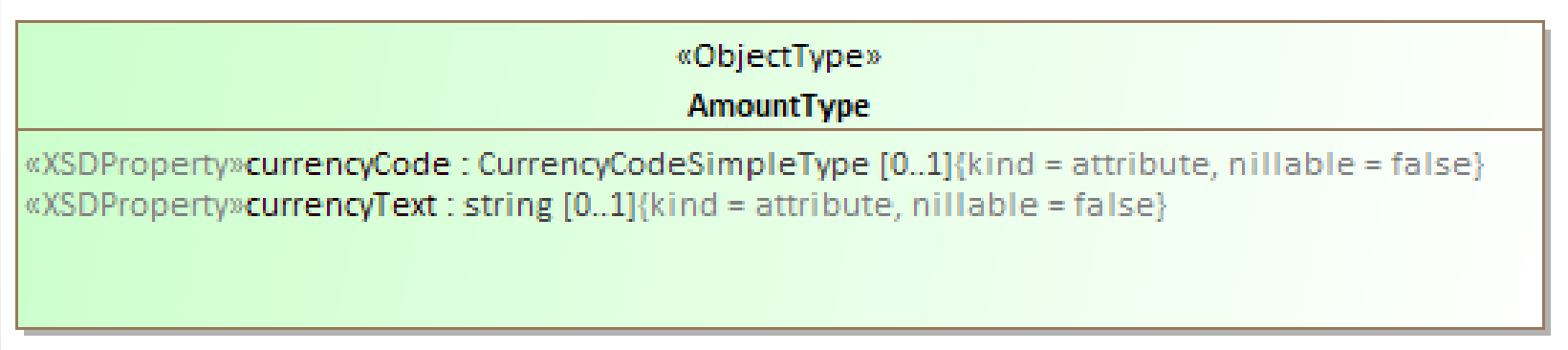
- whiteSpace attribute: value of the whiteSpace component.



XSD Property (Property)

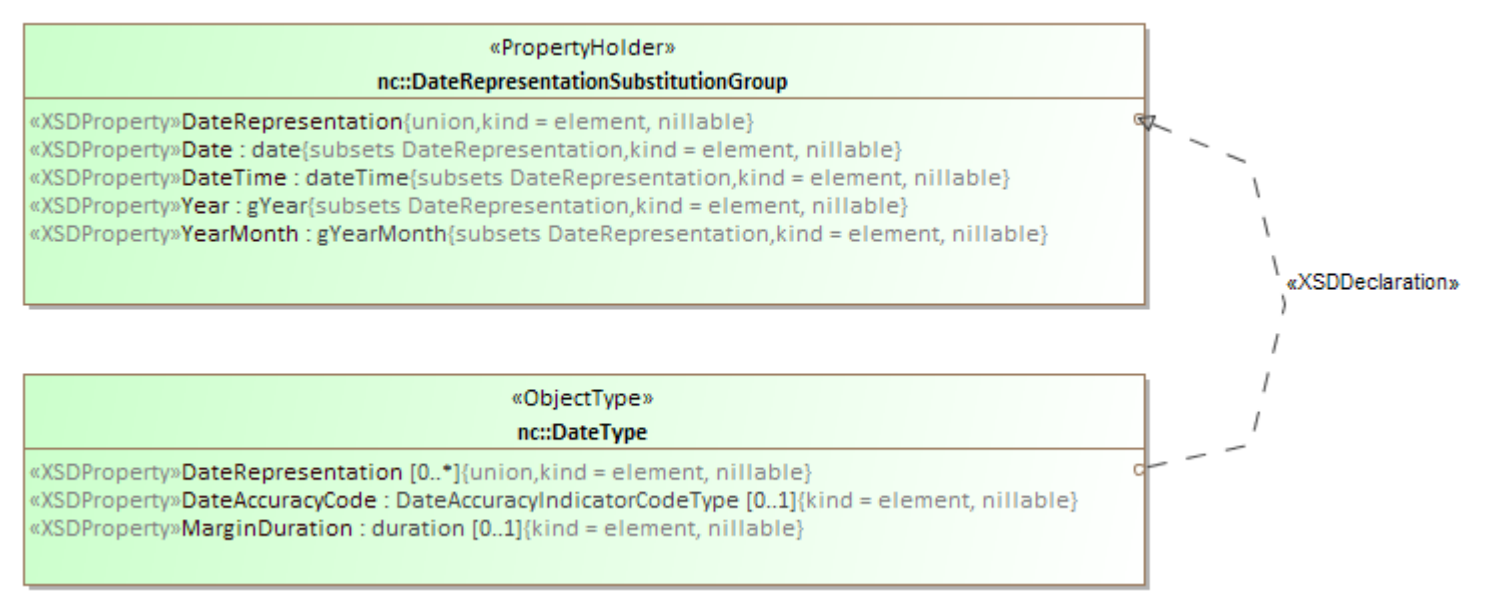
Indicates the implementation of a NIEM property: whether it is an element or attribute declaration, its value constraint property, and its nillable property.

- kind attribute: whether an element or attribute declaration.
- fixed attribute: the value of the value constraint property.
- nillable: the value of the nillable property.



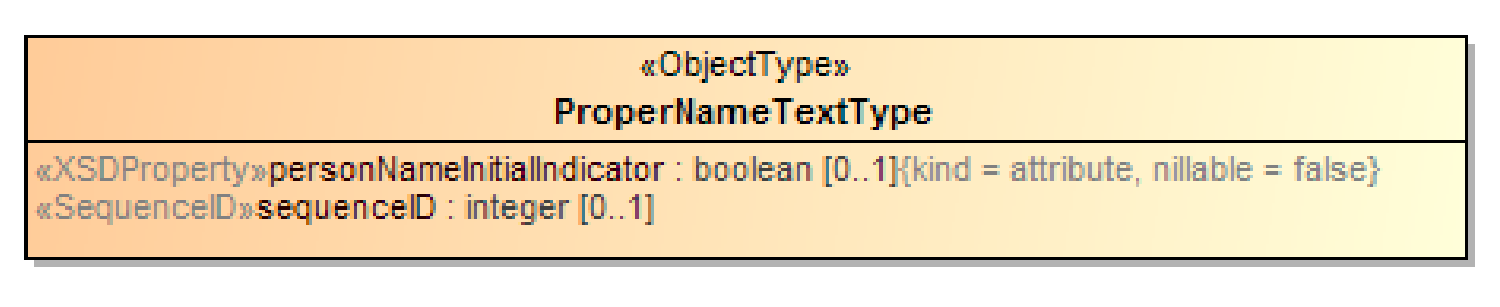
XSD Declaration (Realization)

- Indicates the element or attribute declaration of an element or attribute use.
- client: the element or attribute use
- supplier: the element or attribute declaration



SequenceID (Property)

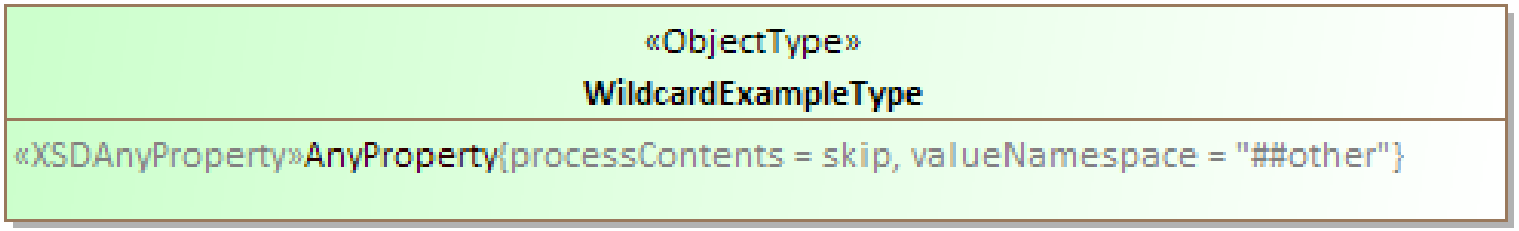
Indicates an attribute use for which the attribute declaration is structures:sequenceID.



XSD AnyProperty (Property)

Indicates a XML Schema wildcard.

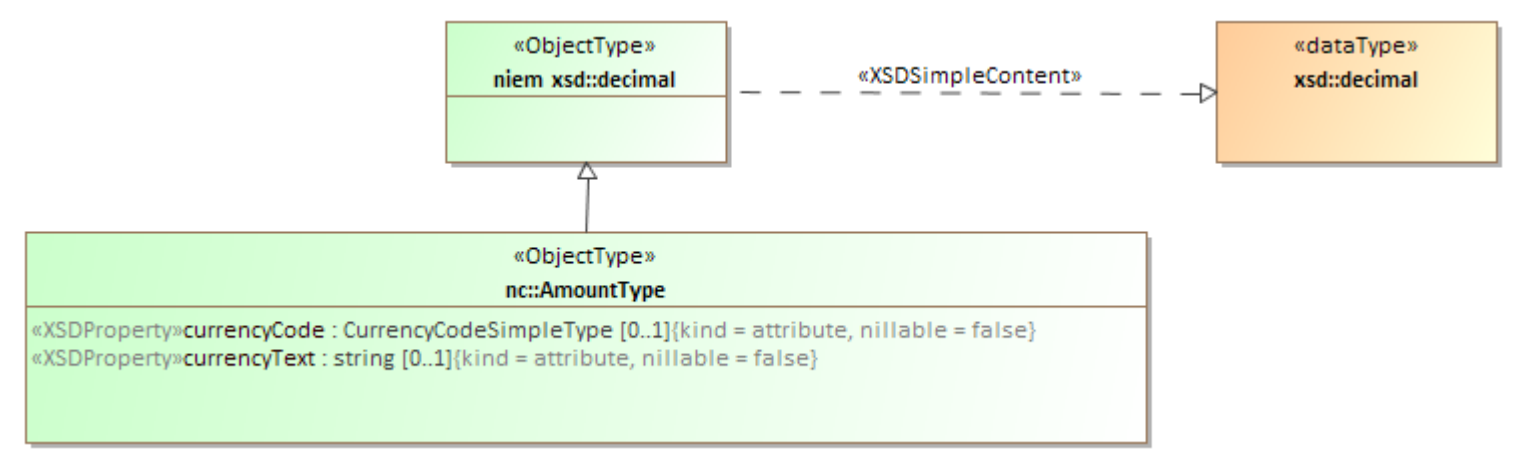
- processContents attribute: the value of the process contents property.
- valueNamespace attribute: the value of the namespace constraint property.



XSD SimpleContent (Realization)

Indicates the content type of a complex type definition.

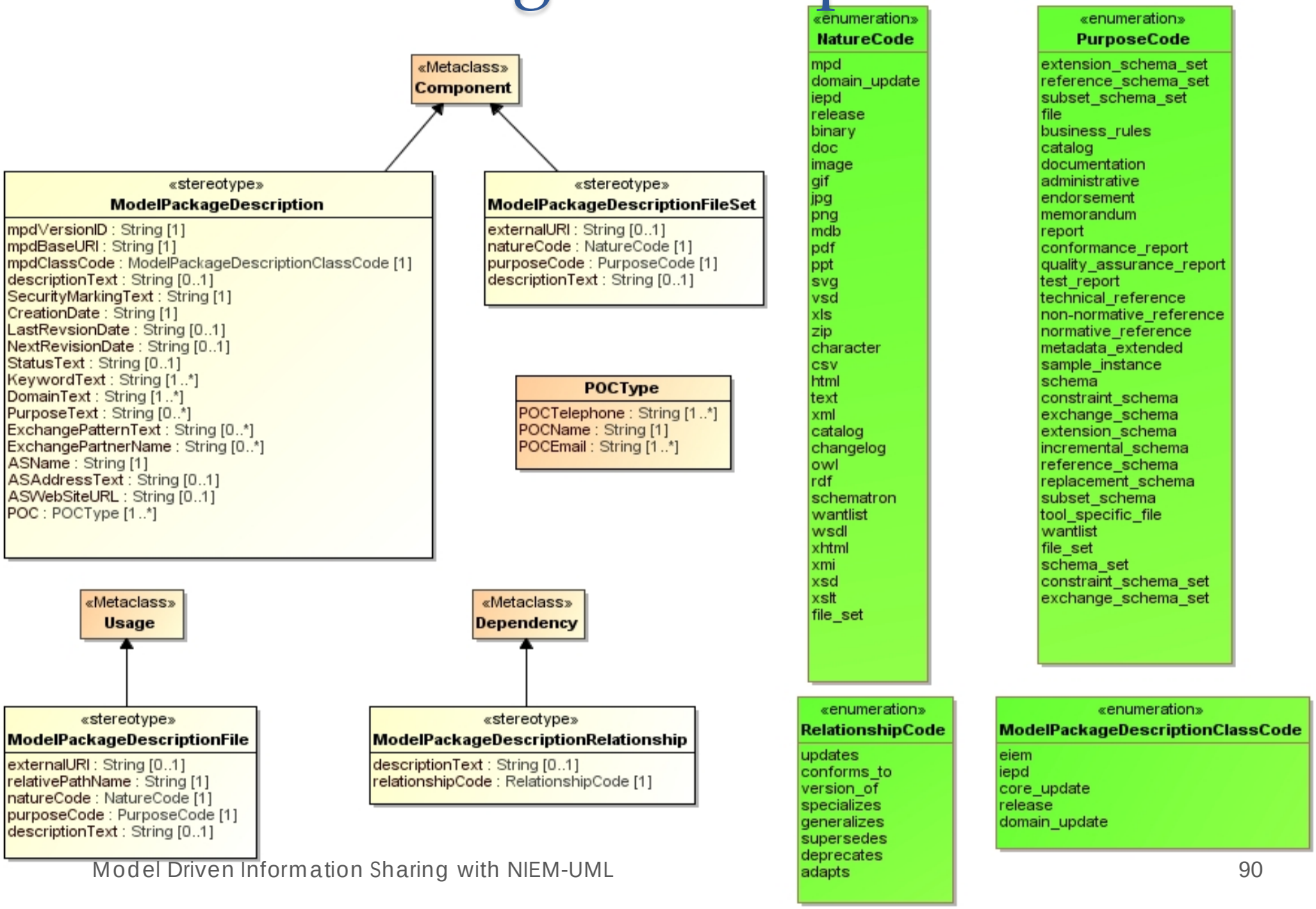
- client: the complex type definition
- supplier: the content type of the complex type definition, a simple type definition



Model Package Description Profile

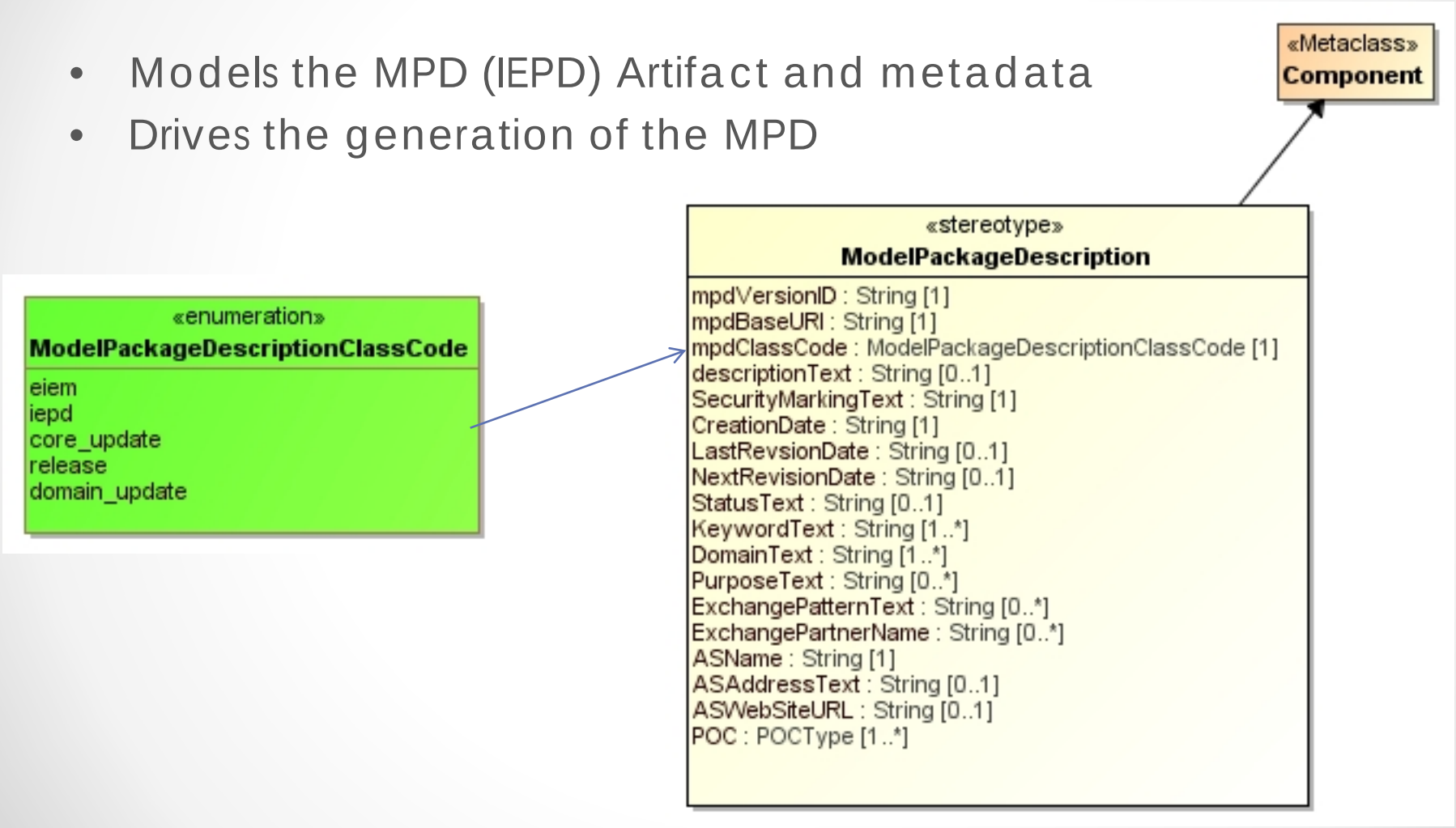
• • •

Model Package Description Model



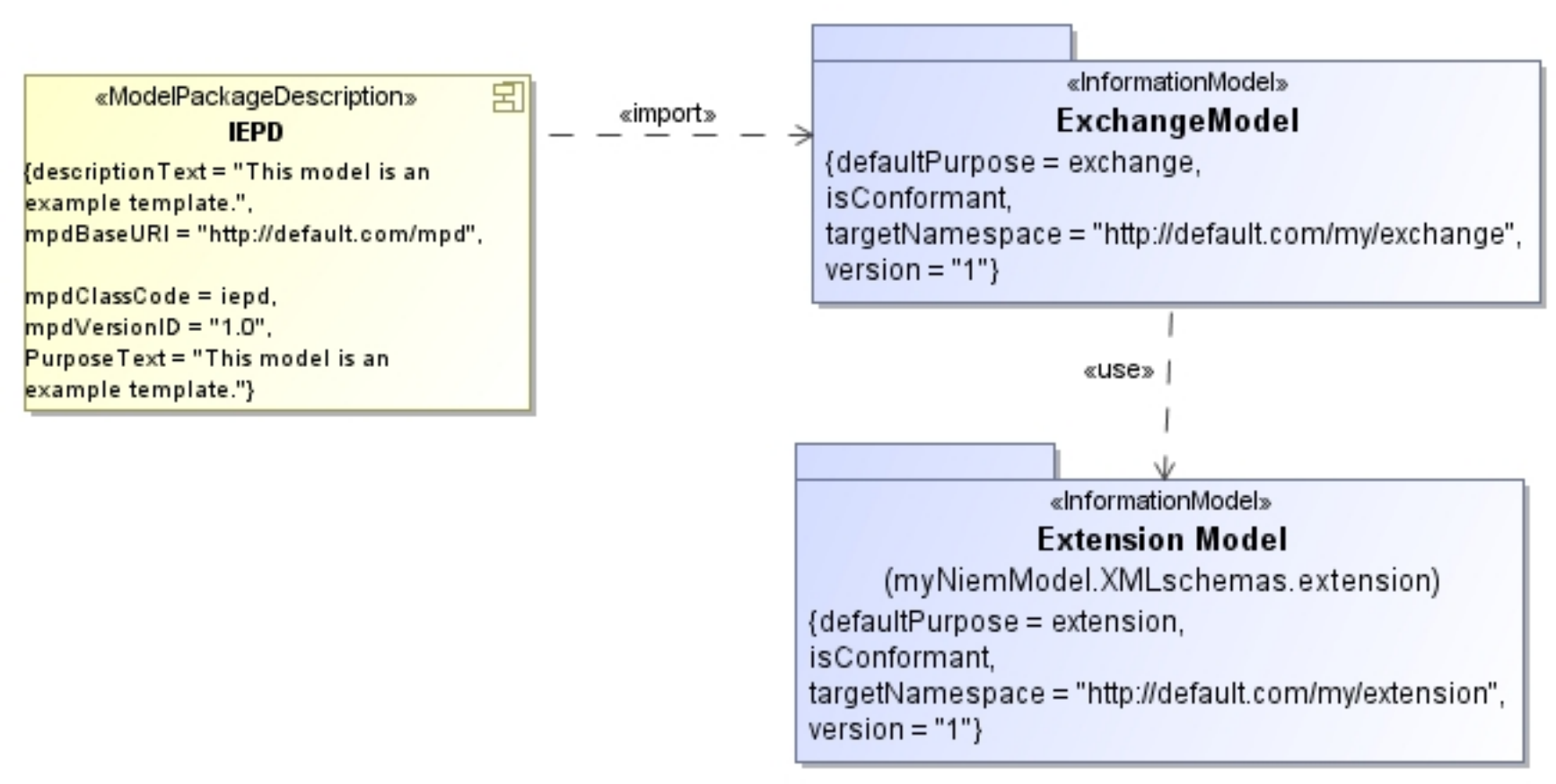
Model Package Description

- Models the MPD (IEPD) Artifact and metadata
- Drives the generation of the MPD



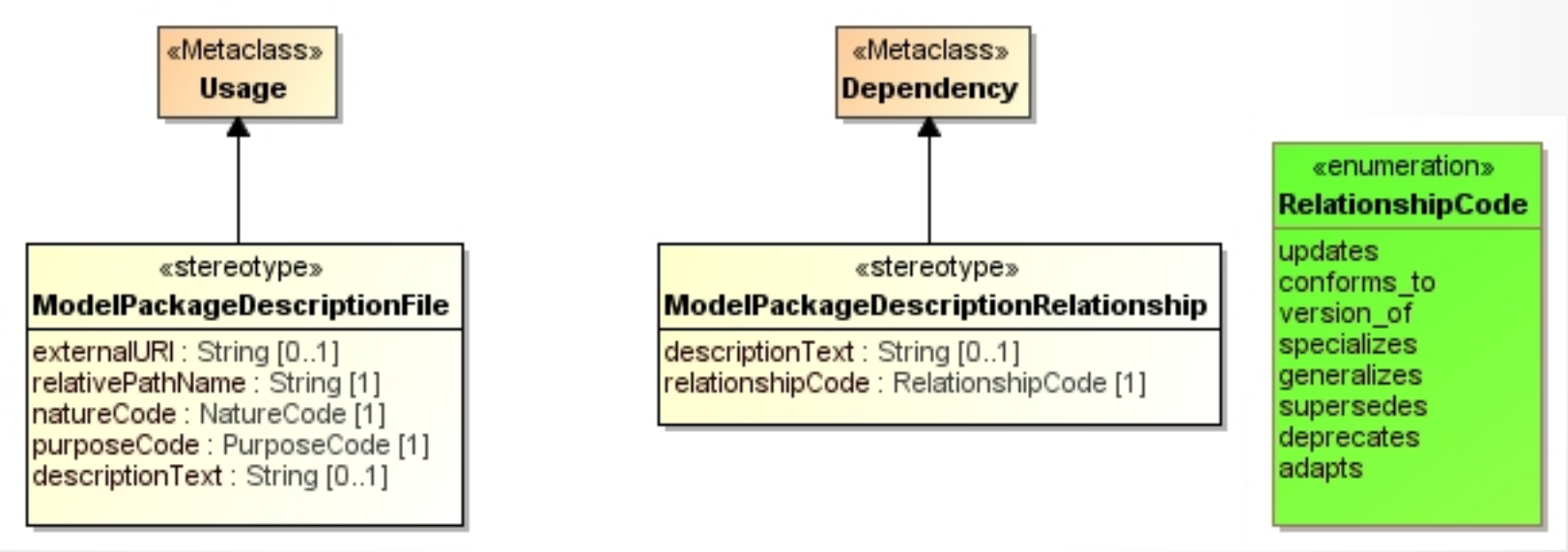
IEPD Example

- A MPD <<imports>> one or more information models that it requires.
- Information models may <<use>> other models, these will be included in the IEPD



Overriding defaults

- **<<ModelPackageDescriptionFile>>** may provide additional information on the information models *or other artifacts* used in the MPD.
- **<<ModelPackageDescriptionRelationship>>** models the relationship between MPDs, for example, between revisions



NIEM Reference Vocabularies

Core (NIEM Core)

Reference (Combined)

common.ansi_d20
common.apco
common.atf
common.cbrncl
common.census
common.dea
common.dod_jcs-pub2.0-misc
common.edxl-cap
common.edxl-de
common.edxl-have
common.edxl
common.fbi
common.fips_10-4
common.fips_5-2
common.fips_6-4
common.geospatial
common.have-codes
common.hazmat
common.icism
common.iso_3166
common.iso_4217
common.iso_639-3
common.itis
common.lasd
common.mmucc_2
common.mn_offense
common.nga
common.nlets
common.nonauthoritative-code
common.post-canada
common.sar
common.twpdes
common.ucr
common.unece_rec20-misc
common.usps_states
common.ut_offender-tracking-misc
core
domains.emergencyManagement
domains.familyServices
domains.infrastructureProtection
domains.intelligence
domains.jxdm
domains.maritime
domains.screening
external.cap
external.de
external.have
external.ogc

More information

Information on NIEM

http://www.niem.gov

Information on NIEM-UML

http://niem-uml.org

Contact Us

Cory-c (at) modeldriven (dot) com

NIEM-UML Vendors

Model Driven Solutions

- Developed the NIEM-UML PIM and First Implementation
- Model Driven Solutions (MDS) is the services division of Data Access Technologies, Inc., a small business headquartered in North Virginia, founded in 1996. Our primary customers are government and large corporations.
- MDS provides a Model Driven approach to business and information systems solutions.
- Perhaps we can provide you with a 2 or 4 day "quick start" on NIEM-UML?

Providing

- Enterprise Architecture
- Business Architecture
- Information Architecture
- Information Sharing
- Services Architecture
- Systems Architecture
- Executable Systems
- Automated Federation
- Open Source Tooling

Using

- Semantic Technologies
- Unified Modeling Language
- NIEM-UML
- Business Process Modeling Notation
- Service Oriented Architecture
- Model Driven Architecture
- Industry Standards
- Open Source & Commercial Products

No Magic Inc.

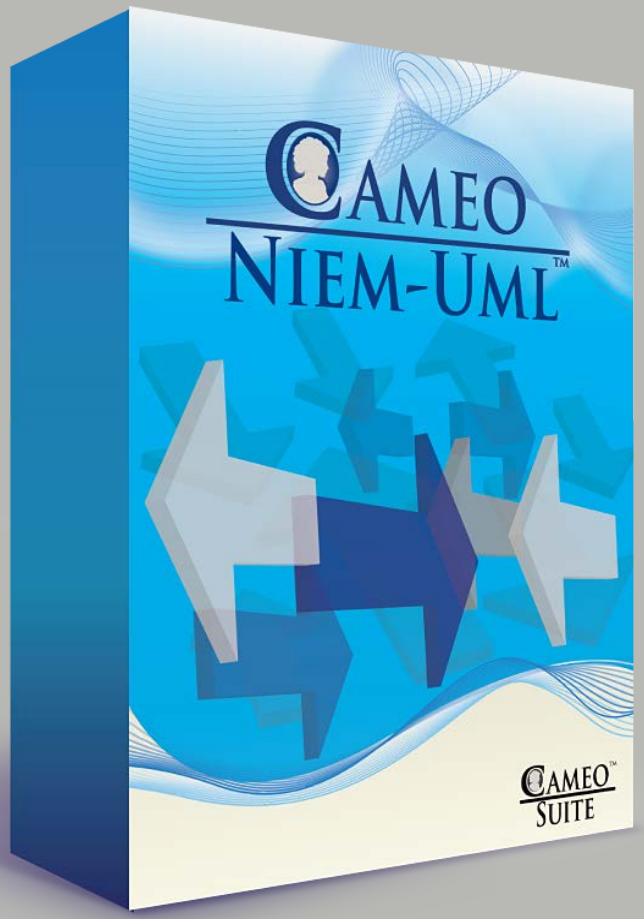
- Founded in 1995
  - Privately owned
- No Magic's Cameo Suite
  - Fully integrated solution suite
- No Magic included in the 2011 Magic Quadrant for Business Process Analysis Tools

- **OMG® standards-compliant products for**
  - UML 2
  - Teamwork Collaboration
  - Enterprise Architecture
  - Systems Engineering
  - Simulation
  - Parametrics
  - NIEM
  - SOA
  - Ontology
  - BPMN 2.0
  - Data Modeling
  - Requirements Management
  - Model Driven Development
  - Interoperability
  - And many other 3rd party and open source

Cameo NIEM-UML Plugin

Requirements:

- MagicDraw 17.0.1 (or later) - Standard through Enterprise Editions
- QVT Plugin



Questions and Comments