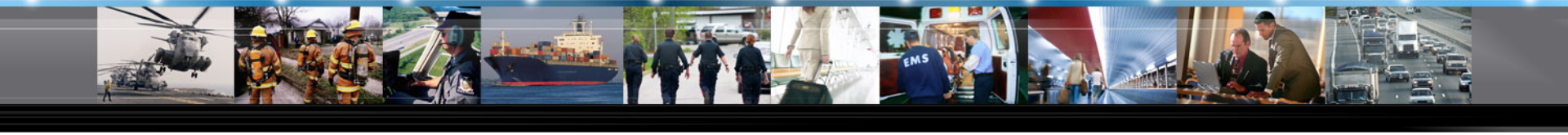
Interoperability Standards for Information Sharing and Safeguarding



PM-ISE

Key Considerations

Information Sharing and Safeguarding focuses on ‘Data-in-Motion (query response, pub-sub, Broadcast, etc.)’ - cross mission, cross border, cross domain, cross agency, cross industry

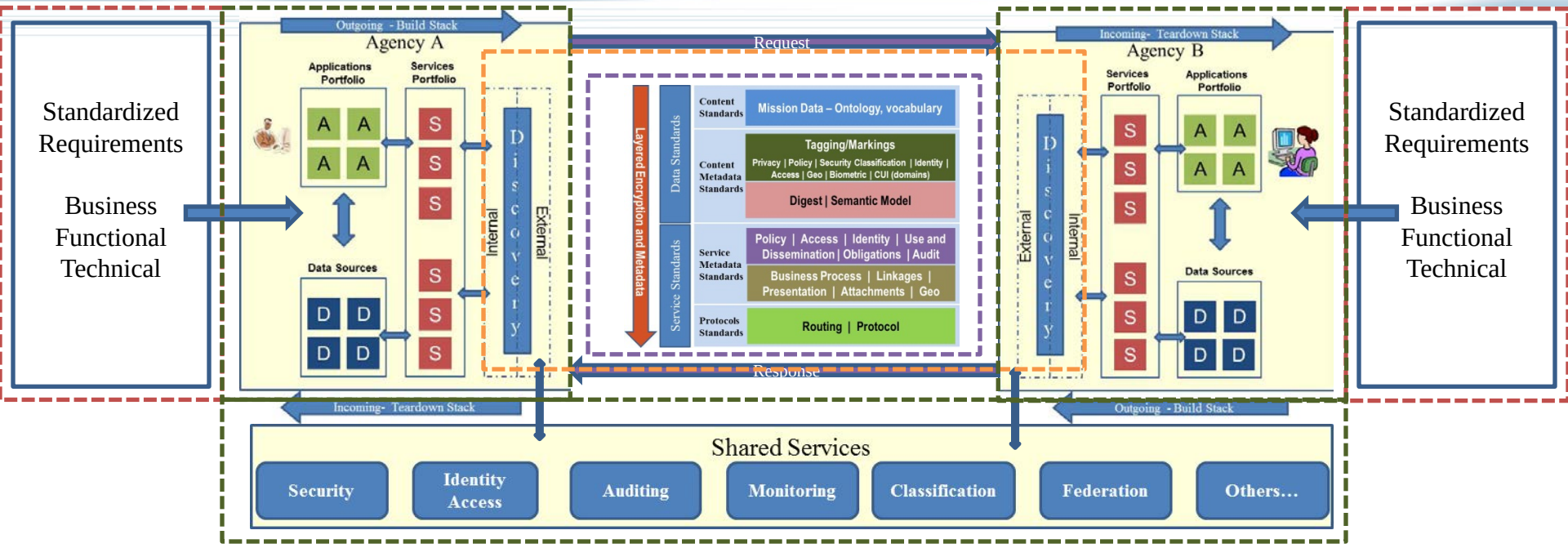
Information Sharing Environment is about enabling mission partners to share information, and make information offered by mission applications, systems and services ‘Assured’, ‘Smart’, ‘Harmonized’, ‘Interoperable’, ‘Secure’, ‘Automated’, ‘Discoverable’, ‘Federated’

Business/Functional and Technical Standards enable Assured Information sharing in an interoperable environment, comprising of an ‘Information Sharing Stack of standards’

Consensus based development and adoption of technical standards required (including existing, under development, and gaps), and the portfolio of business and functional capabilities that implement these standards, driven by complexity of the exchange, and degree of maturity of the partnering organizations and standards

Technical capabilities and standards need to be considered during design/planning phases of projects, within the architecture context to determine needs for implementation in mission applications, or as shared services, or both

Standards Landscape



Corresponding Industry Technical Standards Required

Standardized Requirements

- Standardized Use Cases
- Functional Standards
- Modeling Standards
- Tooling Standards

Architecture Frameworks

- SOA (WS-*)
- Shared Services Portfolio
- Application Security
- Auditing, Monitoring,
- Reporting
- Interfaces
- Runtime (Policy Config)

Discovery

- Taxonomy for federation

Messaging

Architecture Frameworks

- SOA (WS-*)
- Shared Services Portfolio
- Application Security
- Auditing, Monitoring,
- Reporting
- Interfaces
- Runtime (Policy Config)

Standardized Requirements

- Standardized Use Cases
- Functional Standards
- Modeling Standards
- Tooling Standards

Anatomy of A Message Flow

Components of the message exchanged, and the corresponding technical standards

Technical Standards Categories Needed

(Not all messages use all categories)

Layered Encryption and Metadata

Data Standards

Service Standards

Content Standards	Mission Data – Ontology, vocabulary
Content Metadata Standards	Tagging/Markings Privacy | Policy | Security Classification | Identity | Access | Geo | Biometric | CUI (domains) Digest | Semantic Model
Service Metadata Standards	Policy | Access | Identity | Use and Dissemination | Obligations | Audit Business Process | Linkages | Presentation | Attachments | Geo
Protocols Standards	Routing | Protocol

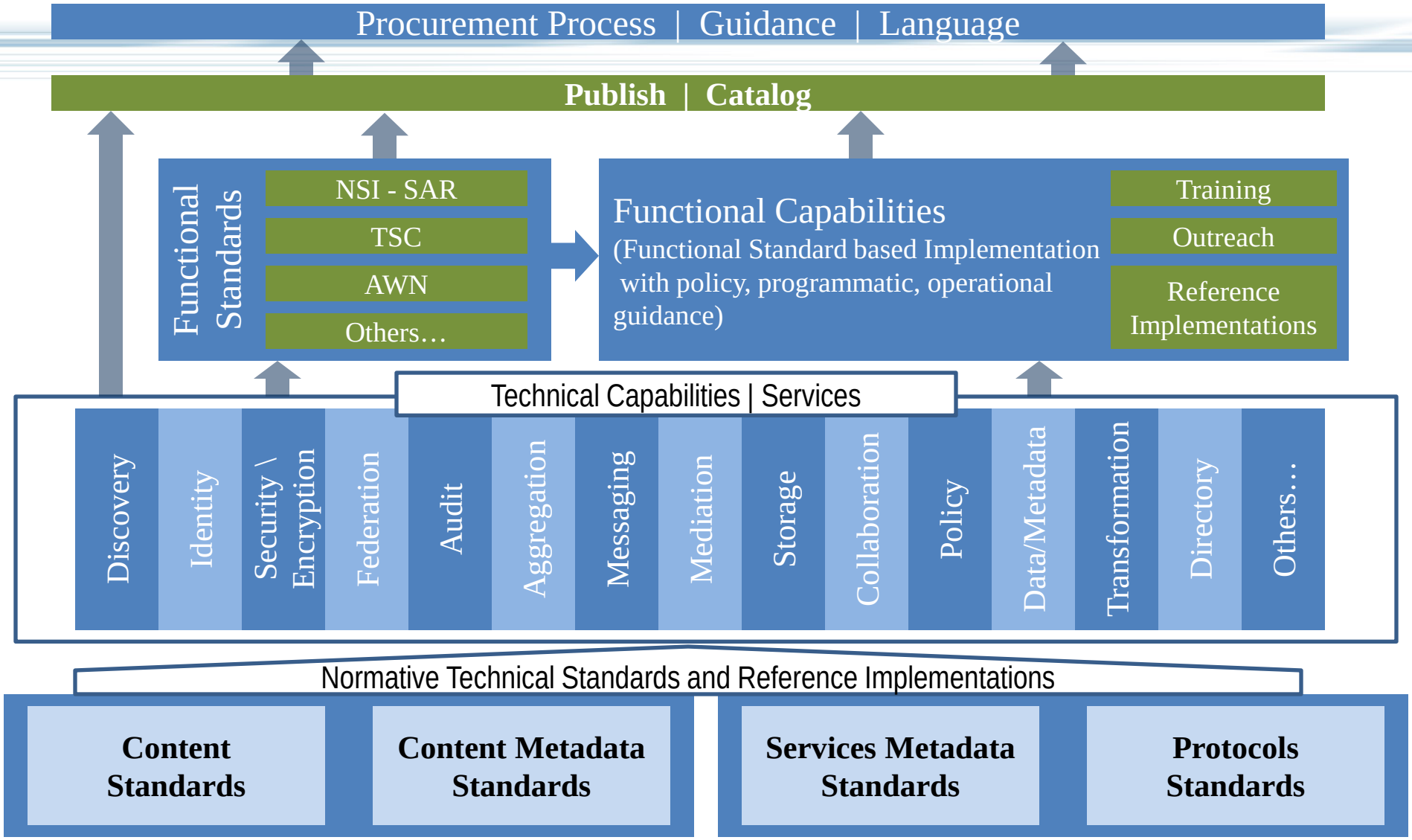
Corresponding Industry Technical Standards

Current State/Gaps

Work with SCC members to address

Additional spreadsheet being developed for details on the current state of standards and potential gaps

ISE Capabilities Model



So how are we going to achieve this…

Standards Coordination Council –

- Government - Standards Working Group, GLOBAL Standards Council, NIEM PMO, NIST
- Standards Organizations - Object Management Group, Open Geospatial Consortium, OASIS, W3C
- Industry - IJIS Institute, ACT-IAC, AFEI

Work with the members of SCC -

- Identify and fill gaps in the standards landscape
- Develop a sequence based on interdependencies among the standards initiatives
- Work with the Industry/SDOs to develop a timeline for each standard critical to ISE

Provide Governance and Support –

- If more than one standard being develop, initiate conversations to help converge standards
- In case of multiple standards for each layer of the stack, develop interoperability profiles to enable information sharing using multiple standards
- Develop plans and incentives to motivate good behavior – for the government to Include standards in procurements, for the vendors to implement the standards, etc.
- Identify pilots to develop reference architectures and implementations to prove the standards

BUILDING BLOCKS

How do you promote responsible information sharing? What do you need to build information sharing across all levels of government, the private sector, internationally, or within your organization? It's a challenge, and we've learned a lot working toward that goal.

Those important lessons we've learned - coupled with best practices from our partners - are incorporated into the following "Building Blocks." Each of the icons below represents one of the fundamental components needed for responsible information sharing. Learn more about us.

HOW CAN I USE BUILDING BLOCKS?
READ AND SHARE CONTENT AND SUCCESS STORIES.

Search Building Blocks

SEARCH

Glossary | FAQ | Contact Building Blocks

GOVERNANCE

BUDGET & PERFORMANCE

ACQUISITION

STANDARDS & INTEROPERABILITY

COMMUNICATIONS & PARTNERSHIPS

NIEM-UML Status and Future

July 28th, 2012

NIEM-UML Background

- The National Information Exchange Model (NIEM) is adopted across a number of domains within the United States and is being considered for adoption internationally
- NIEM-UML is sponsored by the NIEM Project Management Office (PMO) and the Program Manager for the Information Sharing Environment (PM-ISE)
- 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 modeling standard.
- 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 for Magicdraw UML, used as the tooling in this demo.

NIEM-UML Objectives

- **Standards Based**: To leverage standards and standards based tools
- **Simplicity**: To reduce complexity and lower the barrier for entry
- **Reuse**: To facilitate reuse of NIEM models and as a result schemas
- **Agility**: To enable the NIEM profile to be used with other standards, technologies and layers
- **Audience**: Two entry points for tools and modelers – business and schema
- **Clarity**: Ensure that a UML representation of a NIEM model produced by one developer can be interpreted as expected by another.
- **Completeness**: Ensure that a developer can produce a UML representation of any NIEM concept, including semantics, XML Schema structure, and metadata.
- **Practicality**: With minimal effort, a developer can employ the profile in current UML development tools to develop a NIEM model.

NIEM-UML Status

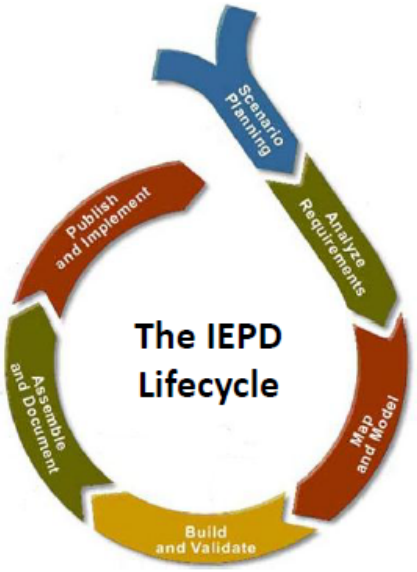
- The final NIEM-UML specification was recommended for adoption by the Object Management Group Government Domain Task Force (http://gov.omg.org/) and architecture board June 2012.
- The specification still has to go through additional OMG adoption processes and approval by the OMG Board of Directors, 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 are available at: http://www.niem-uml.org

Advantages of Using Model Driven Architecture (MDA)

- 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

Efficiency in Creating Information Exchanges

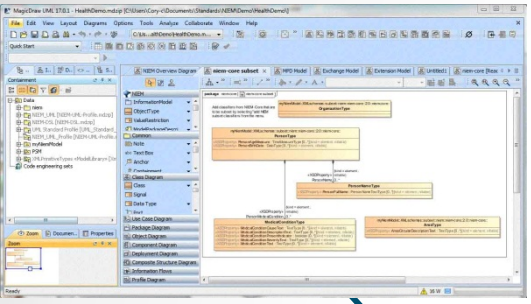
The IEPD Lifecycle



Scenario Planning	Plan the project, establish the process, and identify information exchange business requirements
Analyze Requirements	Selected information exchange is further elaborated to understand and document the business context and data requirements
Map & Model	Associate local objects with types and elements in NIEM. This process is called *mapping* an exchange content model to NIEM
Build & Validate	Create a set of exchange-specific NIEM conformant XML schemas that implement the data model created for the exchange
Assemble & Document	Prepare and package all related files for this IEPD into a single self-contained, self-documented, portable archive file
Publish & Implement	Publish IEPD for search, discovery, and reuse

People & Organizations Sharing Information

Using NIEM-UML to Model Information Exchanges

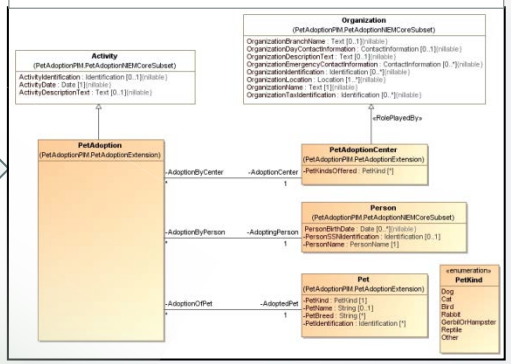


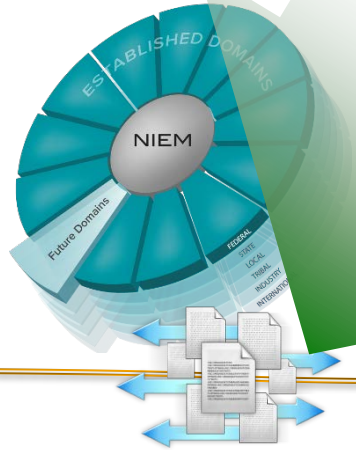
PEOPLE

NIEM-UML Profile

Design and Model

NIEM-UML Information Exchange Model



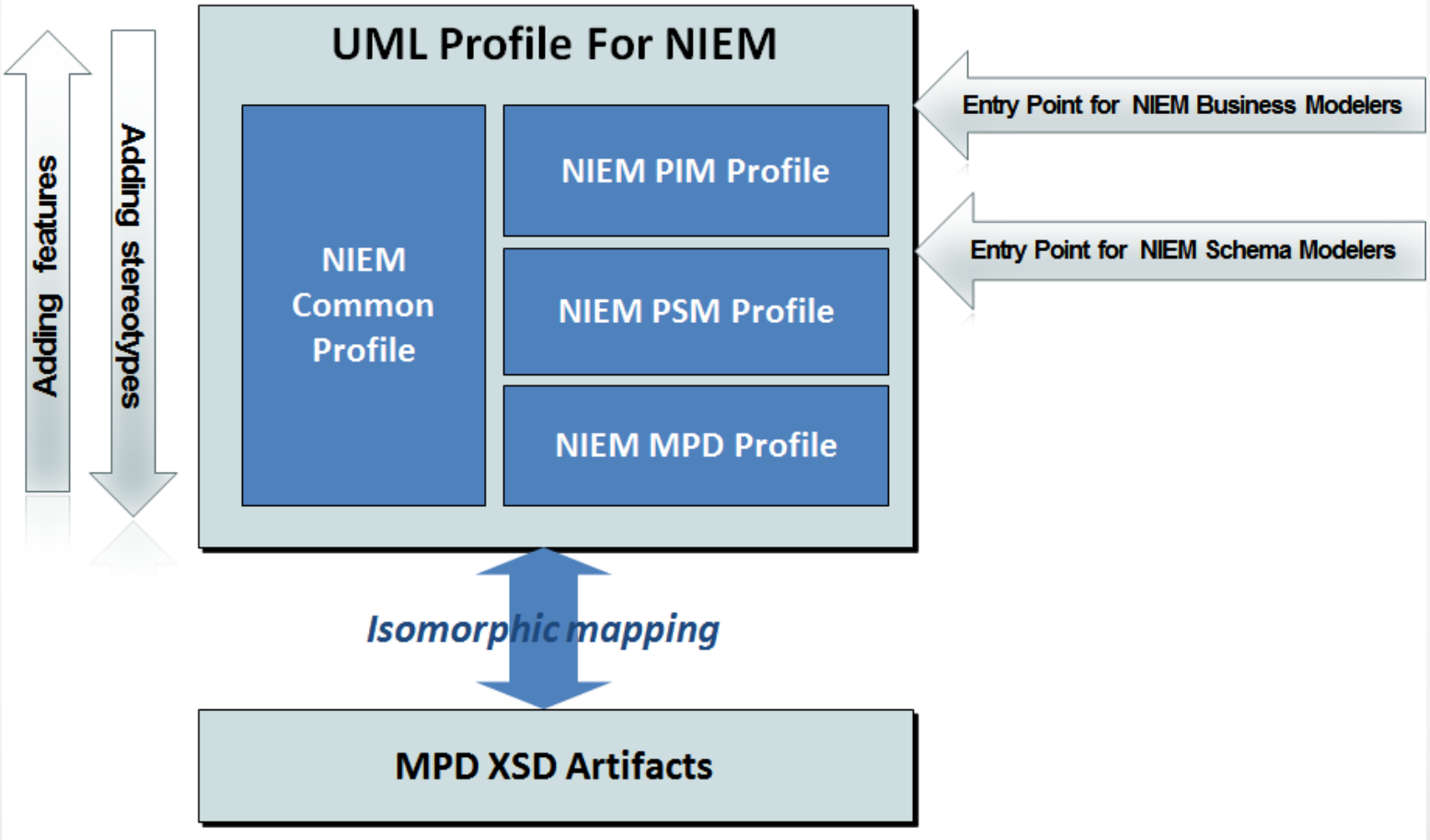


ORGANIZATIONS

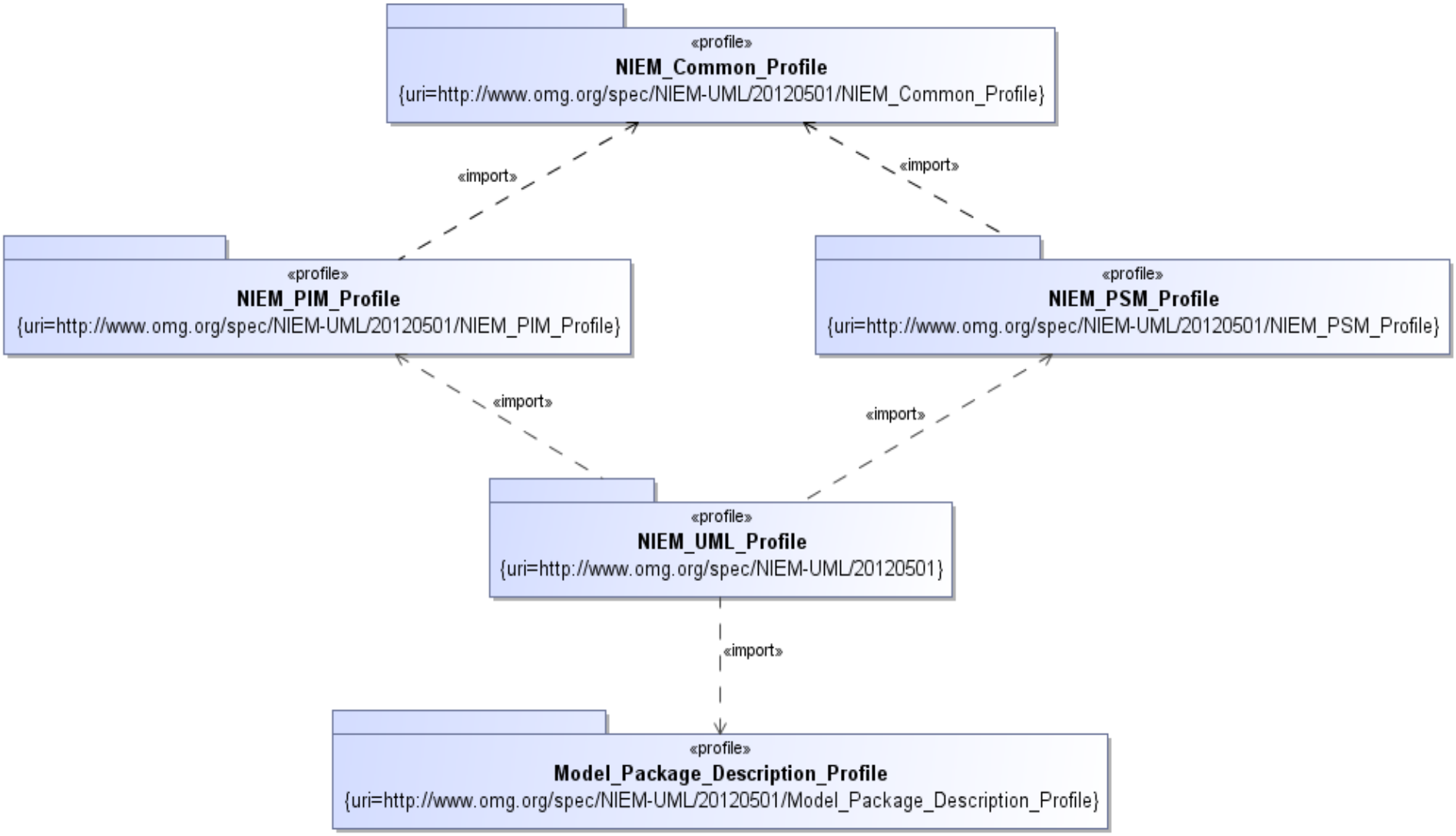
ORGANIZATIONS

XML data "on the wire"

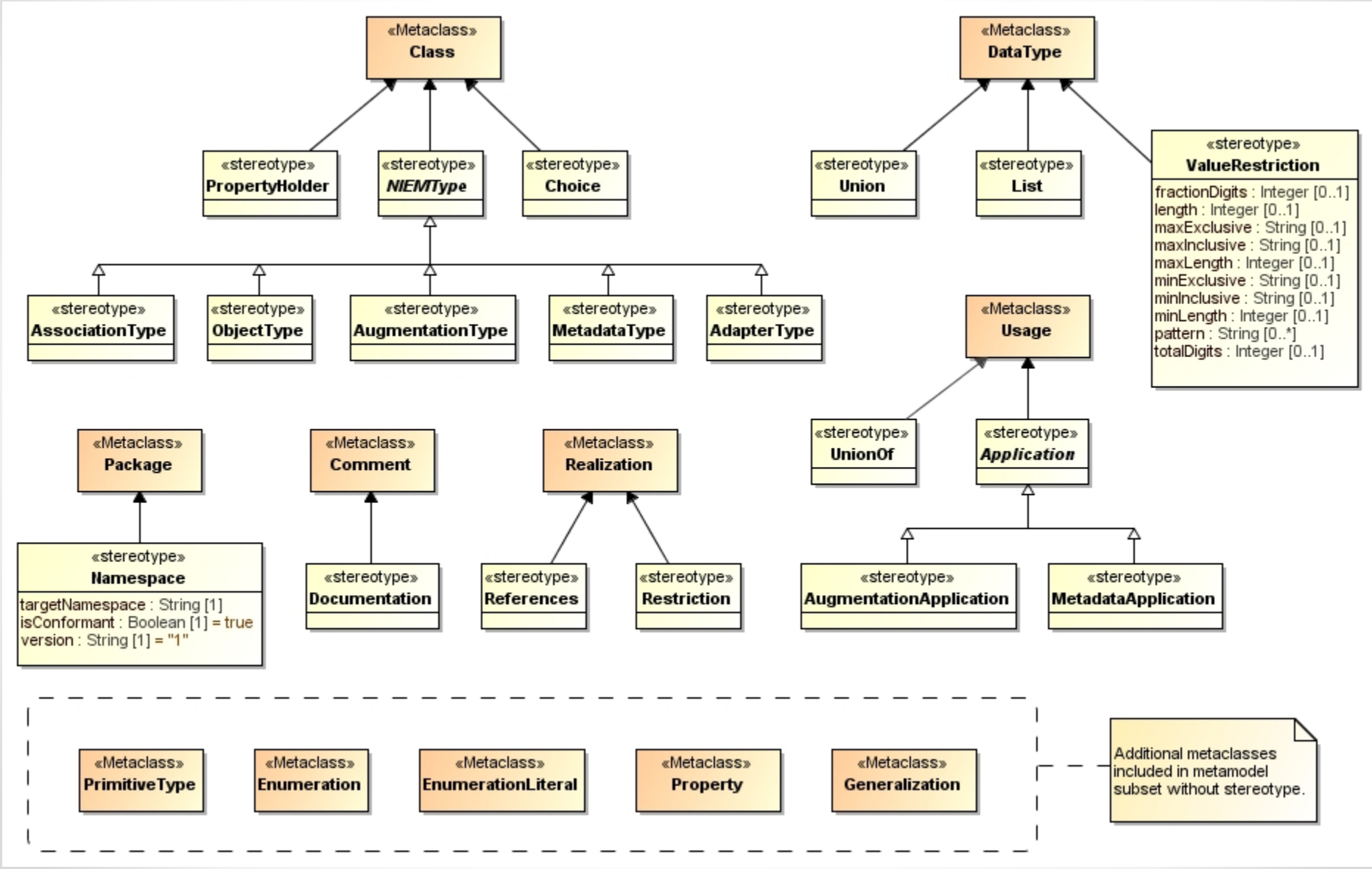
NIEM UML Profile Components



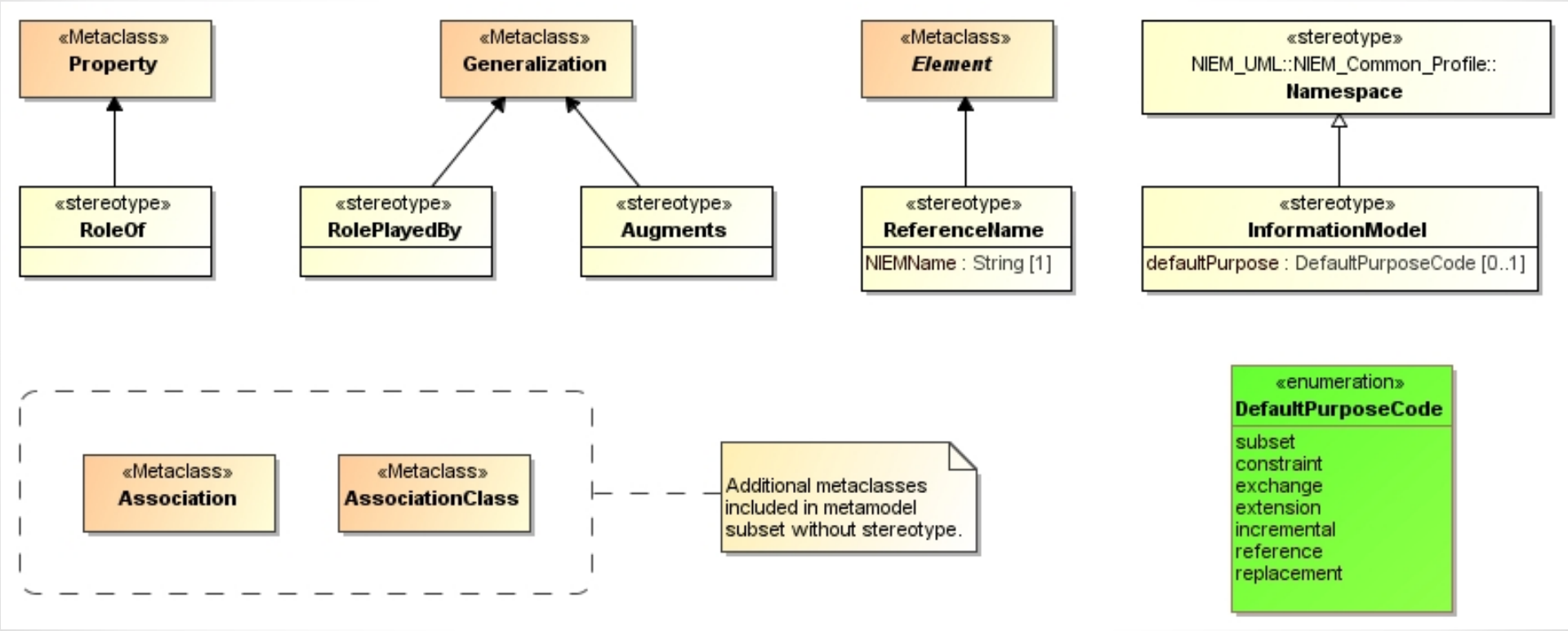
NIEM-UML Profile Structure



Common Profile



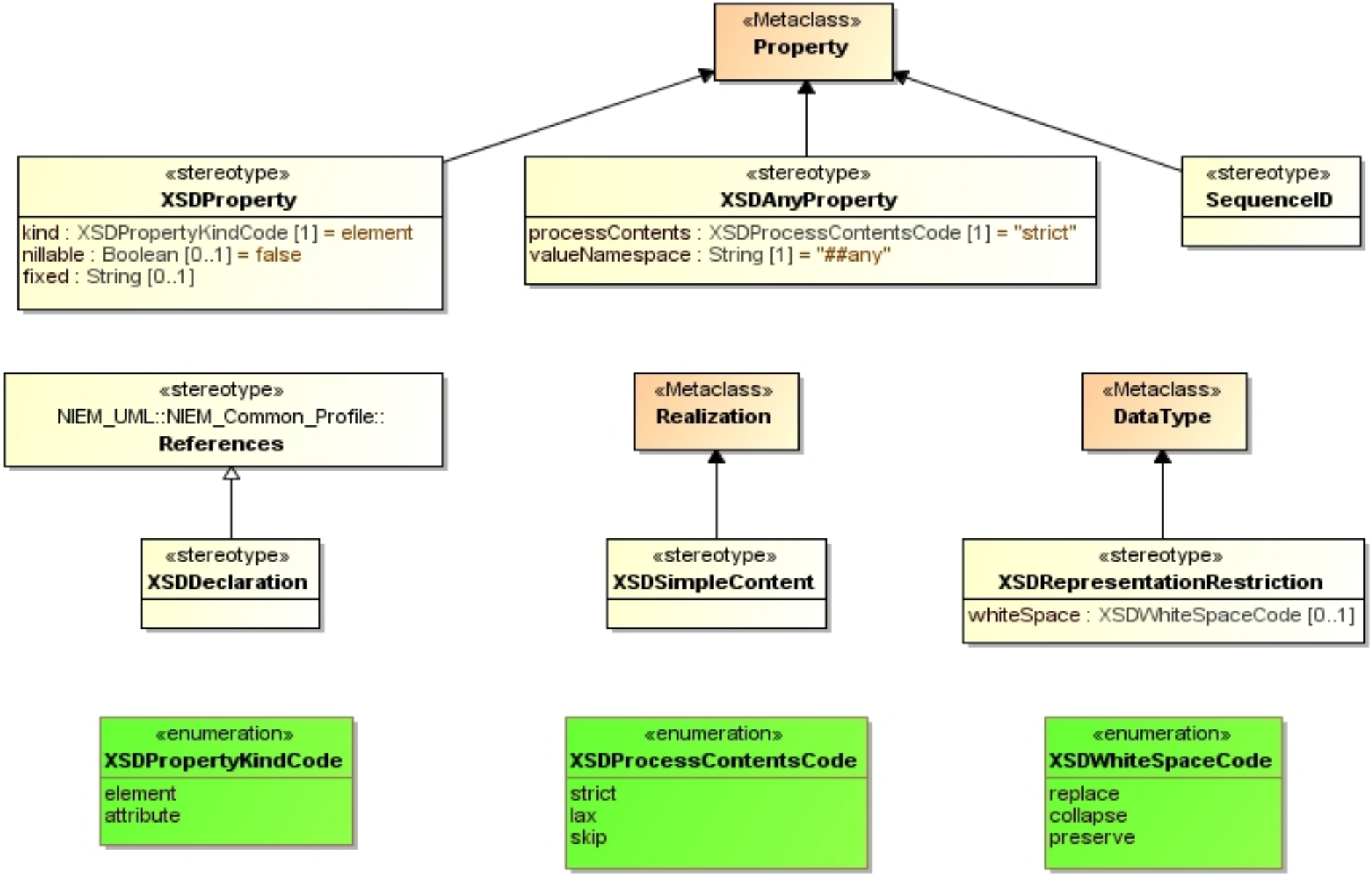
Platform Independent Profile



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 business concepts
- 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

Platform Specific Profile



What is the NIEM PSM Profile?

- A simplified subset of the Unified Modeling Language (UML)
- A set of UML constructs and stereotypes
  - Extends UML to represent NIEM **technical concepts**
  - Technical 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 technical concepts

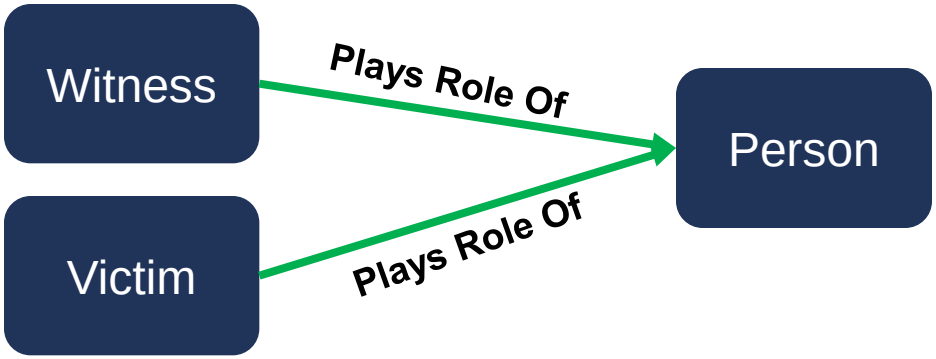
Why Roles?

- Roles are special relationships between an object and a temporary function or purpose of that object
- In this case a specialization does not reflect reality:
  - A Person is only a police officer, a victim, or a witness for a particular period of time in a particular context
  - A Person can act in multiple roles. A single person can play the role of a police officer, a victim, and a witness, all at the same time in a particular Incident
- In NIEM, Roles apply to People, Organizations and Items

Do not use a Specialization to implement Roles
(doing so makes the Role permanent)

Roles in NIEM

- Role types are used to indicate a temporary Role that an object plays within an exchange



- NIEM does not support lists of roles for an object; each role must be independently defined
- Role types include a reference to the object playing the role, as well as additional properties that are specific to the role
- *Specialization* should not be used to implement roles – In NIEM, a *Specialization* is used when an object uses a previously defined type to define a more granular type, and not to define a temporary role of an object

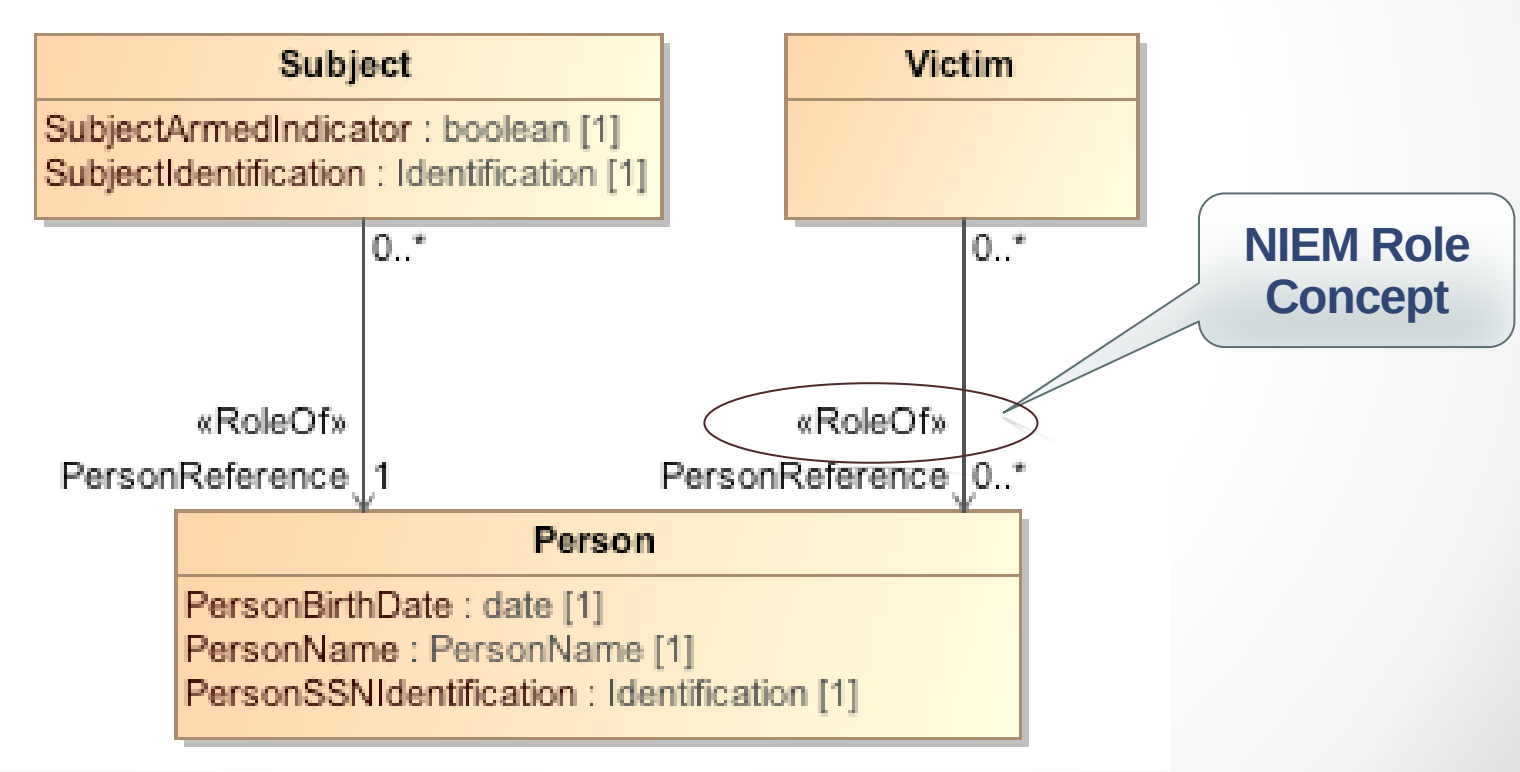
Roles in NIEM

- Role types are used to represent roles which allow for additional role-related information to be associated to a role
- *RoleOf* references must exist within the role type to identify the object playing the role
- Three types of *RoleOf* reference exist:
  - RoleOfPersonReference *(For ex: Victim, Subject)*
  - RoleOfOrganizationReference *(For ex: CriminalOrganization, LeinHolder)*
  - RoleOfItemReference *(For ex: Weapon, CrashVehicle)*
- The *RoleOf* references are all of type `s:ReferenceType` which allows the use of the `s:ref` attribute to link to an object with a matching `s:id` attribute

NIEM Roles

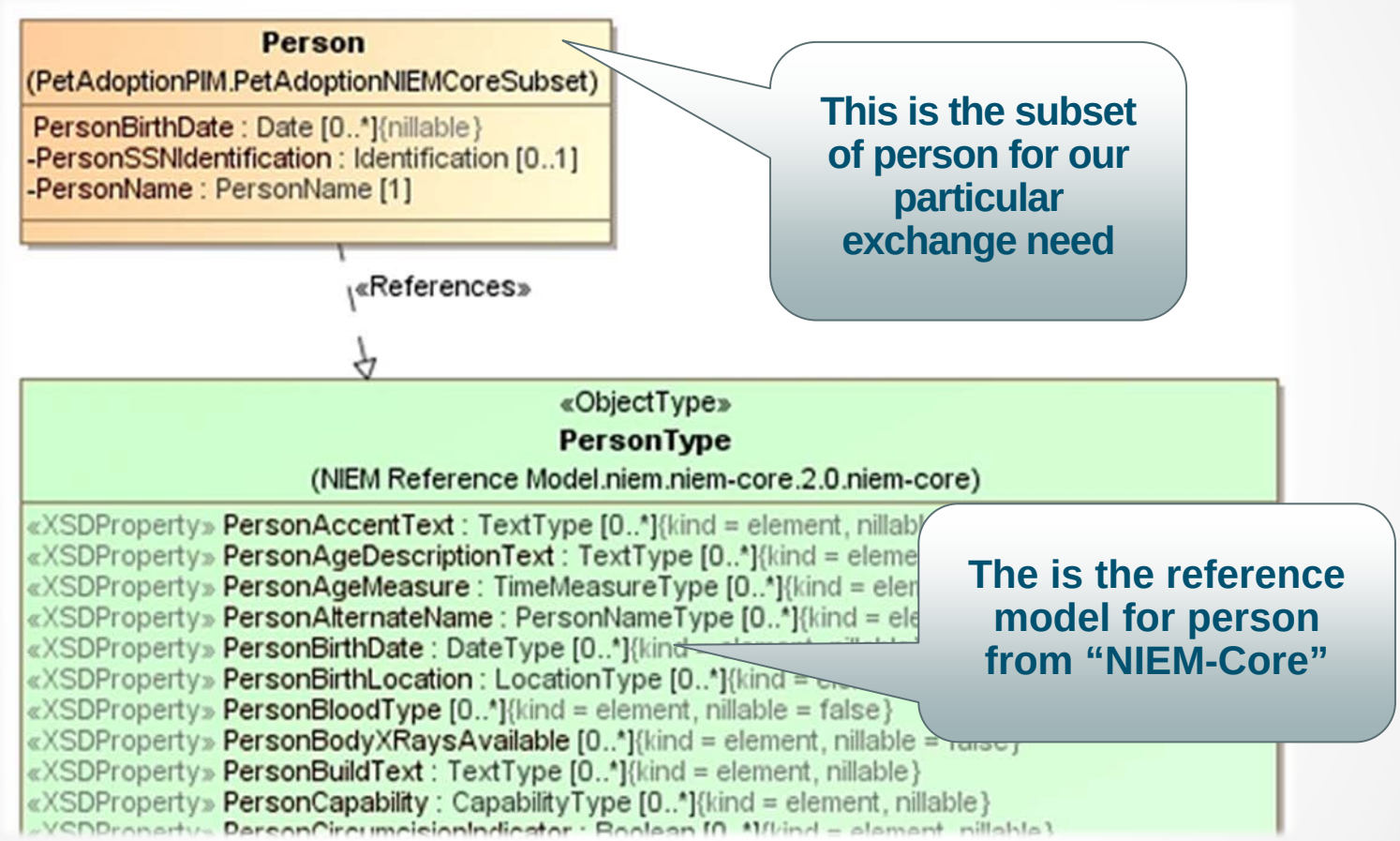
NIEM Roles

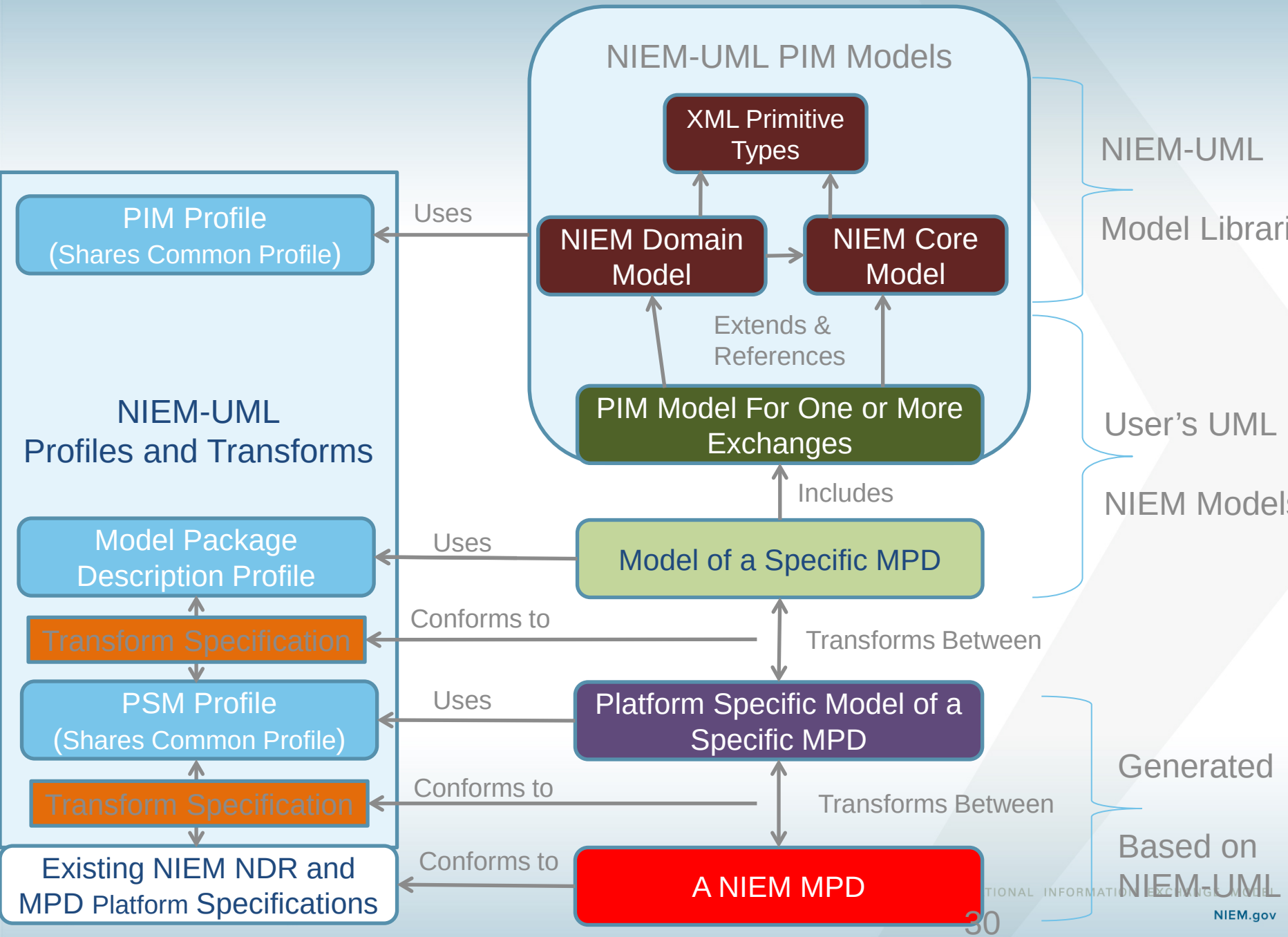
UML represents roles in their simpler form as UML association ends (The names on the ends of lines in a class diagram) or properties. To represent roles that are complex types a class or data type is used.

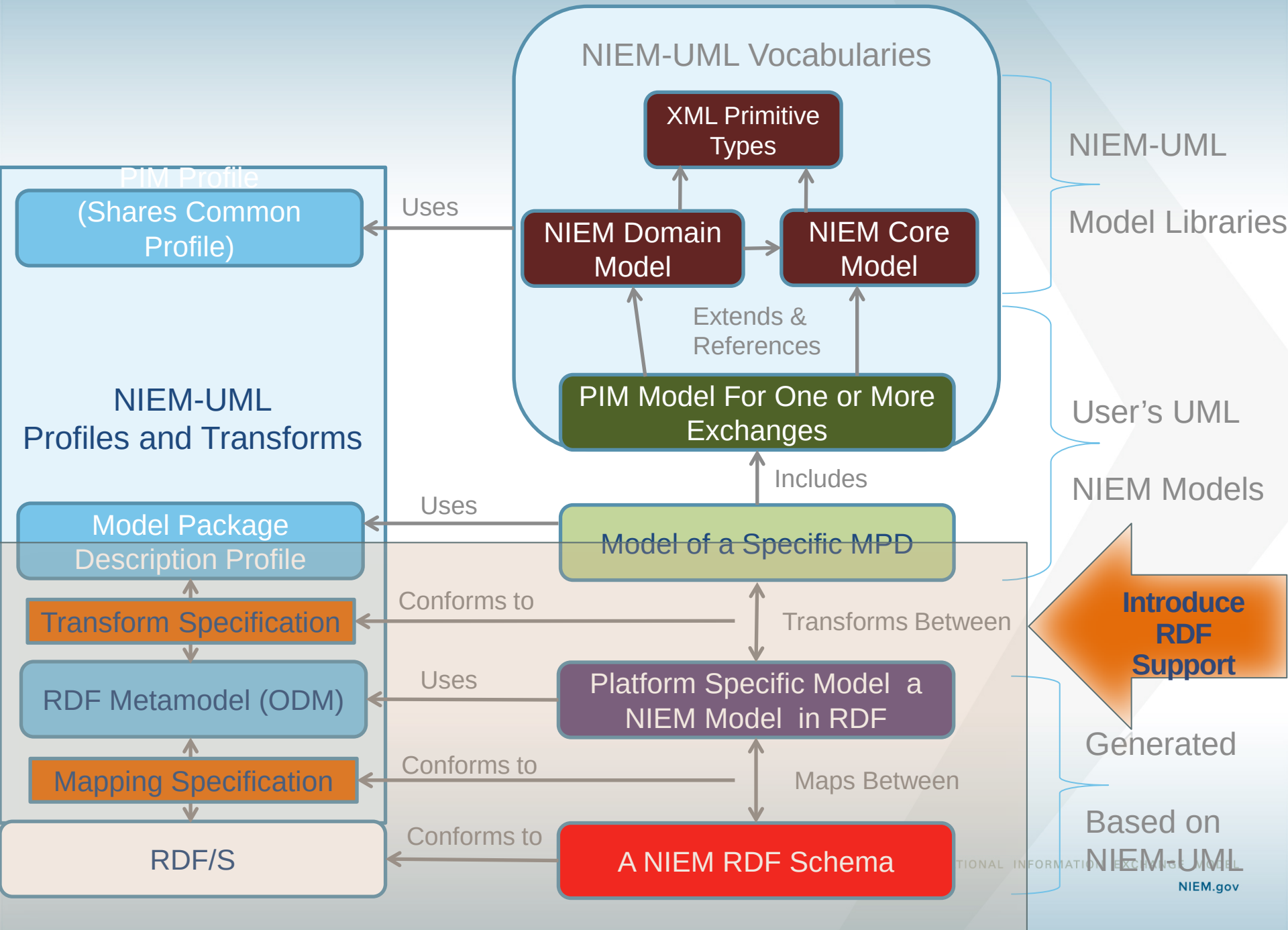


Efficiency in Maintaining Exchanges

Modifying an Existing Information Exchange




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


NIEM-UML Vocabularies
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
Generated
Based on NIEM-UML
PIM Profile
(Shares Common Profile)
Uses
NIEM-UML Profiles and Transforms
Includes
Model Package Description Profile
Uses
Model of a Specific MPD
Introduce RDF Support
Transform Specification
Conforms to
Transforms Between
RDF Metamodel (ODM)
Uses
Platform Specific Model a NIEM Model in RDF
Mapping Specification
Conforms to
Maps Between
RDF/S
Conforms to
A NIEM RDF Schema
NIEM.gov

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 creating information exchange using NIEM
- 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
- Model Driven Architecture also provides for other aspects of the information sharing solution, such as: business processes, SOA services and back-end system integration

An UML Profile for the GRA SSP

1. Leverage SoaML
2. Provide for a Platform Independent Model (PIM) of a service which corresponds to the Service Description
3. Provide a Platform Specific Models to a service which would result in Service Interface Descriptions and other platform specific service artifacts.
4. Meet all of the normative requirements described in the GRA SSG.
5. Allow generation of SSP artifacts from the model. To achieve this it should allow modeling the following key aspects of a GRA SSP
   - The metadata for the service
   - The capabilities provided by the service
   - The real worlds effects realized through the service
   - The business use cases associated with the service
   - The information model of the service
   - The behavior model of the service
6. The information model of the UML Profile for GRA SSP should leverage NIEM-UML

Pet Adoption Example

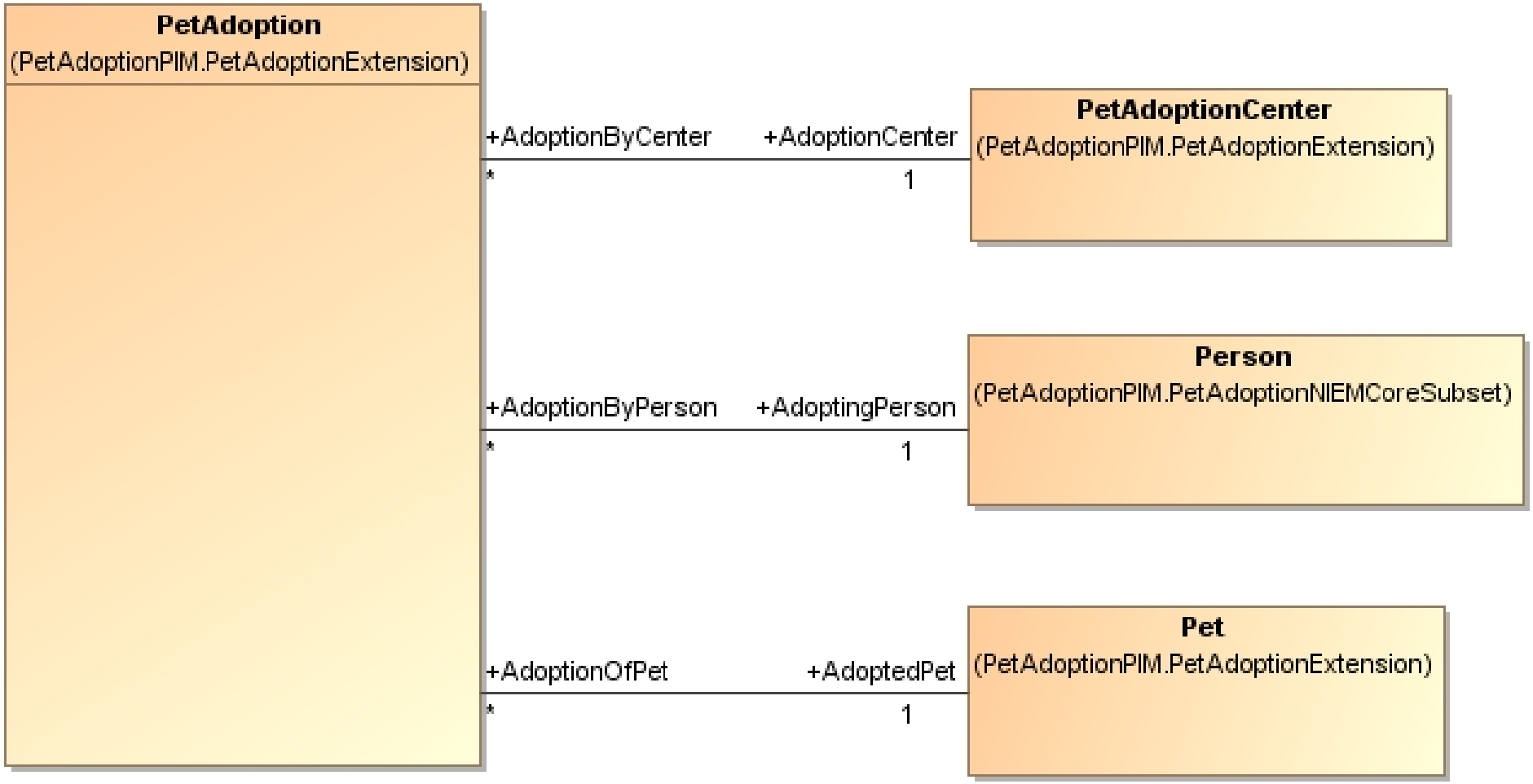
Data Exchange of adoptions by pet rescue centers

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

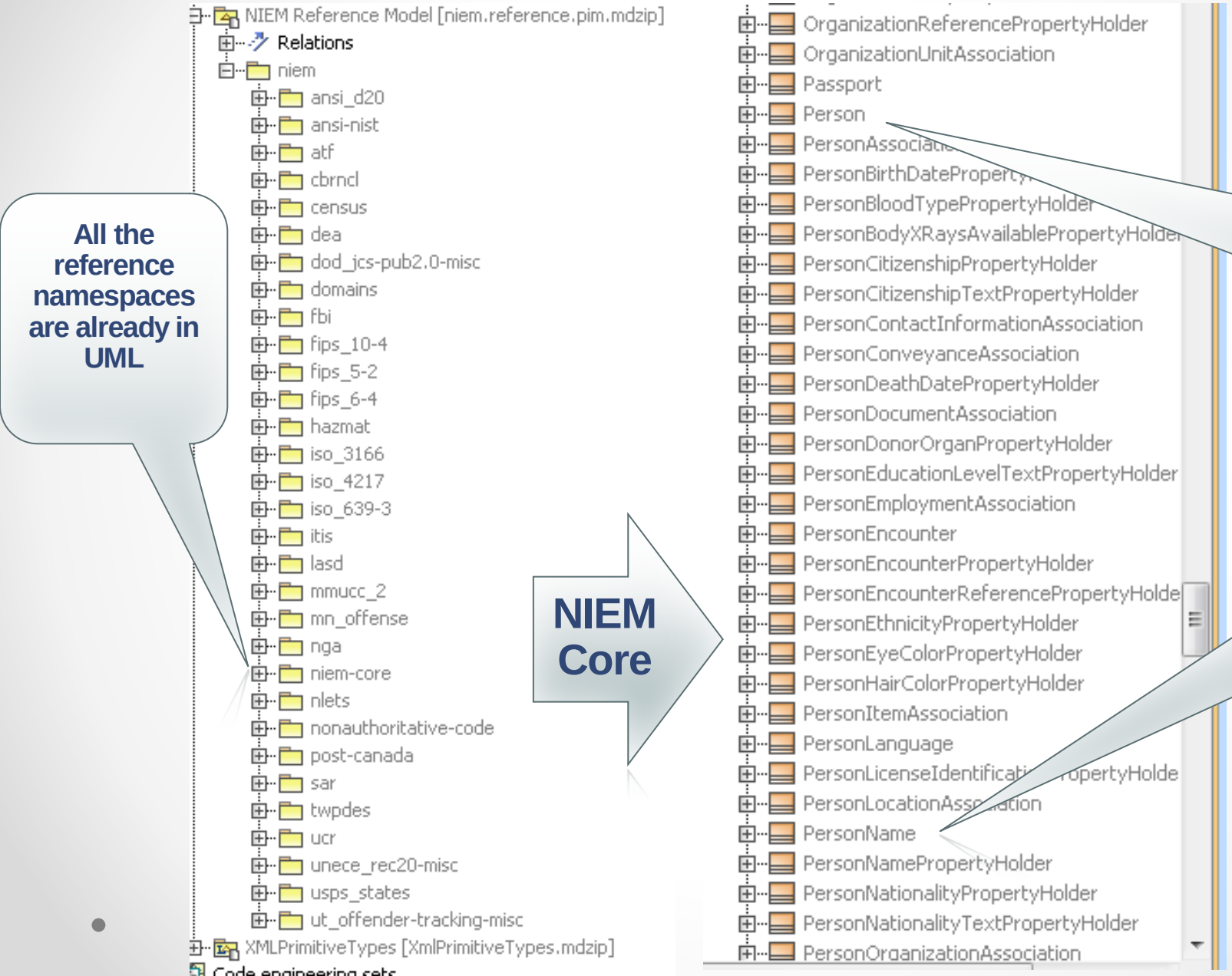
Information to Exchange

- 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!



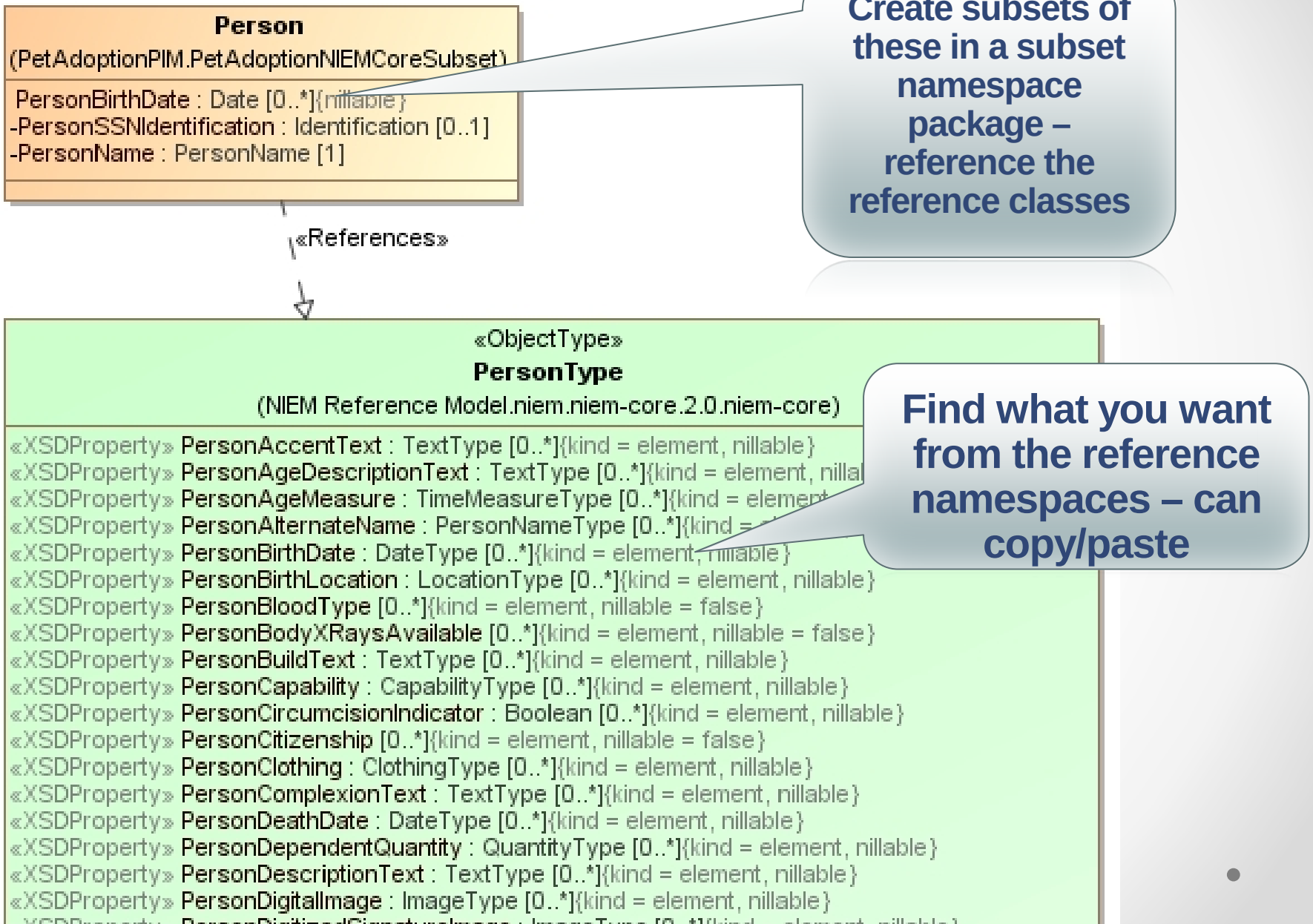
All the reference namespaces are already in UML

NIEM Core

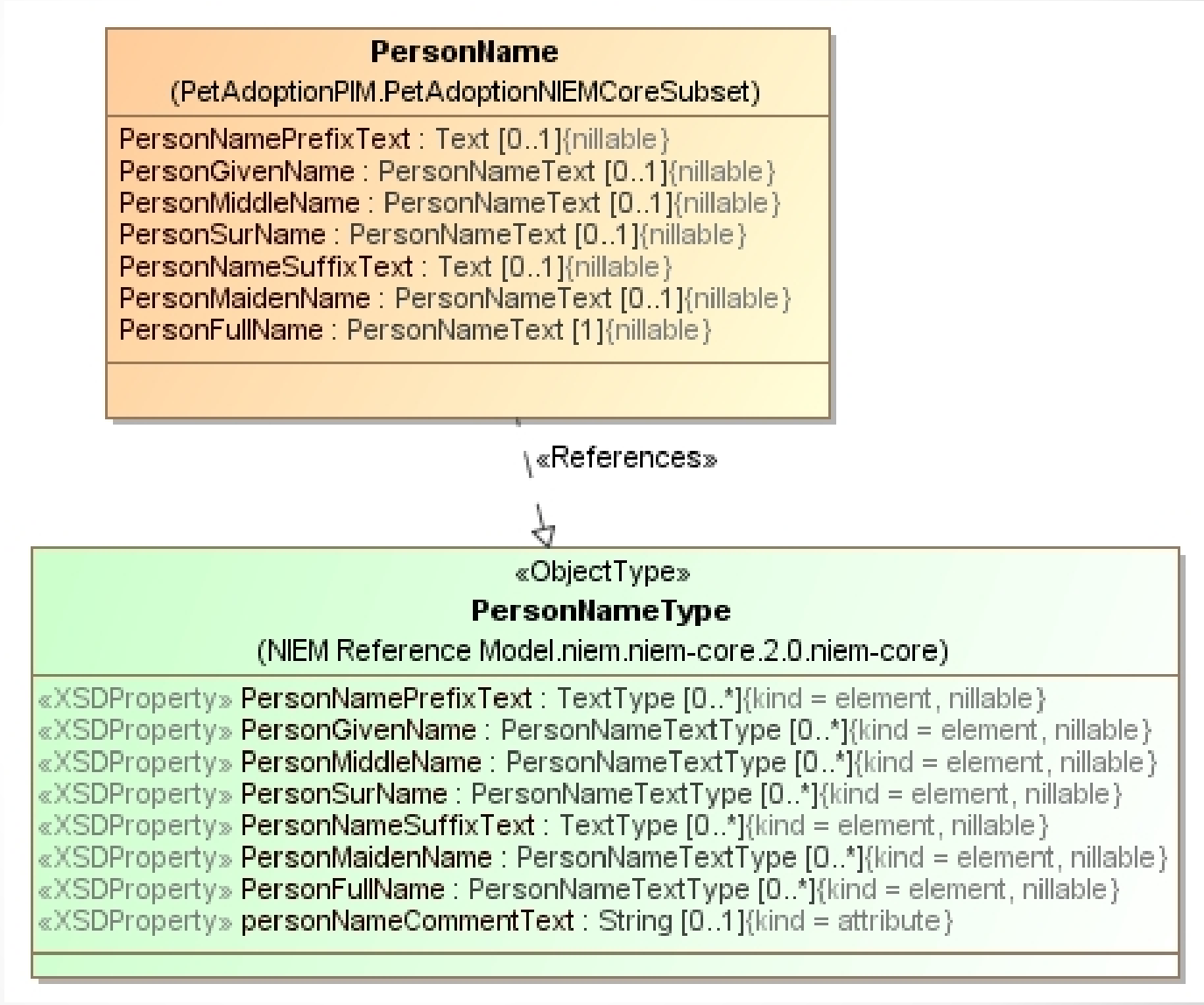
Find what you want to reuse in the reference namespaces

Find what you want to reuse in the reference namespaces

Model Reuse of NIEM Core

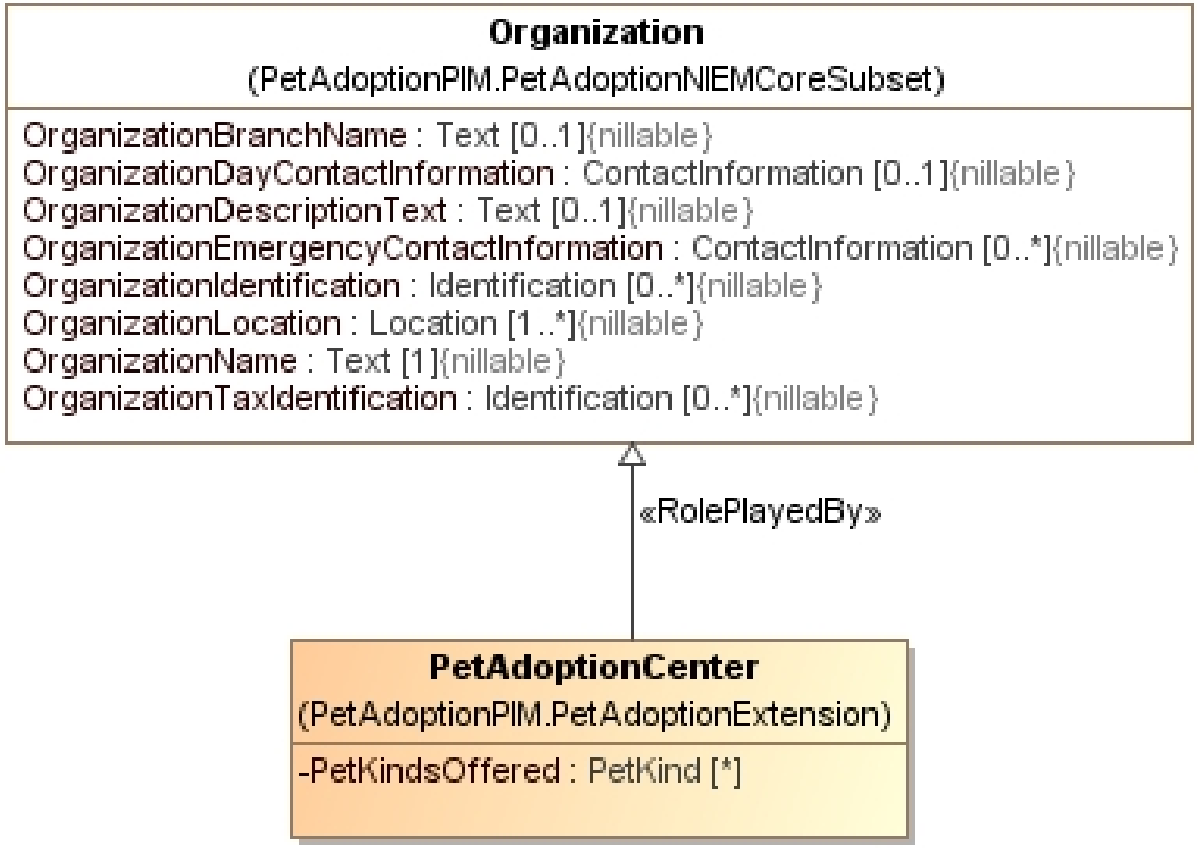


Repeat as required



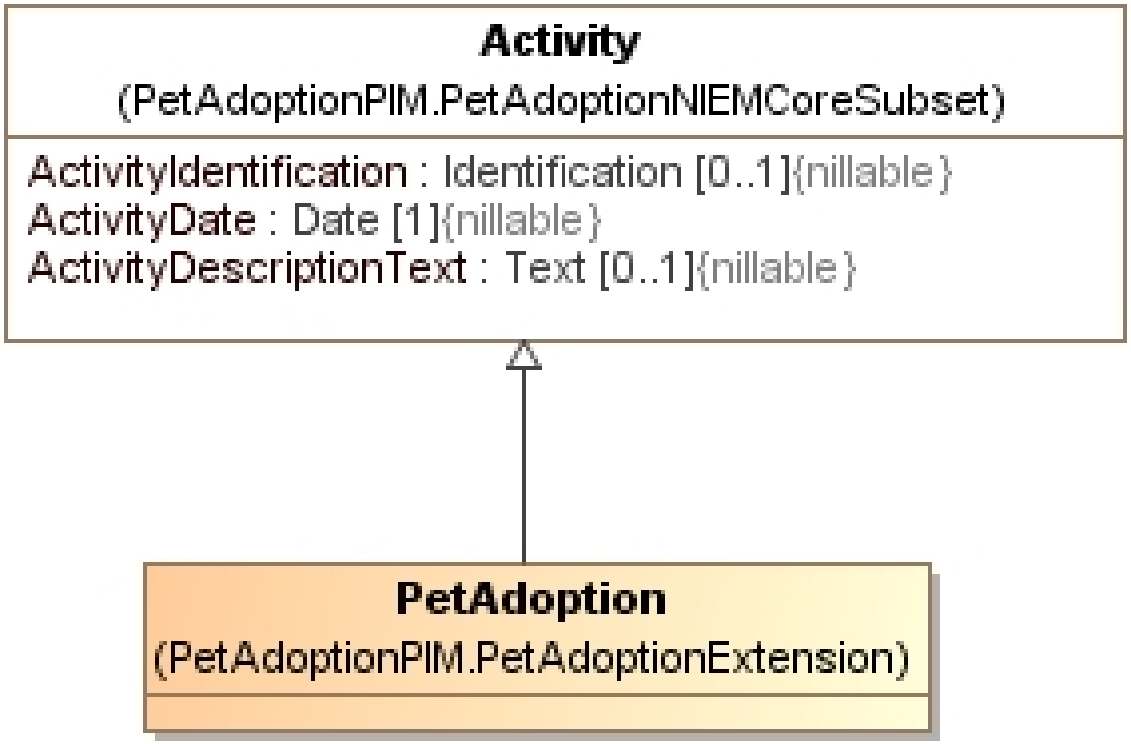
Roles of organizations

- 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”



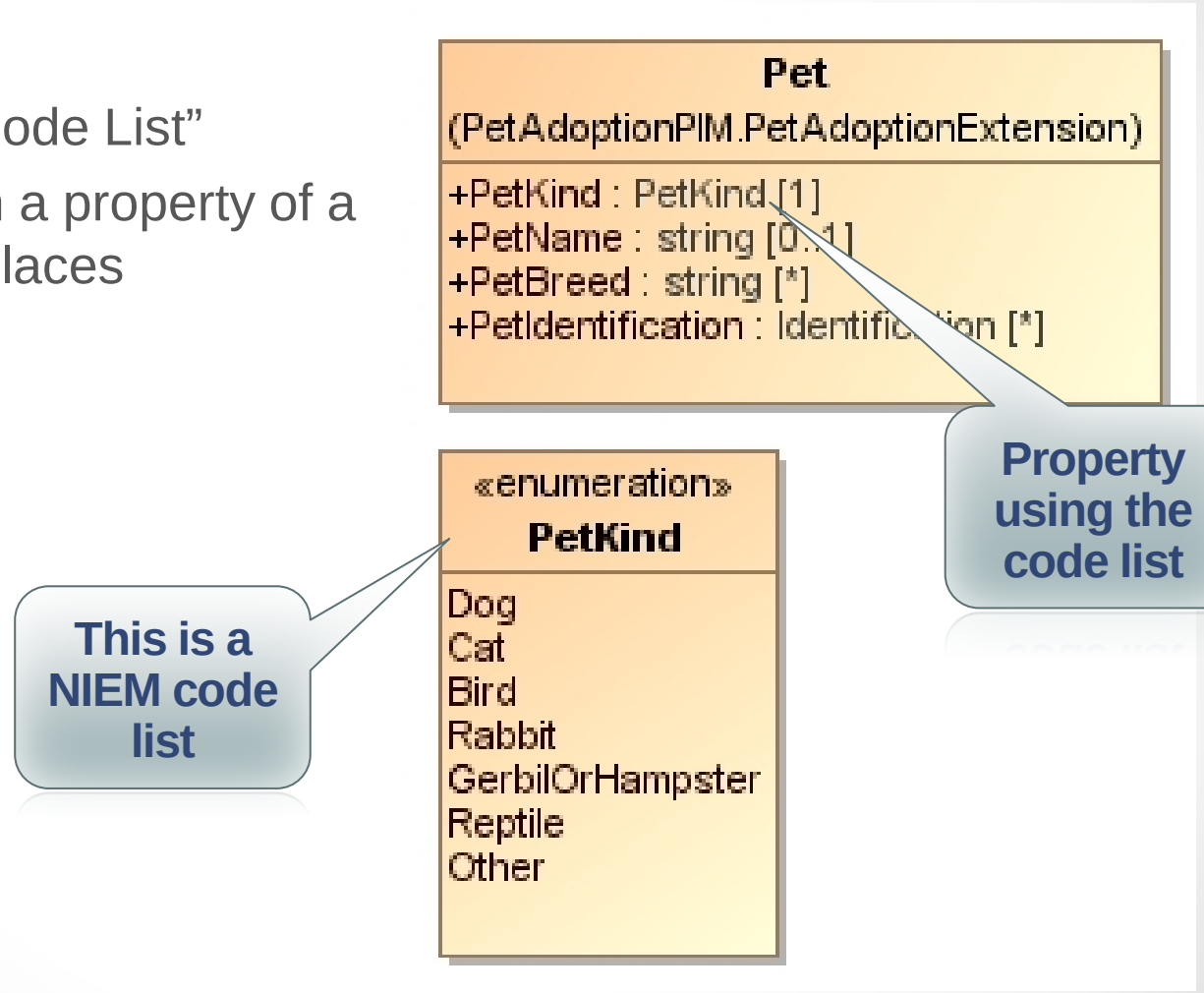
Adoptions as a NIEM Activity

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

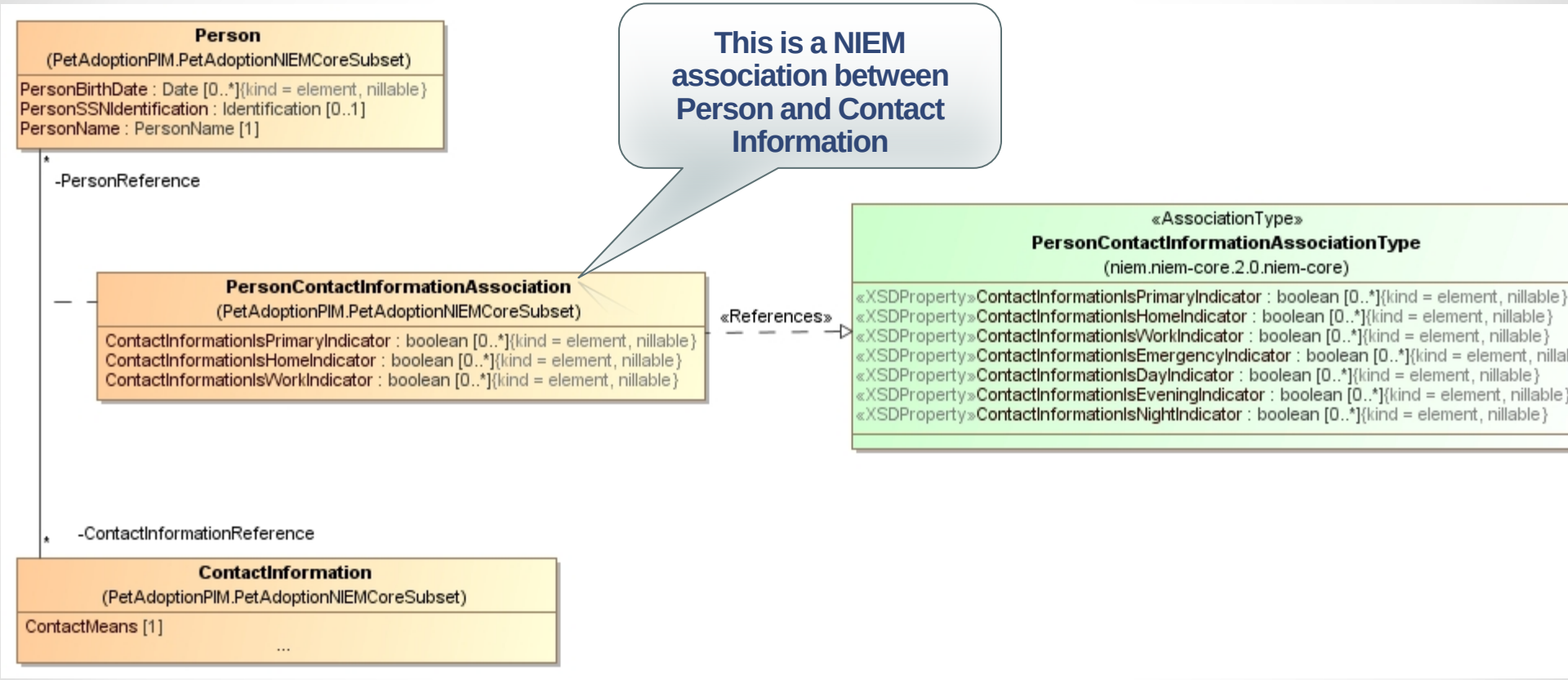


What Kinds Of Pets Are Adopted?

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

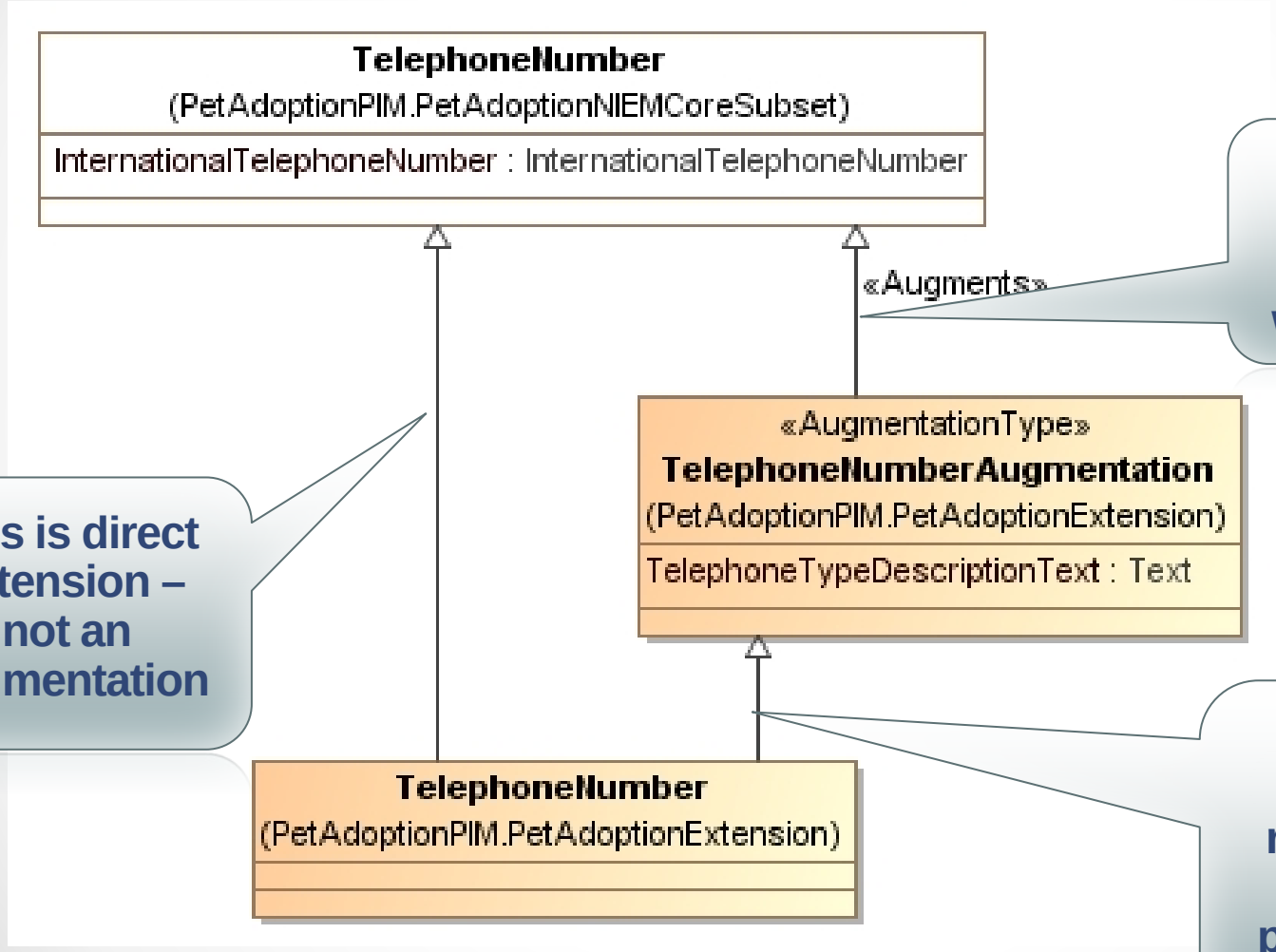


NIEM Associations



Associations Connect Objects – in this case people and contact information

Augmentations – Phone Number ++

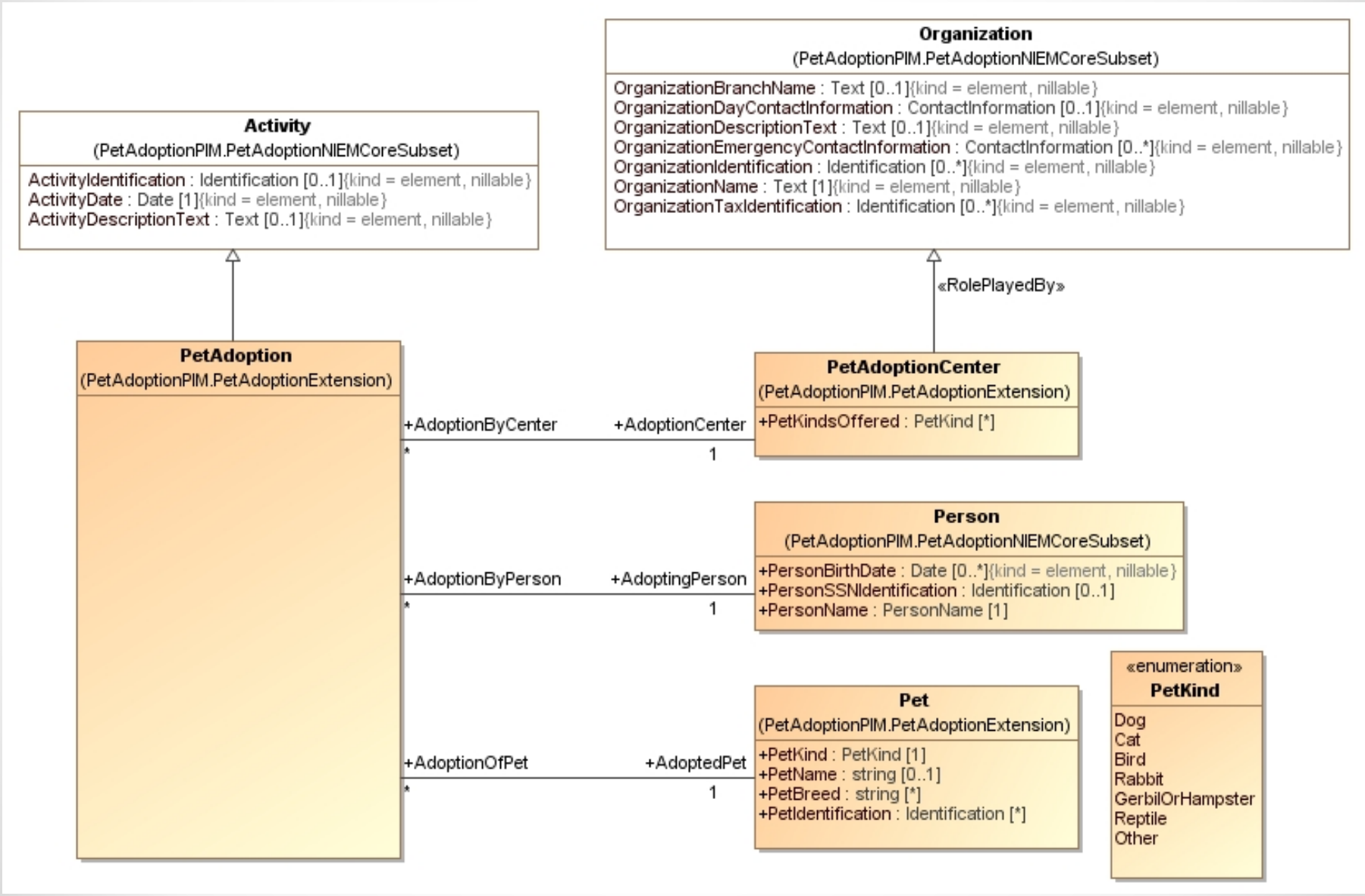


Optionally, an augmentation can be restricted to what it "applies to"

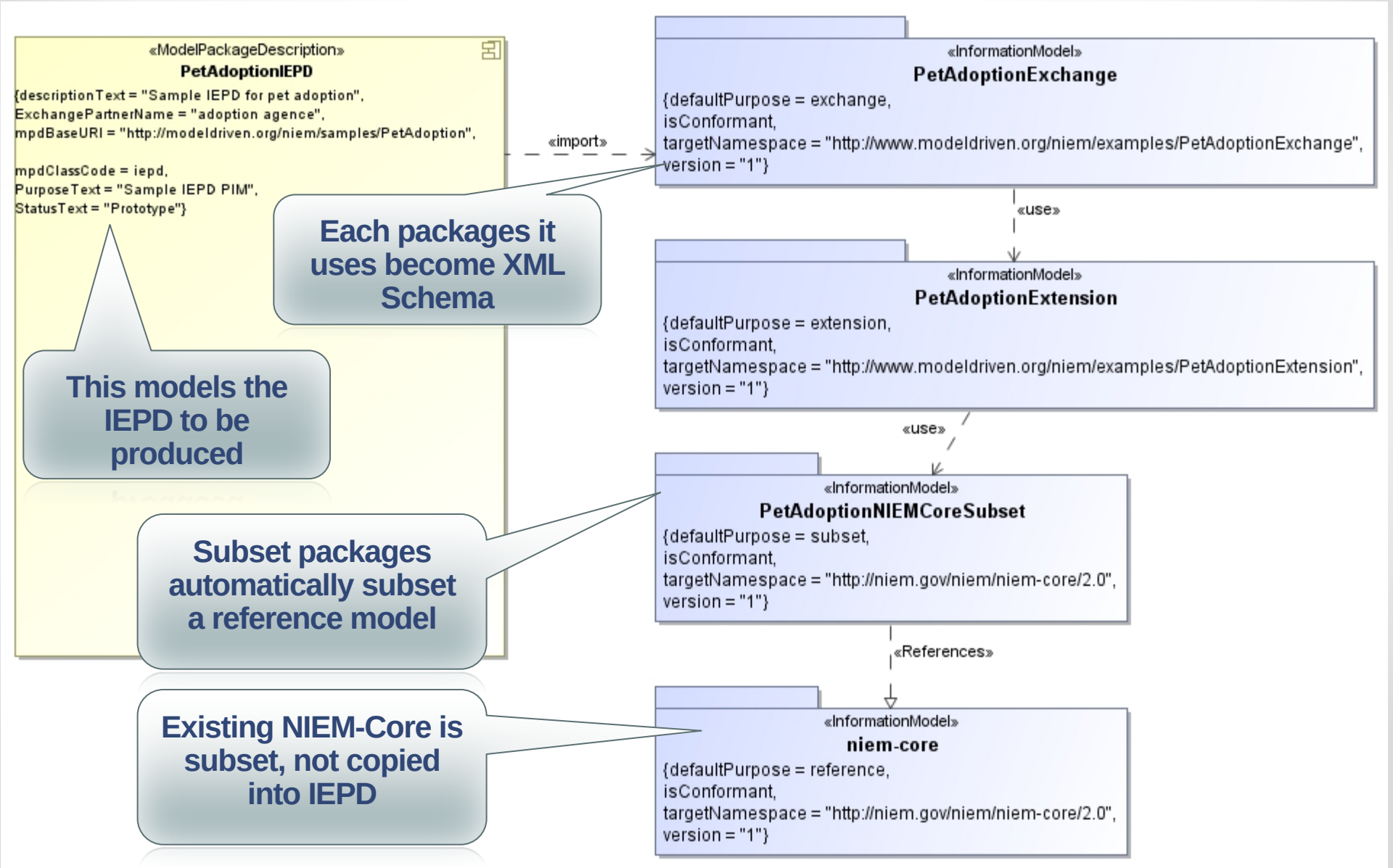
This is direct extension – not an augmentation

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

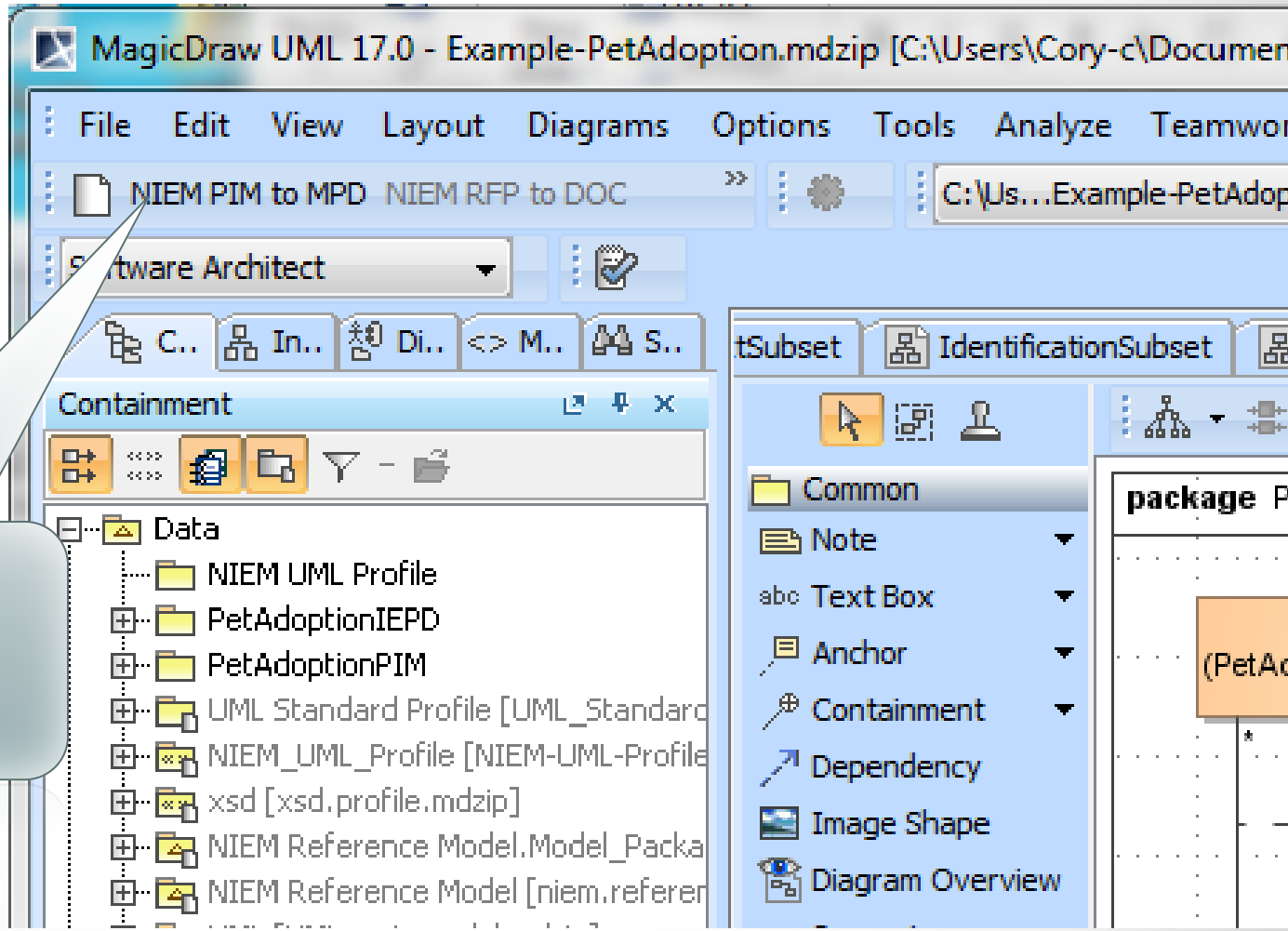
Completed High-Level Model



Adding the IEPD Metadata

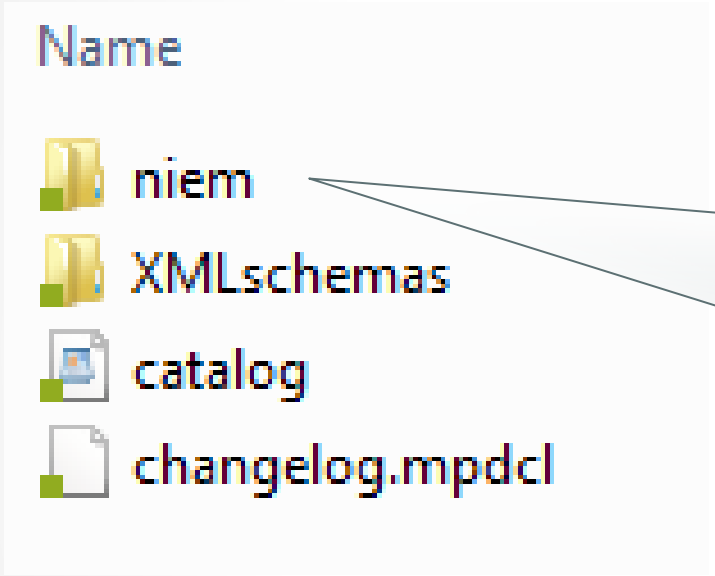


Create the IEPD from the model



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

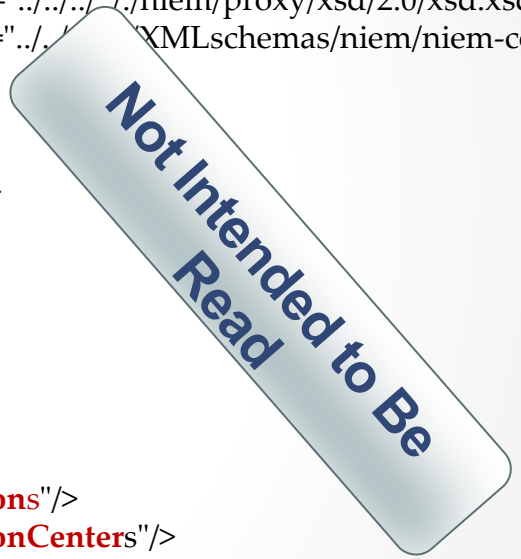
All the IEPD artifacts are then created by the MDA Automation



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?

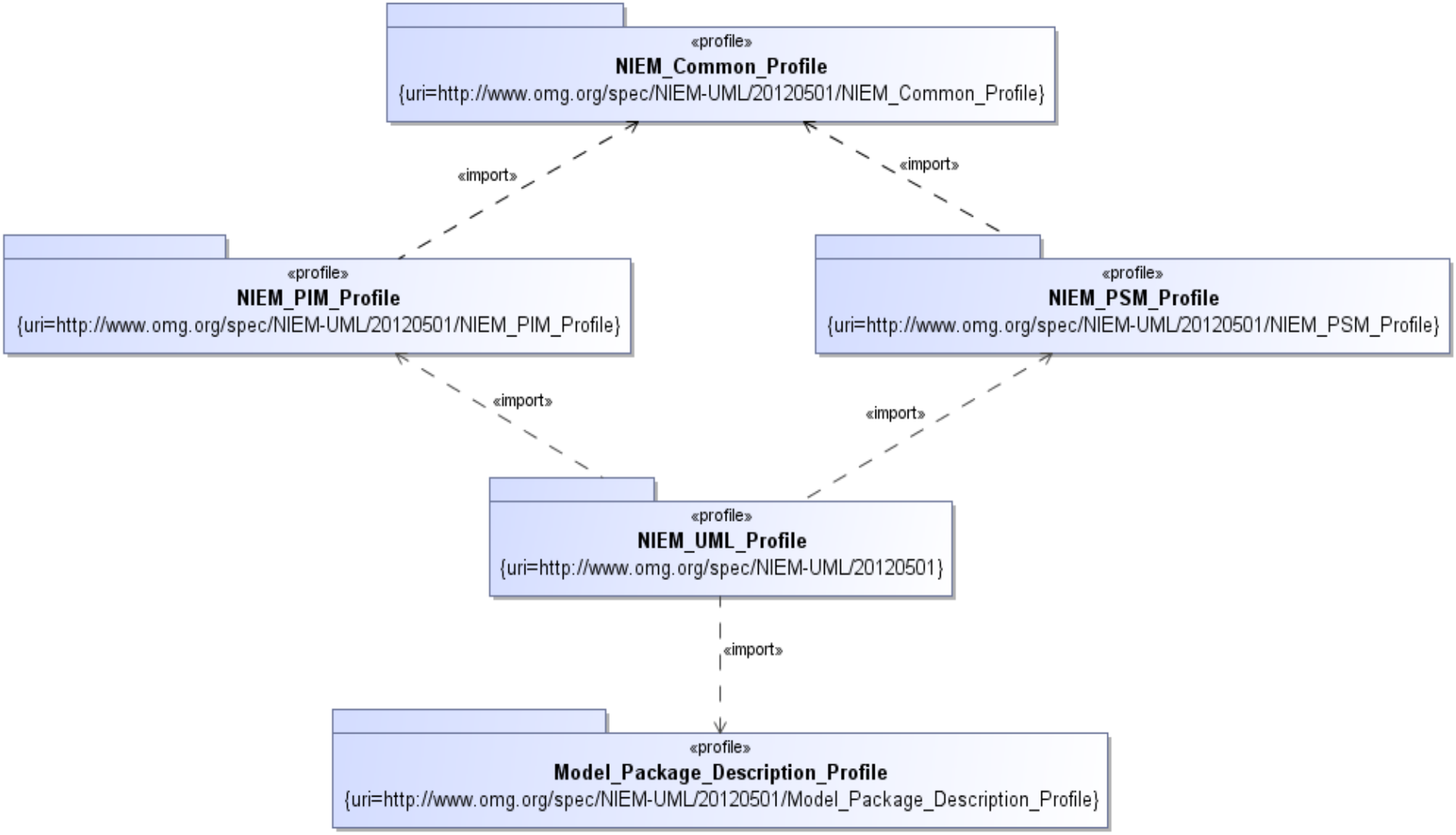
- NIEM Logical Concepts
  - Not the XSD and NDR Details
- The PIM and Common Profile
- The Model Package Description Profile
- A UML Tool (Class diagram subset)

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

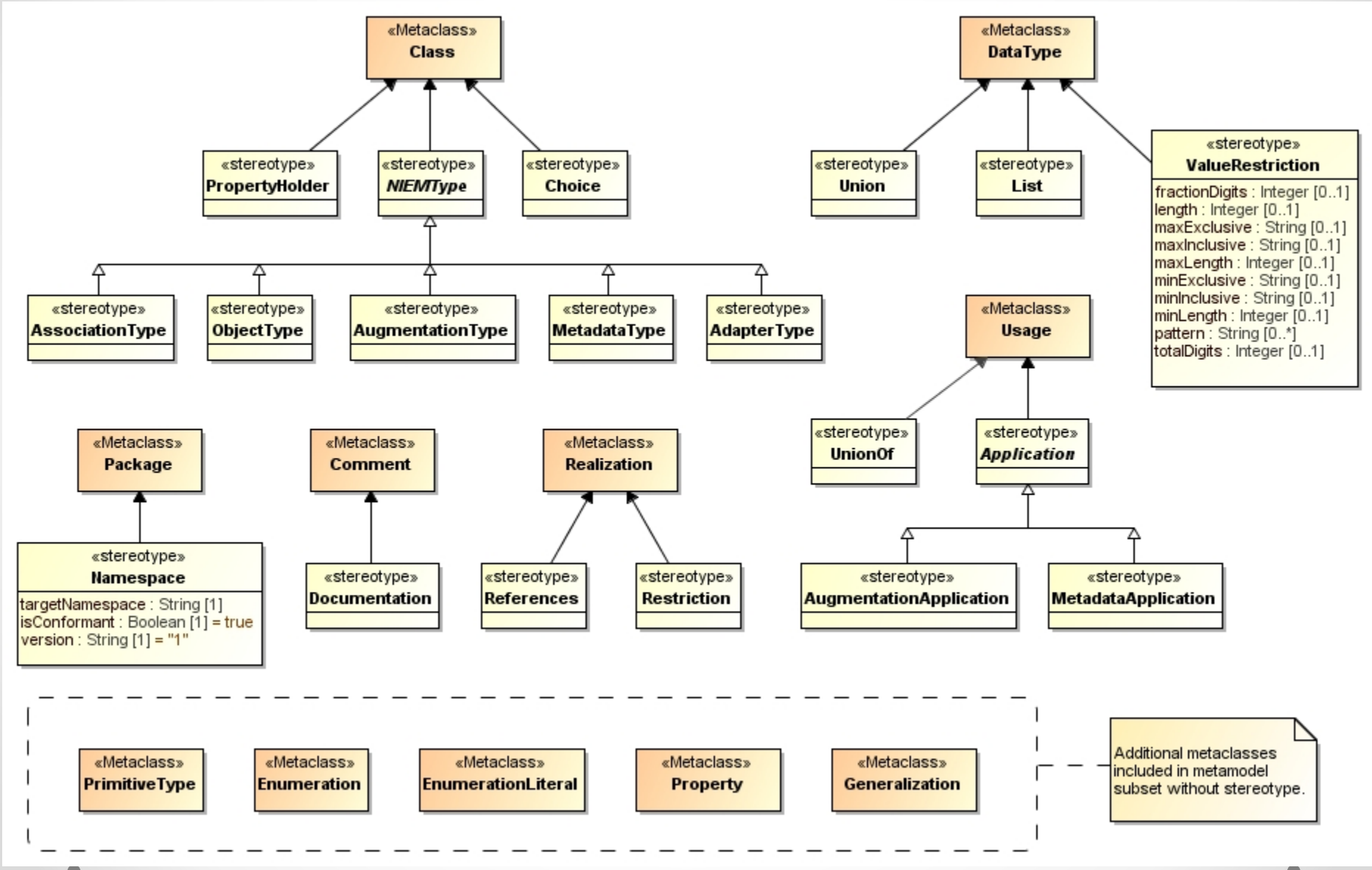
NIEM-UML Profiles

Getting Down to Detail

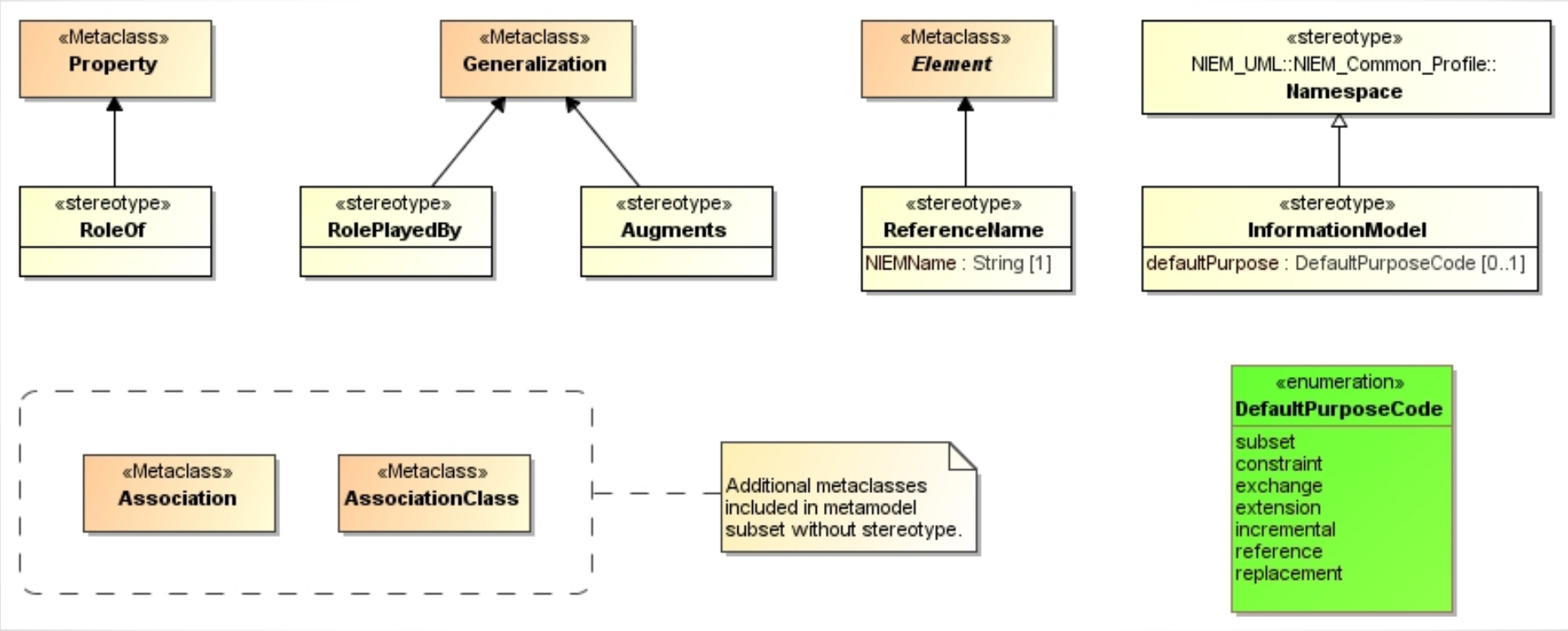
NIEM-UML Profile Structure



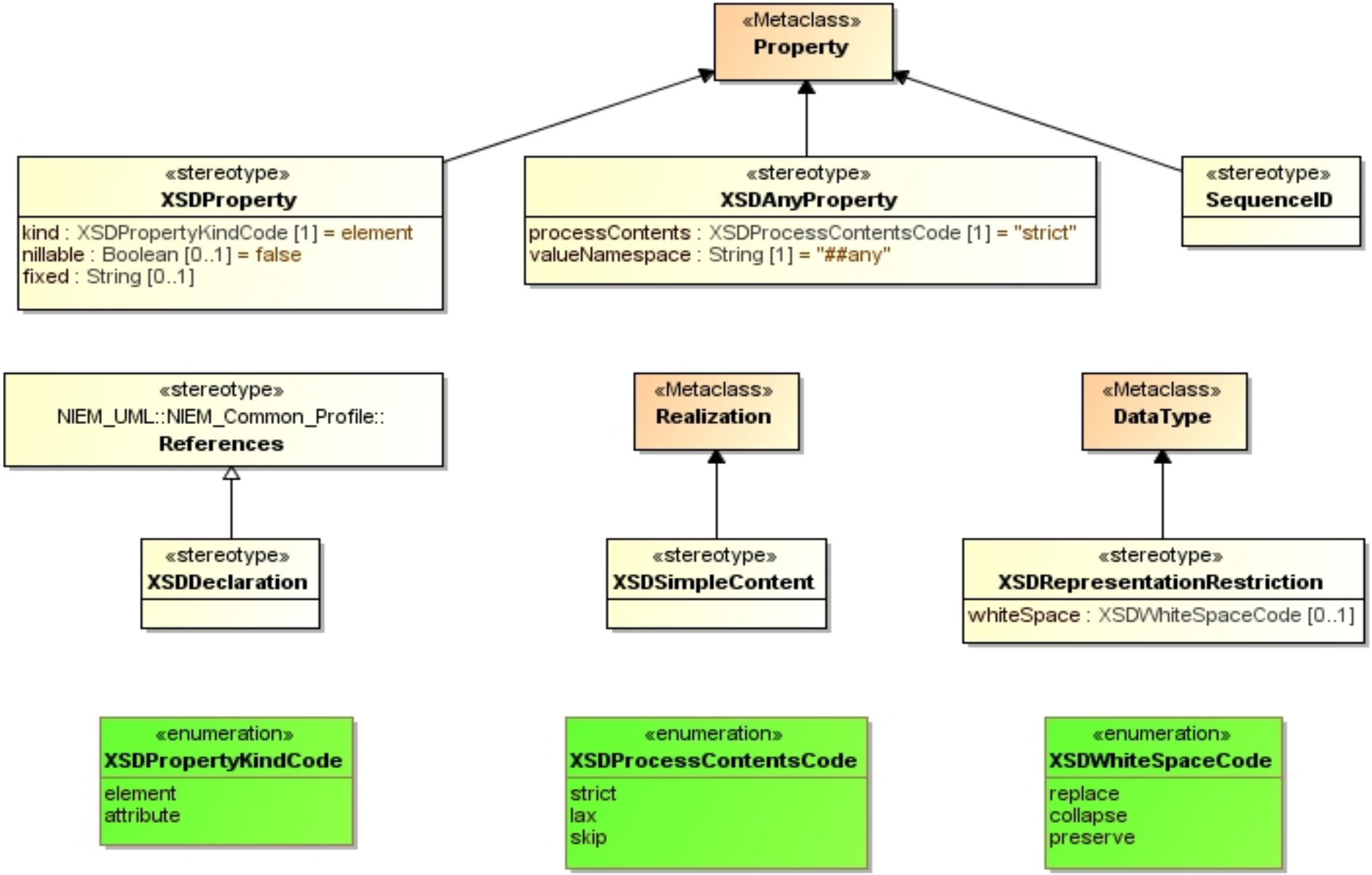
Common Profile



Platform Independent Profile



Platform Specific Profile

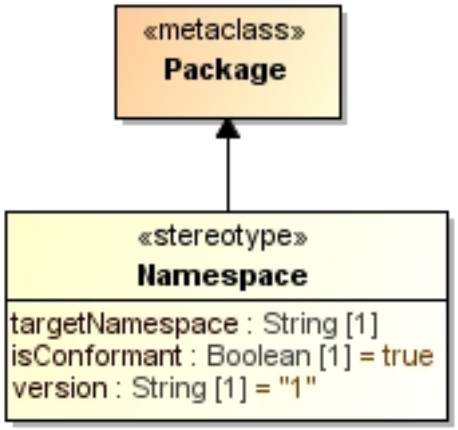


Namespaces

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.

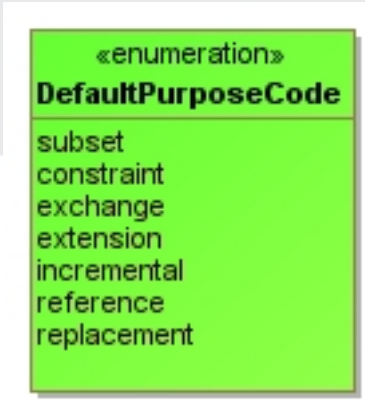
Example

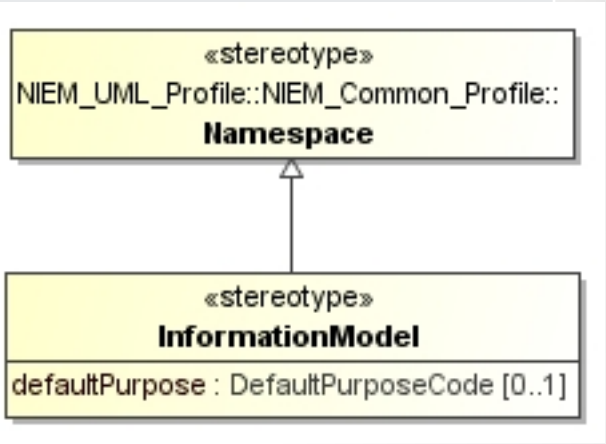


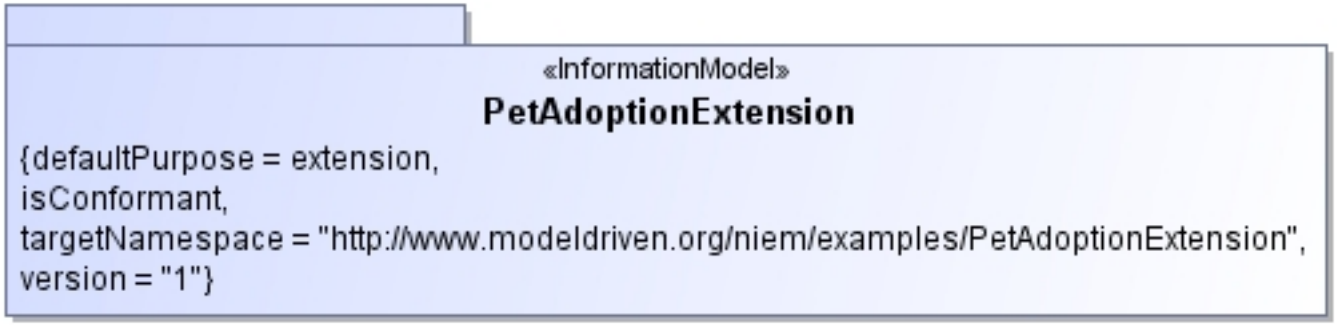
Information Models (PIM Only)

InformationModel (Package)

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



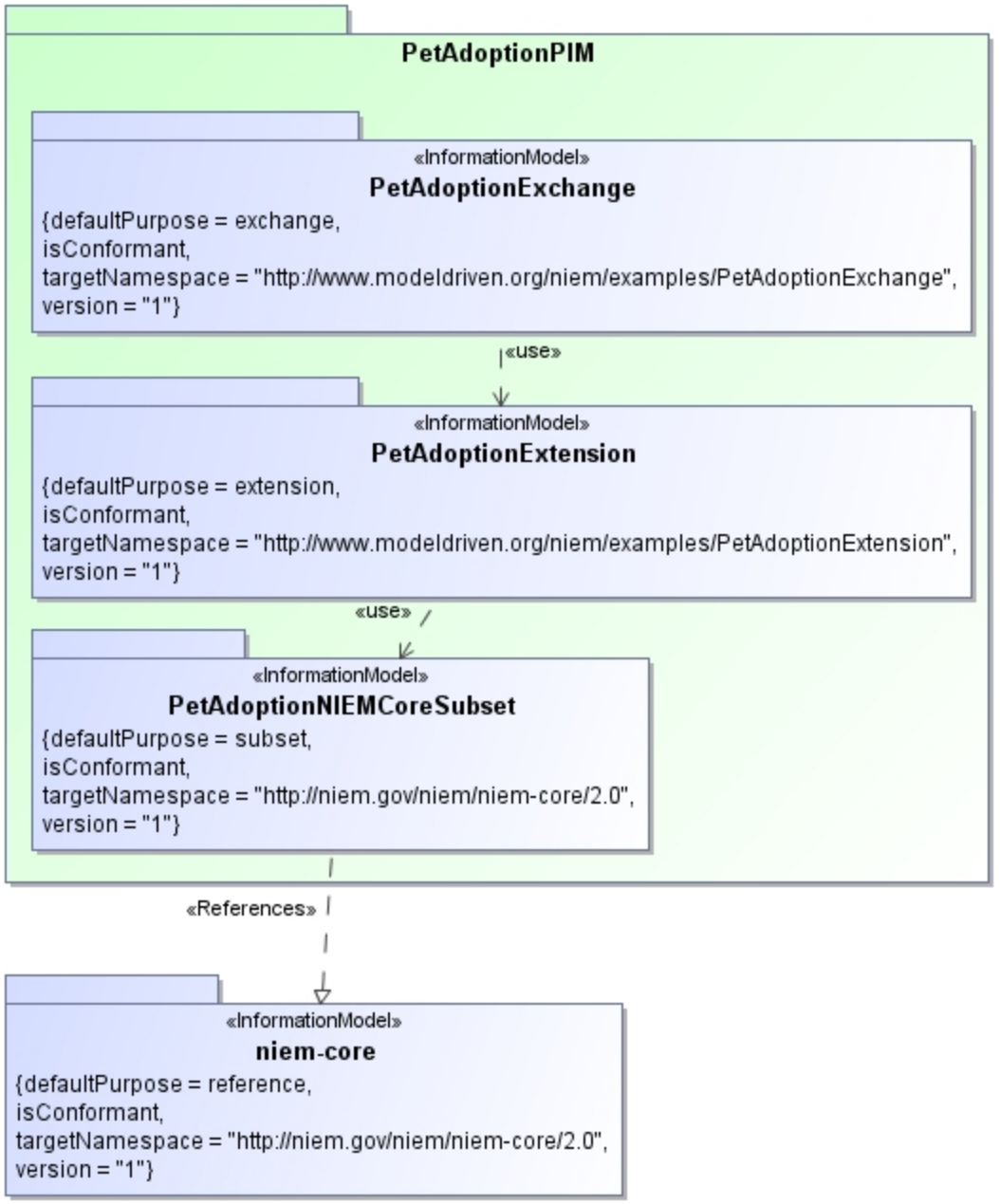




Relations between namespaces (PIM Only)

InformationModel (Package)

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



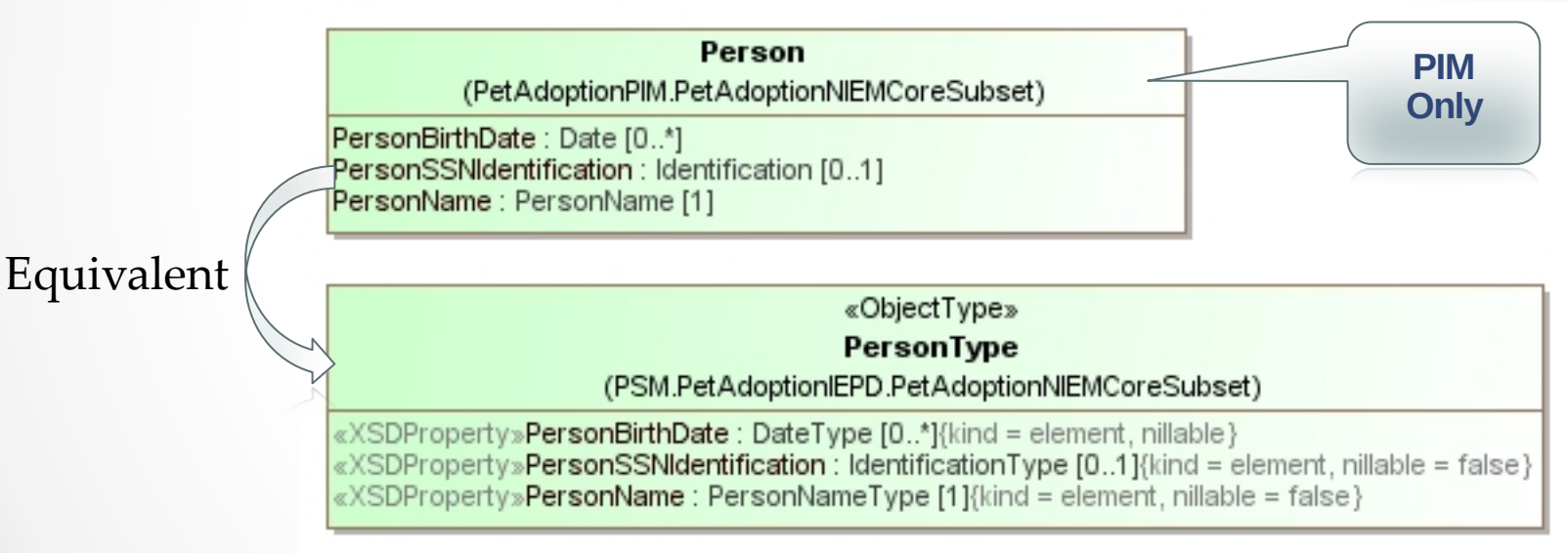
Representation of Complex Types

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.

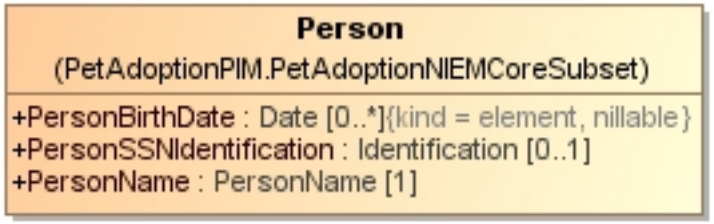
NIEM Object Types

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

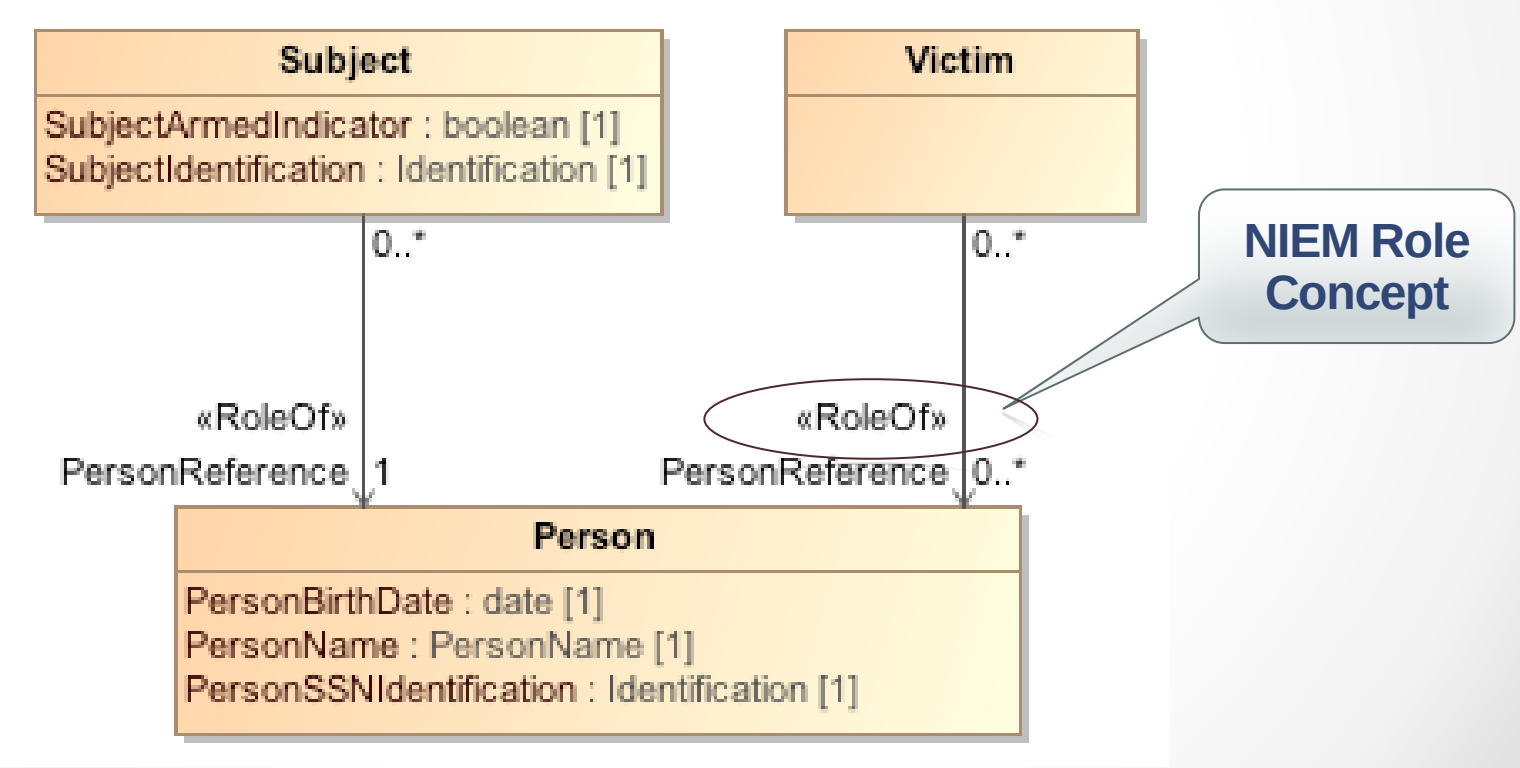
Every element becomes global for reuse (

```
<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

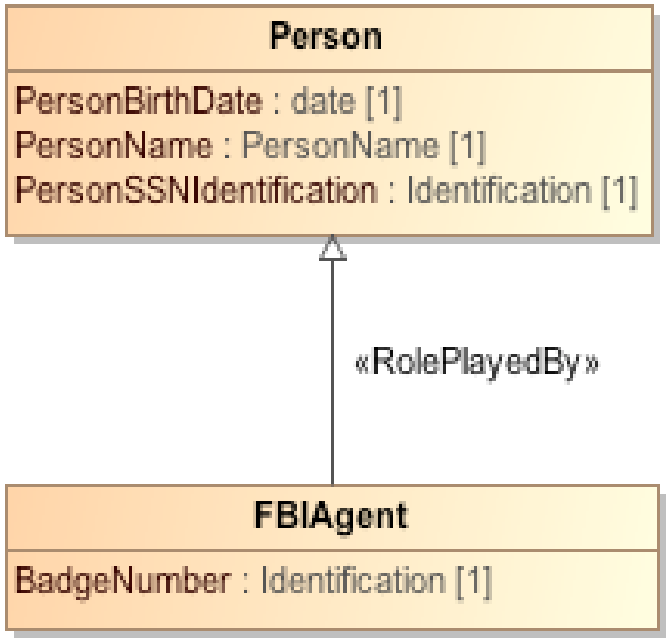
NIEM Roles

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

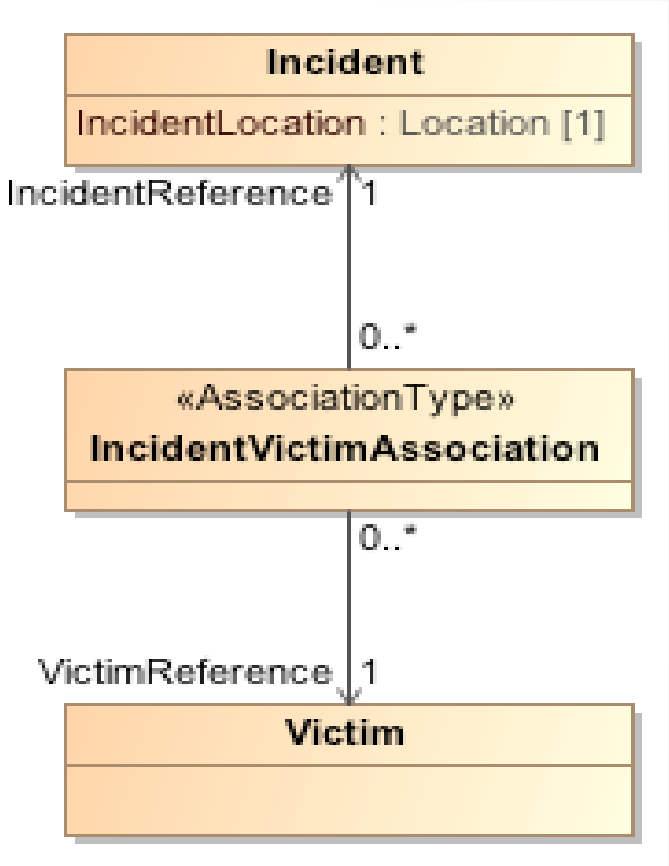
RolePlayedBy
Where a role represents an optional extension to an object type a <<RolePlayedBy>> stereotype of generalization may be used – maps to a NIEM RoleOf property.



NIEM Associations

NIEM Associations

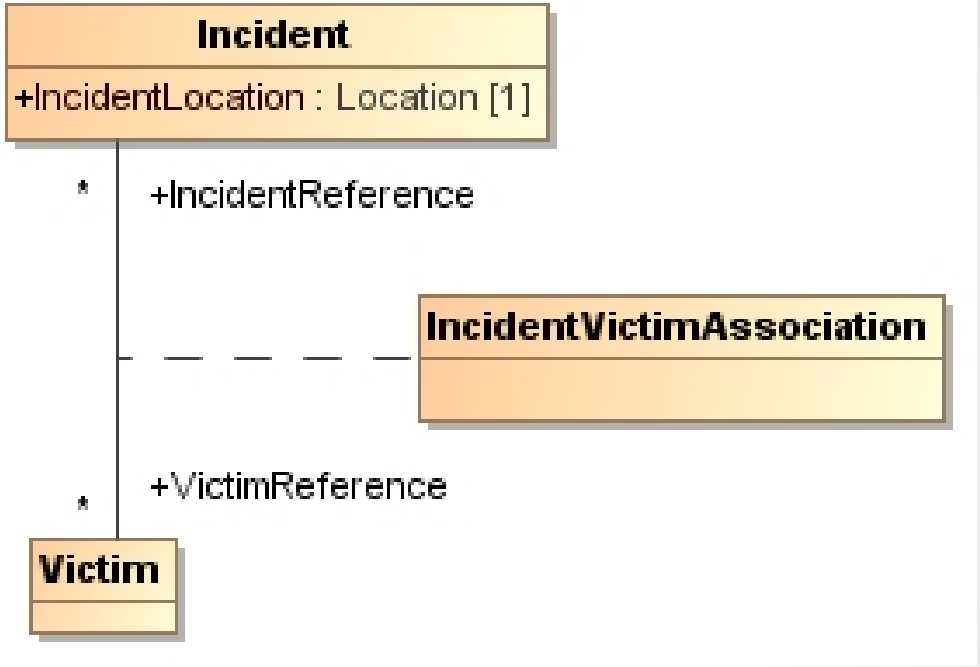
A UML Class stereotyped as an <<AssociationType>> represents a NIEM association using the rules of complex types. Each end of the NIEM association is represented as an independent UML association (an association line in a class diagram). The end is named on the related object side of the UML association 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.



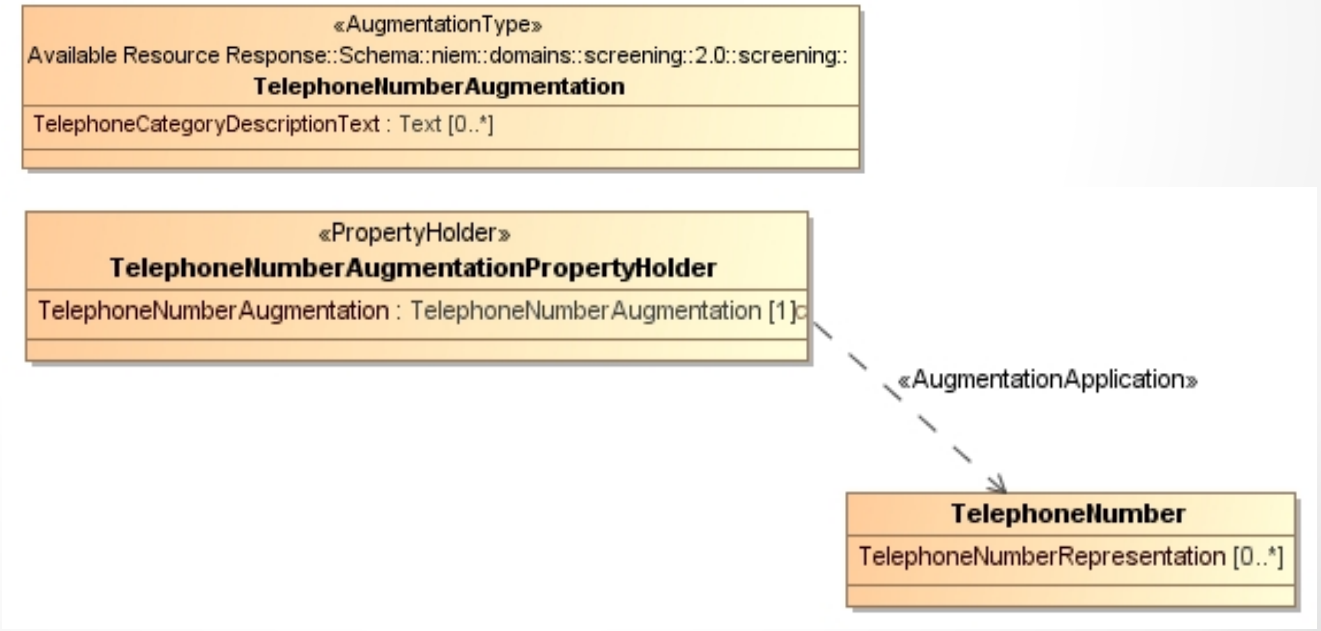
NIEM Metadata

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

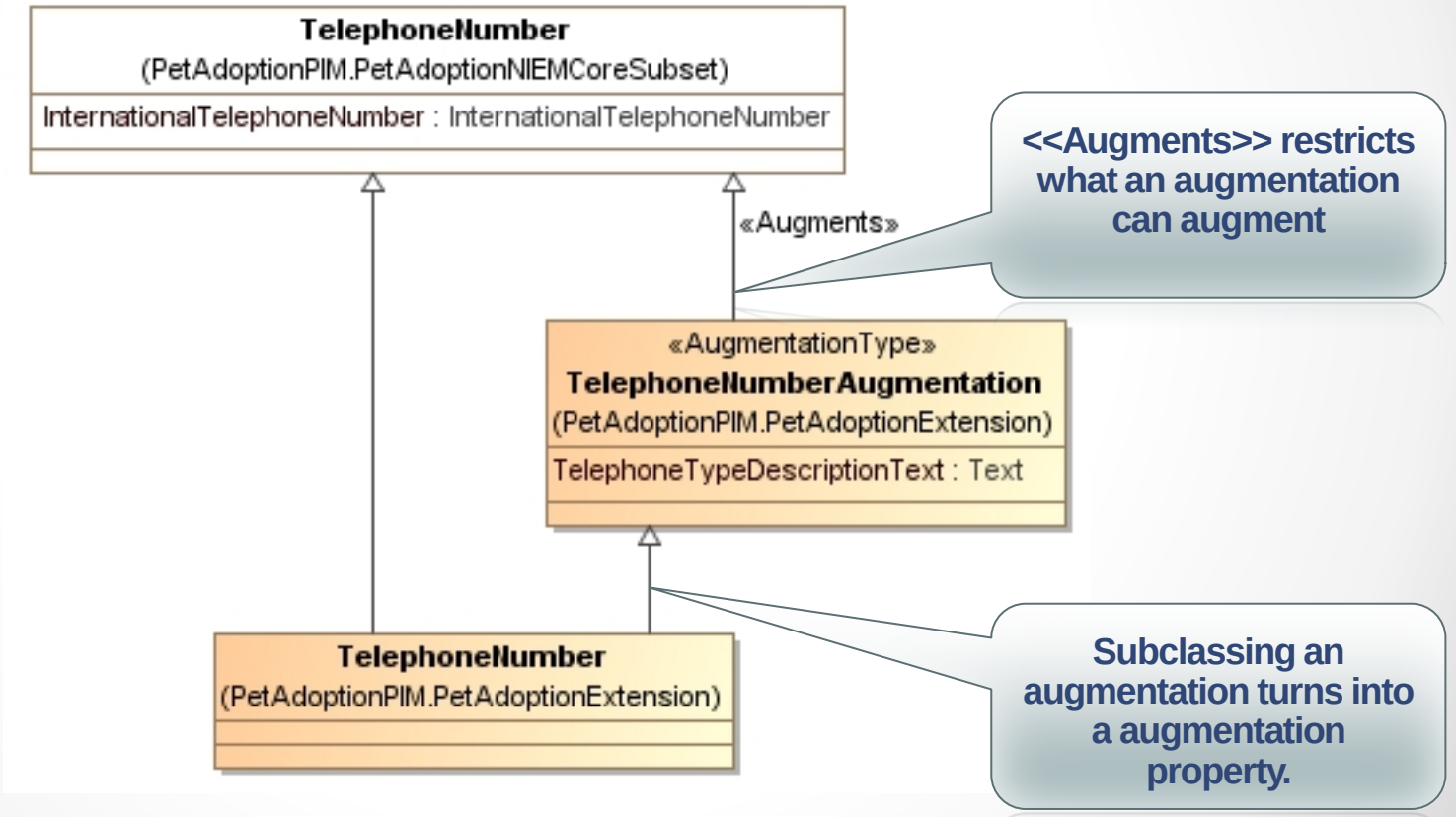
An Augmentation type is represented as a UML class with the <<AugmentationType>> Stereotype. A property typed by an augmentation type may have an <<AugmentationApplication>> dependency which restricts the class of objects that may contain a property typed by an augmentation (this is sometimes called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



NIEM Augmentations

NIEM Augmentations (PIM Only)

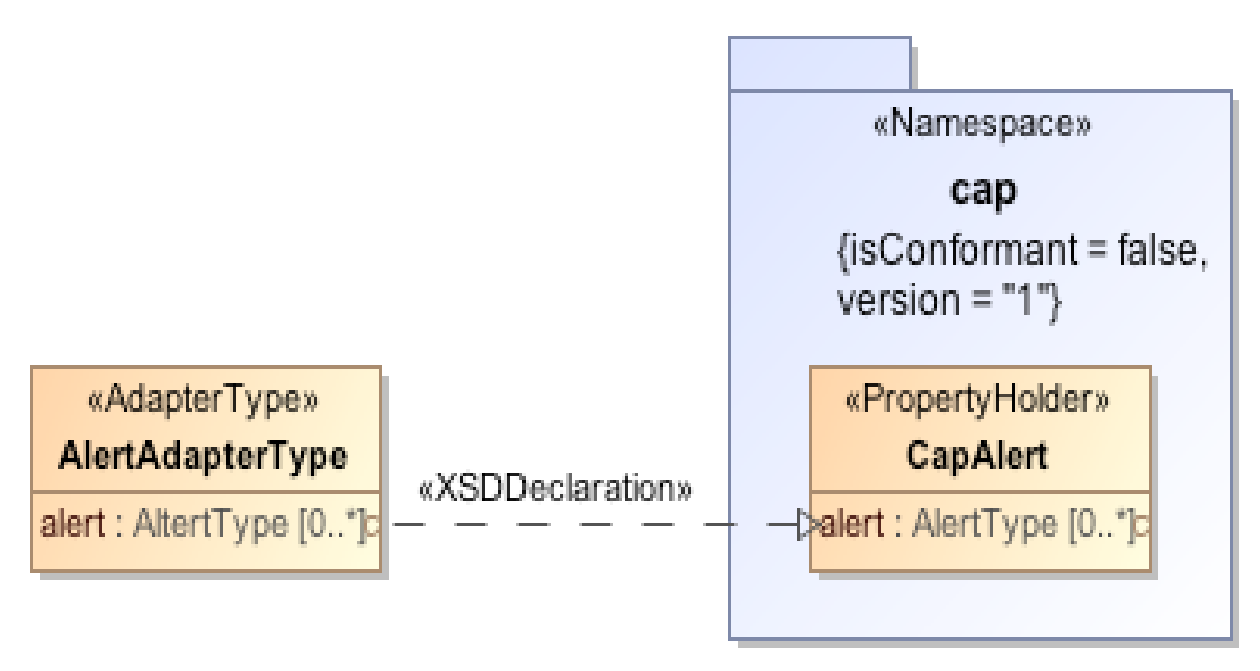
A generalization can be marked to <<Augments>> an <<AugmentationType>>. Inheriting an augmentation creates an augmentation property.



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]

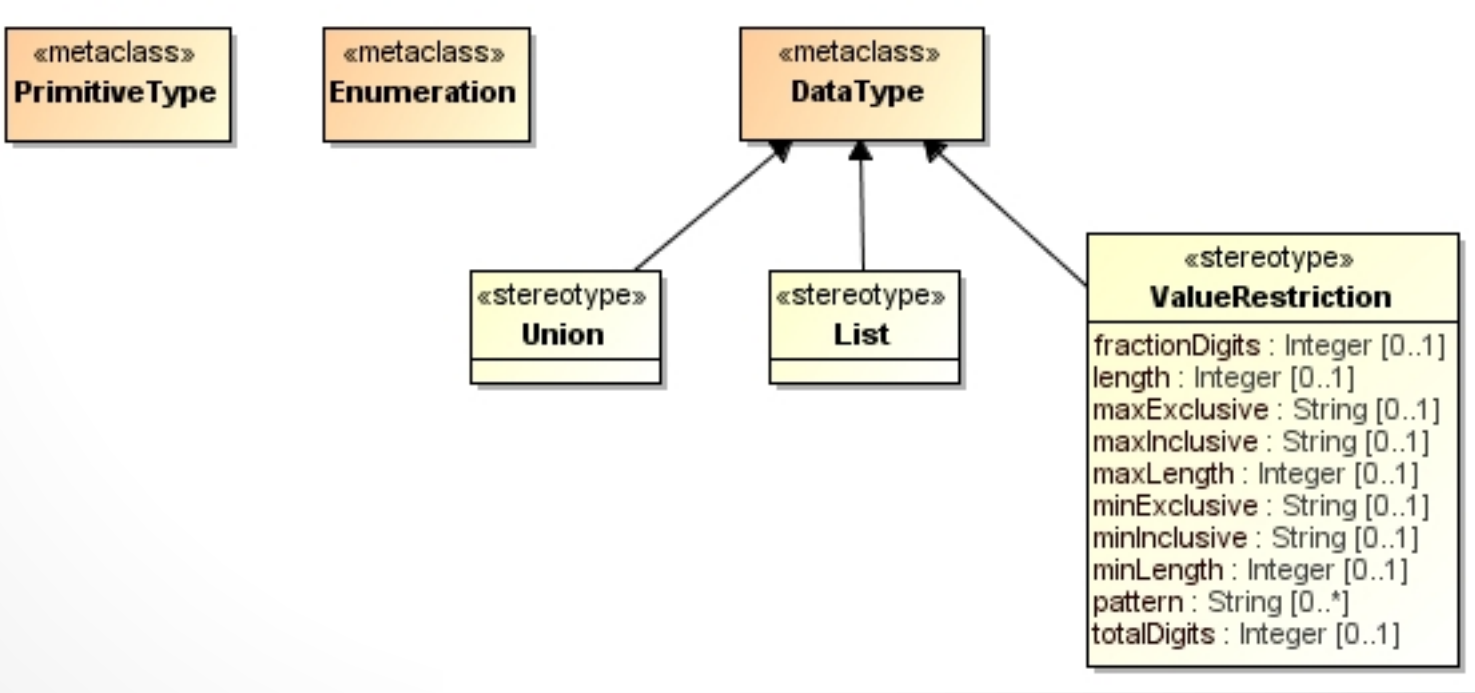


Modeling Simple Types

DataType and Related Elements

DataType (UML)

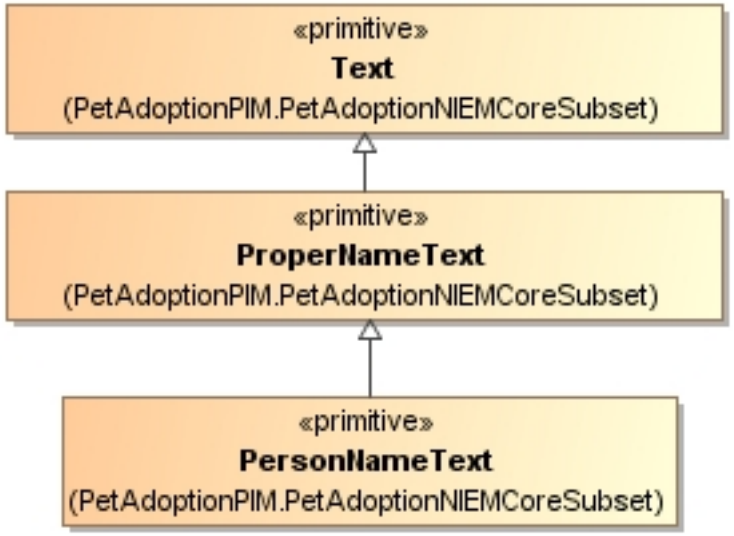
- DataTypes represents a simple type, which is implemented in XML Schema as a simple type definition component.
- PrimitiveType and Enumeration are kinds of DataTypes
- ValueRestriction, Union, and List stereotypes may be applied to DataType
- Realization or Generalization may relate DataTypes



DataType and Related Elements

PrimitiveType (UML)

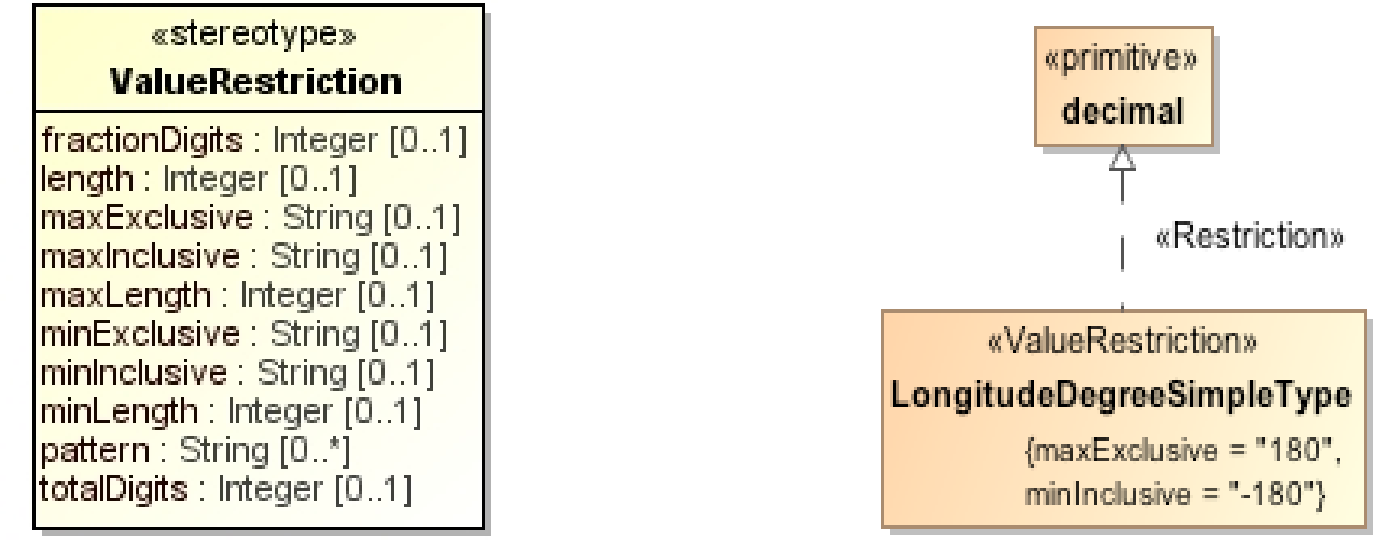
- Represents XML types and subtypes of them.



DataType and Related Elements

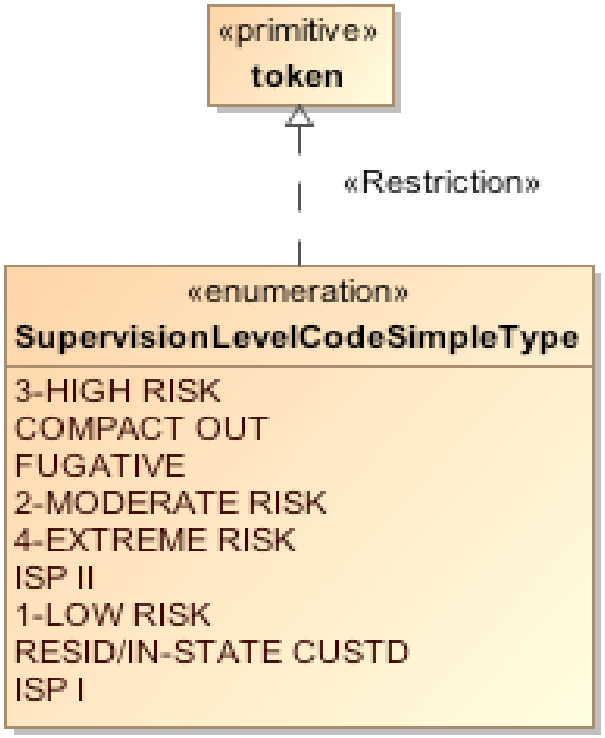
ValueRestriction (DataType)

- 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

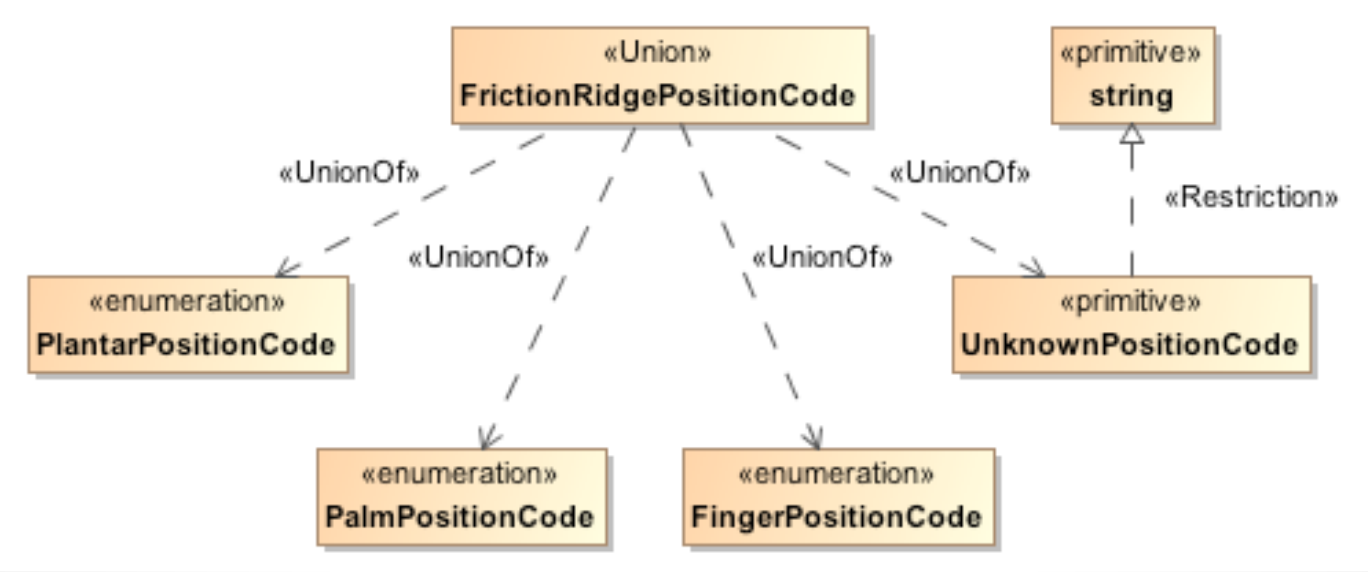
NIEM Code Types
Code types are represented as UML enumerations. Each code value is one value of the enumeration. An enumeration may <<Realize>> another DataType to indicate restriction.



DataType and Related Stereotypes

Union (DataType)

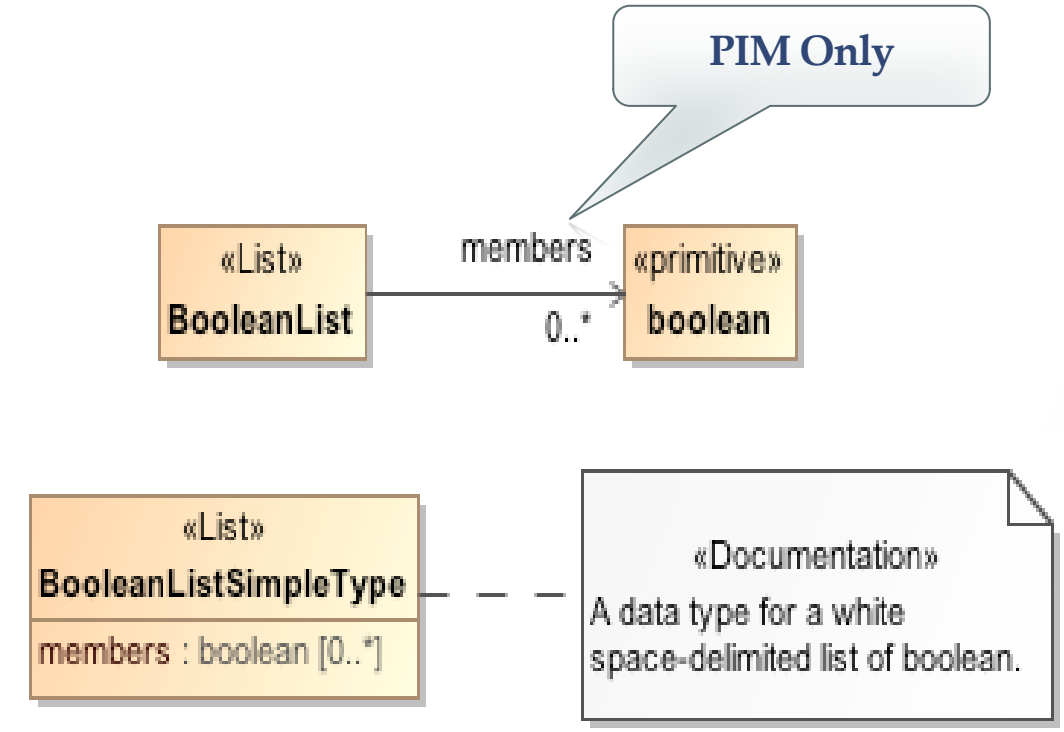
- Union represents a union simple type, represented in XML Schema as a simple type definition component for which the variety property is “union”. A dependency marked as <<UnionOf>> references each member of the union.



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

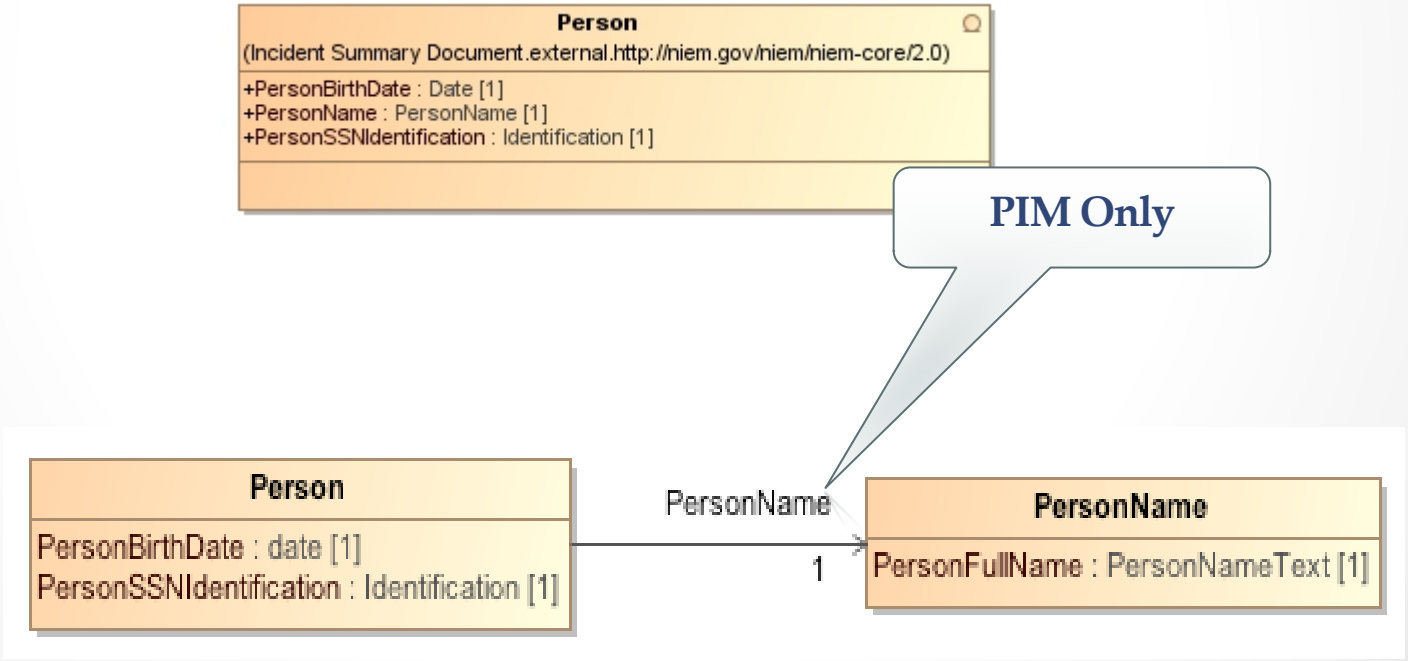


Modeling NIEM Properties

NIEM Properties

NIEM Properties

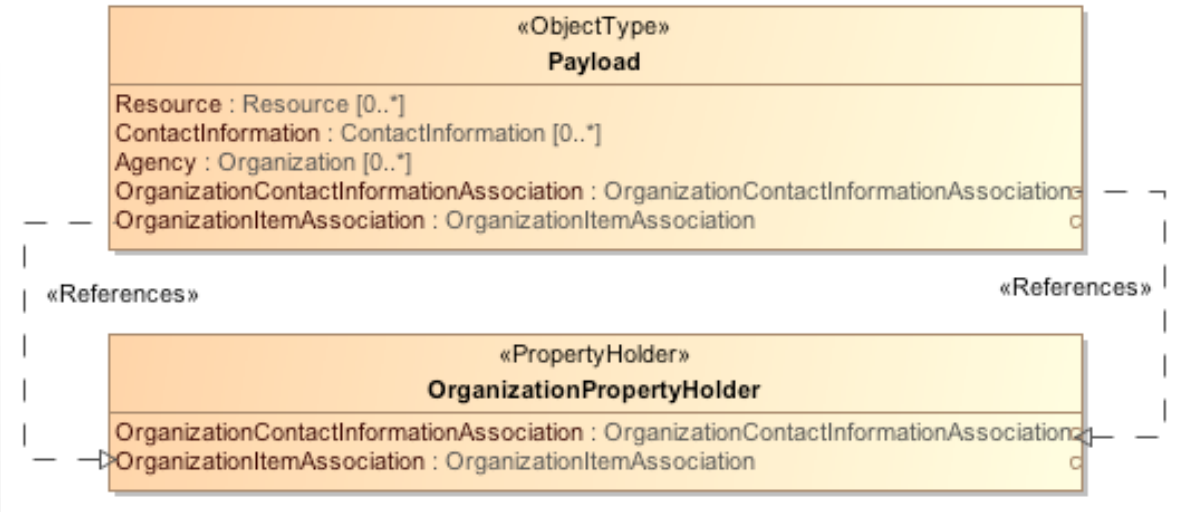
Non-reference properties:
Properties are represented as properties of UML classes or as ends of associations. Information from the UML property or association end definition includes the name, type and cardinality. Associations are only used in a PIM.
.



NIEM Property Reuse

NIEM Property Reuse and Subset Schema

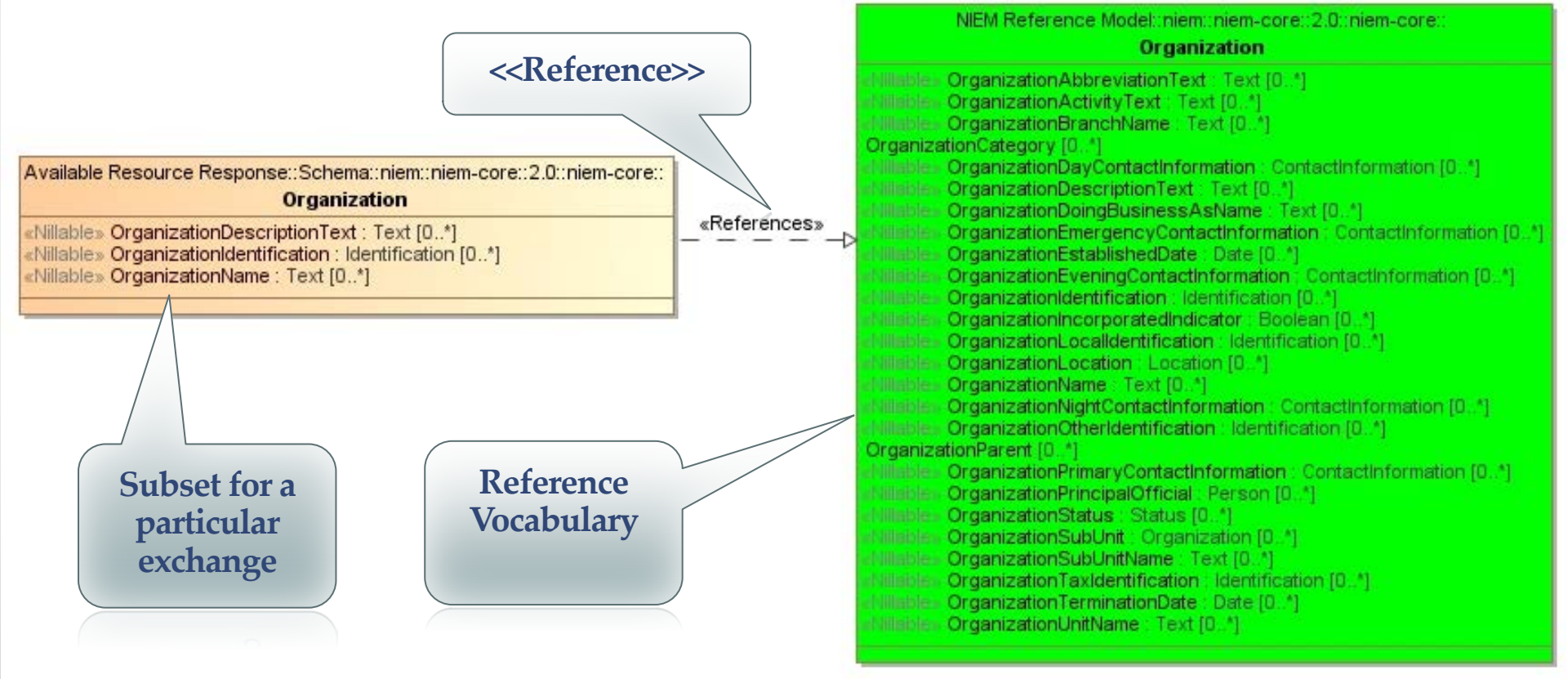
UML has no notion of properties independent of any class and the normal way to handle this in UML is to define classes, perhaps abstract, that are inherited. To be consistent with UML all properties are defined within a class (or data type). The <<References>> stereotype of realization is used to import 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

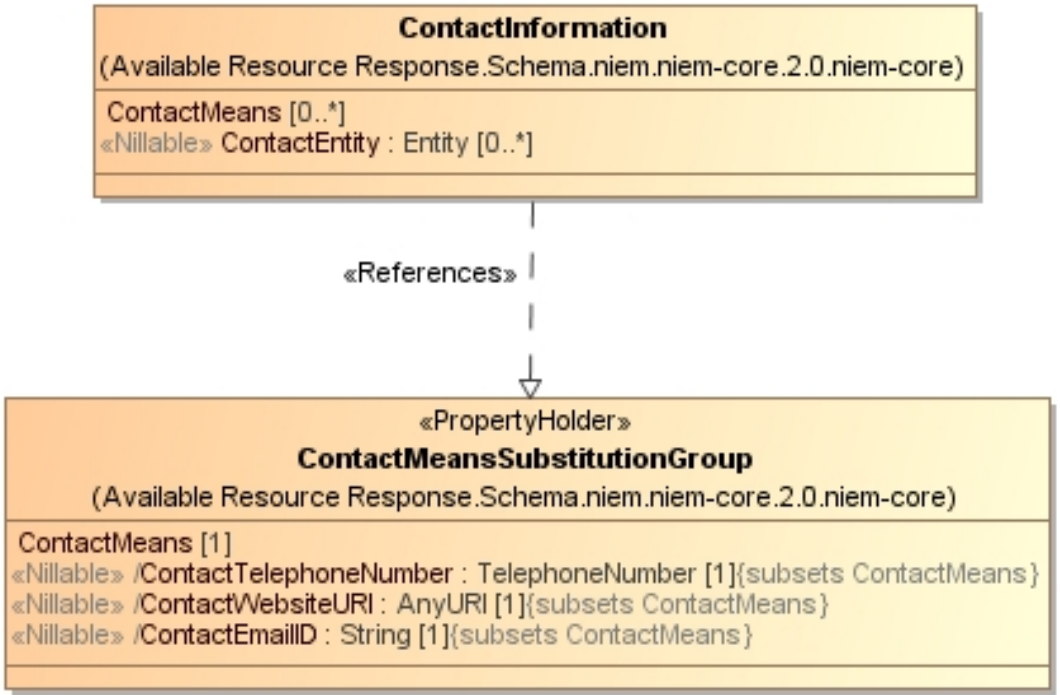
Classes and packages may subset NIEM reference vocabularies.



NIEM Substitution Groups

NIEM Properties

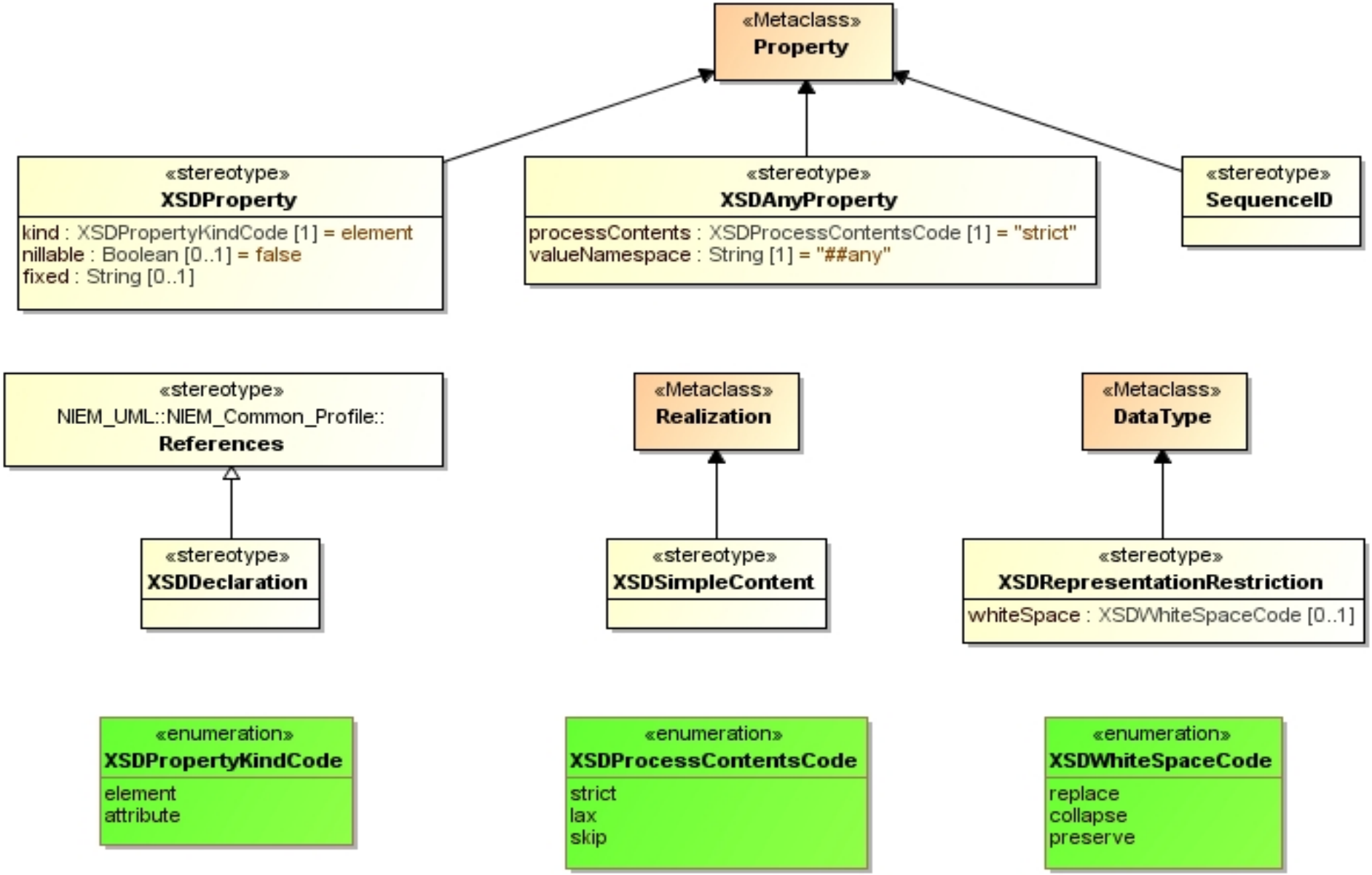
A substitution group is represented by UML property subsetting. A property that subsets another will be substitutable for the base property. All subset properties within a name space are normally grouped together into a single class with the name of the base property combined with the suffix "SubstitutionGroup" (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

• • •

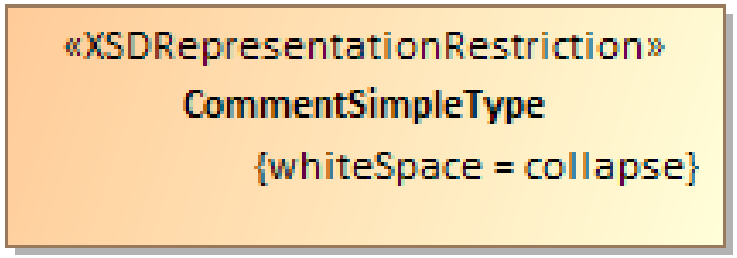
Platform Specific Profile



XSD Representation Restriction (DataType)

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

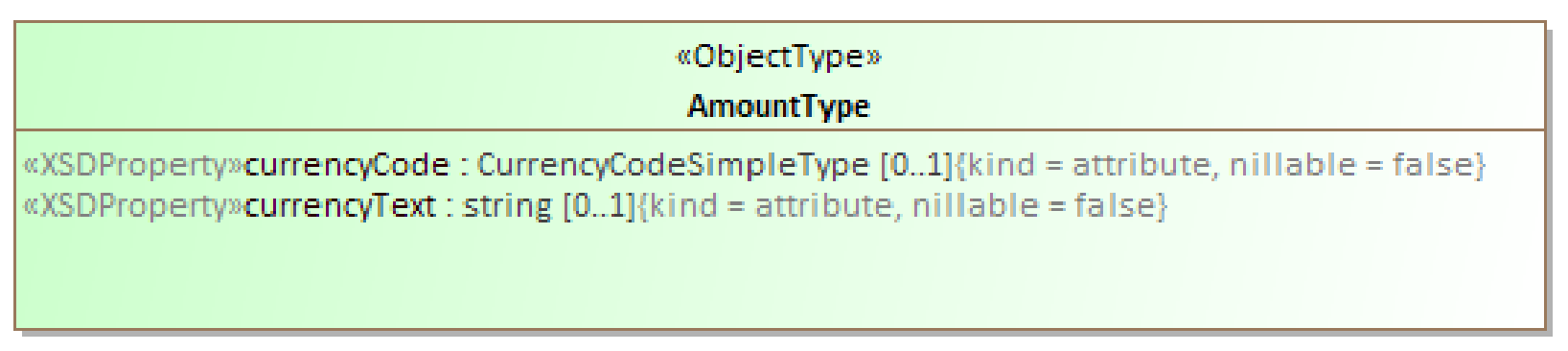
- whiteSpace attribute: value of the whiteSpace component.



XSD Property (Property)

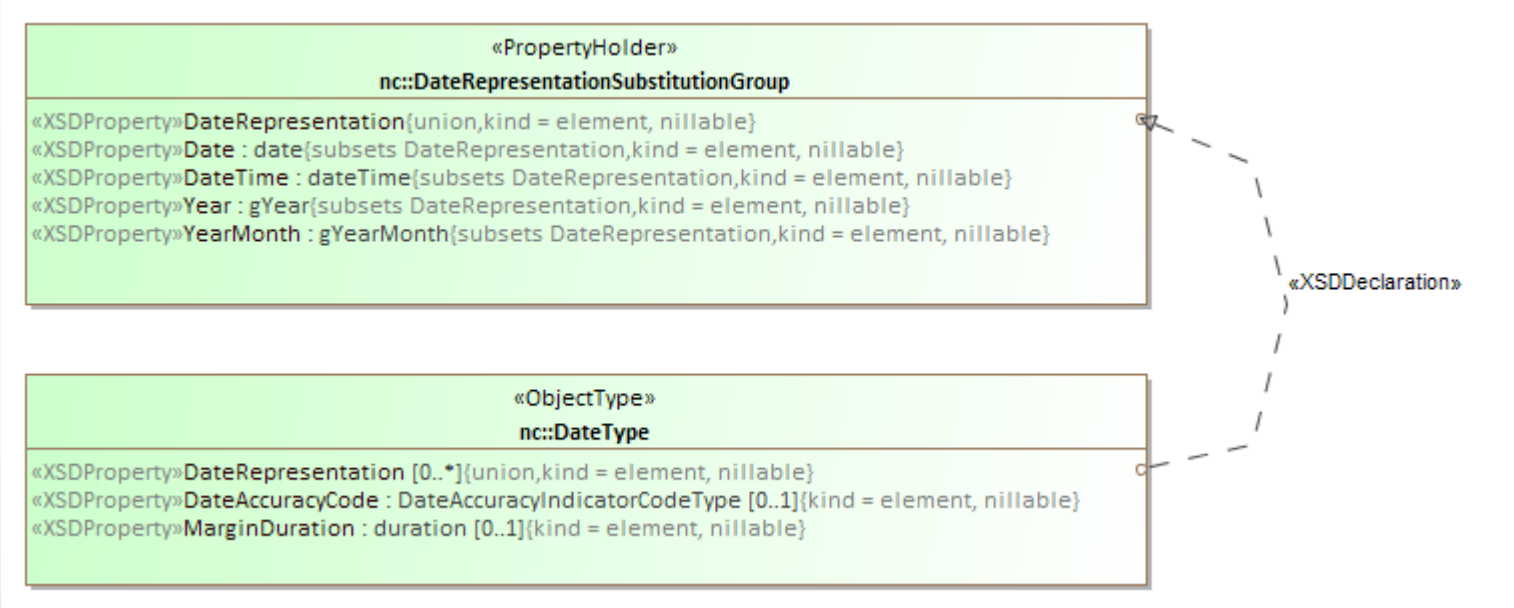
Indicates the implementation of a NIEM property: whether it is an element or attribute declaration, its value constraint property, and its nillable property.

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



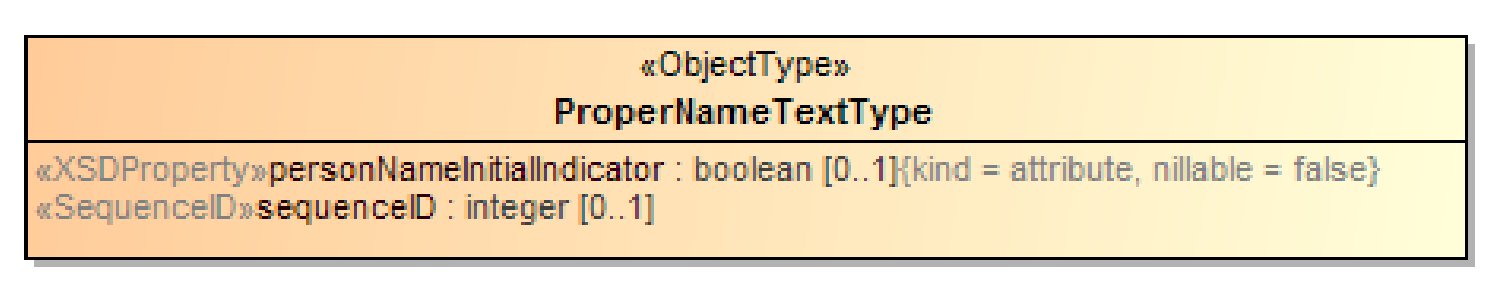
XSD Declaration (Realization)

- 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)

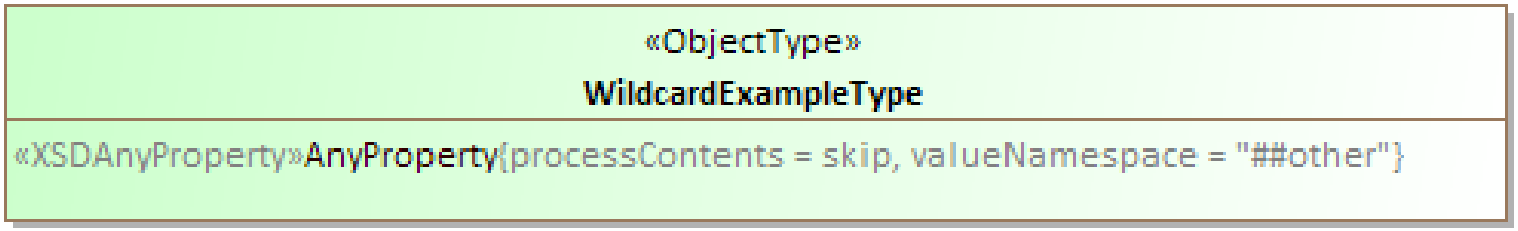
Indicates an attribute use for which the attribute declaration is structures:sequenceID.



XSD AnyProperty (Property)

Indicates a XML Schema wildcard.

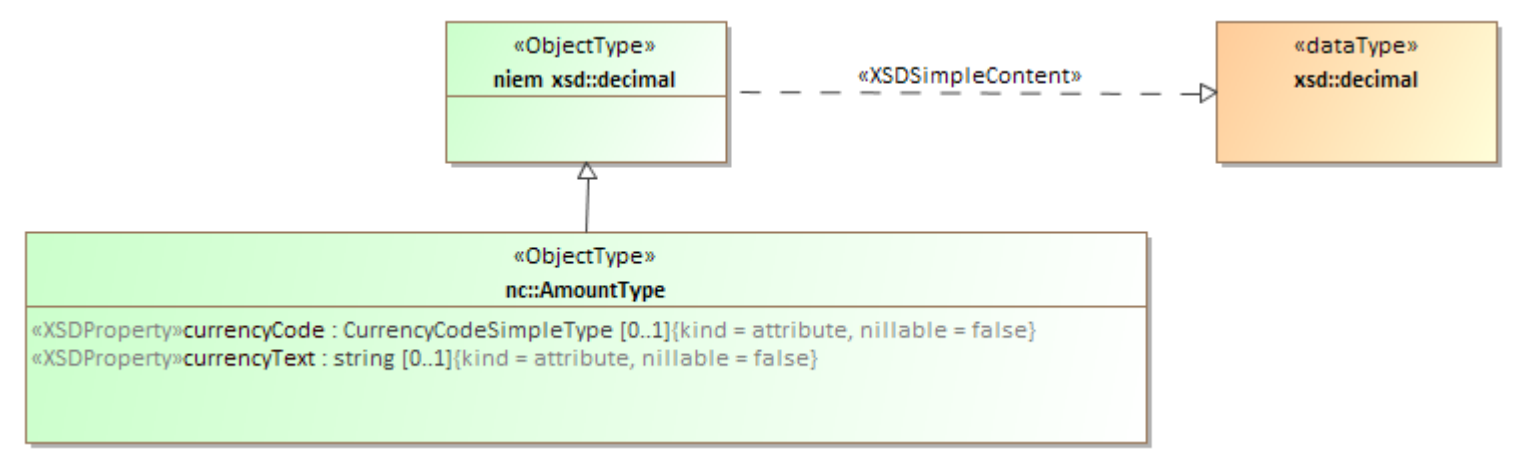
- processContents attribute: the value of the process contents property.
- valueNamespace attribute: the value of the namespace constraint property.



XSD SimpleContent (Realization)

Indicates the content type of a complex type definition.

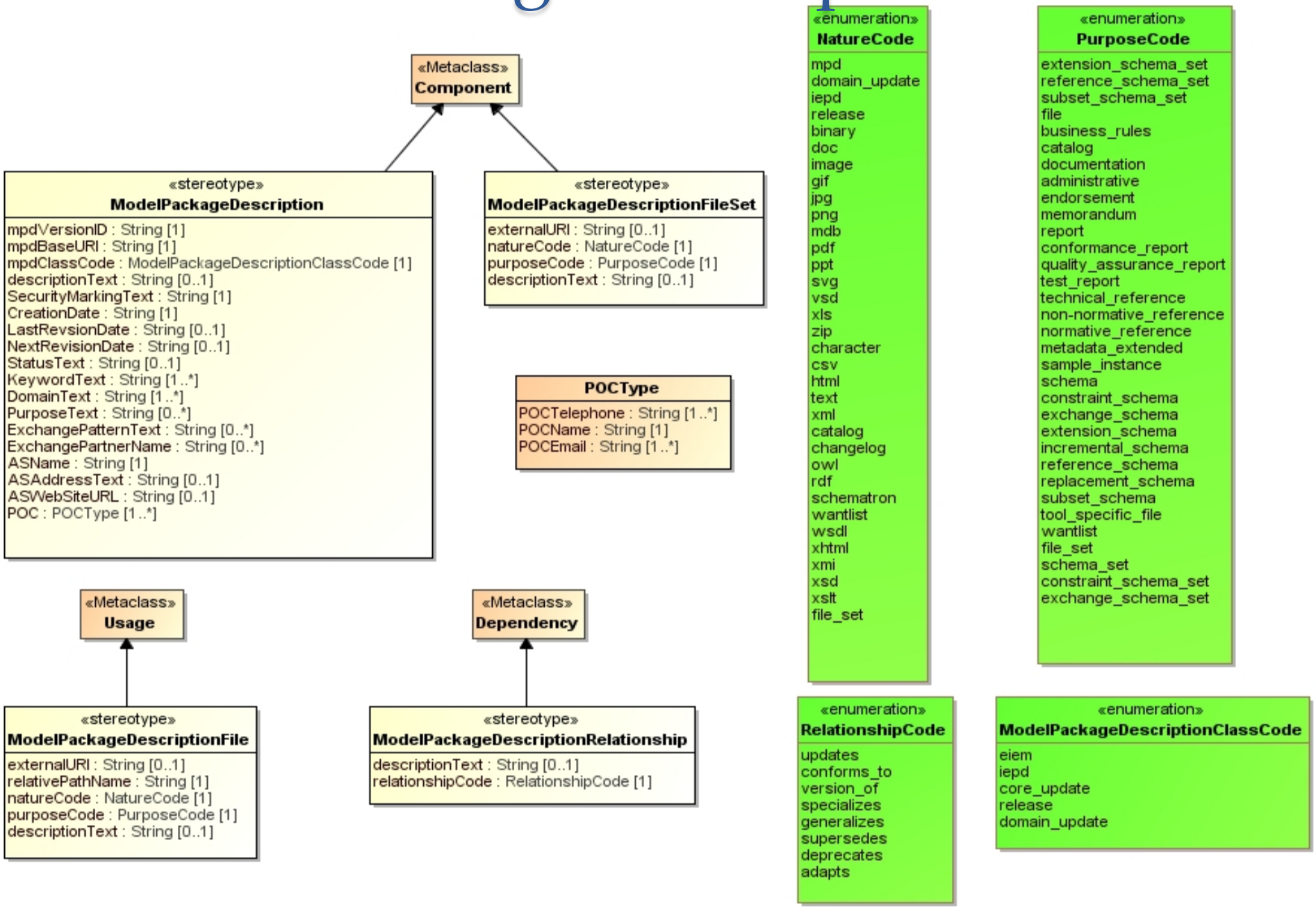
- client: the complex type definition
- supplier: the content type of the complex type definition, a simple type definition



Model Package Description Profile

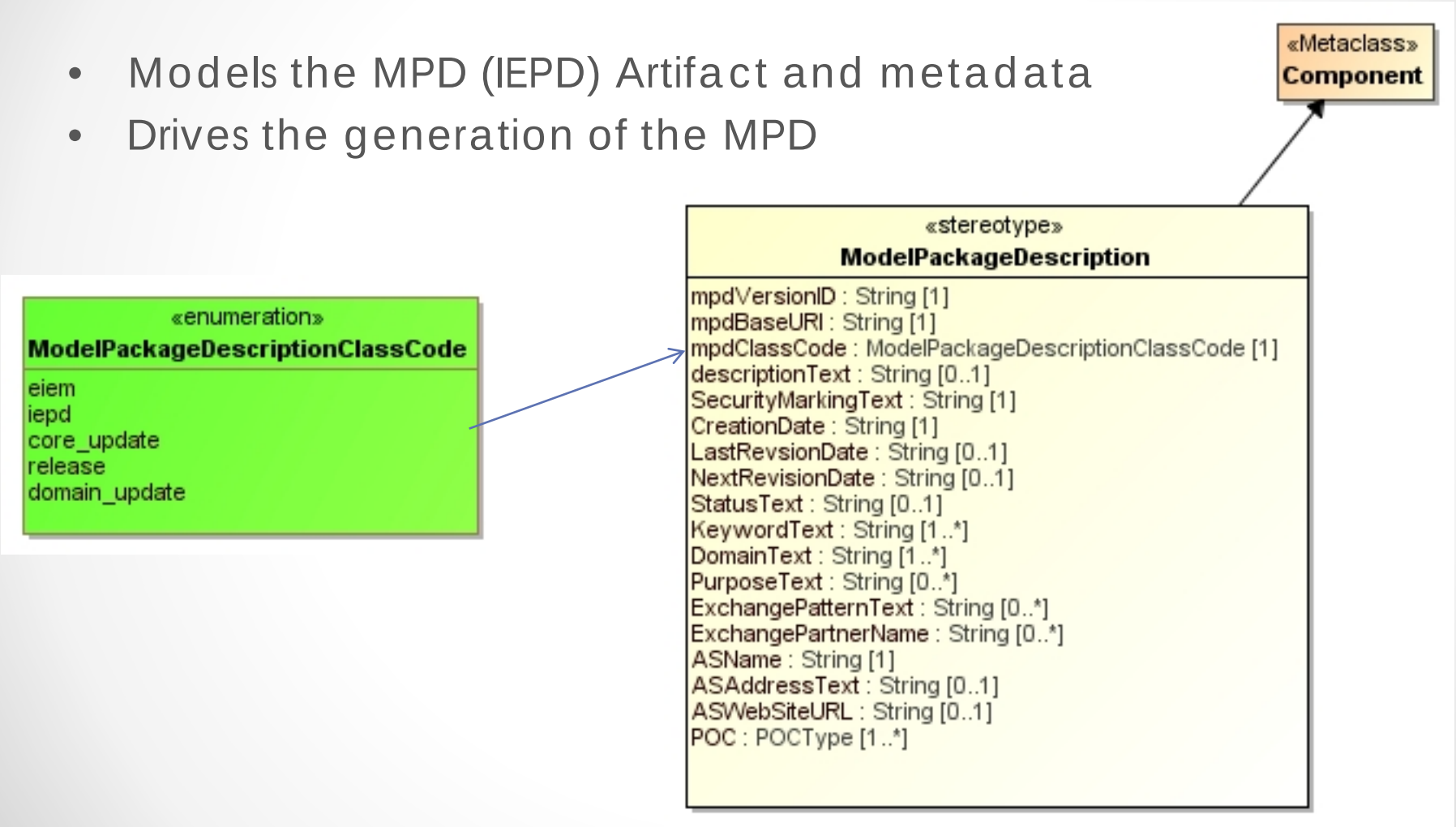
...

Model Package Description Model



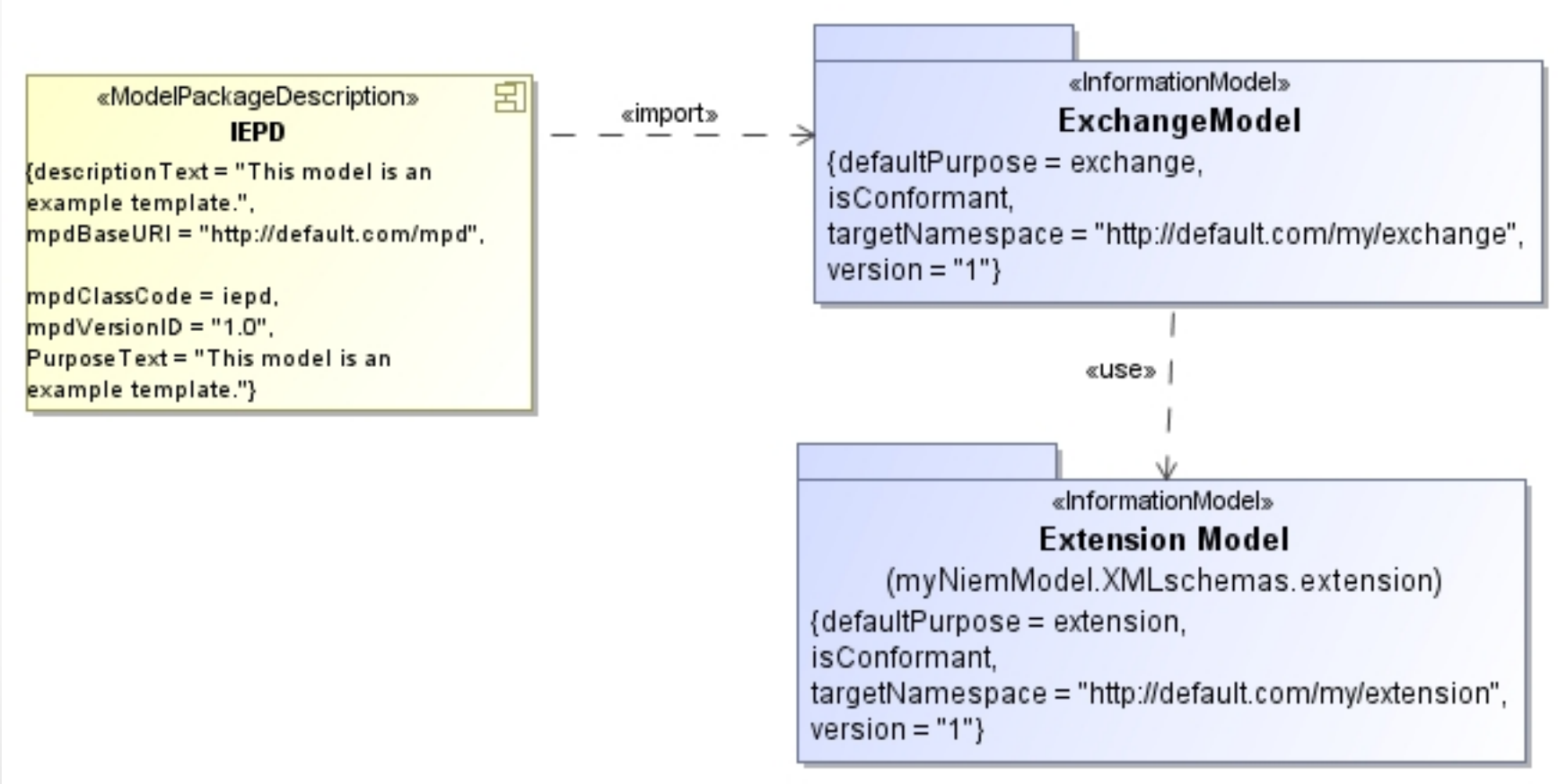
Model Package Description

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



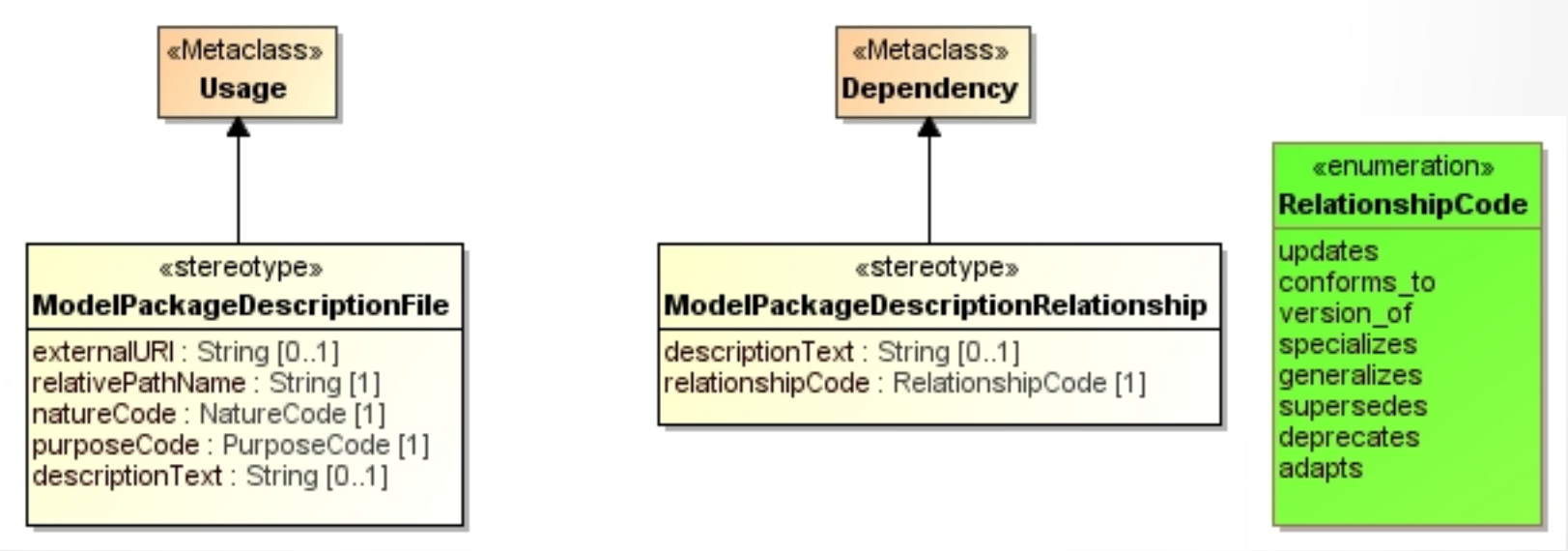
IEPD Example

- 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