Date: May 2015

Information Exchange Packaging Policy Vocabulary (IEPPV)

Version 1.0

OMG Document Number: formal/2015-05-06
Standard document URL: http://www.omg.org/spec/IEPPV
Machine consumable files:
Normative:
http://www.omg.org/spec/IEPPV/20140801/ieppv-umlprofile.xmi
http://www.omg.org/spec/IEPPV/20140801/ieppv.rdf
http://www.omg.org/spec/IEPPV/20140801/ieppv-odmuml.xmi
http://www.omg.org/spec/IEPPV/20140801/ieppv-odm.xmi

USE OF SPECIFICATION - TERMS, CONDITIONS & NOTICES

The material in this document details an Object Management Group specification in accordance with the terms, conditions and notices set forth below. This document does not represent a commitment to implement any portion of this specification in any company's products. The information contained in this document is subject to change without notice.

LICENSES

The companies listed above have granted to the Object Management Group, Inc. (OMG) a nonexclusive, royalty-free, paid up, worldwide license to copy and distribute this document and to modify this document and distribute copies of the modified version. Each of the copyright holders listed above has agreed that no person shall be deemed to have infringed the copyright in the included material of any such copyright holder by reason of having used the specification set forth herein or having conformed any computer software to the specification.

Subject to all of the terms and conditions below, the owners of the copyright in this specification hereby grant you a fully-paid up, non-exclusive, nontransferable, perpetual, worldwide license (without the right to sublicense), to use this specification to create and distribute software and special purpose specifications that are based upon this specification, and to use, copy, and distribute this specification as provided under the Copyright Act; provided that: (1) both the copyright notice identified above and this permission notice appear on any copies of this specification; (2) the use of the specifications is for informational purposes and will not be copied or posted on any network computer or broadcast in any media and will not be otherwise resold or transferred for commercial purposes; and (3) no modifications are made to this specification. This limited permission automatically terminates without notice if you breach any of these terms or conditions. Upon termination, you will destroy immediately any copies of the specifications in your possession or control.

PATENTS

The attention of adopters is directed to the possibility that compliance with or adoption of OMG specifications may require use of an invention covered by patent rights. OMG shall not be responsible for identifying patents for which a license may be required by any OMG specification, or for conducting legal inquiries into the legal validity or scope of those patents that are brought to its attention. OMG specifications are prospective and advisory only. Prospective users are responsible for protecting themselves against liability for infringement of patents.

GENERAL USE RESTRICTIONS

Any unauthorized use of this specification may violate copyright laws, trademark laws, and communications regulations and statutes. This document contains information which is protected by copyright. All Rights Reserved. No part of this work covered by copyright herein may be reproduced or used in any form or by any means--graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems--without permission of the copyright owner.

DISCLAIMER OF WARRANTY

WHILE THIS PUBLICATION IS BELIEVED TO BE ACCURATE, IT IS PROVIDED "AS IS" AND MAY CONTAIN ERRORS OR MISPRINTS. THE OBJECT MANAGEMENT GROUP AND THE COMPANIES LISTED ABOVE MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS PUBLICATION, INCLUDING BUT NOT LIMITED TO ANY WARRANTY OF TITLE OR OWNERSHIP, IMPLIED WARRANTY OF

MERCHANTABILITY OR WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE OR USE.
IN NO EVENT SHALL THE OBJECT MANAGEMENT GROUP OR ANY OF THE COMPANIES LISTED ABOVE BE LIABLE FOR ERRORS CONTAINED HEREIN OR FOR DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL, RELIANCE OR COVER DAMAGES, INCLUDING LOSS OF PROFITS, REVENUE, DATA OR USE, INCURRED BY ANY USER OR ANY THIRD PARTY IN CONNECTION WITH THE FURNISHING, PERFORMANCE, OR USE OF THIS MATERIAL, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

The entire risk as to the quality and performance of software developed using this specification is borne by you. This disclaimer of warranty constitutes an essential part of the license granted to you to use this specification.

RESTRICTED RIGHTS LEGEND

Use, duplication or disclosure by the U.S. Government is subject to the restrictions set forth in subparagraph (c) (1) (ii) of The Rights in Technical Data and Computer Software Clause at DFARS 252.227-7013 or in subparagraph (c)(1) and (2) of the Commercial Computer Software - Restricted Rights clauses at 48 C.F.R. 52.227-19 or as specified in 48 C.F.R. 227-7202-2 of the DoD F.A.R. Supplement and its successors, or as specified in 48 C.F.R. 12.212 of the Federal Acquisition Regulations and its successors, as applicable. The specification copyright owners are as indicated above and may be contacted through the Object Management Group, 109 Highland Avenue, Needham, MA 02494, U.S.A.

TRADEMARKS

IMM®, MDA®, Model Driven Architecture®, UML®, UML Cube logo®, OMG Logo®, CORBA® and XMI® are registered trademarks of the Object Management Group, Inc., and Object Management Group™, OMG™, Unified Modeling Language™, Model Driven Architecture Logo™, Model Driven Architecture Diagram™, CORBA logos™, XMI Logo™, CWM™, CWM Logo™, IIOP™, MOF™, OMG Interface Definition Language (IDL)™, and OMG SysML™ are trademarks of the Object Management Group. All other products or company names mentioned are used for identification purposes only, and may be trademarks of their respective owners.

COMPLIANCE

The copyright holders listed above acknowledge that the Object Management Group (acting itself or through its designees) is and shall at all times be the sole entity that may authorize developers, suppliers and sellers of computer software to use certification marks, trademarks or other special designations to indicate compliance with these materials.

Software developed under the terms of this license may claim compliance or conformance with this specification if and only if the software compliance is of a nature fully matching the applicable compliance points as stated in the specification. Software developed only partially matching the applicable compliance points may claim only that the software was based on this specification, but may not claim compliance or conformance with this specification. In the event that testing suites are implemented or approved by Object Management Group, Inc., software developed using this specification may claim compliance or conformance with the specification only if the software satisfactorily completes the testing suites.

OMG’S ISSUE REPORTING PROCEDURE

All OMG specifications are subject to continuous review and improvement. As part of this process we encourage readers to report any ambiguities, inconsistencies, or inaccuracies they may find by completing the Issue Reporting Form listed on the main web page *http://www.omg.org*, under Documents, Report a Bug/Issue (*http://www.omg.org/report_issue.htm*).

Table of Contents

Preface

About the Object Management Group

OMG

Founded in 1989, the Object Management Group, Inc. (OMG) is an open membership, not-for-profit computer industry standards consortium that produces and maintains computer industry specifications for interoperable, portable, and reusable enterprise applications in distributed, heterogeneous environments. Membership includes Information Technology vendors, end users, government agencies, and academia.

OMG member companies write, adopt, and maintain its specifications following a mature, open process. OMG's specifications implement the Model Driven Architecture® (MDA®), maximizing ROI through a full-lifecycle approach to enterprise integration that covers multiple operating systems, programming languages, middleware and networking infrastructures, and software development environments. OMG's specifications include: UML® (Unified Modeling Language™); CORBA® (Common Object Request Broker Architecture); CWM™ (Common Warehouse Metamodel); and industry-specific standards for dozens of vertical markets.

More information on the OMG is available at *http://www.omg.org/*.

OMG Specifications

As noted, OMG specifications address middleware, modeling and vertical domain frameworks. All OMG specifications are available from the OMG website at:

http://www.omg.org/spec

Specifications are organized by the following categories:

Business Modeling Specifications

Middleware Specifications

- **CORBA/IIOP**
- **Data Distribution Services**
- **Specialized CORBA**

IDL/Language Mapping Specifications

Modeling and Metadata Specifications

- **UML, MOF, CWM, XMI**
- **UML Profile**

Modernization Specifications

Platform Independent Model (PIM), Platform Specific Model (PSM), Interface Specifications

- **CORBAServices**
- **CORBAFacilities**

OMG Domain Specifications

CORBA Embedded Intelligence Specifications

CORBA Security Specifications

Signal and Image Processing Specifications

All of OMG's formal specifications may be downloaded without charge from our website. (Products implementing OMG specifications are available from individual suppliers.) Copies of specifications, available in PostScript and PDF format, may be obtained from the link cited above or by contacting the Object Management Group, Inc. at:

OMG Headquarters
109 Highland Avenue
Needham, MA 02494
USA
Tel: +1-781-444-0404
Fax: +1-781-444-0320
Email: *pubs@omg.org*

Certain OMG specifications are also available as ISO standards. Please consult *http://www.iso.org*

Typographical Conventions

The type styles shown below are used in this document to distinguish programming statements from ordinary English. However, these conventions are not used in tables or section headings where no distinction is necessary.

Times/Times New Roman - 10 pt.: Standard body text

Helvetica/Arial - 10 pt. Bold: OMG Interface Definition Language (OMG IDL) and syntax elements.

`Courier - 10 pt. Bold:` Programming language elements.

Helvetica/Arial - 10 pt: Exceptions

Note – Terms that appear in *italics* are defined in the glossary. Italic text also represents the name of a document, specification, or other publication.

Issues

The reader is encouraged to report any technical or editing issues/problems with this specification to *http://www.omg.org/report_issue.htm*.

1 Scope

This specification provides an Information Exchange Packaging Policy Vocabulary (IEPPV). The vocabulary is intended to improve the accuracy, fidelity, clarity, and consistency in the specification and design of data assembly (e.g., aggregation, transformation, filtering/redaction, and tagging) and processing (e.g., parsing, validating, transformation, and marshaling) rules corresponding to information sharing and safeguarding (ISS) policies. The IEPPV provides the precision to enable modeling and policy development tools to automate the transformation of interface specifications and designs into the one or more machine readable and enforceable policy or rules languages.

Information sharing and safeguarding encompasses a broad policy environment that includes, but is not limited to access management and control, identity management, credential/attribute management, tagging/marking, authorization, Key Management, Encryption and logging/auditing. The IEPPV specifically addresses policies governing the assembly and processing of data and information elements to ensure that interfaces conform to information sharing agreements and the policies for protection of sensitive (e.g., private, confidential and classified, and legally significant) information.

This specification also provides two implementations of vocabulary:

- UML Profile that enables the modeling of Information Packaging Specifications that aligns to other architecture models; and
- Web Ontology Language (OWL) that will enable users to analyze rules resulting from the serialization of the UML Model.

1.1 Organization of this Specification

This specification includes seven Clauses and seven Annexes:

- Clause 1: Provides an overview of the specification, including: *Scope*; *Objectives*; *Within the Context of the Information Exchange Framework*; *Problem Statement*; *IEF History and Pedigree*; *Support for Trusted Semantic Interoperability*; *and IEF Concept.*
- Clause 2: Defines the compliance points for the IEPPV: Compliance Point 1: Information Payload Specification; Compliance Point 2a: Basic Message Specification; Compliance Point 2b: Full Message Specification; Compliance Point 2c: Information Specification; and Compliance Point 3: Information Exchange Specification.
- Clause 3: Identifies *Normative References* for this specification.
- Clause 4: Identifies *Terms and their Definitions* used in various parts of the specification. This Clause does not include concepts and properties comprising the IEPPV.
- Clause 5: Identifies any special *Symbols/Acronyms* used in the development of this specification.
- Clause 6: Provides *Additional Information* about this specification.
- Clause 7: Identifies and defines the classes (concepts), properties, and restrictions underpinning the Information Exchange Policy Vocabulary.
- Annex A: (Normative): Provides the Taxonomy of the IEPPV.
- Annex B: (Normative): Described the UML Profile for the IEPPV.
- Annex C (informational): *Domain Model* for the storage of Information Exchange Policies defined using the IEPPV.
- Annex D: (Informational): Provides a *UML Example Model* using the UML Profile.

- Annex E: (Informational): Provides the Bibliography for this specification.
- Annex F: (Informational): Glossary of Terms and Acronyms.
- Annex G: Describes how the specification elements address the RFP requirements.

1.2 Objectives

The primary objective for this IEPPV specification is the provision of a vocabulary that will provide consistent and tools agnostic concepts for the expression of rules governing:

1. **Information Packaging**: The assembling (aggregating, transforming, tagging/marking and redacting/filtering) of data and information elements and formatting them for a specific information exchange requirement.
2. **Information Processing**: The parsing, validation, transformation and marshaling of information and data elements to information or data store(s).

The UML Profile provided as Annex B, will support the development of modeling tools that will:

1. Align Information Packaging and Processing to other architectural aspects of an information exchange specification (e.g., interface, System, communications/networks, security, operations, mission/operation).
2. Improve the traceability of information interoperability from policy instruments (e.g., legislation, regulation, policy and service level agreements).
3. Provide an architecture-driven approach for the specification and design of information sharing agreements.
4. Provide Model Driven Architecture support for the serialization of packaging and processing models into a machine executable form.
5. Provide the ability to model information sharing and safeguarding.
6. Reduce life-cycle and training costs through the reuse, repurposing and sharing of data patterns.
7. Improve retention and reuse of corporate information and knowledge through architecture and modeling
8. Improve the communication between, and understanding of, stakeholders.

The proposed integration of IEPPV into the Unified Profile for DODAF and MODAF (UPDM) version 3 will tie information packaging to related elements in broader system, operational, enterprise architectures (e.g., data, interface, system, platform, capability, program and organization definitions). The proposed integration will have IEPPV replace Shared Operational Picture Exchange Services Profile (SOPES) profile integrated into UPDM V2.1 (http://www.omg.org/specs/UPDM).

1.3 Information Exchange Framework

The IEPPV is being developed under the umbrella of the Information Exchange Framework (IEF), which is an OMG initiative to develop a series of specifications for policy vocabularies and enabling-services (decision and enforcements points) for the automation of information sharing and safeguarding policies. The IEPPV is specifically targeting the packaging (assembly and formatting) and processing of data and information elements exchanged between information systems.

The IEPPV is the first in a family of information exchange vocabulary specifications (Figure 1.1) that will enable the specification of information packaging rules deriving from business, operational, and security policy; enable users to align policy instruments to the target information domains; and enable the automation of key information sharing and safeguarding tasks. The IEF is also seeking policy-driven capability in the areas of (Figure 1.1):

- Identity Management
- Credentials Management
- Access Management
- Dissemination and QoS
- User Defined Services
- Auditing Services

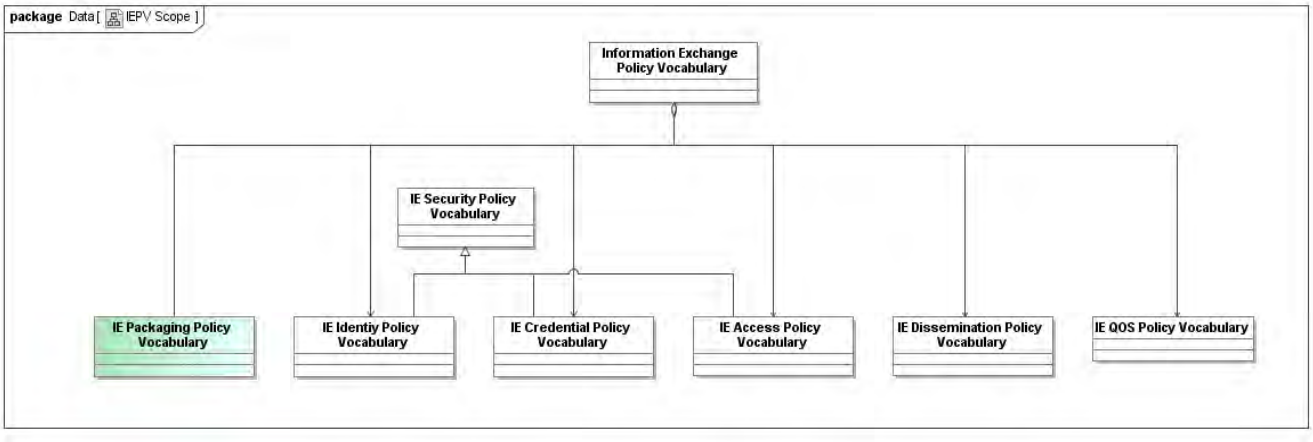

Figure 1.1 - IEF Policy Domain

Where policy languages exist (e.g., SAML and XACML) the IEF would be seeking language implementation of the policy vocabularies in the existing standards. The IEF is also seeking modeling language implementations of the vocabularies (e.g., the UML profiles provided in Annex B), to align the transformation of policy to an executable form to other architecture and engineering models. These modeling language implementations would be an intermediate stage in the serialization of policies or rules for incorporation into services that automate information sharing and safeguarding activities. The IEF will also be seeking to adopt and integrate existing standards in any of the policy and service areas (e.g., DDS Dissemination and QOS policies).

This initial specification addresses the expression of rules as derived from user defined policies for the assembly of information content in a secure and trusted manner. The IEPPV limits its scope to the packaging (i.e., assembly (e.g., aggregation, transformation, tagging/marking, and redaction) and formatting) and the inverse processing (parsing, transformation, and marshaling) of information shared between information systems. The IEPPV strays from its principle focus to provide the ability to specify the information dissemination services to be used for the exchange of the resulting information element or message. These additional elements were included to address Mandatory Requirements in the RFP (Mandatory Requirement (6) in Annex G).

The IEPPV specifically addressed the requirements of the Information Exchange Policy Vocabulary (IEPV) Request for Proposal (RFP): mars/2011-03-15.

1.4 Information Sharing Policy Development

Semantic interoperability is defined as the ability for information systems to exchange information in such a manner that the meaning and intent is properly and consistently interpreted by the receiving system; in other words, the interpretation of a receiving system must be the same interpretation as intended by the sending system. Semantic interoperability requires two or more systems to derive the same interpretation from a common content. For this to occur, users must:

1. Specify the structure and syntax of the information exchange. There are numerous existing standards, including:
    a. National Information Exchange Model (NIEM)
    b. Over the Horizon Gold (OTH-Gold)
    c. Unites States Message Text Format (USMTF)
    d. Multilateral Interoperability Programme (MIP) Information Model (MIM) XML
    e. Emergency Data Exchange Language (*EDXL*)
    f. Common Alerting Protocol (CAP)
2. Specify the assembly and processing rules for information elements contained in an information exchange in a clear, concise, and unambiguous manner. Rules that specify:
    a. The Assembly of releasable datasets:
        i. The patterns for the aggregation of data and information elements;
        ii. The transforms for the conversion of user data elements to sharing agreement standards;
        iii. The Tagging and marking of information aggregates and message elements to address operational needs (e.g., privacy, confidentiality, security, legal issues, and Quality of service;
        iv. Filtering or redaction of data and information elements to assure that datasets are releasable within the operational context.
    b. Assignment of the information exchanges to the appropriate dissemination channels.
    c. Formatting data to standard Message protocols (e.g., NIEM).
    d. The inverse of the assembly, the processing of received messages or datasets:
        i. The parsing (separation) of messages into their constituent information and data elements;
        ii. The validation that the data fulfills sharing agreement requirements;
        iii. The transformation of received data into user data standards;
        iv. The marshaling (assignment and transfer) of data and information to the appropriate data store(s).
3. Capture and retain information about the rules governing the operation of transactional interfaces in a manner that enables certification and accreditation.

Items 2 and 3 (above) are aspects of the system life-cycles that are not well serviced by traditional development practices, frameworks, tools, and technologies. The translation of policies to executable rules is typically based on textual requirements that are encoded in software. In addition, information systems have developed, emerged, and evolved with varying degrees of independence, largely based on the operational needs of an organization. These systems rarely took full account of broader community interoperability requirements. In addition the requirement to separate information by sensitivity (classification, confidentiality and privacy, legal significance and caveat) has further contributed to and

justified the development of stove-piped capabilities and a lack of interoperability. The goal of many organizations and communities (e.g., First responders, Emergency Management, Public Safety, National Security, Intelligence, and Military) is to link these disparate and partitioned systems in a manner that they can provide sustained, responsible, and reliable information during international and inter-agency operations; providing timely, knowledge-based decision making by leadership and decision makers at all levels; instigating the need for better practices, tools, and technologies.

Although not exclusive to the target communities (above), the IEPPV is focused on the delivery of secure, adaptive, and transaction based information sharing sought by organizations responding to dynamic real-world events. The more stable information sharing requirements of the business community would likely use a subset of the concepts in the vocabulary and target implementation at ETL (Extract, Transform, and Load) tools.

Into this requirement domain, the IEPPV provides a domain agnostic vocabulary that can be implemented in multiple policy automation strategies. The IEPPV provides the structural concepts for expressing rules governing the packaging (assemble and formatting) of information and inverse processing of received messages or datasets in accordance with user Information sharing and safeguarding Policy. This specification provides a UML and OWL representation of the IEPPV vocabulary. The UML profile (Annex B) will enable users to develop policy models that translate policy into corresponding rules that are aligned to the user's data environment. The use of UML to develop the policy models provides the option to use Model Drive Architecture (MDA) transformation to serialize the models as interface code or policy/rules languages that can be executed by multiple services (i.e., decision and enforcement points) or platforms.

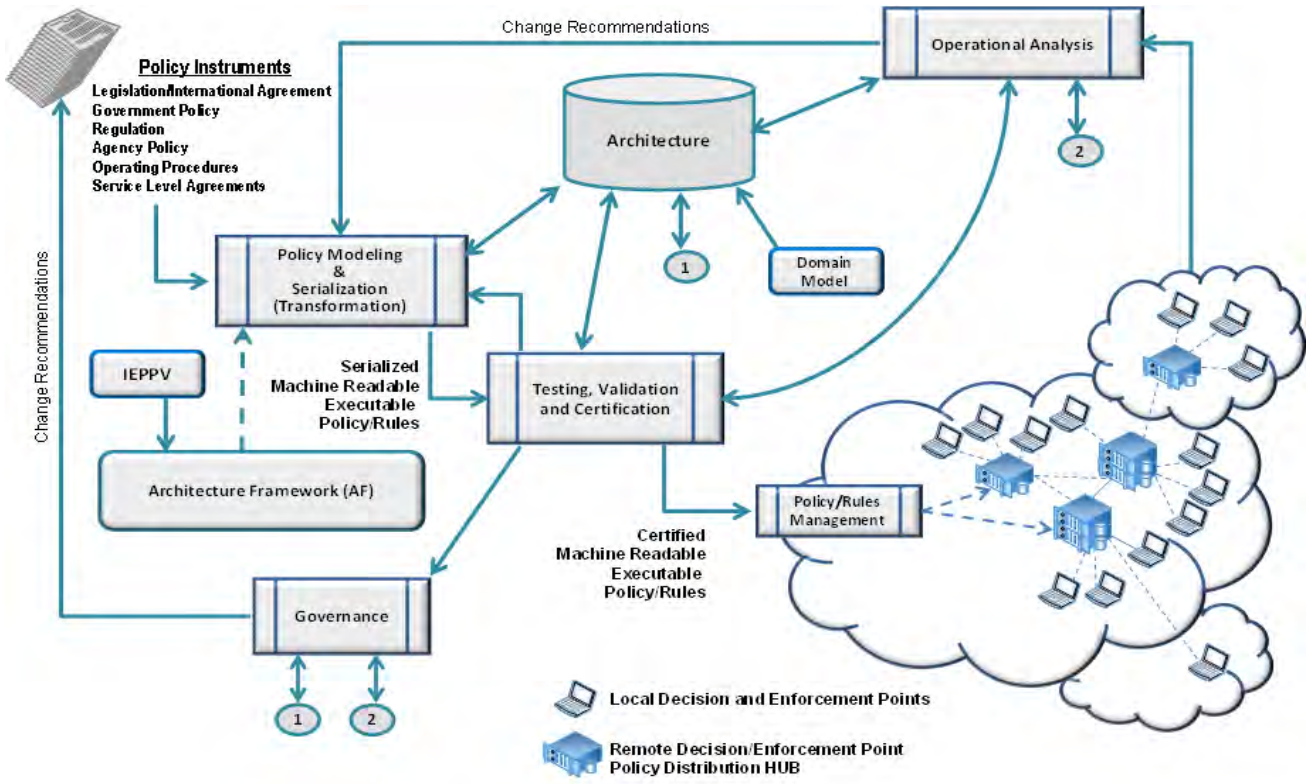

Figure 1.2 - Policy Life-Cycle

**Note: Governance is informed by the information derived from Architecture and Operational Analysis. These information flows illustrated as connectors 1 & 2 (enclosed in ovals) in Figure 1-2.*

As illustrated in Figure 1.2, IEF and in this case the IEPPV, is seeking a systematic process for translating information sharing and safeguarding policy instruments (e.g., legislation, regulation, policy, and service level agreements) into a machine consumable form that can be automated in the operational (/runtime) environment. This specification offers one option, a model based transformation using the UML profile (Annex B) to model user policy in a manner that aligns the policy to the specification data environment. The IEPPV UML Profile is used to define permissible patterns for assembling data and information elements into releasable datasets that conform to the originating policy. These policy models can then be transformed into a serialized form that is machine consumable and automated by platform specific implementations of policy decision and enforcement points linked to user data stores.

Key elements in a policy life-cycle include:

- **Policy Instruments**: typically unstructured textual documents that express information sharing and safeguarding policy.
- **Policy modeling and serialization**: implements the IEPPV profile and other Architecture Views to develop policy models that align information sharing policy with operational need and data domains. Using UML to develop the user policy models will enable the use of QVT (*Query/View/Transformation*) or other MDA approaches to serialize the policy model to one or more machine readable and enforceable languages (e.g., XACML).
- **Testing, Validation, and certification**: testing, modeling and Simulation and analysis tools that enable users to validate and verify that policy models and machine readable serialization conform to the originating policies.
- **Policy/Rules Management**: the deployment, management, and administration of policies/rules in the operational domain.
- **Operational Analysis**: procedure and tools used to determine the effectiveness and efficiency of ISS policy in the operational domain.
- **Governance**: the system of rules, practices, and processes by which ISS policies are directed and controlled.
- **Decision and Enforcement Points**: applications and services that combine to enforce ISS policy.

The OWL implementation is intended to be used with reasoning applications to provide services to assess or validate the composite of policies being instantiated within an operational environment. These services might include the identification of conflicting rules, or combinations of rule sets (that may have been developed separately) that may cause situations where privacy or security considerations may be breached. These types of application may also spawn the development of analytical and business intelligence services that enable:

- Governance and Stewardship
- Certification and Accreditation (C&A)
- Threat Risk Assessments (TRA)
- Statement of Sensitivity (SoS)
- Modeling and Simulation (M&S)
- Pre and Post Mission Scenario Analysis
- Design and Operational Audits (e.g., Security)

2 Compliance

2.1 Introduction

The Information Exchange Packaging Policy Vocabulary compliance points are on the complexity of the message to be supported by the target implementation. The compliance points are derived from:

- **SOPES (all compliance points)**: The Shared Operational Picture Exchange Services (SOPES) Information Exchange Data Model (IEDM) provides a set of platform independent concepts for the expression of governing the assembly and processing of datasets, including:
  - Contract (renamed InformationExchangeSpecification)
  - Semantic (renamed to SemanticElement in the IEPPV)
  - Transactional (renamed to TransactionalElement in the IEPPV)
  - Wrapper (Wrapper Element)

  The IEPPV extends SOPES IEDM by adding concepts to the expression of rules for:
  - The transformation of data elements
  - The adding of tags and marking to assembled information elements
  - For the filtering/redaction of information elements during the assembly process
- **LEXS (Compliance Point 2)**: Logical Entity Exchange Specification (LEXS: http://lexs.codeplex.com/) that defines an XML message structure for complex information environments. LEXS added concepts including:
  - Information Package
  - Structured Payload (e.g., NIEM Message)
  - Metadata
  - Digest
  - Linkages

  Compliance Point 2 provides implementers with 3 options as to the complexity of message structure they want to support.

The compliance points are structured in a manner that a basic data exchange (CP-1) to the full complexity of a multiple payload and attachment message.

2.2 Selecting a Compliance Point

The IEPPV is a vocabulary specification. It defines a set of concepts that combine to express the rules governing packaging, processing, and dissemination of information. The compliance points allow the implementers to select the level of message complexity they need to support. CP1 through CP2c build on the concepts defined in the previous levels.

CP-3 provides a set of concepts that enable users to assign information elements to specific information dissemination services.

2.3 Compliance Points

The following compliance points have been set for the IEPPV:

1. CP1 (Mandatory): Information Payload Specification
2. CP2a (Optional): Basic Message Specification (single Information Payload only)
3. CP2b (Optional): Extended Message Specification (single Package only)
4. CP2c (Optional): Full Message Specification (multiple Packages)
5. CP3 (Optional): Information Exchange Specification

Implementations of this specification must address CP-1.

2.3.1 Compliance Point 1 (Mandatory): Information Payload Specification

Compliance Point 1 (CP-1) forms the foundation of this specification, and is mandatory to all implementations. CP-1 provides a set of concepts for expressing rules governing the assembly and processing of a releasable dataset.

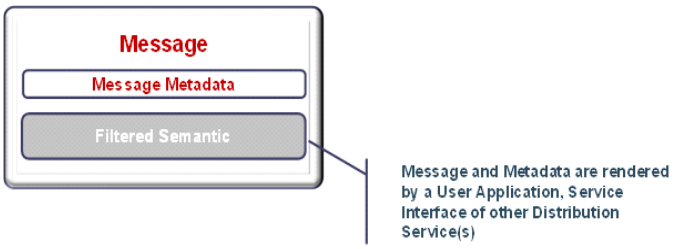

Figure 2.1 - Compliance Point 1

The enforcement of the rules derived from CP-1 concepts will result in the assembly of an unformatted dataset. CP-1 vocabulary concepts are defined in sub clause 7.3.

2.3.2 Compliance Point 2

Compliance Points 2a, b, and c extend the packaging policy vocabulary to include message structure and formatting rules and instructions. These three compliance points enable the specification of the message structure and format for the message content specified by the instructions contained within one or more Filtered-Semantics. The three CP-2 compliance points enable varying levels of complexity within the message structure.

These compliance points do not address how the middleware builds or routes the information. Middleware can integrate policy driven services like those proposed for IEF standardization or integrate the vocabulary into their own internal policy or scripting languages. This specification seeks conformance to the vocabulary, properties, and restrictions expressed in Clause 7.

2.3.2.1 Compliance Point 2a (Optional): Basic Message Specification

CP-2a adds a basic Message Specification to the policy specification, but only includes the minimum set of elements (metadata, payload, and attachments). This is the simplest form of a Message Specification. Compliance to CP-2a requires the concepts built into CP-1.

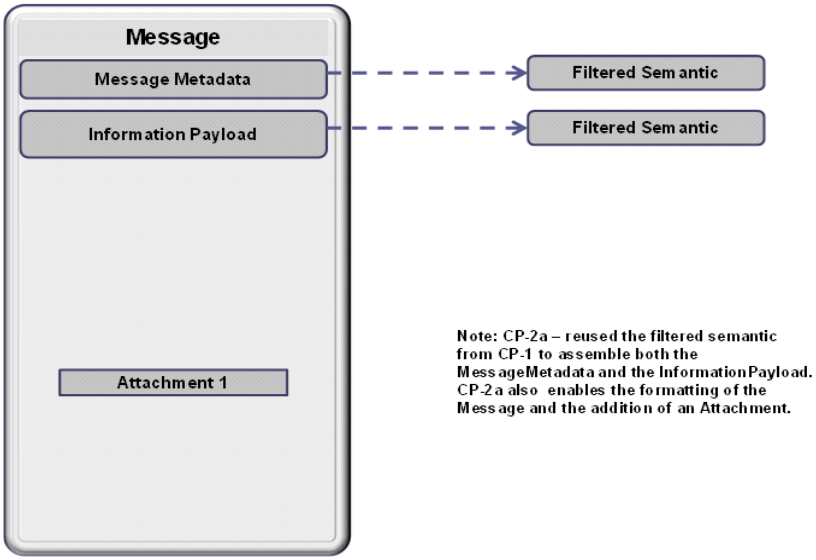

Figure 2.2 - Compliance Point 2a

The enforcement of the rules supported by CP-2a concepts will result in the generation of a basic message including the MessageMetadata, one informationPayload and up to 1 Attachment. CP-2a vocabulary concepts are defined in sub clause 7.4.

2.3.2.2 Compliance Point 2b (Optional): Extended Message Specification

CP-2b extends the scope of the Message Specification, building on CP1 and CP2a requirements. It increases the capability provided by CP2a by including support for several additional elements sometime integrated into an Information Package:

1. Message Metadata
2. Submitter Metadata
3. One Information Package to be included in the message; including:
    a. Package Metadata
    b. Digest
    c. Information Payload
    d. Rendering Instruction
4. Attachments

CP2b provides for the specification of complex message types required by certain diverse communities (e.g., Justice and Law Enforcement). Compliance to CP2b includes all concepts defined by CP1 and CP2a.

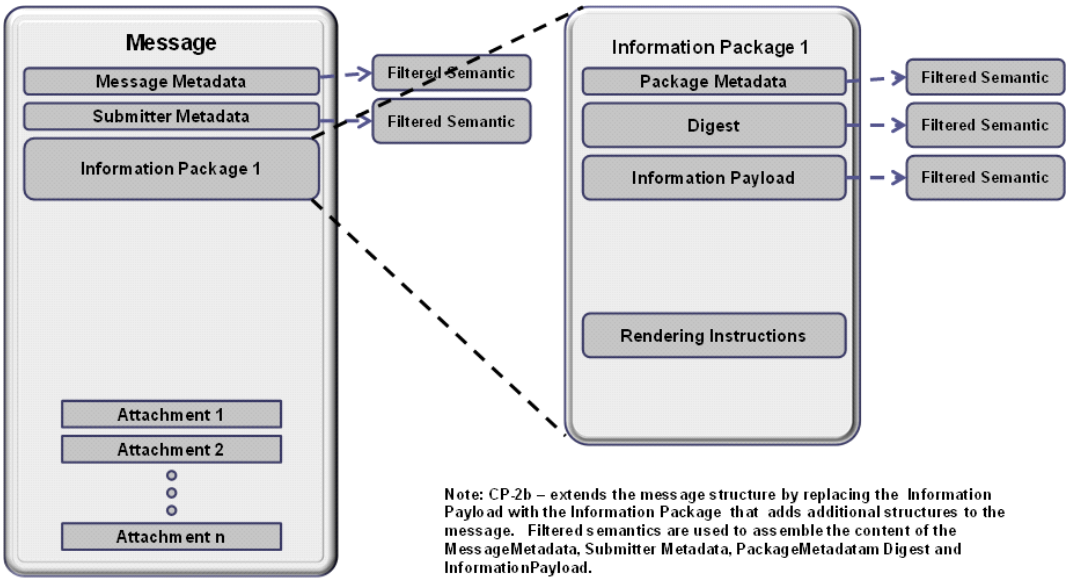

Figure 2.3 - Compliance Point 2b

The enforcement of the rules supported by CP-2b concepts will result in the generation of a message including the MessageMetadata, SubmitterMetadata, one InformationPackage and multiple Attachments. CP-2b vocabulary concepts are defined in sub clause 7.5.

2.3.2.3 Compliance Point 2c (Optional): Information Specification

CP-2c further extends the Message Specification defined in CP-2b. Compliance Point 2c adds the ability to include multiple Information Packages to the message. Elements added to CP-2c include:

1. Multiple Information Packages
2. Addition of Information Package elements, including:
    a. Linkages
    b. Attachment Summaries
    c. Narrative Text

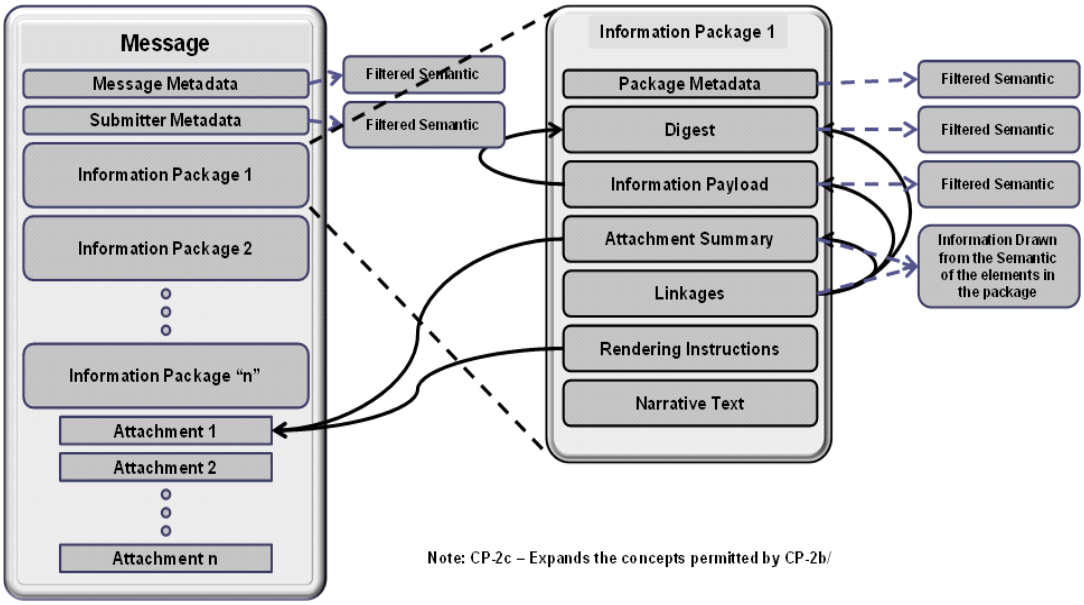

Figure 2.4 - Compliance Point 2c

The enforcement of the rules supported by CP-2c concepts will result in the generation of a message including the MessageMetadata, SubmitterMetadata, multiple InformationPackages, and multiple Attachments. Several additional concepts are included in the CP-2c InformationPackage. CP-2c vocabulary concepts are defined in sub clause 7.6.

2.3.3 Compliance Point 3 (Optional): Distribution Specification

Compliance Point 3 adds the ability to specify a basic distribution specification, which includes:

1. Session
2. Session Specification
3. Release Instructions
4. Quality of Service Requirements

Compliance to CP3 includes one of compliance to CP1, CP-2a, CP-2b, or CP-2c, and the Distribution Specification. The Distribution Specification directs the use of a specific Information Dissemination Service to be used.

2.4 Domain Vocabularies

This specification does not direct conformance to any specific domain or community vocabulary. The policy vocabulary, specified herein, defines concepts that will enable users to translate business policy into information processing and assembly rules independent of the operational and business domain. Domain vocabulary is integrated into the expression of rules. The IEPPV allows users to systematically express and align business policy to individual business (/operational) domains. This approach applies to both domain specific vocabularies and Metadata (tag-values).

If expressed in a modeling language, such as UML (see Annex C – UML Profile), this alignment may be directly integrated into an Enterprise, Business, Information, or Security Architecture. In this case, the domain specific concepts become the class names on the various ***Specifications***, ***SemanticElements***, ***TransactionalElements***, ***WrapperElements***, and ***Attributes***. The SOPES IEDM (formal/2011/05/04) is an information exchange model that conforms to the IEPPV policy vocabulary.

3 Normative References

The following normative documents contain provisions which, through reference in this text, constitute provisions of this specification. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply.

1. OMG Unified Modeling Language (OMG UML), Superstructure, Version 2.4.1, formal/2011-08-06 (http://www.omg.org/spec/UML/2.4.1/Superstructure/PDF)
2. OMG MOF 2 XMI Mapping Specification, Version 2.4.1. formal/2011-08-09 (http://www.omg.org/spec/XMI/2.4.1/PDF)
3. Unified Profile for DODAF and MODAF (UPDM) version 2.1 Formal/2013-08-04 http://www.omg.org/spec/UPDM/2.1
4. UML 2.3.1 OCL Specification (http://www.omg.org/spec/OCL/2.3.1/)
5. Shared Operational Picture Exchange Services (SOPES) Information Exchange Data Model (IEDM) Version 1.0 (http://www.omg.org/cgi-bin/doc?formal/2011-05-04.pdf), Annex A – Modeling Profile.
6. OMG Ontology Definition Metamodel (ODM), Version 1.0 (available at http://www.omg.org/spec/ODM/1.0/)
7. OWL 2 Web Ontology Language Quick Reference Guide, W3C Recommendation 27 October 2009, available at http://www.w3.org/TR/2009/REC-owl2-quick-reference-20091027/
8. OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax, W3C Recommendation 27 October 2009, available at http://www.w3.org/TR/2009/REC-owl2-syntax-20091027/
9. Resource Description Framework (RDF): Concepts and Abstract Syntax, W3C Recommendation 10 February 2004, available at http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/
10. RDF Vocabulary Description Language 1.0: RDF Schema, W3C Recommendation 10 February 2004, available at http://www.w3.org/TR/2004/REC-rdf-schema-20040210/
11. LEXS, Logical Entity Exchange Specification, http://lexs.codeplex.com/
12. NIEM, National Information Exchange Model, https://www.niem.gov/Pages/default.aspx
13. OMG SOPES IEDM, Shared Operational Picture Exchange Services Information Exchange Data Model, formal/2010-05-04, http://www.omg.org/spec/SOPES/1.0/PDF
14. Shared Operational Picture Exchange Services (SOPES) Information Exchange Data Model (IEDM) Version 1.0 (http://www.omg.org/cgi-bin/doc?formal/2011-05-04.pdf), Annex A – Modeling Profile.
15. JC3IEDM, Joint Consultation Command and Control Information Exchange Data Model, https://mipsite.lsec.dnd.ca/Public%20Document%20Library/Forms/AllItems.aspx?RootFolder=%2fPublic%20Document%20Library%2f04-Baseline_3.1%2fInterface-Specification%2fJC3IEDM&FolderCTID=0x012000CDEC559A618DF74781A1E0AE00DB1626

16. http://www.omg.org/cgi-bin/doc?ab/2012-11-01 - an MS Excel spreadsheet defining the metadata, and
17. http://www.omg.org/cgi-bin/doc?ab/2012-11-02 - the corresponding RDF/XML serialized OWL ontology
18. UML Profile for NIEM (NIEM_UML) 1.0 , http://www.omg.org/spec/NIEM_UML/
19. Simple Knowledge Organization System (SKOS) Reference, http://www.w3.org/TR/2009/REC-skos-reference-20090818/

4 Terms and Definitions

The focus of this specification is the development of a formal vocabulary (terms and definitions) for the specification and design of information/data packaging policy (/business rules and Constraints). The definitions for the Information Exchange Packaging Policy Vocabulary elements are included in Clause 7 and Annex A.

To assist the reader who may not be familiar with the information sharing and safeguarding domain, Annex F provides a glossary of these terms and acronyms. These definitions are provided for information purposes only.

5 Symbols

There are no additional symbols defined for this specification. All symbols used in this specification are based on standard UML.

6 Additional Information

6.1 Intended Audience

This specification will be of interest to end users, analysts and integrators who will use this profile to define information exchange specifications and tool vendors interested in developing tools support for the development and sustainment of information interoperability solutions. End users, auditors, and developers will have a clearer understanding of the semantic and business rules (sharing and safeguarding) for information exchange.

6.2 Acknowledgements

The following organizations are the direct submitters to this specification:

- Advanced System Management Group (ASMG) Ltd.

Contributors (/Contributing Entities)

The following organizations contributed tools, knowledge or resources to the development of this specification:

- Sandpiper Software; who provided the Visual Ontology Modeling (VOM) Tools instrumental to the development of this specification.
- Thematix Partners LLC; provided knowledge and expertise in the development of formal vocabularies and ontologies central to the development of this specification.

The following companies submitted and/or supported parts of this specification:

- Advanced System Management Group (ASMG) Ltd.
- Thematix Partners LLC

In particular the submitter would like to acknowledge the participation and contribution from the following individuals: Michael Abramson (ASMG), Jean Claude Lecomte (ASMG), Simon Brameld (ASMG), Michael Wiwchar (ASMG), Eric Penwill (ASMG), Elisa Kendall (Thematix).

The authors of this IEPPV Specification are therefore greatly indebted to organizations and authors who have contributed to all SOPES and IEF specifications over the years. Some of these are listed above.

The following organizations identified support for the concepts and content included in this specification.

1. MITRE
2. Raytheon
3. Centre for Security Sciences (CSS), Defence Research and Development Canada (DRDC)
4. KDM Analytics
5. Model Driven Solutions
6. IBM Canada
7. Atego
8. MIAB Systems Ltd
9. Lecomte Systems
10. PKH Enterprises (US)

6.3 Additional Materials

N/A

6.4 Vocabulary Architecture

6.4.1 Introduction to IEPPV

The IEPPV specification reuses a subset of UML 2 and provides additional extensions needed to address requirements specific to the IEPV RFP (mars/2011-03-15). The IEPPV submitters used the RFP requirements as the basis for this specification. This specification documents the language architecture in terms of the parts of UML 2 that are reused and the extensions to UML 2. This clause explains design principles and how they are applied.

6.4.2 ODM

The IEPPV was modeled using UML coupled with a profile that implements the Ontology Definition Metamodel (ODM) profiles for the Resource Description Framework (RDF) and OWL, and generates the RDF/XML artifacts as OWL 2.0-compliant documents. The resulting ontologies have been tested using the W3C RDF Validators and several OWL-DL compliant reasoning tools.

Metadata developed for the IEPPV utilizes the OMG Architecture Board recommendation for specification metadata, available at http://www.omg.org/techprocess/AB/SM/20120614/SpecificationMetadata.owl.

6.4.3 Philosophy

The IEPPV was developed using a model-driven approach. A simple description of the work process is:

- The IEPPV Vocabulary was developed using Ontology Definition Metamodel (ODM) Diagrams;
- The Vocabulary was expressed as a UML Profile (ANNEX C) for modeling information packaging rules;
- The conformance levels were finalized;
- The OWL representation of the Vocabulary was generated using a MDA transformation based on the ODM Diagrams in Clause 7;
- The XMI representation of the model was generated from the UML Tool;
- The Profile diagrams, stereotype descriptions, and documentation were added; and
- The specification was generated from the model.

This approach allowed the team to concentrate on architecture issues rather than documentation production. Consistency was automatically maintained by the UML tool.

6.4.4 Core Principles

The fundamental design principles for IEPPV include:

- **Requirements-driven –** IEPPV is intended to satisfy the requirements of the IEPV RFP Mandatory Requirements.
- **Reuse of existing specifications** – IEPPV reuses UML wherever practical to satisfy the requirements of the IEPV RFP (mars/2011-03-15) and leverages features from UML to provide a robust modeling capability. Consequently, IEPPV is intended to provide a path for tool vendors to develop a model based information packaging and protection solution. The vocabulary is seeking to provide a vocabulary that frames many of the community-derived Extensible Mark-up Language (*XML*) based exchange standards/specifications **(**e.g., **NIEM** and **EDXL)** and messaging specifications **(**e.g.**, LEXS, ATOM**, and **EDXL/DE)**. In addition the IEPPV seeks to support transformation to multiple standardized policy languages, including (references in Annex F):
  - Security Assertion Markup Language 2.0 (*SAML* 2.0)
  - eXtensible Access Control Markup Language (XACML 1.0)
  - Ponder
- **Partitioning -** The package is the basic unit of partitioning in this specification. The packages partition the model elements into logical groupings that minimize circular dependencies among them.
- **Architecture –** The IEPPV will be directly tied into architecture frameworks through the UPDM.

6.4.5 Ontology Development Approach

The IEPPV has been designed from the outset as an Ontology Definition Metamodel (ODM)[1] compliant ontology. By this we mean that the basic model was developed as a UML model with the ODM RDF and OWL Profiles applied. In addition to the UML/XMI for the model itself, the normative artifacts include a Web Ontology Language (OWL) 2.[2]0 compliant ontology, serialized as an RDF/XML document. Primarily because of the use of qualified cardinality restrictions, development necessitated the use of a version of the profile that supports OWL 2, currently in work by the ODM RTF. This specification was produced using changes to the ODM 1.0 profile that have already been approved by the ODM 1.1 RTF.

Our approach included ontology modeling with subject matter expert review of both the diagrams and related text definitions, generation of OWL from the model, and validation of the resulting ontology through a combination of the OWL editor[3,4,5,6] from Stanford Center for Biomedical Informatics Research at Stanford University, the Information Systems Group in the Department of Computer Science at Oxford University, and Clark & Parsia. Issues uncovered through reasoning over the model were then corrected in the UML environment and the process of generation / validation was repeated using both reasoners to ensure the accuracy of the results. The combination of ODM-based UML visualization and OWL 2 reasoning support enabled us to produce what we believe is a high-quality, logically consistent ontology for use by our community.

6.4.6 Ontology Architecture and Namespaces

The ontology architecture for IEPPV is designed to facilitate reuse and ontology evolution to the degree possible. An approach that provides very high-level, abstract conceptual knowledge designed to facilitate mapping is an important design goal. It depends on (1) basic terminology and ontology metadata, such as the OMG Architecture Board's Specification Metadata recommendation, and (2) may ultimately require the use of a number of external modules, representing concepts for units of measure, depending on the message payload requirements, and concepts defining dates, times, calendars, and schedules.

The namespaces and their well-known prefixes corresponding to external elements required for use of the IEPPV include all of those listed in Table 6.1.

Table 6.1 - Prefix and Namespaces for referenced/external vocabularies

Namespace Prefix	Namespace
rdf	http://www.w3.org/1990/02/22-rdf-syntax-ns#
rdfs	http://www.w3.org/2000/01/rdf-schema#
owl	http://www.w3.org/2000/07/owl#
xsd	http://www.w3.org/2001/02/XML-Schema#
dct	http://purl.org/dc/terms/
skos	http://www.w3.org/2004/02/slos/core#
sm	http://www.omg.org/techprocess/ab/SpecificationMetadata/

1. http://www.omg.org/spec/ODM/1.0/
2. http://www.w3.org/TR/2012/REC-owl2-syntax-20121211/
3. http://protege.stanford.edu/
4. http://bmir.stanford.edu/
5. http://www.hermit-reasoner.com/
6. http://clarkparsia.com/pellet

The namespace approach taken for IEPPV is based on OMG guidelines and is constructed as follows:

- A standard OMG prefix, http://www.omg.org/spec/
- The family name, IEF
- The abbreviation for the specification: IEPPV

Note that the URI/IRI strategy for the ontology takes a “slash” rather than “hash” approach, in order to accommodate server-side applications. Though not technically necessary, this specification does mandate namespace prefixes to be used. These are constructed as follows with the components separate by “-”:

- The specification family name ief
- The specification abbreviation: ieppv

The namespace itself for this specification is: http://www.omg.org/spec/IEF/IEPPV/IEPPV1-0/, and corresponding namespace prefix is ief-ieppv. The version IRI for the specification is http://www.omg.org/spec/IEF/IEPPV/20131101/IEPPV1-0/.

6.5 Specification Metadata

The OMG Architecture Board has recommended a metadata strategy, initially designed to support ontology, vocabulary, and other content oriented models. The IEPPV and other current OMG content models have adopted this recommendation for two reasons: (1) such metadata is needed to document the model specified herein, and (2) to support the Architecture Board in vetting the efficacy of this recommendation. The recommendation extends the Dublin Core Metadata Terms standard[7] and the W3C Simple Knowledge Organization System (SKO[8]S), and is partially derived from ISO/IEC FDIS 11179-3 Information technology - Metadata registries (MDR) - Part 3: Registry metamodel and basic attributes 3rd Edition and ISO/IEC FCD 24706 Metadata for technical standards and specifications document[9]s [3], tailored to support the OMG process.

For the purposes of this specification, we have incorporated recommendations for module and file-level metadata in both the ODM/UML and OWL model files, and will augment this with specification level metadata.

7. http://www.dublincore.org/
8. http://www.w3.org/standards/techs/skos#w3c_all
9. http://www.metadata-standards.org/ -- home page of ISO JTC 1 SC32 WG 2, where the ISO standard documents are available RFP Requested Discussions

7 Information Exchange Packaging Policy Vocabulary Specification

7.1 Introduction

This defines the Information Exchange Packaging Policy Vocabulary concepts, properties, and restrictions.

7.1.1 Modeling Conventions

The IEPPV is modeled using an ODM Profile. The colors applied to the elements in the diagrams possess no specific architectural meaning, they are provided to assist the reader to more rapidly identify different types of elements diagrams:

- Object of primary interest colored Green
- Objects drawn from other Ontologies are colored Yellow
- Object Properties are colored Mauve
- Unions are colored blue
- Restrictions are colored Orange

7.1.2 IEPPV Model Overview

This sub clause provides an overview for the IEPPV Model presented in sub clauses 7.2 through 7.6.

7.1.2.1 IEPPV Ontology Dependencies

As illustrated, the Information exchange Packaging Policy Vocabulary imports concepts from:

1. Specification Metadata (SM) Ontology
2. DCMI Metadata Terms
3. Simple Knowledge Organizations System (SKOS)

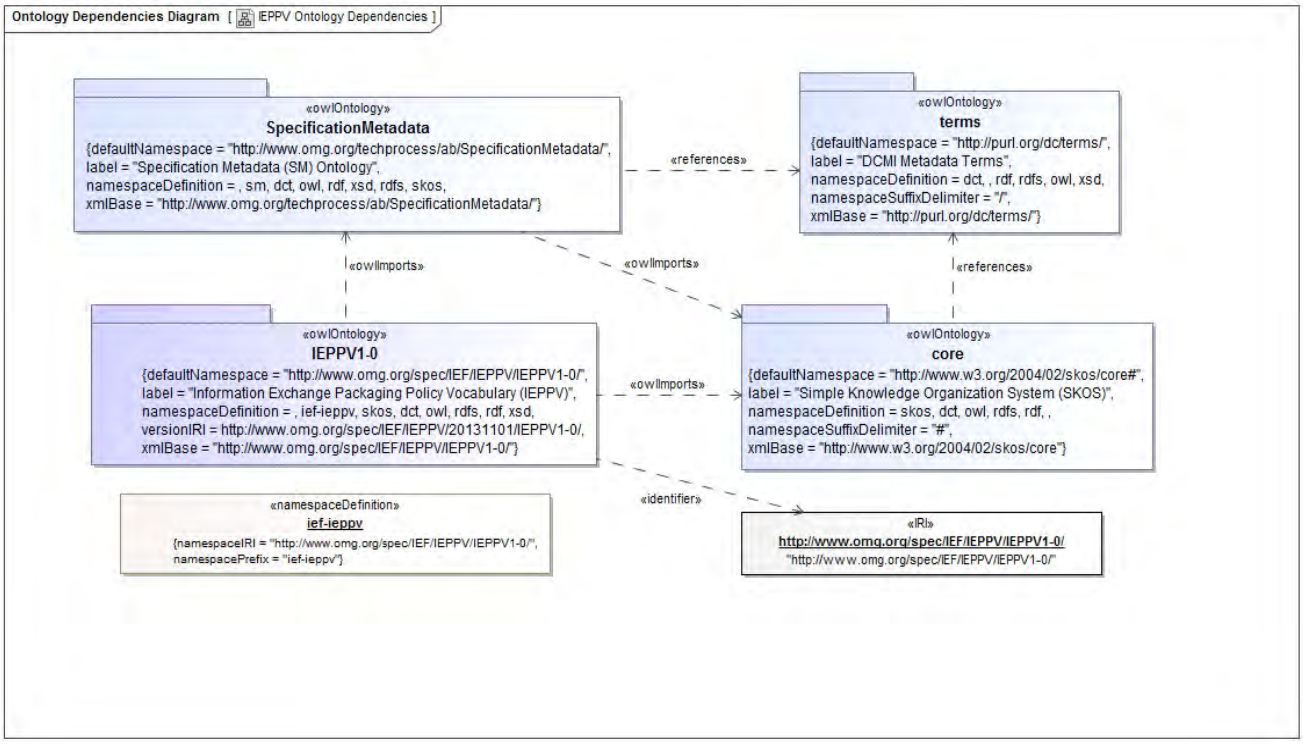

Figure 7.1 - IEPPV Ontology Dependencies

7.1.2.2 IEPPV Scope

The IEPPV addresses the requirements specified in the Information Exchange Policy Vocabulary (IEPV) RFP (mars/2011-03-15). The IEPPV is the first in a family of policy vocabularies intended to address a broad range of Information sharing and safeguarding (ISS) requirements. This family of ISS policy vocabularies, Information Exchange Policy Vocabularies (IEPV), will include specifications that address multiple ISS policy areas beyond the information packaging, including:

1. Identity Management
2. Credential Management
3. Access Management
4. Distribution and Dissemination Policy
5. Quality of Service

The IEPPV specifically address rules governing the Packaging (assembly (e.g., aggregation, transformation, filtering/ redaction and tagging/labeling/marking), and Formatting) and Processing (e.g., Parsing, validation, transformation, and marshaling) of information and data elements. The IEPPV also provides the concepts for a simple rules set for assigning the packaged information (ReleasableDataSet (unformatted) or Message (formatted) to the specified dissemination service.

As illustrated the Vocabulary is packaged in with respect to the compliance point in Clause 2. The Information Exchange Agreement binds the packaging of information (content) with the services specified to distribute or disseminate the information.

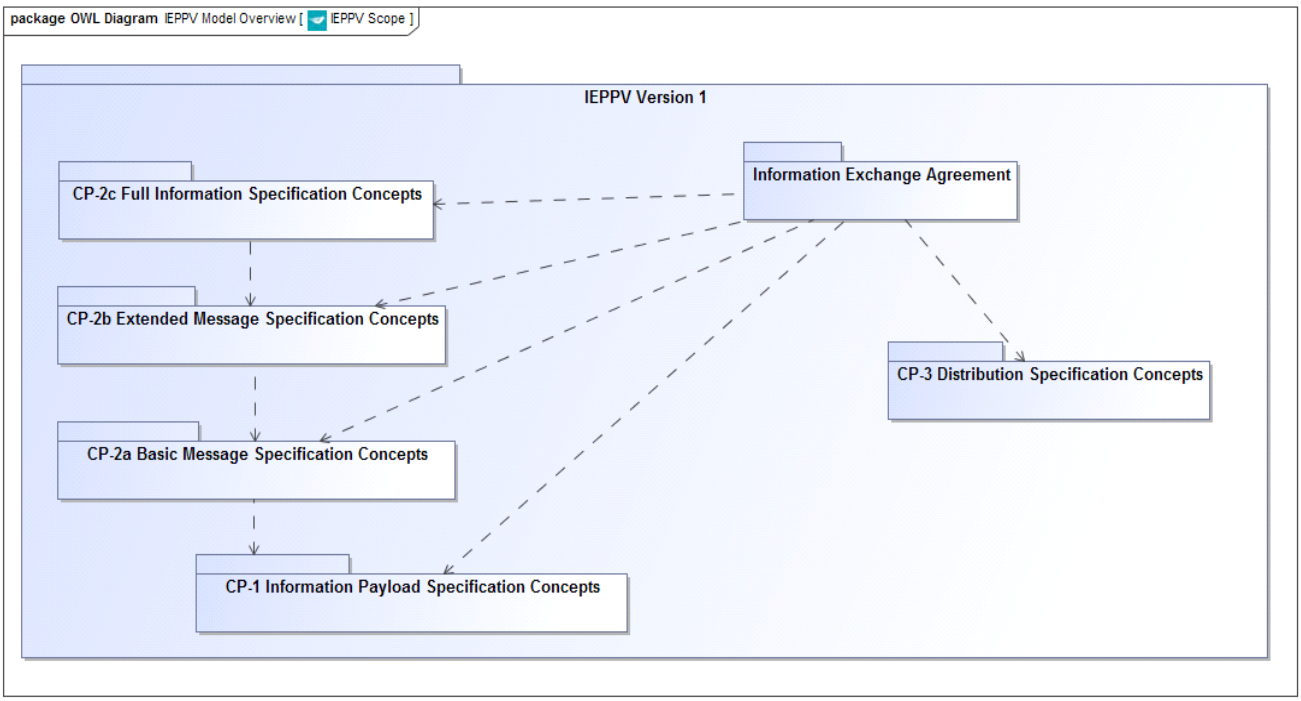

Figure 7.2 - IEPPV Scope

7.1.3 Concepts

This sub clause provides the definitions for concepts used throughout the ODM model provided in sub clauses 7.2 though 7.7.

AcknowledgeInstruction: An instruction to the recipient of an information exchange directing the issuance of an acknowledgment to the receipt of the information to the provider of the information.

ActionInstruction: An instruction directing the producer or receiver of a message to take a specific action, (1) message specific rules governing the release of the information, or (2) message specific actions to be taken upon receipt of the message.

AttachmentElement: A binary file or (e.g., PDF file, image or video) or document, and information about the binary or document, such as the size and type and description.
Source: Logical Entity Exchange Specification (LEXS): Attachment (N): A binary, such as an image or PDF file or video, as well as information about the binary, such as the size and type and description.

AttachmentFormattingInstruction: An instruction to the provider of information defining the rules for formatting the data set in accordance with the agreed protocol for the exchange.

AttachmentRenderingInstruction: An instruction to the recipient of an information exchange defining the rules for rendering or displaying an attachment or set of attachments.

AttachmentSemantic: A Semantic that specifies the rules for assembling the attachments to a message. It also provides the rules for generating an attachment summary and linkages.

AttachmentSpecification: A specification of the rules governing attachment of binary information elements to an information exchange or message.

AttachmentSummary: A summary or list of attachments for a specific data package.

AttachmentSummaryRenderingInstruction: An instruction to the recipient of an information exchange defining the rules for rendering or displaying an attachment summary.

Attribute: A defined property of an entity, object, triple, schema, etc.
Source: A Dictionary of Computing. Oxford University Press, 2008. Oxford Reference Online. Oxford University Press.

BinaryDataRenderingInstruction: An instruction to the recipient of an information exchange defining the rules for rendering or displaying binary data.

Container: A receptical for results of an aggregation of data and information elements. Derived from http://www.merriam-webster.com/dictionary/container, a receptacle (as a box or jar) for holding goods.

DataCreatorMetadata: Metadata tags and markings that identify the creator of data or information elements.

DataElement: Representation of information (data) in a formalized manner suitable for communication, interpretation, or processing by humans or by automated means. In the context of IEPPV, data elements are atomic facts.
Derived from UPDM.

DataOwnerMetadata: Tags and markings that identify the owner or steward of the data or information elements.

Digest: An information structure, format and syntax common to all communities. It provides the ability for systems to handle heterogeneous data without having to understand the specific context and or semantics of the source. As long as the entities relevant to the packaged data items are represented in the Digest, users will be able to discover, link, map, etc. the information within.
Source: the concept for digest is derived from and intended to support the Logical Entity eXchange Specification (LEXS). http://130.207.211.107/content/lexs-overview.

The Digest provides the common level of understanding, it does not mean that all sources have to populate all elements, or that all consumers have to use all elements; merely that at a schema level all applications understand the Digest. Implementers only need to build one module in order to produce or consume a basic set of data understandable by many. It also means that implementers do not have to develop large applications for each exchange, but rather build one that handles the basics and then additional smaller modules in order to produce or consume more complex exchanges. The objective of the Digest is to present the most common characteristics of real-world objects that can be supported by any data source or data consumer. Digest-level data objects may be further augmented or described with additional details in included packages or narrative text integrated into the message. The information in the digest must be semantically complete for both the data source or data consumer; the information package contents may rely on the digest to complete its semantics. The enforcement of a “Digest Semantic” by a software service will result in the generation of the digest for the instance of the Information Package. In other applications, where the digest is not used, the “Payload” comprises the entire data portion of the message content.

DigestFormattingInstruction: An instruction to the provider of information specifying the rules for formatting the data set for a Digest in accordance with the agreed protocol for the exchange.

DigestSemantic: A SemanticElement that specifies the rules for assembling data and information elements for a Digest.

DigestSpecification: A specification and set of rules governing the preparation (generation) of a digest.

DiscardInstruction: An instruction to the recipient of an information exchange specifying the rules for destruction or discarding of data included within an information package or message.

DistributionSpecification: A specification of the rules governing the assignment of InformationElements to a specific information dissemination service (e.g., User Application, Service Interface, and Middleware).

DoNotForwardInstruction: An instruction to the recipient of an information exchange specifying that the information must not be forwarded to any other recipient or destination.

DoNotPersistInstruction: An instruction to the recipient of an information exchange directing the recipient not to persist any of the information or data in a payload or message.

DynamicFilter: Rules for a data or domain filter whose parameters may be configured at run-time.

EnclosedTransactionalElement: A TransactionalElement included as part of the build pattern of a TransactionalElement or SemanticElement.

EnclosingTransactionalElement: A TransactionalElement that includes one or more TransactionalElements or WrapperElements.

EncryptInstruction: An instruction or set of instructions to the producer of the information directing that the message or elements of the message need to be encrypted prior to release.

Entity: Independent, separate, or self-contained existence.
Source: Merriam-Webster Dictionary

File: A collection of information, referred to by file name; for example, a user-created document, program data, or the program itself. With a program, the information is held on backing store (i.e., usually on magnetic disk) in order (a) to enable it to persist beyond the time of execution of a single job and/or (b) to overcome space limitations in main memory. Files with a very brief existence (i.e., in case (b) above, or where they simply carry information between one job and the next in sequence) are called work files. See also master file, data file.
Source: A Dictionary of Computing. Ed John Daintith and Edmund Wright. Oxford University Press, 2008. Oxford Reference Online.

Filter: A profile or script containing the rules to restrict the assembly of data or information elements.
Source: Defined for the Information Exchange Policy Vocabulary. Derived from “A general term used to describe software which examines some content and prevents it from reaching its destination, based on a number of rules stored in a script or a profile.” A Dictionary of the Internet. Darrel Ince. Oxford University Press, 2009. Oxford Reference Online.

FilteredSemanticElement: Specifies rules for the assignment of one or more DynamicFilters to a specified SemanticElement.
Source: Derived from SOPES IEDM V1

FilteredTransactionalElement: Rules specifying the WrapperAttributes that are filterable at runtime.

FilterRule: A rule or rules governing the inclusion of or rejection of data or information elements based on the value of a specified attribute, or values of specified attributes.

FormattingInstruction: An instruction to the provider of information defining the rules for formatting a generated data set.

ForwardInstruction: An instruction to the recipient of an information exchange to forward the information to authorized recipients in accordance with any provided list, or in accordance with specified information sharing agreements.

HandlingInstruction: An instruction to the recipient of an information exchange specifying how this information must be handled.

Identifier: Identifies the element (TransactionalElement or WrapperElement) that holds a unique identifier or key needed for the construction of a data set. This subtended class would contain, as a minimum, the base global unique identifier (e.g., database key, foreign keys, or unique identifier) that would differentiate which Transactional or Wrapper instance (information element instances) is included in the construction of the composite (e.g., foreign key relationships). There exists one and only one identifier for each SemanticElement or TransactionalElement.
Source: Derived from UML Profile for DODAF and MODAF (UPDM) V2.0, formal/2012-01-03 and Shared Operational Picture Exchange Services (SOPES) Information Exchange Data Model (IEDM) Version 1.0, formal/2011-05-04

InformationElement: An item of information that flows between operational activities and nodes. For IEPPV, an information element refers to a grouping of data elements (including other information elements) providing meaning within the context of an operation or situation.
Derived from:
MODAF: A formalized representation of information subject to an operational process.
DoDAF: Information that is passed from one operational node to another. Associated with an information element are such performance attributes as timeliness, quality, and quantity values. (DoDAF) Information Exchange: The collection of information elements and their performance attributes such as timeliness, quality, and quantity values. (DoDAF).
Note: Within the architectural context of the UPDM, SOPES, and IEPPV, the Information element provides a description of, or specification for, the data or information processed or exchanged. The Information element does not refer to the instance data or information being processed or exchanged, as this can only be determined at run-time.

InformationExchangeSpecification: Specifies the information elements shared as part of a specific information sharing agreement and the information dissemination services to be used.

InformationPackage: A standard representation of structured, semi-structured, and binary information applicable to an information sharing agreement. Packages may contain metadata, a Digest, a Structured Payload, Rendering Instructions, and optional linkages depending on the established agreements.

InformationPackageFormattingInstruction: An instruction to the provider of information defining rules for formatting the elements of a Data Package in accordance with the agreed protocol for the exchange.

InformationPackageMetadata: Tags and markings that identify and describe the contents of an information package.

InformationPackageMetadataFormattingInstruction: An instruction to the provider of information defining the rules for formatting the Data Package Metadata in accordance with the agreed protocol for the exchange.

InformationPackageMetadataSemantic: A SemanticElement that specifies the rules for assembling the data elements to be included within Information Package Metadata.

InformationPackageReleaseInstruction: An instruction to the producer of an information exchange specifying instructions (e.g., Encrypt) pertaining to the release of the information package or message.

InformationPackageRenderingInstruction: An instruction to the recipient of an information exchange defining the rules for rendering or displaying an Information Package.

InformationPackageSpecification: The rules and constraints governing the construction preparation of an information or data package.

InformationPayload: A formatted dataset without protocols and metadata required for an information exchange. Derived from: Body (payload) The part of a cell or packet in a network that holds the information supplied by the end-user for transmission from the sender to the receiver. A Dictionary of Computing. Ed John Daintith and Edmund Wright. Oxford University Press, 2008. Oxford Reference Online.
Data Payload: Refers to the "actual data" in a packet or file minus all headers attached for transport and minus all descriptive meta-data. In a network packet, headers are appended to the payload for transport and then discarded at their destination. In a key-length-value structure, the key and length are descriptive data about the value (the payload) http://www.pcmag.com/encyclopedia_term/0,1237,t=payload&i=48909,00.asp.

InformationPayloadFormattingInstruction: An instruction to the provider of information defining the rules for formatting the information payload in accordance with the agreed protocol for the information exchange.

InformationPayloadSpecification: The rules governing the assembly and processing of a structured dataset for an information exchange.

InformationSpecification: Specifies the InformationElements that are included as part of the Information Exchange Agreement.
Source: Defined for the Information Exchange Packaging Policy Vocabulary.

Instruction: The description of an operation that is to be performed by a computer or human operator. Derived from: "The description of an operation that is to be performed by a computer. It consists of a statement of an operation to be performed and some method of specifying the operands (or their locations) and the disposition of the result of the operation." A Dictionary of Computing. Ed John Daintith and Edmund Wright. Oxford University Press, 2008.

Message: A formatted InformationElement transferred by a message switching system (or Network). Messages may be of any length, from a few bits to a complete file, and no part of a message is released to its final recipient until all of the message has been received at the network node adjacent to the destination.
Source: A Dictionary of Computing. Ed John Daintith and Edmund Wright. Oxford University Press, 2008. Oxford Reference Online.

MessageElement: An identifiable part of a message structure containing contextually relevant data or information elements. Message elements are integrated and formatted in accordance with contract or information exchange specification rules and instructions prior to release.

MessageFormattingInstruction: An instruction to the provider of information defining the rules for formatting the elements of a Message in accordance with the agreed protocol for the exchange.

MessageMetadata: Set of tags and markings (including their established Values) that describe the content of a message.

MessageMetadataFormattingInstruction: An instruction to the provider of information defining the rules for formatting the elements of MessageMetadata in accordance with the agreed protocol for the exchange.

MessageMetadataRenderingInstruction: An instruction to the recipient of an information exchange defining the rules for rendering or displaying message metadata.

MessageMetadataSpecification: The rules governing the assembly of message metadata.

MessageRenderingInstruction: An instruction to the recipient of an information exchange defining the rules for rendering or displaying a message.

MessageSensitivity: Metadata Tag or marking that provides an indication of the sensitivity of the information with reference to privacy, confidentiality or security.

MessageSpecification: Specifies the rules and constraints governing the assembly of a community compliant structured or semi-structured message in accordance with a specified message protocol (e.g., LEXS, EDXL-DE, and ATOM).

MessageTimeStamp: Metadata Tag indicating when the Message was created.

MessageType: Metadata tag that identifies the type of message being exchanged.

Metadata: Data (tags and markings) that describes other data.
Source: A Dictionary of the Internet. Darrel Ince. Oxford University Press, 2009. Oxford Reference Online. Oxford University Press.

MetadataRenderingInstruction: An instruction to the recipient of an information exchange defining the rules for rendering or displaying metadata.

MetadataSemantic: A SemanticElement that specifies the rules for assembling the metadata elements.

MetadataSpecification: The rules governing the assembly of metadata to be attached to a message, package, information elements of an exchange covered by the contract.

NarrativeText: Identifies the location and rules for attaching a narrative of free text field to a message or package of information elements.

PackageMetadataSpecification: The rules governing the assembly of metadata and tags for an information package.

Participant: A List of entities to produce or receive the information or message.
DODAF: Any entity - human, automated, or any aggregation of human and or automated - that participates in an information exchange agreement.

PersistenceInstruction: An instruction to the recipient of an information exchange indicating that the information may be persisted in local stores.

PrivacyMetadata: Tags and or markings that support the enforcement of privacy policy.

PublisherMetadata: Tags and markings that support the publishing of sharable information to a data registry, repository, or publication-subscription middleware infrastructure. This metadata provides the structures required to represent the data as well as that associated with publishing and storage data. The data registry, repository, or middleware receives and records the published metadata in a manner for users and systems to discover the associated information elements.
Derived from: Logical Entity Exchange Specifications 4.0 (LEXS) User Guide (http://130.207.211.107/sites/all/lexs/docs/lexs-4.0/LEXS_4_UserGuide%209-27-2011.pdf)

QualityOfServiceInstruction: An instruction or set of instructions to the producer or publisher of the information specifying the quality of services requirements for the exchange of the information.

ReceiptInstruction: An instruction to the recipient of an information exchange to perform a particular operation, or multiple operations, upon the receipt of that information.

ReleasableDataSet: The assembly of data elements resulting from the enforcement of rules enclosed by a SemanticElement or FilteredSemanticElement.

ReleaseInstruction: An instruction or set of instructions to the producer or publisher of the information specifying actions to be taken prior to the release of the information. (e.g., encryption requirements).

RenderingInstruction: An instruction or set of instructions to the receiver of information describing the rules for rendering or displaying the information.

RetentionInstruction: An instruction to the recipient of an information exchange defining the rules regarding the allowable persistence of the information.

RetrievalMetadata: Tags and markings included in a message or information package that assists in the retrieval of that information.

Safeguard: Policies, rules, services, and technologies that serve to guard or protect data and information elements from malicious or inadvertent release of sensitive or protected information. Derived from http://www.thefreedictionary.com/safeguard, one that serves as protection or a guard.

SearchMetadata: Refers to metadata that broadly identifies the information elements being sought and results in a response that returns possible candidates for the user to examine further. This metadata provides the characteristics of a query to the registry, repository or publication-subscription infrastructure that responds with information pertaining to the sharable information elements, topics or channels that can be accessed. The response provides the information needed to request specific information elements, topic or channel subscription.
The intent is that the requesting entity can narrow the search by reviewing the search response and then request more detailed information on a specific information element, topic, or channel. Depending on the implementation, metadata could include a text-string and request for a text search on unstructured data in a registry or repository (e.g., report), or on structured data, such as a name, attachment or narrative element. A data item metadata search looks for one or more information elements containing information matching the criteria described in the SearchMetadata.
Derived from Logical Entity Exchange Specifications 4.0 (LEXS) User Guide (http://130.207.211.107/sites/all/lexs/docs/lexs-4.0/LEXS_4_UserGuide%209-27-2011.pdf)

SecurityFilter: A specialization of a filter that provides the rules that restrict the assembly of data and information elements based on the values of a security tag or label.

SecurityMetadata: Tags and markings that assist in the enforcement of security policy and malicious or inadvertent release of classified information to unauthorized recipients.

SecurityPolicy: A set of objectives, rules of behavior for users and administrators, and requirements for the configuration, operation and management of computer systems to enhance the security of organization or enterprise people, operations, and systems.

Note – This specification is focused on the specification of policies and rules for the packaging and release of information for authorized recipients. A Security Policy might include requirements or processes for:

1. Virus detection and prevention
2. Firewall use and configuration
3. Password strength and management
4. Host System administration practices
5. Access Control rules
6. Use of Access Logs
7. Use of screen locking software
8. Logging out of unattended workstations
9. Physical security

10. Account termination

11. Procedures for granting and revoking system access

SemanticAttribute: An attribute assigned to a semantic element.
Derived from UPDM

SemanticElement: Composite of rules governing the assembly of data elements in accordance with commitments defined by an information exchange agreement and policies pertaining to the safeguarding of sensitive information.
Derived from SOPES IEDM V1: Semantic

Session: The software connection to the information dissemination services to be used for the exchange of information under the informationExchangeSpecification.
Derived from the Seven Layer Reference Model:
1. Session Layer - Identifies the service of binding two presentation service entities together logically and controls the dialogue between them as far as message synchronization is concerned.
2. Presentation Layer - Provides a set of services that may be selected by the application to enable it to interpret the meaning of the data exchanges. Such services include management of the entity exchange, display, and control of the structured data. The presentation layer is the heart of the seven layer proposal, enabling disparate terminal and computer equipment to intercommunicate.
A Dictionary of Computing. Ed John Daintith and Edmund Wright. Oxford University Press, 2008. Oxford Reference Online. Oxford University Press.

SessionSpecification: Specifies the rules governing communications between the data services and information distribution services (or middleware).

SourceData: Raw data (sometimes called source data or atomic data) is data that has not been processed for use. A distinction is sometimes made between data and information to the effect that information is the end product of data processing.
Source: http://searchdatamanagement.techtarget.com/definition/raw-data.

Specification: A detailed precise presentation of something. Within the context of the IEPPV, a detailed and precise presentation of rules governing the assembly or processing of information elements.
Derived from http://www.merriam-webster.com/dictionary/specification: a detailed precise presentation of something or of a plan or proposal for something.

StaticFilter: A filter created at design-time that cannot be modified at run-time.

StructuredDataRenderingInstruction: An instruction to the recipient of an information exchange defining the rules for rendering or displaying structured data.

SubmitterMetadata: Tags and markings identifying the submitter of the information.

SubtendedElement: An Element (TransactionalElement or WrapperElement) forming part of another element (TransactionalElement or SemanticElement). Wrapper is always a subtended information element since it cannot exist outside of a TransactionalElement definition.

SubtendedElementAttribute: An attribute assigned to a SubtendedElement.

SubtendedTransactional: A TransactionalElement included as part of another TransactionalElement or SemanticElement (aka Supporting Transactional).

Table: A collection of records. Each record may store information associated with a key by which specific records are found, or the records may be arranged in an array so that the index is the key. In commercial applications the word table is often used as a synonym for matrix or array.
Source: A Dictionary of Computing. Ed John Daintith and Edmund Wright. Oxford University Press, 2008. Oxford Reference Online. Oxford University Press.

TimeStamp: A tag or mark indicating the time when the message was created.

TransactionalAttribute: An attribute assigned to a TransactionalElement.
Derived from UPDM.

TransactionalElement: Specifies a reusable pattern comprising rules governing the assembly and processing of data and information elements.
Derived from SOPES IEDM V1: Transactional.

Transformation: The conversion of data from one form to another. In this instance the specification of rules governing the conversion or transformation of data.
Source: A Dictionary of Computing. Ed John Daintith and Edmund Wright. Oxford University Press, 2008. Oxford Reference Online. Oxford University Press.

TransformationResultingAttribute: An attribute resulting from a transformationElement.

Triple: An RDF triple consists of three components:

- the subject, which is an IRI or a blank node;
- the predicate, which is an IRI;
- the object, which is an IRI, a literal, or a blank node.

An RDF triple is conventionally written in the order subject, predicate, object.
Source: http://www.w3.org/TR/2013/CR-rdf11-concepts-20131105/#section-triples

Tuple: An ordered set with an unspecified but finite number (n) of elements.
Source: A Dictionary of Computing. Ed John Daintith and Edmund Wright. Oxford University Press, 2008. Oxford Reference Online.

ValidateInstruction: An instruction to the recipient of an information exchange containing criteria for the validation of the content and semantics of the message or information payload.

WatchPoint: A trigger mechanism used by an application to commence the assembly of a TransactionalElement. A data model assigns this tagged value to a WrapperElement aggregation arc in the Transactional pattern. Additions to the underlying data store for this WrapperElement triggers the application to start building the composite.
Derived from SOPES IEDM V1: Wrapper.

WatchPointTransactionalElement: A TransactionalElement with an associated Watchpoint data event that triggers the assembly of enclosing TransactionalElements and SemanticElements.
Source: Derived from SOPES IEDM V1.

WrapperAttribute: An attribute assigned to a WrapperElement.
Source: derived from UPDM.

WrapperElement: A logical construct that wraps or encapsulates the definition of a data set, table entity, triple, file, etc. A Wrapper directly maps to a data instance (e.g., row of data in a database application) in the logical data model and the physical data model. Derived from Derived from SOPES IEDM V1: Wrapper

7.1.4 Object Properties

The following objectProperties are used to define the relationships between concepts in the following ODM Model.

assign: To specify additional rules or restrictions to an information element.
Derived from: to fix or specify in correspondence or relationship, http://www.merriam-webster.com/dictionary/assign.

comprises: to be made up of (something), to include or consist of (something).
Source: http://www.merriam-webster.com/dictionary/comprise

contains: To have within a larger container concept. Derived from: to have (something) inside, to have or include (something).
Source: http://www.merriam-webster.com/dictionary/contain

encloses: to surround (something), to put something around (something), to include along with something else in a parcel or envelope.
Source: http://www.merriam-webster.com/dictionary/enclose

includes: to make (someone or something) a part of something, to take in or comprise as a part of a whole or group.
Source: http://www.merriam-webster.com/dictionary/include

governsFormattingOf: to control or direct actions to be taken in the formatting of information.

governsReleaseOf: to control or direct actions to be taken during the release of information.

owns: to have (something) as property, to legally possess (something).
Source: http://www.merriam-webster.com/dictionary/own

produces: To generate, compute or produce a transformation of attributes and generate a result.
Derived from: to make (something) especially by using machines, to cause (something) to exist or happen, to cause (a particular result or effect), http://www.merriam-webster.com/dictionary/produce

references: To identify an association between one element and another.
Derived from: to mention (something or someone) in speech or in writing, to refer to (something or someone) http://www.merriam-webster.com/dictionary/references

Note – Although minimum cardinality is identified as 1 in some uses of this property, in some environments the reference infers the existence of Unique / or Globally Unique identifier or keyed relationship between the concepts (e.g., foreign key relationships in between tables in a relational construct).

restricts: to confine within bounds.
Source: http://www.merriam-webster.com/dictionary/restricts

resultsIn: an effect generated through the execution of a process, procedure, or rules.
Derived from http://www.merriam-webster.com/dictionary/results: to proceed or arise as a consequence, effect, or conclusion.

specifies: To explicitly state the policies, rules and instructions for generating a specific output.
Derived from http://www.merriam-webster.com/dictionary/specify: to name or state explicitly or in detail, to include as an item in a specification.

7.2 Information Exchange Agreement

The Information Exchange Agreement includes concepts within the Vocabulary that are used to bind the Information Packaging Concepts in CP-1 and CP-2s to the Dissemination concepts in CP-3.

7.2.1 Information Exchange Specification Concepts

The InformationExchangeSpecification includes concepts for the specification of rules that bind the information packaging and processing concepts in CP-1 and CP-2 (a, b and c) to the Distribution Concepts provided in CP-3.

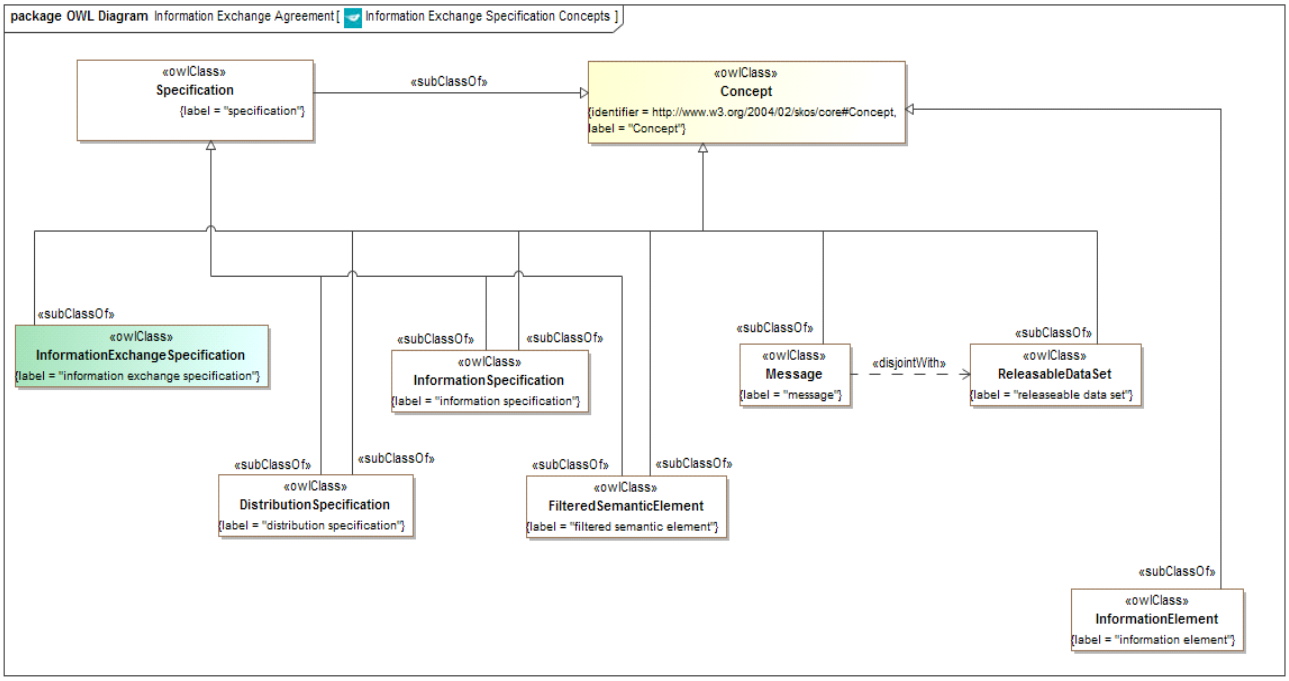

Figure 7.3 - Information Exchange Specification Concepts

7.2.2 Information Exchange Specification

The following figure illustrates the relationships between concepts for the expression of rules that bind the concepts in CP-1 and CP-2 to the concepts in CP-3.

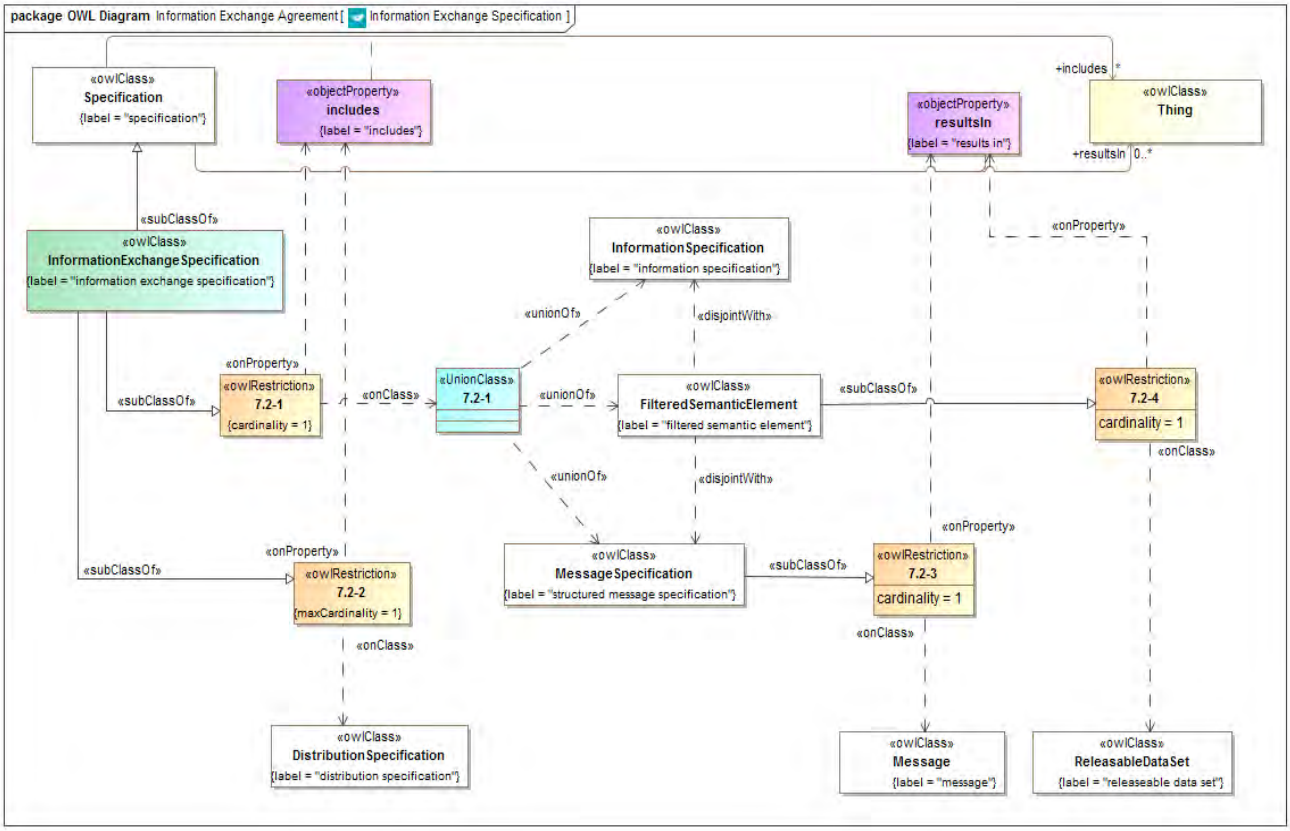

Figure 7.4 - Information Exchange Specification

The following rules apply to the Information Exchange Specification:

1. InformationExchangeSpecification includes one and only one includes.
2. InformationExchangeSpecification includes a maximum one DistributionSpecification.
3. MessageSpecification resultsIn one and only one Message.
4. FilteredSemanticElement resultsIn one and only one ReleasableDataSet.

* The name used to identify the rules is derived from the restriction encompassed by the rule.

7.2.3 Information Specification Concepts

The InformationSpecification includes concepts for the expression of rules that bind one or more InformationElements to an Information Exchange Agreement.

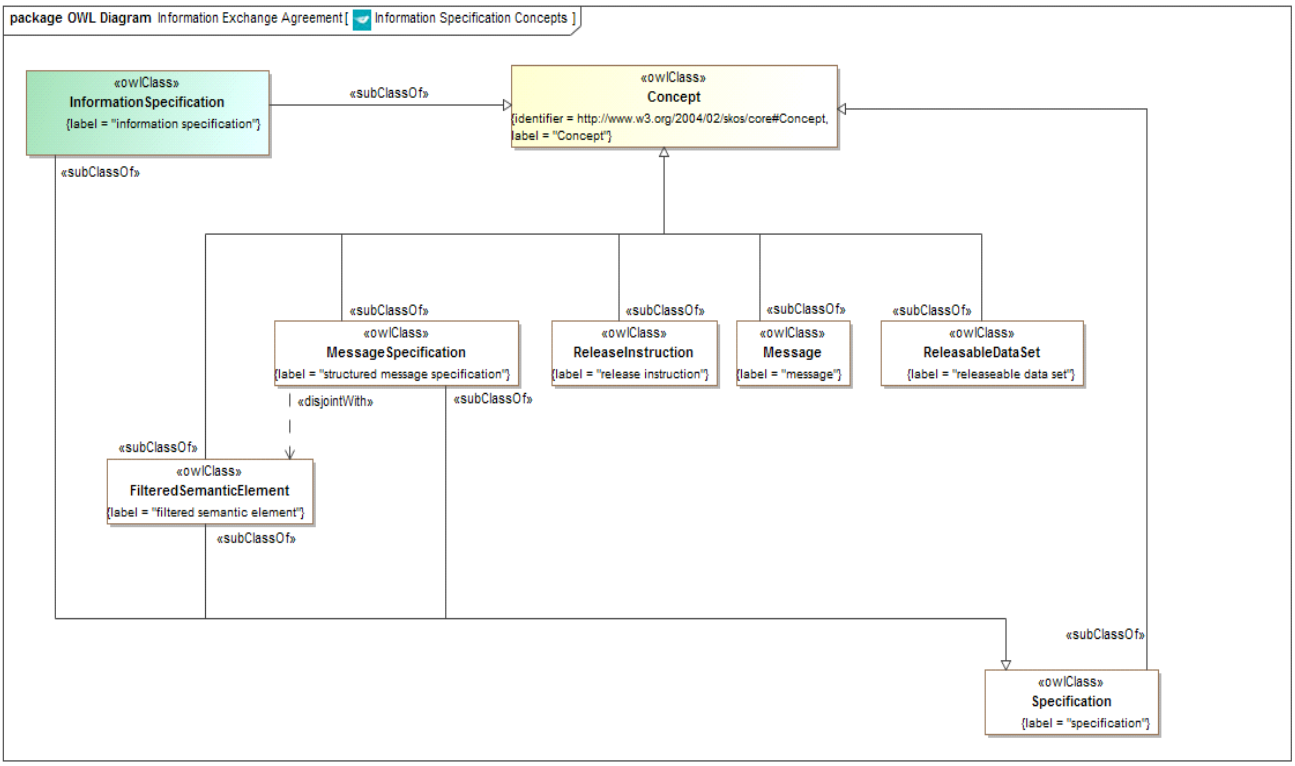

Figure 7.5 - Information Specification Concepts

7.2.4 Information Specification

The following figure illustrates the relationships between concepts for the expression of rules that bind InformationElements to an Information Exchange Agreement.

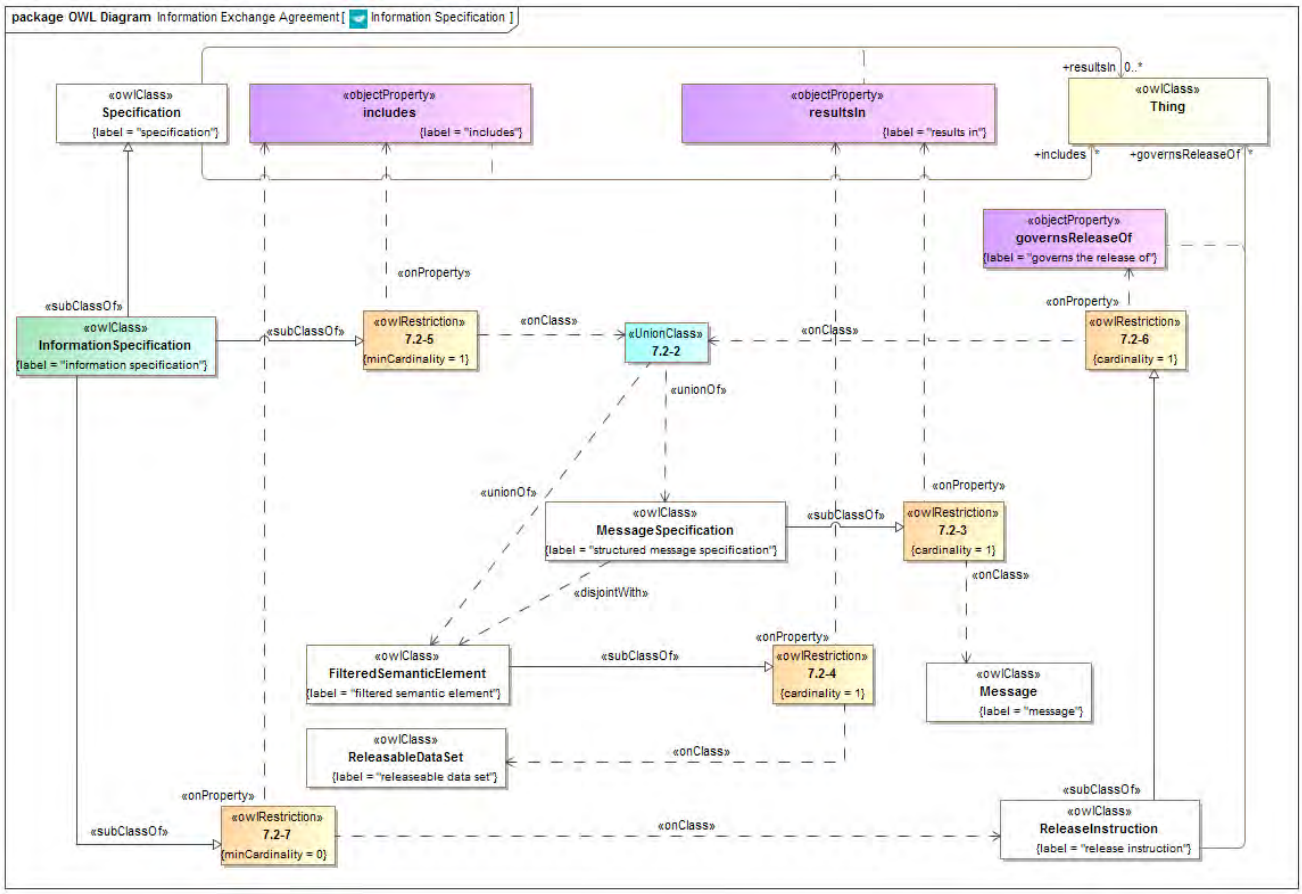

Figure 7.6 - Information Specification

The following rules apply to the Information Specification:

1. MessageSpecification resultsIn one and only one Message.
2. FilteredSemanticElement resultsIn one and only one ReleasableDataSet.
3. InformationSpecification includes at least one of (FilteredSemanticElement or MessageSpecification).
4. ReleaseInstruction governsReleaseOf one and only one of (FilteredSemanticElement or MessageSpecification).
5. InformationSpecification includes an optional set of ReleaseInstruction.

* The name used to identify the rules is derived from the restriction encompassed by the rule.

7.3 CP-1 Information Payload Specification Concepts

This sub clause identifies the concepts within the vocabulary that apply to Compliance Point 1. These concepts combine to enable the expression of rules governing the packaging and processing of DataElements and InformationElements involved in an information exchange. It enables the expression of the rules for assembly patterns which align user policy to a specific Information Domain.

Note – CP-1 is mandatory for all compliance points.

7.3.1 Filtered Semantic Element Concepts

The FilteredSemanticElement identifies concepts within the Vocabulary that combine to define rules that assign run-time configurable filters to SemanticElements. The addition of run-time configurable filters provided users the ability to tailor the assembly of data for a specific recipient during operations.

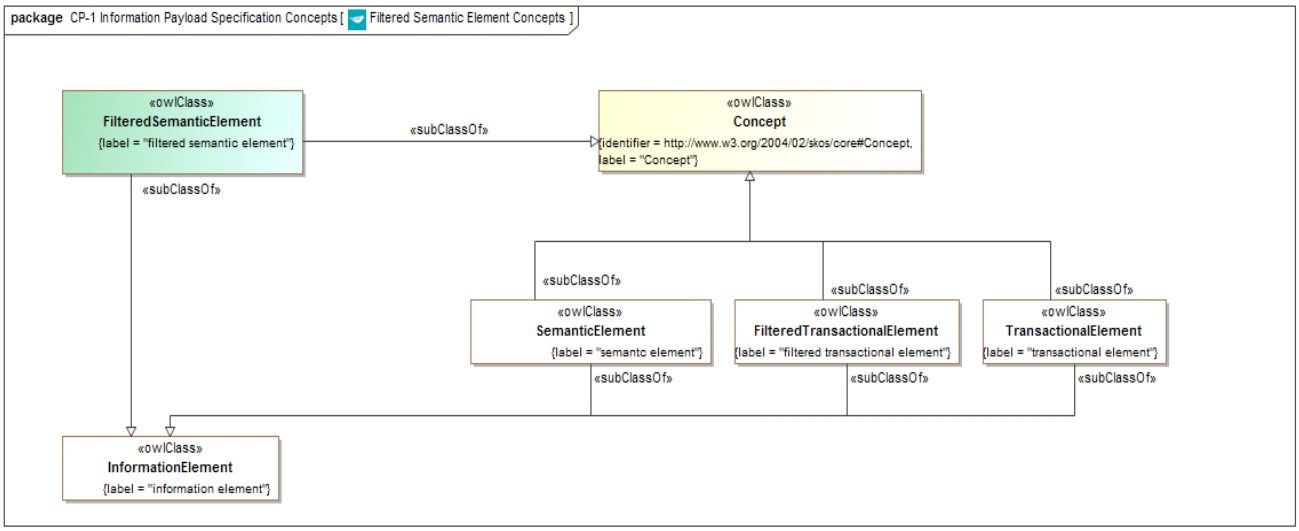

Figure 7.7 - Filtered Semantic Element Concepts

7.3.2 Filtered Semantic Element

The following figure illustrates the relationships between concepts used in the expression of rules that identify which FilteredTransactionals are used to identify the Attributes that are filterable in the run-time environment. It is the FilteredTransactional that identifies the filterable Attributes. The Filtered Semantic aligns a specific SemanticElement to its runtime filters.

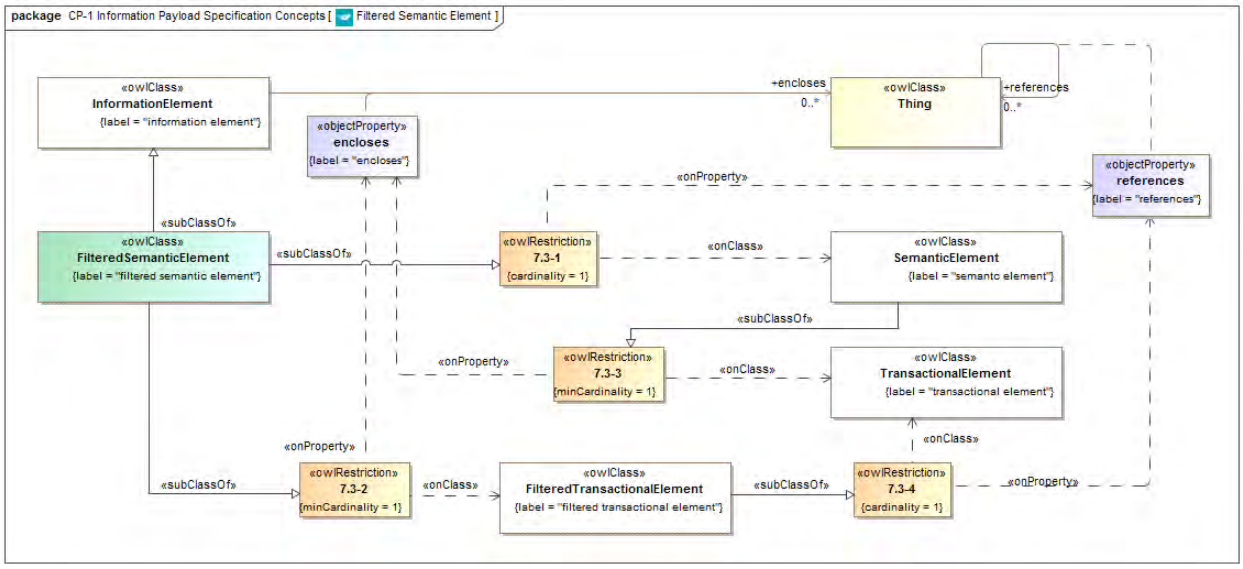

Figure 7.8 - Filtered Semantic Element

The following rules apply to the Filtered Semantic Element:

1. FilteredSemanticElement references one and only one SemanticElement.
2. FilteredSemanticElement encloses at least one FilteredTransactionalElement.
3. SemanticElement encloses at least one TransactionalElement.
4. FilteredTransactionalElement references one and only one TransactionalElement.

* The name used to identify the rules is derived from the restriction encompassed by the rule.

7.3.3 Filtered Transactional Element Concepts

The FilteredTransactionalElement specifies concepts within the Vocabulary that combine to express rules that assign the filters to its rules and attributes.

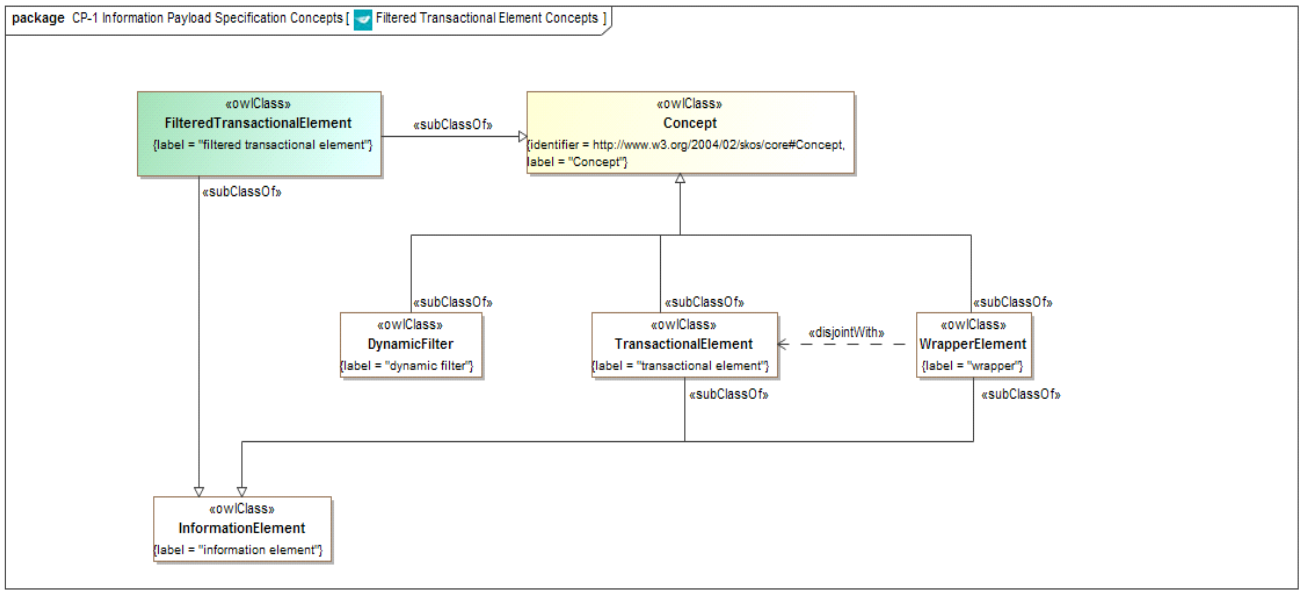

Figure 7.9 - Filtered Transactional Element Concepts

7.3.4 Filtered Transactional Element

The following figure illustrates the relationships between concepts used in the expression of rules that align specific filters to its FilterRules and the Attributes to be used.

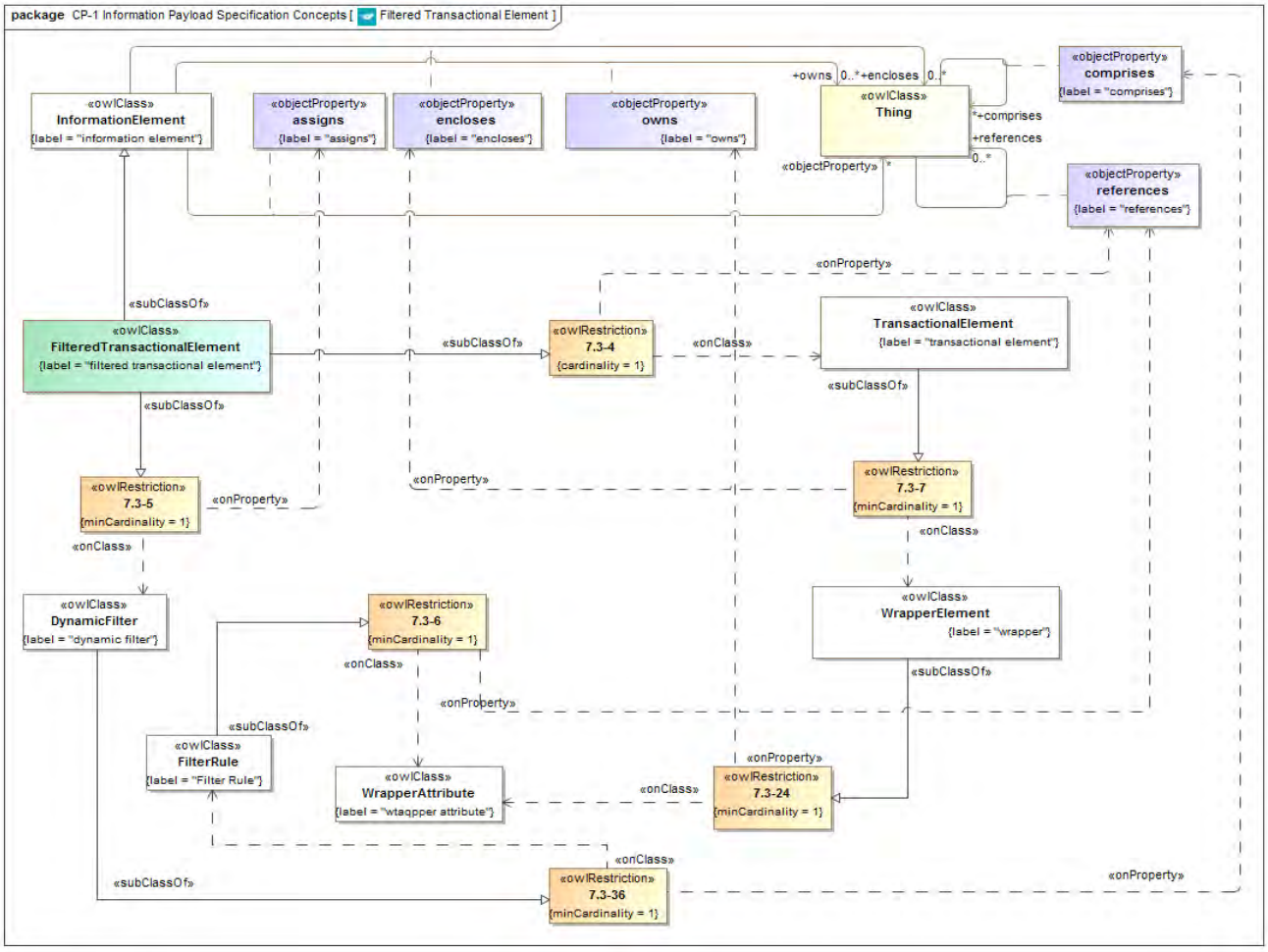

Figure 7.10 - Filtered Transactional Element

The following rules apply to the Filtered Transactional Element:

1. WrapperElement owns at least one WrapperAttribute.
2. DynamicFilter comprises at least one FilterRule.
3. FilteredTransactionalElement references one and only one TransactionalElement.
4. FilteredTransactionalElement assigns at least one DynamicFilter.
5. FilterRule references at least one WrapperAttribute.
6. TransactionalElement encloses at least one WrapperElement.

* The name used to identify the rules is derived from the restriction encompassed by the rule.

7.3.5 Semantic Element Concepts

The TransactionalElement identifies the concepts within the Vocabulary that enable the expression of rules that align TransactionalElements to a SemanticElement. SemanticElement use the building blocks (TransactionalElements) to aggregate rules sets that enable the assembly and processing of specified messages by an information exchange agreement.

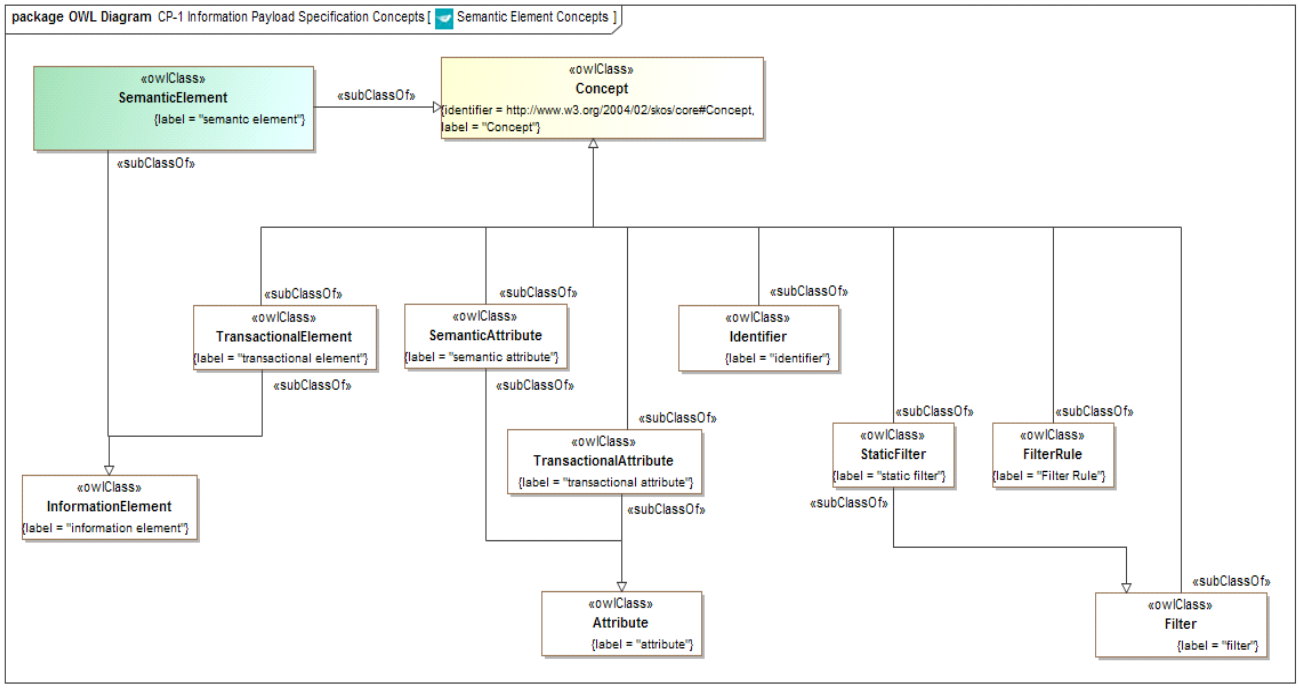

Figure 7.11 - Semantic Element Concepts

7.3.6 Semantic Element (Foundation)

The following figure illustrates the relationships between concepts that are used in the expression of rules that align TransactionalElements to a SemanticElement. The TransactionalElements are the building blocks (or re-usable patterns) for assembling and processing the data associated with InformationElements specified in an information exchange agreement. The SemanticElements also provides the rules that identify which TransactionalElement contains (or holds) the element (key or identifier) that identifies the specific data instances to be assembled into a releasable the dataset.

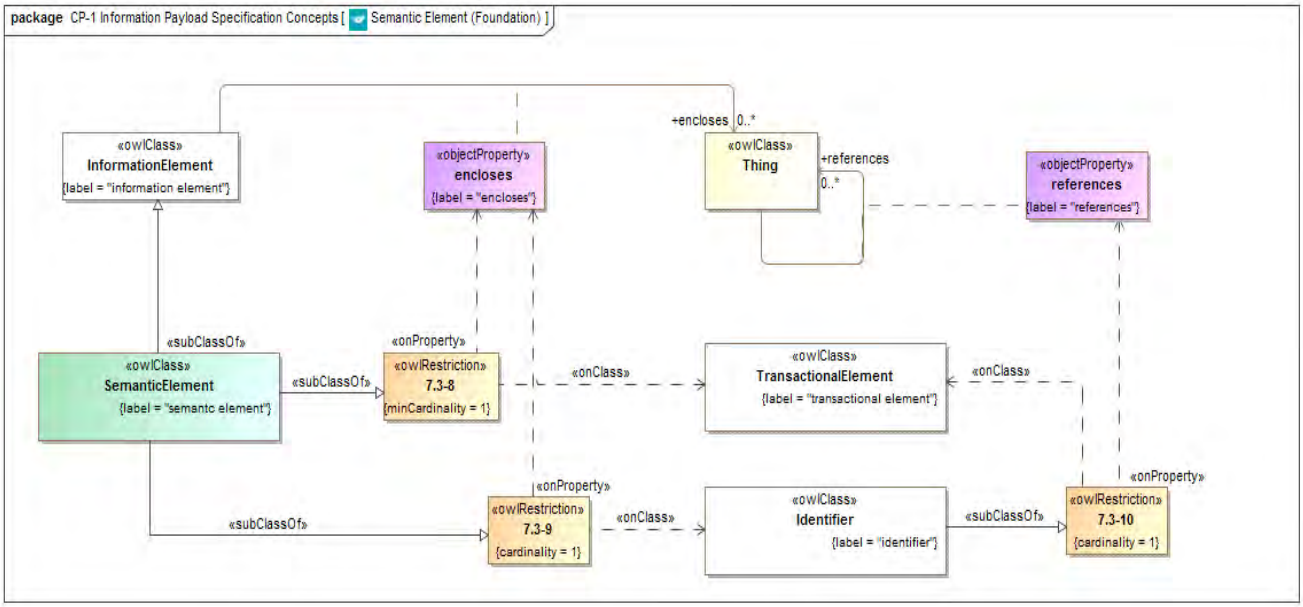

Figure 7.12 - Semantic Element (Foundation)

The following rules apply to the Semantic Element (Foundation):

1. Identifier references one and only one TransactionalElement.
2. SemanticElement encloses at least one TransactionalElement.
3. SemanticElement encloses one and only one Identifier.

* The name used to identify the rules is derived from the restriction encompassed by the rule.

7.3.7 Semantic Element (Attribution)

The following figure illustrates the relationships between concepts used in the expression of rules that explicitly assign TransactionalAttributes to a SemanticElement's Attributes. The ability to explicitly specify these associations permits:

1. The selective aggregation of attributes or selective redaction of data elements.
2. The translation between Community, business, logical, and physical name spaces.

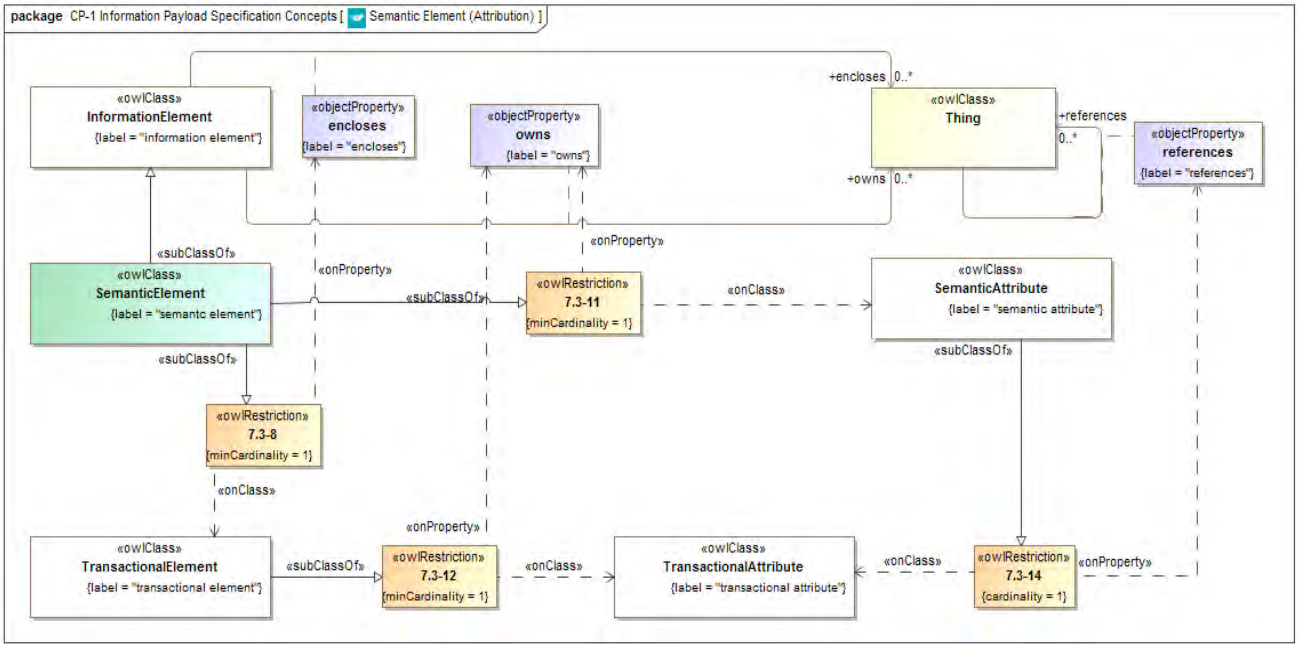

Figure 7.13 - Semantic Element (Attribution)

The following rules apply to the Semantic Element (Attribution):

1. SemanticElement owns at least one SemanticAttribute.
2. TransactionalElement owns at least one TransactionalAttribute.
3. SemanticAttribute references one and only one TransactionalAttribute.
4. SemanticElement encloses at least one TransactionalElement.

* The name used to identify the rules is derived from the restriction encompassed by the rule.

7.3.8 Semantic Element (Static Filters)

The following figure illustrates the relationships between concepts used in the expression of rules that define StaticFilters within a SemanticElement. Information Elements defined by the TransactionalElements are redacted based values of one or more Attribute at run-time. StaticFilters in the SemanticElement (Pattern) are unalterable at runtime.

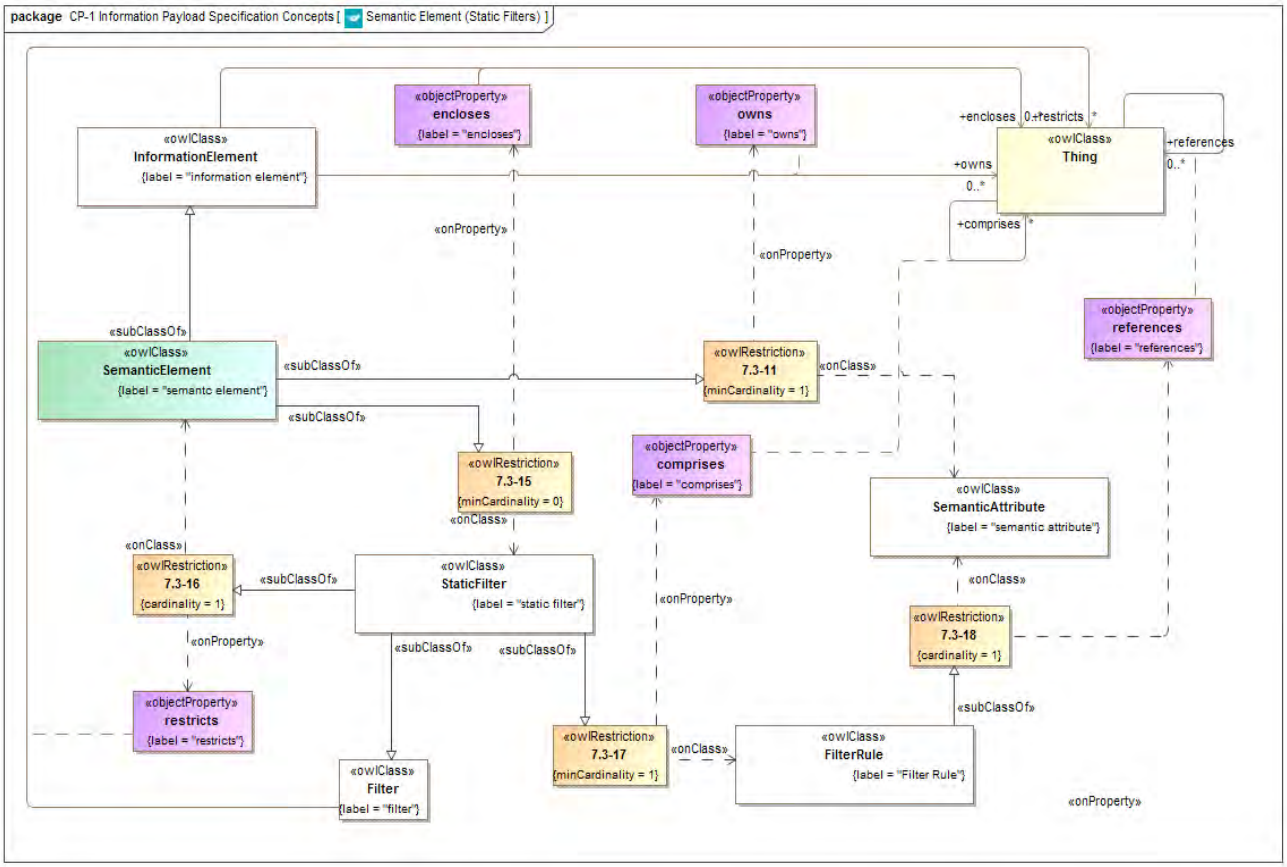

Figure 7.14 - Semantic Element (Static Filters)

The following rules apply to the Semantic Element (Static Filters):

1. SemanticElement owns at least one SemanticAttribute.
2. SemanticElement encloses an optional set of StaticFilter.
3. StaticFilter restricts one and only one SemanticElement.
4. StaticFilter comprises at least one FilterRule.
5. FilterRule references one and only one SemanticAttribute.

* The name used to identify the rules is derived from the restriction encompassed by the rule.

7.3.9 Transactional Element Concepts

The TransactionalElement identifies the concepts within the Vocabulary that enable the expression of rules for the assembly (aggregation, transformation, tagging/marking, and redaction) and processing (parsing, validation, transformation, and marshaling) of data and information elements.

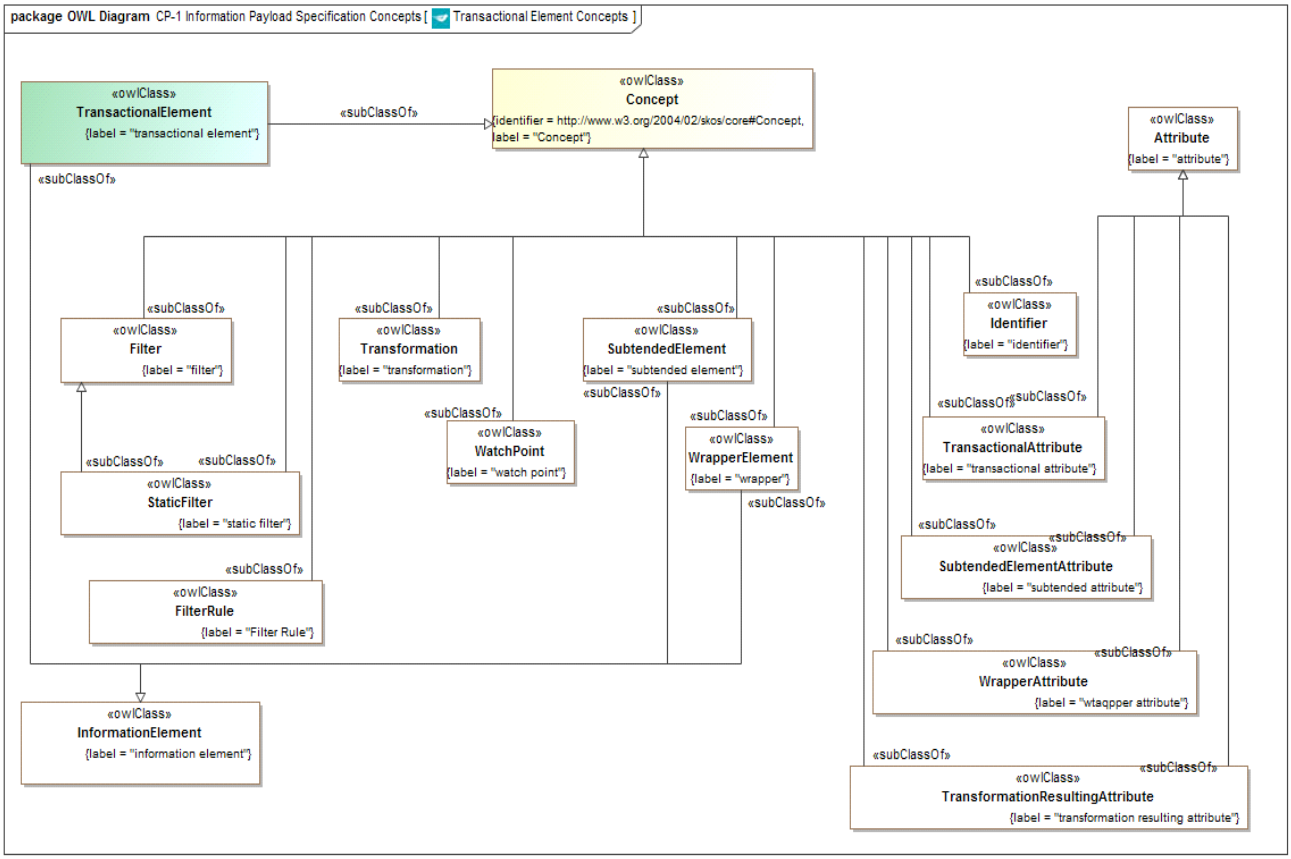

Figure 7.15 - Transactional Element Concepts

7.3.10 Transactional Element (Foundation)

The following figure illustrates the relationships between concepts used in the expression of rules that identify which TransactionalElements are used in the assembly and processing of releasable data for a specific information exchange agreement. The TransactionalElements are the building blocks of SemanticElements.

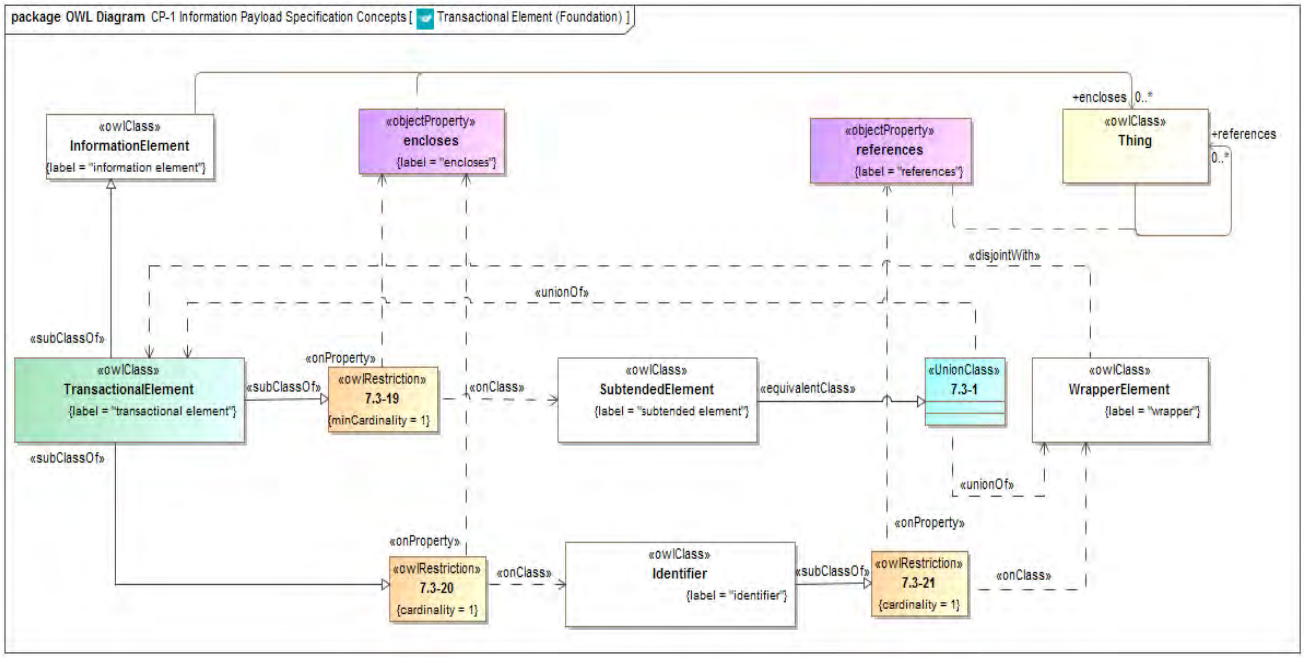

Figure 7.16 - Transaction Element (Foundation)

The following rules apply to the Transactional Element (Foundation):

1. TransactionalElement encloses at least one SubtendedElement.
2. TransactionalElement encloses one and only one Identifier.
3. Identifier references one and only one WrapperElement.

* The name used to identify the rules is derived from the restriction encompassed by the rule.

7.3.11 Transactional Element (Attribution)

The following figure illustrates the relationships between concepts used in the expression of rules that specifically assign Wrapper and TransactionalAttributes to the enclosing TransactionalAttributes includes in the releasable dataset. The ability to explicitly specify these associations permits:

1. The selective aggregation of attributes or selective redaction of data elements.
2. The translation between logical and physical name spaces.

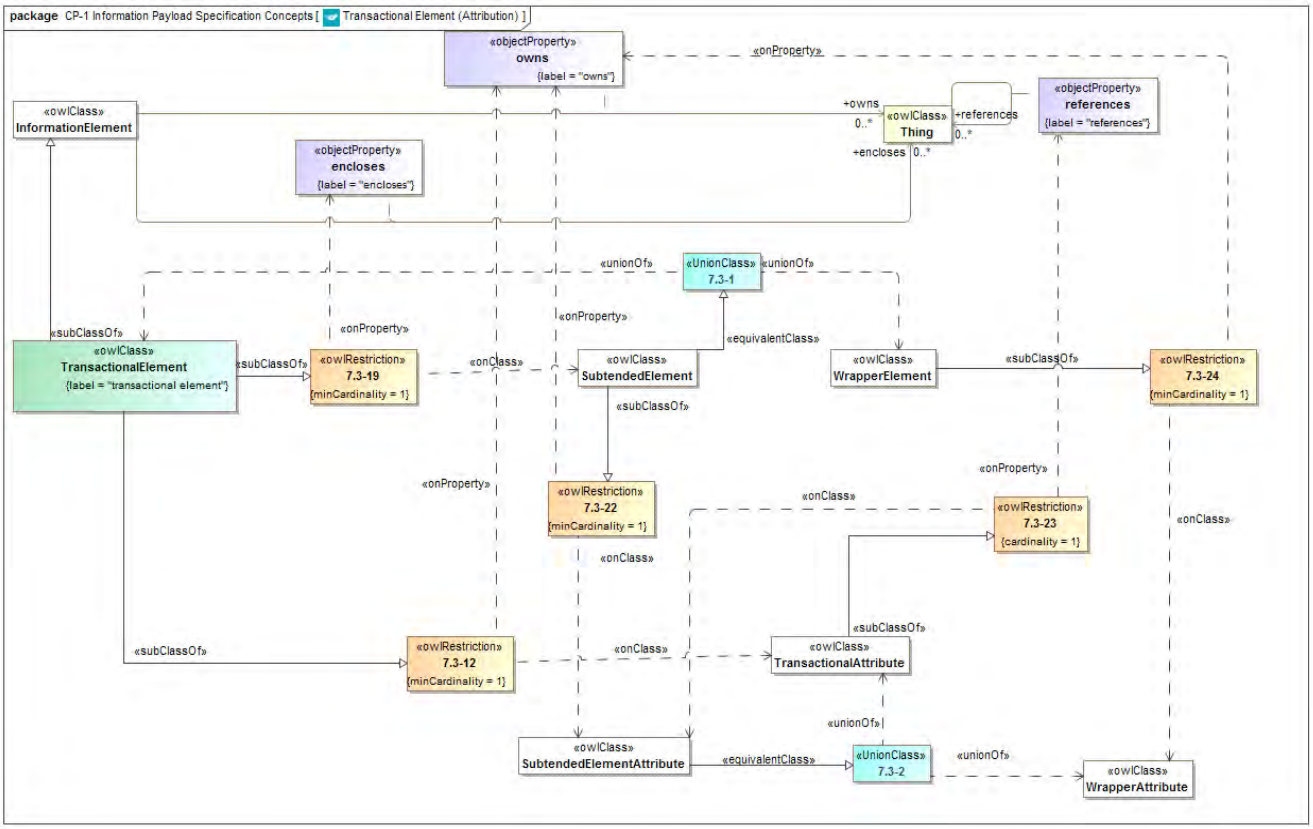

Figure 7.17 - Transactional Element (Attribution)

The following rules apply to the Transactional Element (Attribution):

1. TransactionalElement owns at least one TransactionalAttribute.
2. TransactionalElement encloses at least one SubtendedElement.
3. SubtendedElement owns at least one SubtendedElementAttribute.
4. TransactionalAttribute references one and only one SubtendedElementAttribute.
5. WrapperElement owns at least one WrapperAttribute.

* The name used to identify the rules is derived from the restriction encompassed by the rule.

7.3.12 Transactional Element (Static Filters)

The following figure illustrates the relationships between concepts used in the expression of rules that define StaticFilters within a TransactionalElement. Information Elements defined by the TransactionalElements or WrapperElements are redacted based on values of one or more Attributes at run-time. StaticFilters in the SemanticElement (Pattern) unalterable at runtime.

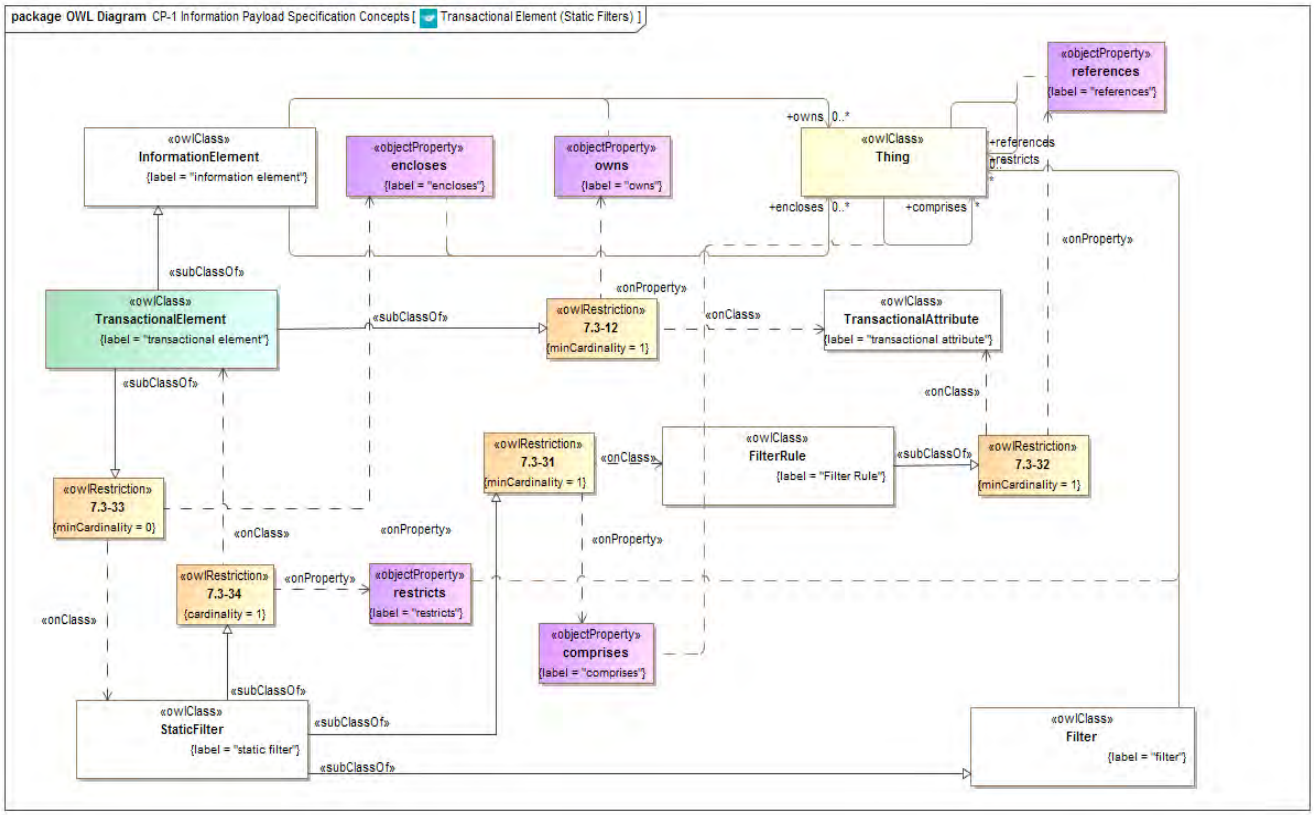

Figure 7.18 - Transactional Element (Static Filters)

The following rules apply to the Transactional Element (Static Filters):

1. TransactionalElement owns at least one TransactionalAttribute.
2. StaticFilter comprises at least one FilterRule.
3. FilterRule references at least one TransactionalAttribute.
4. TransactionalElement encloses an optional set of StaticFilter.
5. StaticFilter restricts one and only one TransactionalElement.

* The name used to identify the rules is derived from the restriction encompassed by the rule.

7.3.13 Transactional Element (Transformation)

The following figure illustrates the relationships between concepts used in the expression of rules that direct the transformation of SubtendedElementAttribute to conform to the data domain of information exchange agreement. All data transformations are specified as part of TransactionalElements.

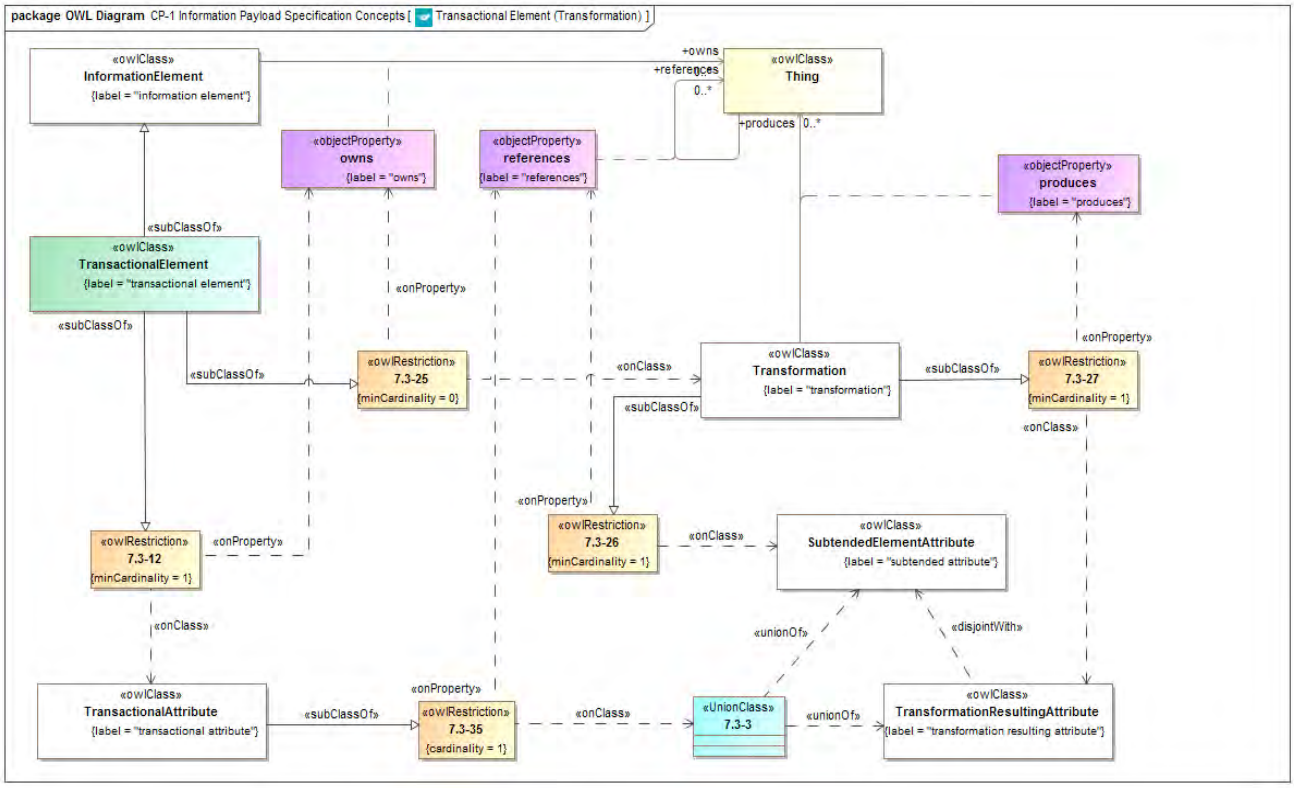

Figure 7.19 - Transactional Element (Transformation)

The following rules apply to the Transactional Element (Transformation):

1. TransactionalElement owns at least one TransactionalAttribute.
2. TransactionalElement owns an optional set of Transformation.
3. Transformation references at least one SubtendedElementAttribute.
4. Transformation produces at least one TransformationResultingAttribute.
5. TransactionalAttribute references one and only one of (SubtendedElementAttributy or TransformationResultingAttribute).

* The name used to identify the rules is derived from the restriction encompassed by the rule.

7.3.14 Transactional Element (Watchpoint)

The following figure illustrates the relationships between concepts used in the expression of rules that establish the identity of the subtended elements where changes in their data trigger the assembly of the TransactionalElement and the Semantics to which they are enclosed.

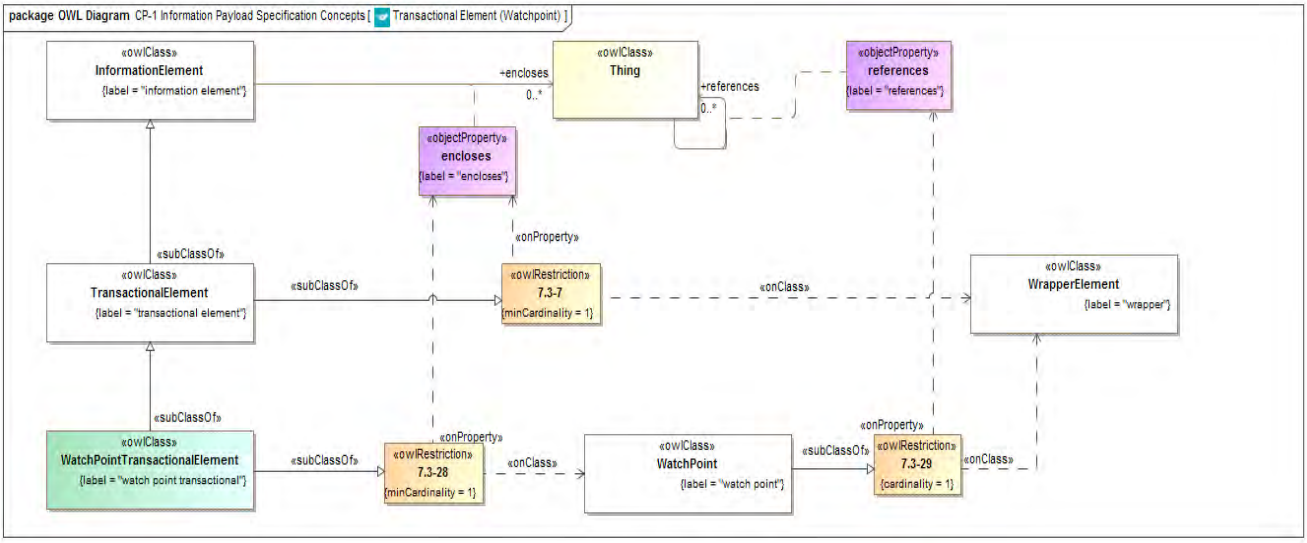

Figure 7.20 - Transactional Element (Watchpoint)

The following rules apply to the Transactional Element (Watchpoint):

1. WatchPointTransactionalElement encloses at least one WatchPoint.
2. WatchPoint references one and only one WrapperElement.
3. TransactionalElement encloses at least one WrapperElement.

* The name used to identify the rules is derived from the restriction encompassed by the rule.

7.3.15 Wrapper Element Concepts

The WrapperElements identify the concepts within the Vocabulary that enable the expression of rules that link Transactional Elements to the source data definitions. This mapping is performed through the WrapperElements that wrap the source DataElements.

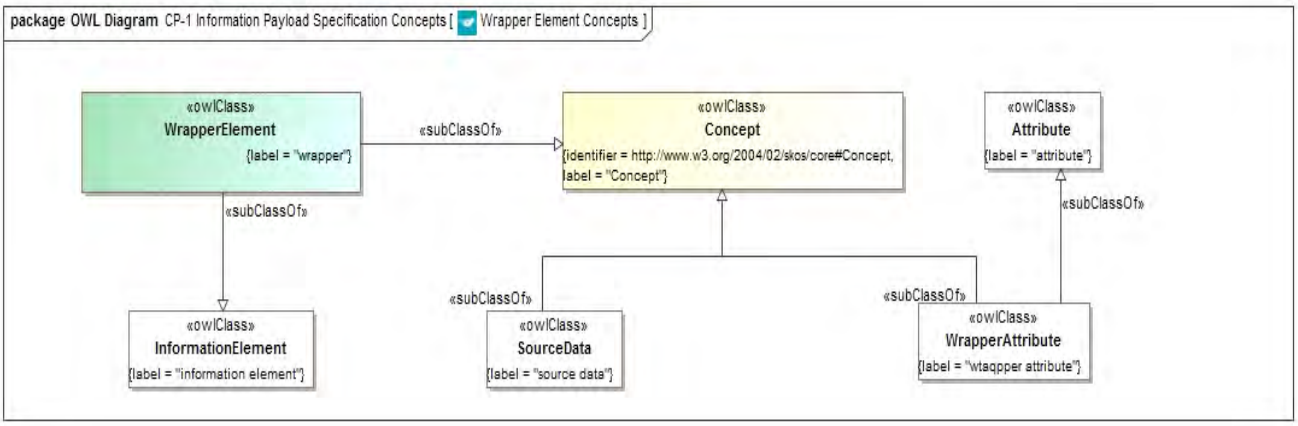

Figure 7.21 - Wrapper Element Concepts

7.3.16 Wrapper Element

The following figure, Wrapper, illustrates the basic object properties and restrictions associated with the expression of the rule governing the association between the data sources and the Wrapper. The Wrapper connects the physical definition of the data to the object domain described in the transactional patterns.

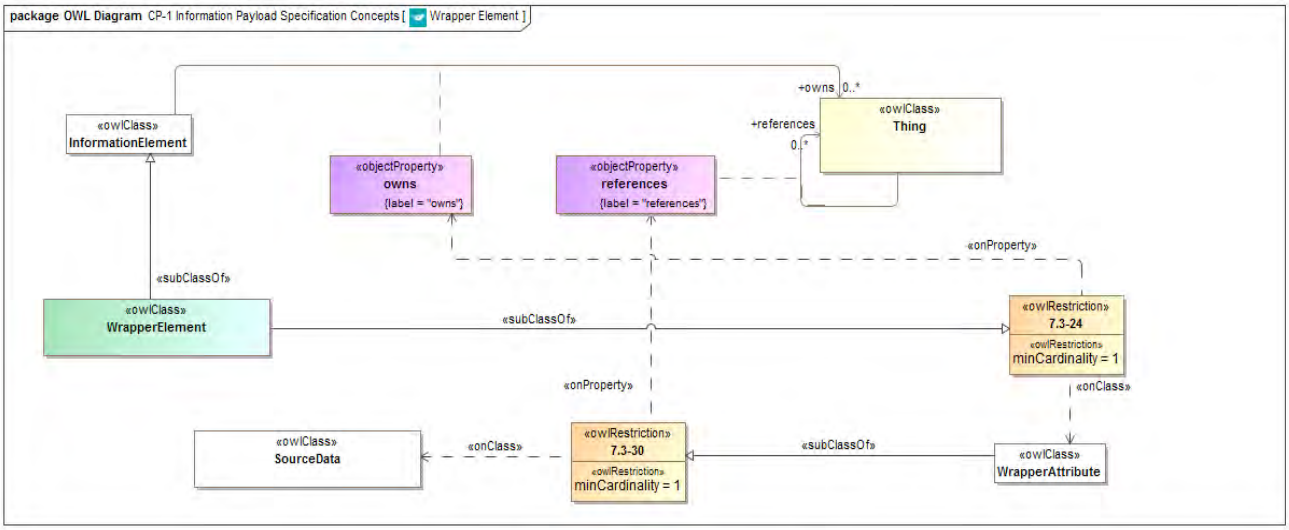

Figure 7.22 - Wrapper Element

The following rules apply to the Wrapper Element:

1. WrapperElement owns at least one WrapperAttribute.
2. WrapperAttribute references at least one SourceData.

* The name used to identify the rules is derived from the restriction encompassed by the rule.

7.4 CP-2a Basic Message Specification Concepts

This sub clause defines the concepts within the Vocabulary that are used to express the rules for the assembly of a basic Message comprising MessageMetadata, one Payload and one Attachment. Each of CP-2b and CP-2c extend this basic pattern.

7.4.1 Message Specification Concepts

The Message Specification identifies concepts within the Vocabulary that combine to express the rules used to specify the assembly of a basic Message structure comprising MessageMetadata, one Payload and one Attachment.

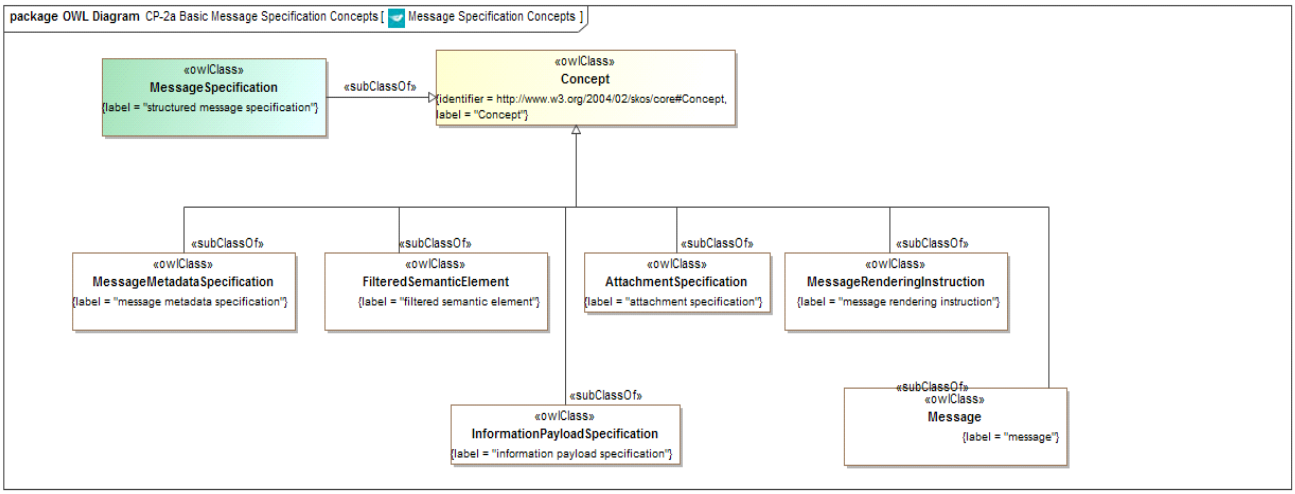

Figure 7.23 - Message Specification Concepts

7.4.2 Message Specification

The following figure illustrates the relationships between concepts used in the expression of rules for the assembly of a basic Message structure.

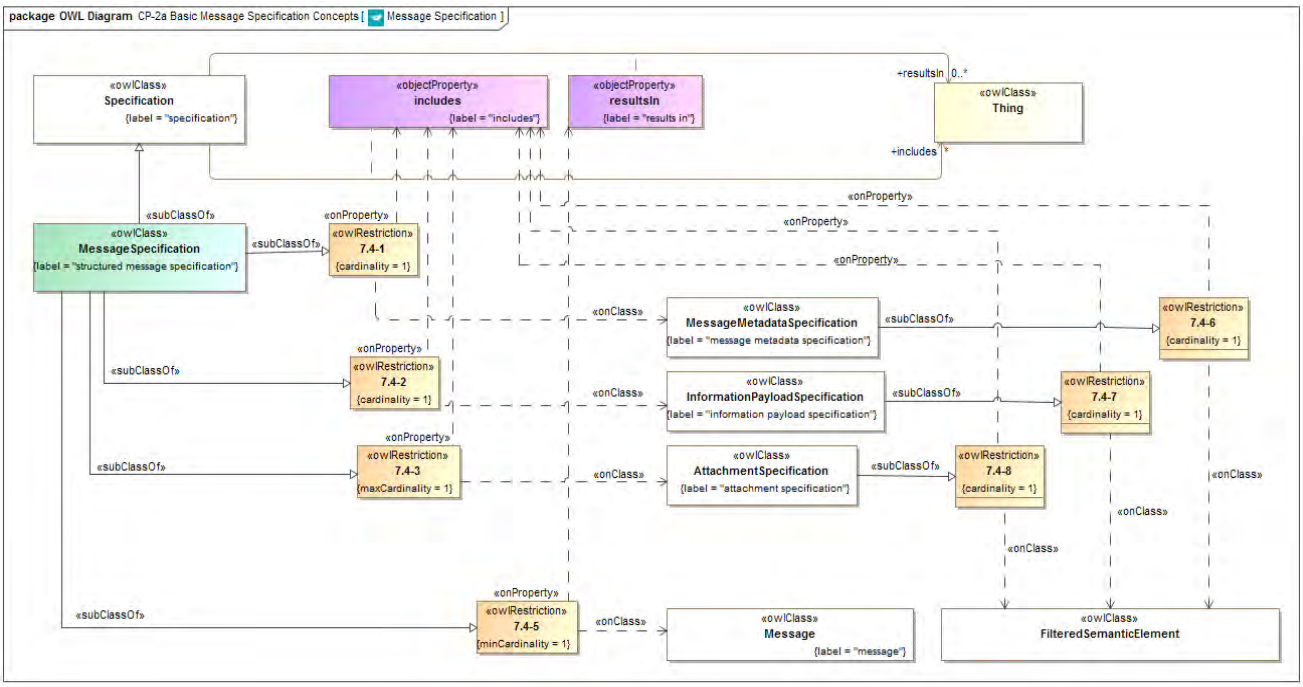

Figure 7.24 - Message Specification

The following rules apply to the Message Specification:

1. MessageSpecification includes one and only one MessageMetadataSpecification.
2. MessageSpecification includes one and only one InformationPayloadSpecification.
3. MessageSpecification includes a maximum one AttachmentSpecification.
4. MessageSpecification resultsIn at least one Message.
5. MessageMetadataSpecification includes one and only one FilteredSemanticElement.
6. InformationPayloadSpecification includes one and only one FilteredSemanticElement.
7. AttachmentSpecification includes one and only one FilteredSemanticElement.

* The name used to identify the rules is derived from the restriction encompassed by the rule.

7.4.3 Message Specification (continued)

The following figure extends the relationships between concepts used in the expression of rules for assembly of a Basic Message structure.

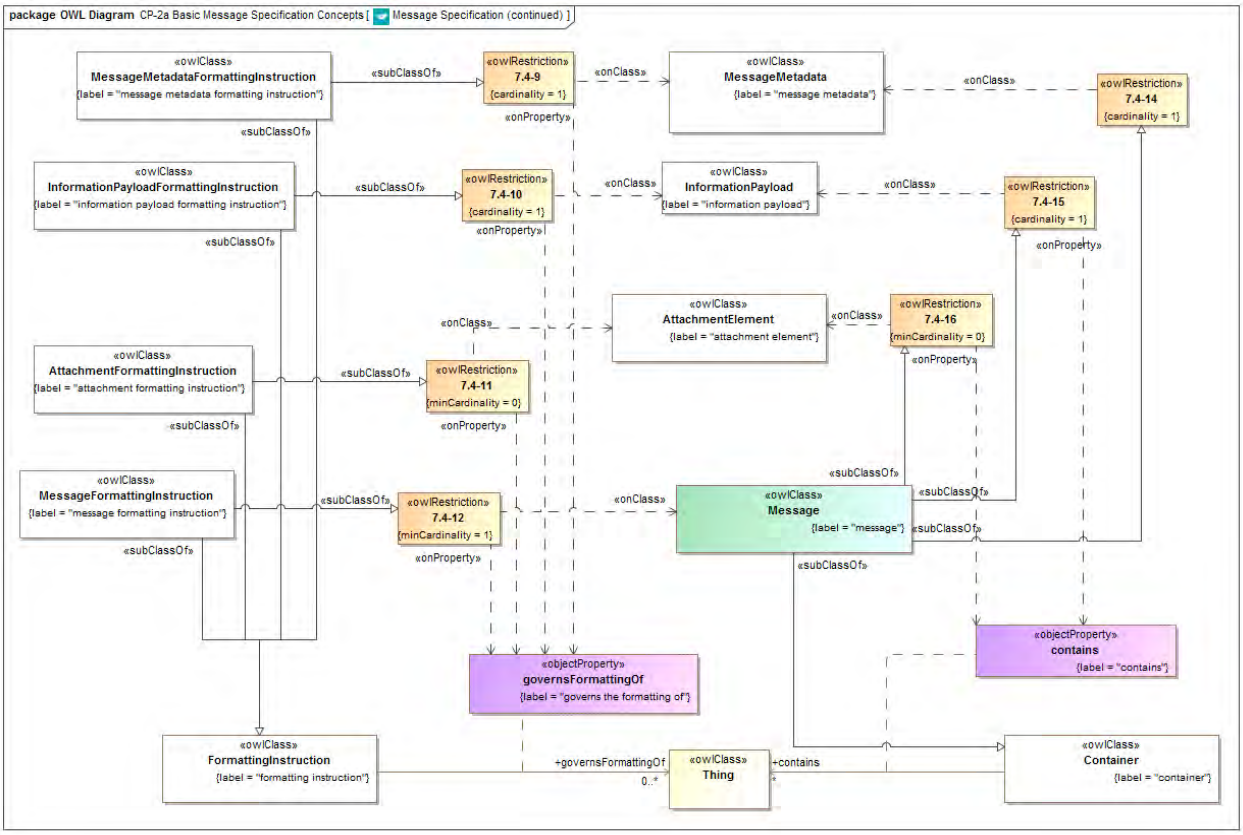

Figure 7.25 - Message Specification (continued)

The following rules apply to the Message Specification (continued):

1. InformationPayloadFormattingInstruction governsFormattingOf one and only one InformationPayload.
2. AttachmentFormattingInstruction governsFormattingOf an optional set of AttachmentElement.
3. MessageFormattingInstruction governsFormattingOf at least one Message.
4. Message contains one and only one MessageMetadata.
5. Message contains one and only one InformationPayload.
6. Message contains an optional set of AttachmentElement.
7. MessageMetadataFormattingInstruction governsFormattingOf one and only one MessageMetadata.

* The name used to identify the rules is derived from the restriction encompassed by the rule.

7.4.4 Message Metadata Specification Concepts

The Attachment Specification identifies concepts within the Vocabulary that combine to express the rules for assembly of message Metadata tags and attaching them to a message.

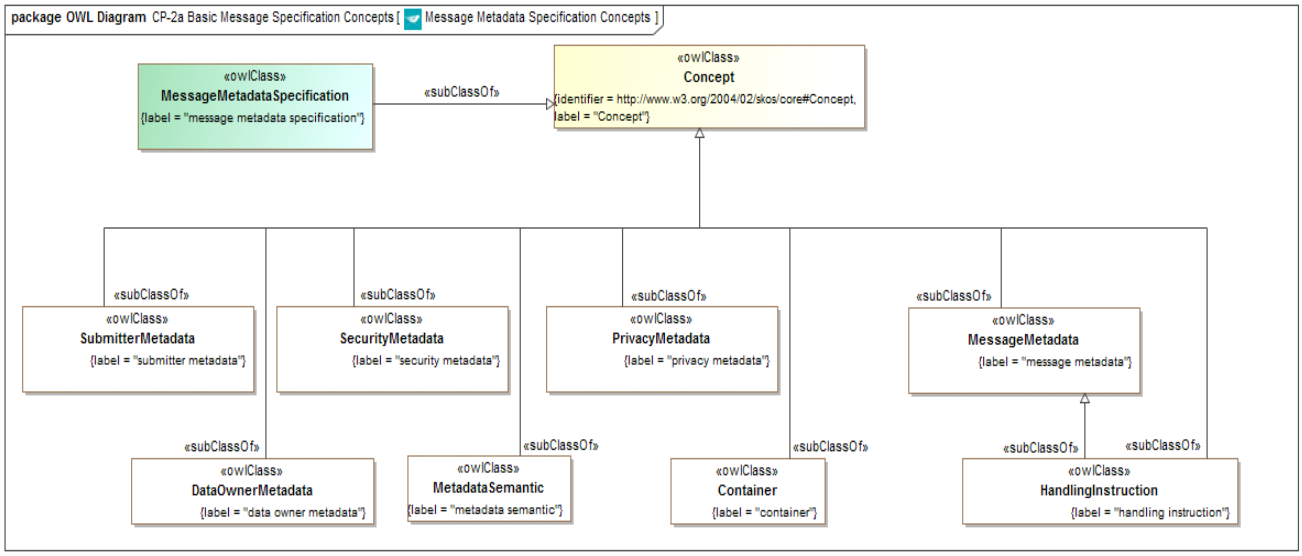

Figure 7.26 - Message Metadata Specification Concepts

7.4.5 Message Metadata Specification

The following figure illustrates the relationships between concepts used in the expression of rules for the assembly of Metadata and attaching them to a message.

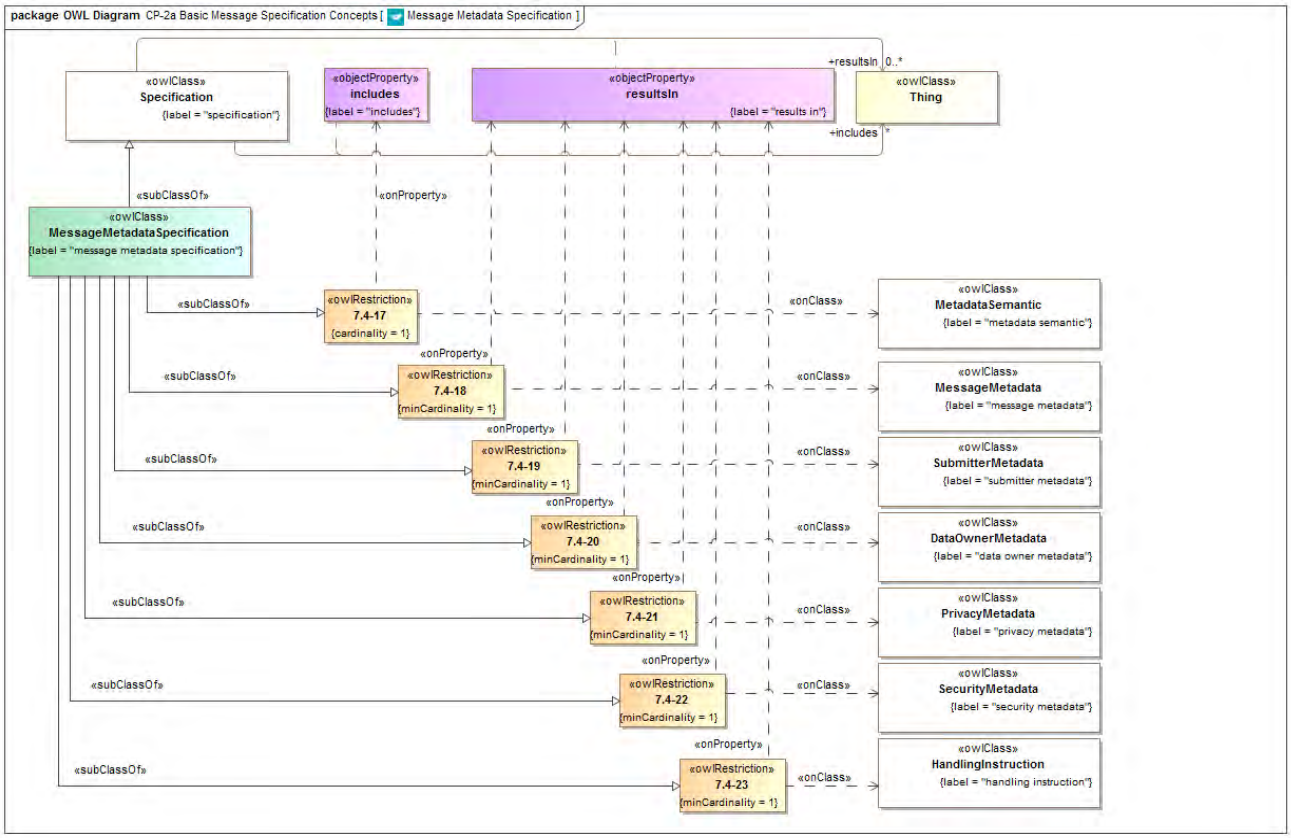

Figure 7.27 - Message Metadata Specification

The following rules apply to the Message Metadata Specification:

1. MessageMetadataSpecification includes one and only one MetadataSemantic.
2. MessageMetadataSpecification resultsIn at least one MessageMetadata.
3. MessageMetadataSpecification resultsIn at least one SubmitterMetadata.
4. MessageMetadataSpecification resultsIn at least one DataOwnerMetadata.
5. MessageMetadataSpecification resultsIn at least one PrivacyMetadata.
6. MessageMetadataSpecification resultsIn at least one SecurityMetadata.
7. MessageMetadataSpecification resultsIn at least one HandlingInstruction.

* The name used to identify the rules is derived from the restriction encompassed by the rule.

7.4.6 Message Metadata Specification (continued)

The following figure extends the relationships between concepts used in the expression of rules for assembly of Metadata tags attaching them to a message.

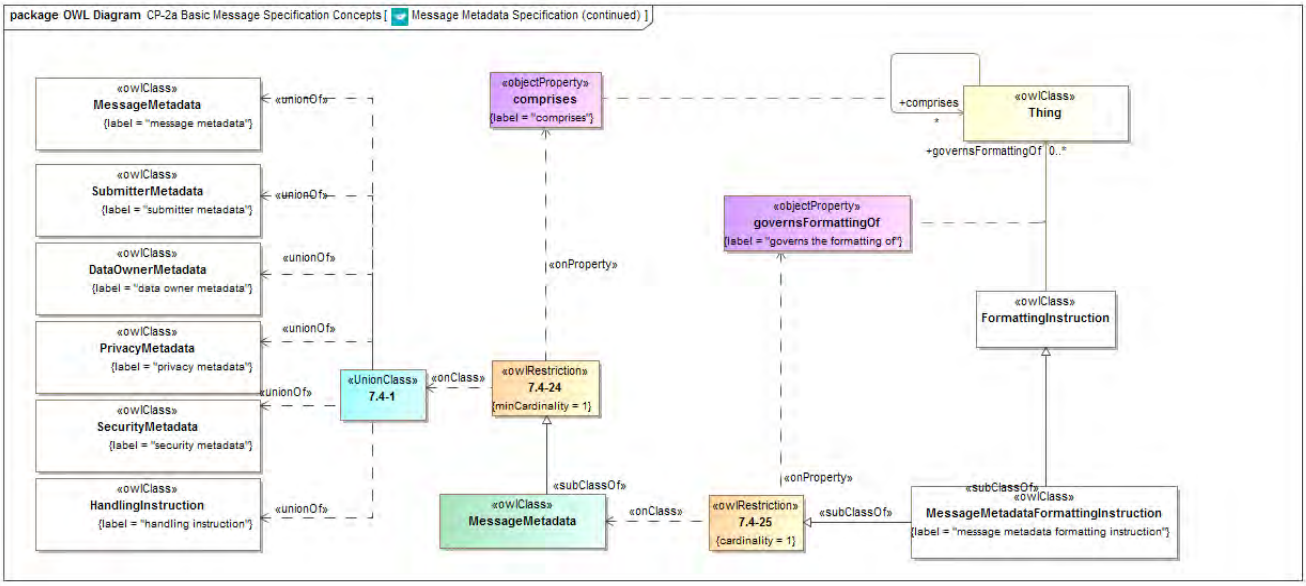

Figure 7.28 - Message Metadata Specification (continued)

The following rules apply to the Message Metadata Specification (continued):

1. MessageMetadata comprises at least one of (MessageMetadata, SubmitterMetadata, DataOwnerMetadata, PrivacyMetadata, SecurityMetadata, or HandlingInstruction).

2. MessageMetadataFormattingInstruction governsFormattingOf one and only one MessageMetadata.

* The name used to identify the rules is derived from the restriction encompassed by the rule.

7.4.7 Attachment Specification Concepts

The Attachment Specification identifies concepts within the Vocabulary that combine to express the rules used to specify the attachment to be included with a Message.

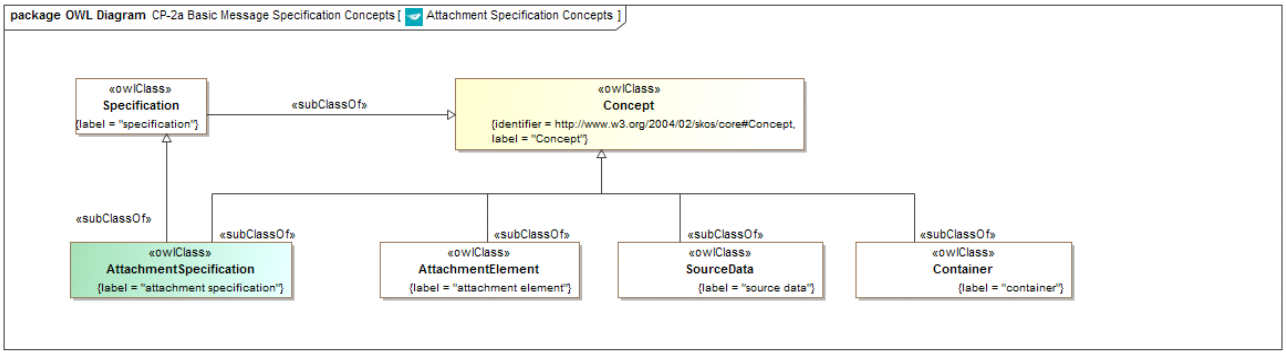

Figure 7.29 - Attachment Specification Concepts

7.4.8 Attachment Specification

The following figure illustrates the relationships between concepts used in the expression of rules for attaching unstructured data/information Attachments to a Message.

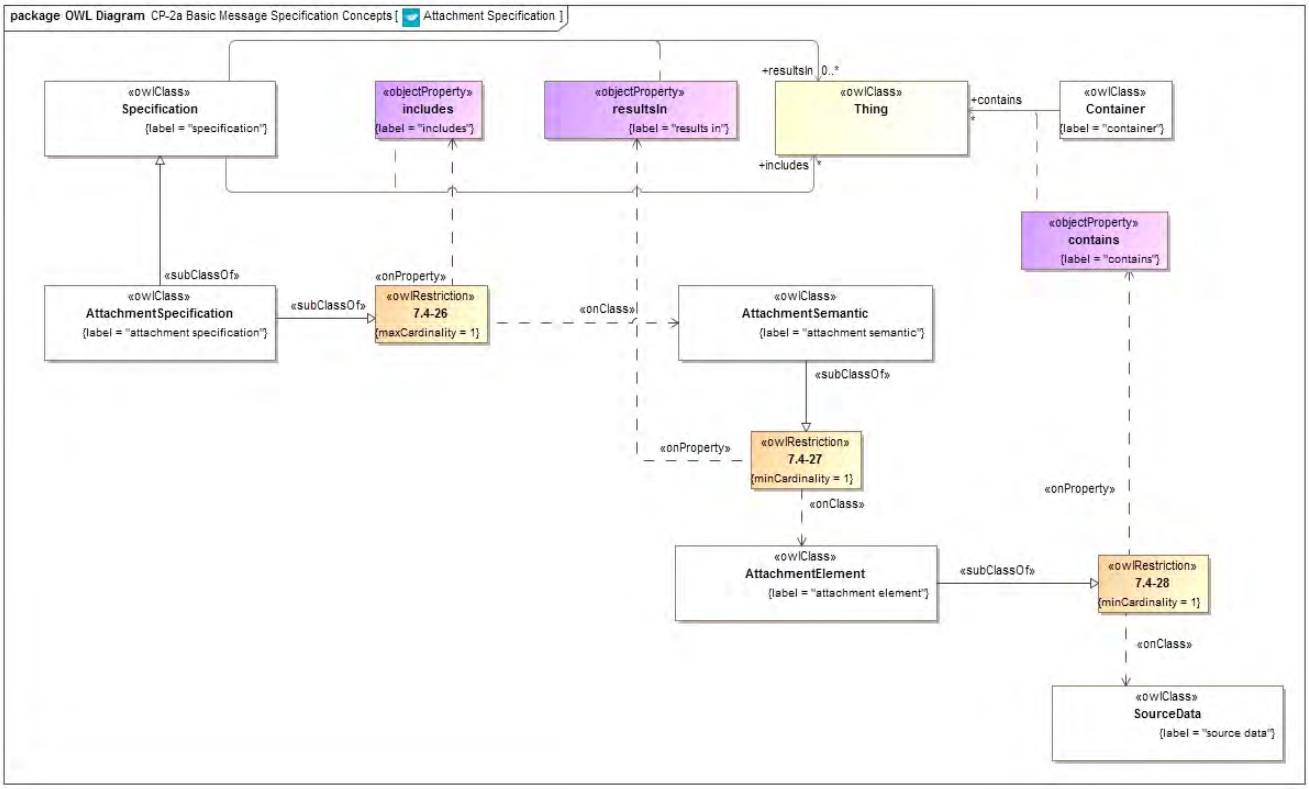

Figure 7.30 - Attachment Specification

The following rules apply to the Attachment Specification:

1. AttachmentSpecification includes a maximum one AttachmentSemantic.
2. AttachmentSemantic resultsIn at least one AttachmentElement.
3. AttachmentElement contains at least one SourceData.

* The name used to identify the rules is derived from the restriction encompassed by the rule.

7.5 CP-2b Extended Message Specification Concepts

This sub clause defines the concepts within the Vocabulary that are used to express the rules for the assembly of a Message comprising MessageMetadata, one InformationPackage, and Multiple Attachments.

7.5.1 Message Specification Concepts

The MessageSpecification extends the concepts supported by the CP-2a and identifies concepts within the Vocabulary that combine to express the rules governing for assembling and processing a message structure comprising Metadata, one InformationPackage and multiple Attachments.

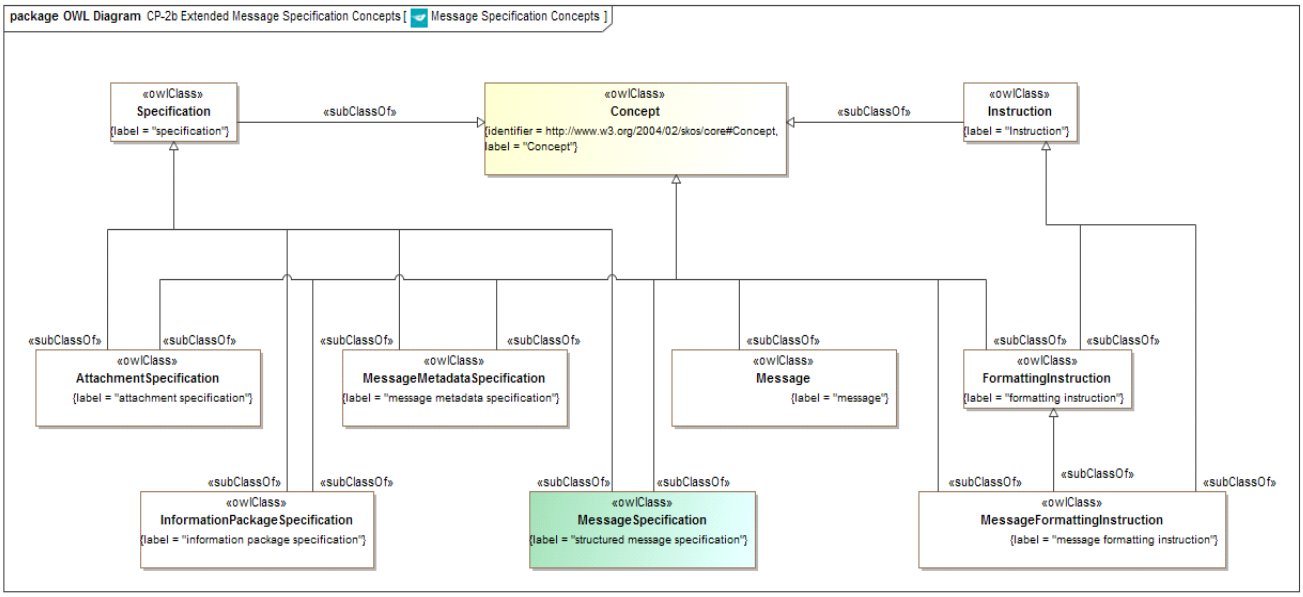

Figure 7.31 - Message Specification Concepts

7.5.2 Message Specification

The following figure illustrates the relationships between concepts used in the expression of rules for the assembly and processing of a message comprising Metadata, one InformationPackage and multiple Attachments.

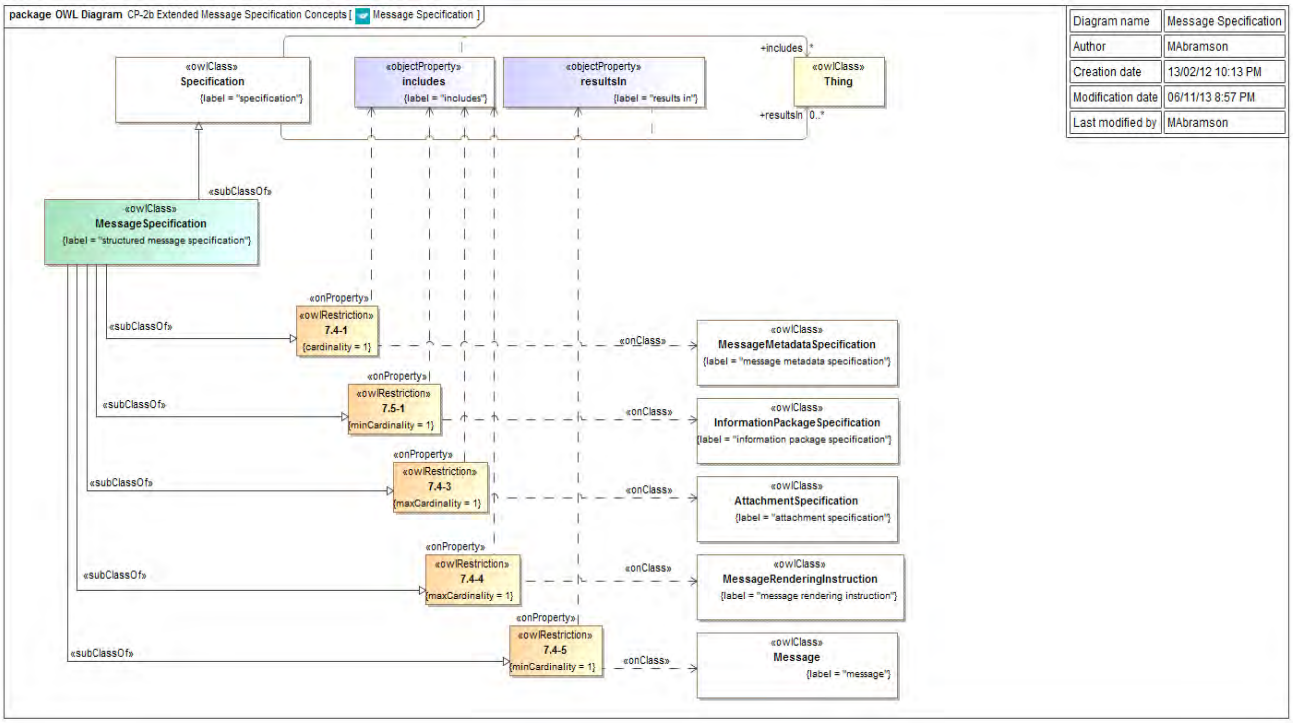

Figure 7.32 - Message Specification

The following rules apply to the Message Specification:

1. MessageSpecification includes one and only one MessageMetadataSpecification.
2. MessageSpecification includes a maximum one AttachmentSpecification.
3. MessageSpecification includes a maximum one MessageRenderingInstruction.
4. MessageSpecification resultsIn at least one Message.
5. MessageSpecification includes at least one InformationPackageSpecification.

* The name used to identify the rules is derived from the restriction encompassed by the rule.

7.5.3 Message Specification (continued)

The following figure illustrates the relationships between concepts used in the expression of rules for the assembly and processing of a message; specifically the assignment formating instructions for each of the message elements.

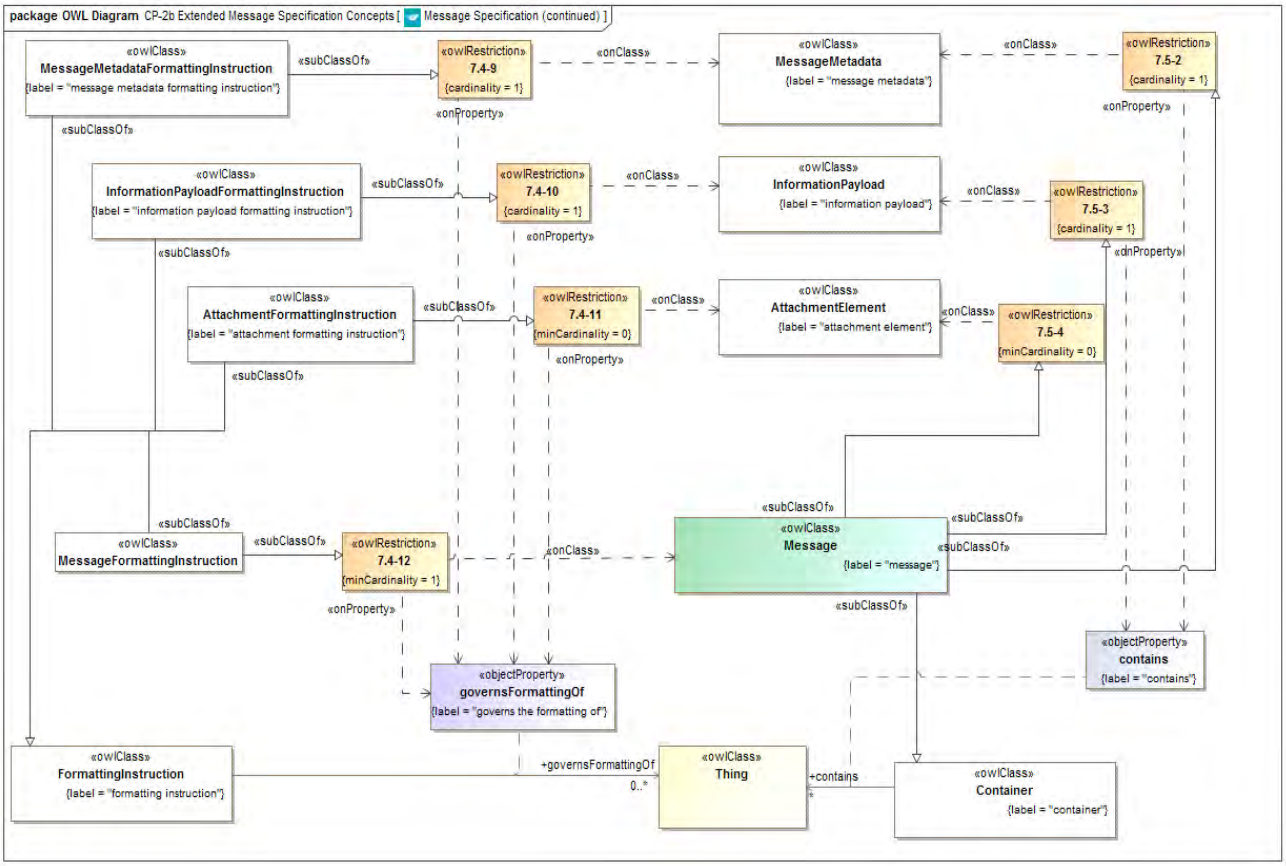

Figure 7.33 - Message Specification (continued)

The following rules apply to the Message Specification (continued):

1. InformationPayloadFormattingInstruction governsFormattingOf one and only one InformationPayload.
2. AttachmentFormattingInstruction governsFormattingOf an optional set of AttachmentElement.
3. MessageFormattingInstruction governsFormattingOf at least one Message.
4. MessageMetadataFormattingInstruction governsFormattingOf one and only one MessageMetadata.
5. Message contains one and only one MessageMetadata.
6. Message contains one and only one InformationPayload.
7. Message contains an optional set of AttachmentElement.

* The name used to identify the rules is derived from the restriction encompassed by the rule.

7.5.4 Information Package Specification Concepts

The InformationPackageSpecification identifies concepts within the Vocabulary that combine to express the rules governing the assembly and processing of information packages.

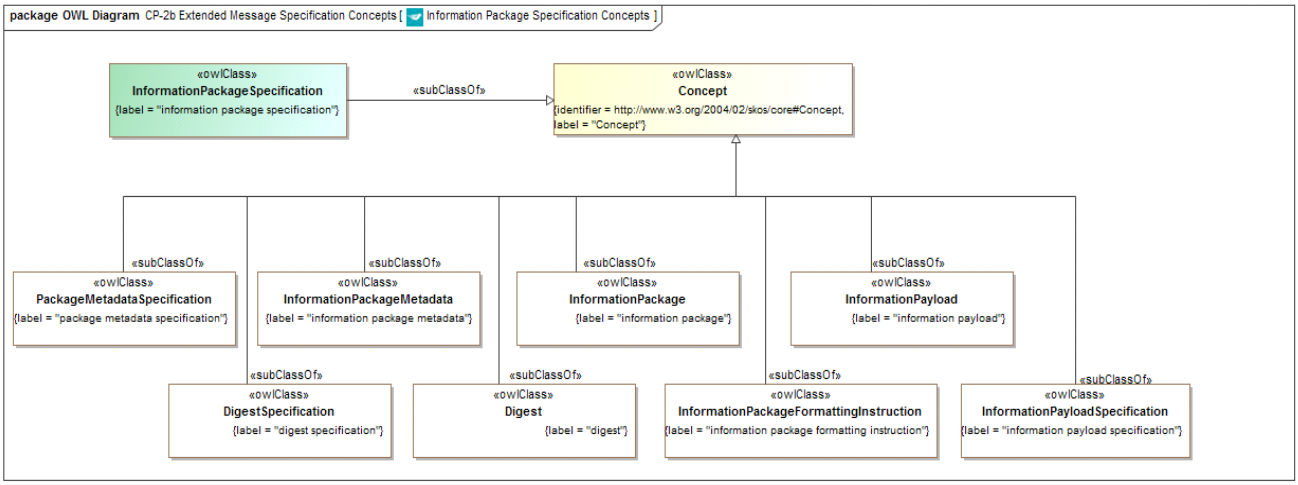

Figure 7.34 - Information Package Specification Concepts

7.5.5 Information Package Specification

The following figure illustrates the relationships between concepts used in the expression of rules for the assembly and processing of payload elements including a package metadata, digest and information payload.

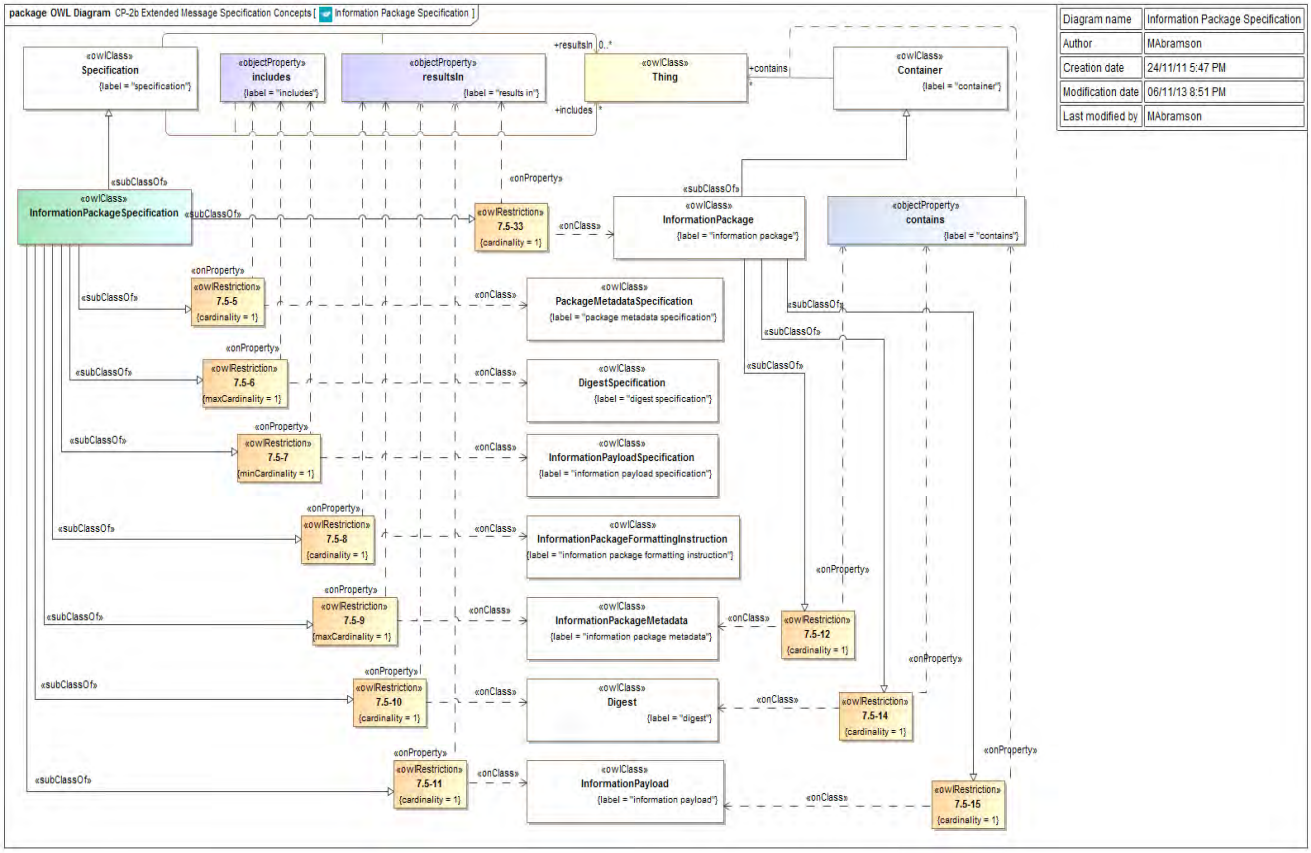

Figure 7.35 - Information Package Specification

The following rules apply to the Information Package Specification:

1. InformationPackageSpecification resultsIn one and only one Digest.
2. InformationPackageSpecification resultsIn one and only one InformationPayload.
3. InformationPackage contains one and only one InformationPackageMetadata.
4. InformationPackage contains one and only one Digest.
5. InformationPackage contains one and only one InformationPayload.
6. InformationPackageSpecification resultsIn one and only one InformationPackage.
7. InformationPackageSpecification includes one and only one PackageMetadataSpecification.
8. InformationPackageSpecification includes a maximum one DigestSpecification.
9. InformationPackageSpecification includes at least one InformationPayloadSpecification.
10. InformationPackageSpecification resultsIn one and only one InformationPackageFormattingInstruction.
11. InformationPackageSpecification resultsIn a maximum one InformationPackageMetadata.

* The name used to identify the rules is derived from the restriction encompassed by the rule.

7.5.6 Information Package Specification (continued)

The following figure illustrates the relationships between concepts used in the expression of rules governing the assembly and processing of an InformationPackage. The figure extends the definition of InformationPackage by providing the relationships from the Specifications to the resulting InformationElements.

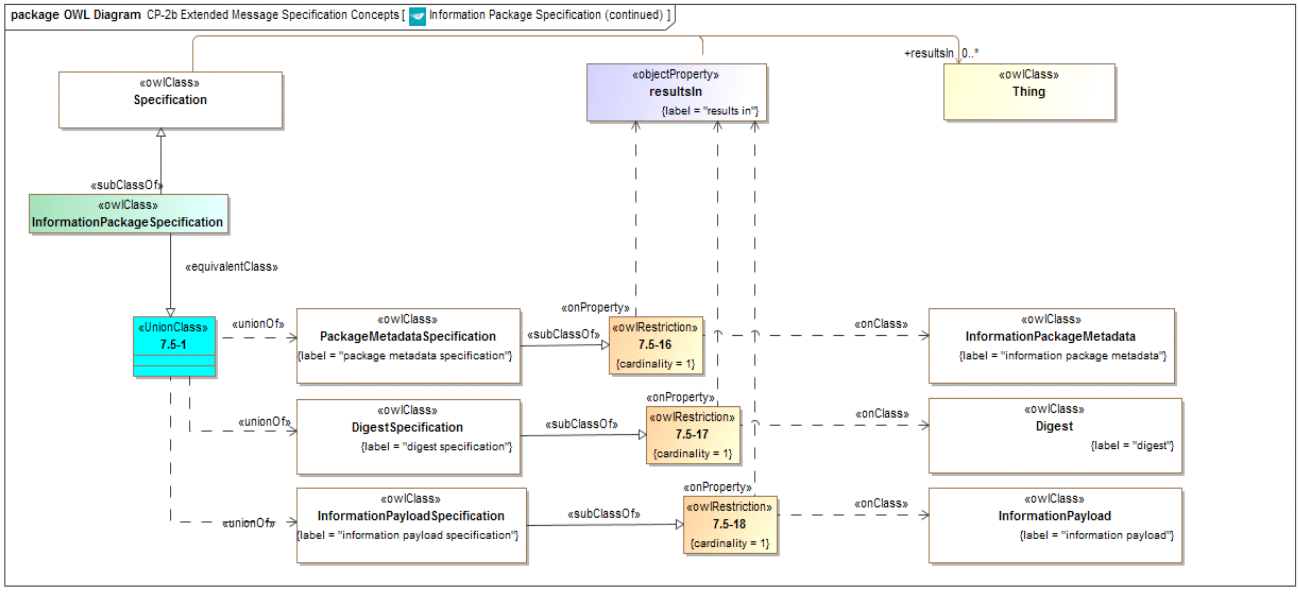

Figure 7.36 - Information Package Specification (continued)

The following rules apply to the Information Package Specification (continued):

1. PackageMetadataSpecification resultsIn one and only one InformationPackageMetadata.
2. DigestSpecification resultsIn one and only one Digest.
3. InformationPayloadSpecification resultsIn one and only one InformationPayload.

* The name used to identify the rules is derived from the restriction encompassed by the rule.

7.5.7 Information Package Specification (formatting)

The following figure illustrates the relationships between concepts used in the expression of rules governing the assembly and processing of an InformationPackage. This figure illustrates the relationships between the InformationPackage and associated FormattingInstructions.

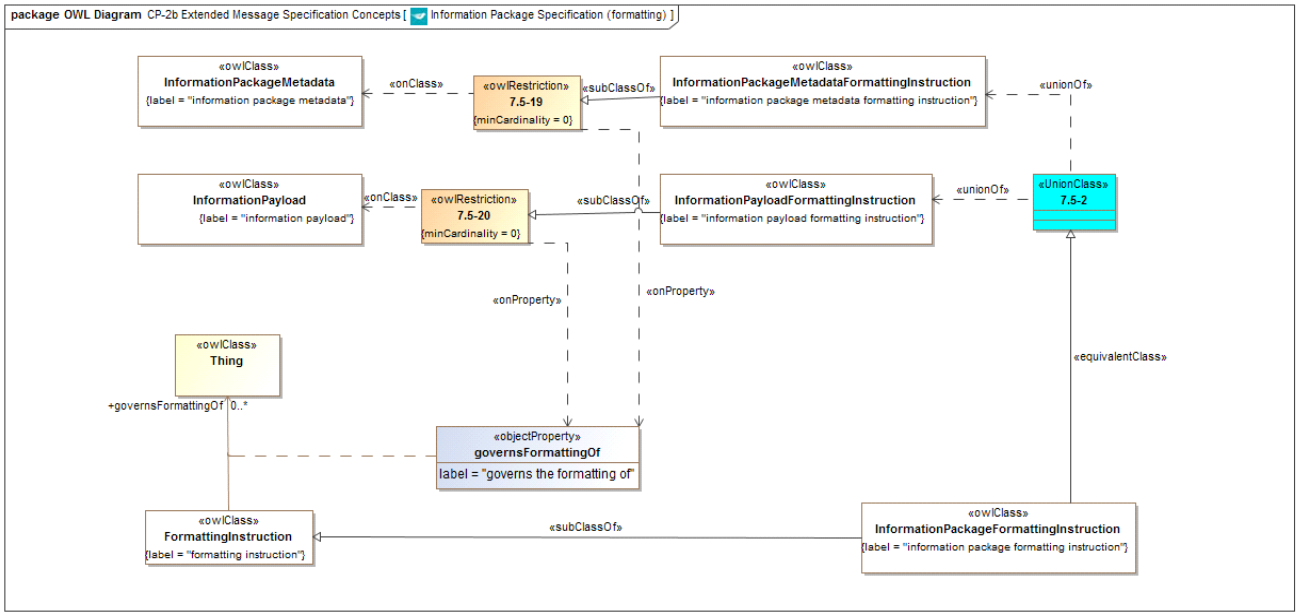

Figure 7.37 - Information Package Specification (formatting)

The following rules apply to the Information Package Specification (formatting):

1. InformationPackageMetadataFormattingInstruction governsFormattingOf an optional set of InformationPackageMetadata.

2. InformationPayloadFormattingInstruction governsFormattingOf an optional set of InformationPayload.

* The name used to identify the rules is derived from the restriction encompassed by the rule.

7.5.8 Information Package Metadata Specification Concepts

The InformationPackageMetadataSpecification identifies concepts within the Vocabulary that combine to express the rules used for assembly and processing the Metadata tags associated with an InformationPackage.

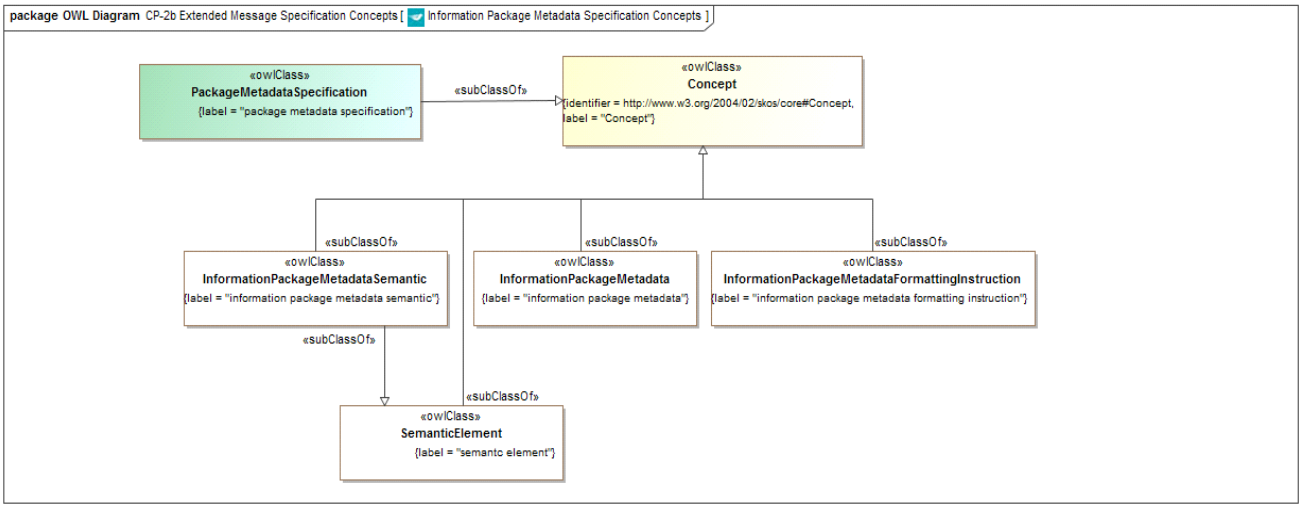

Figure 7.38 - Information Package Metadata Specification Concepts

7.5.9 Information Package Metadata Specification

The Attachment Specification identifies concepts within the Vocabulary that combine to express the rules used to specify the attachment to be included with a Message. CP-2b extends the concepts provided in CP-2a by permitting multiple attachments to the Message structure.

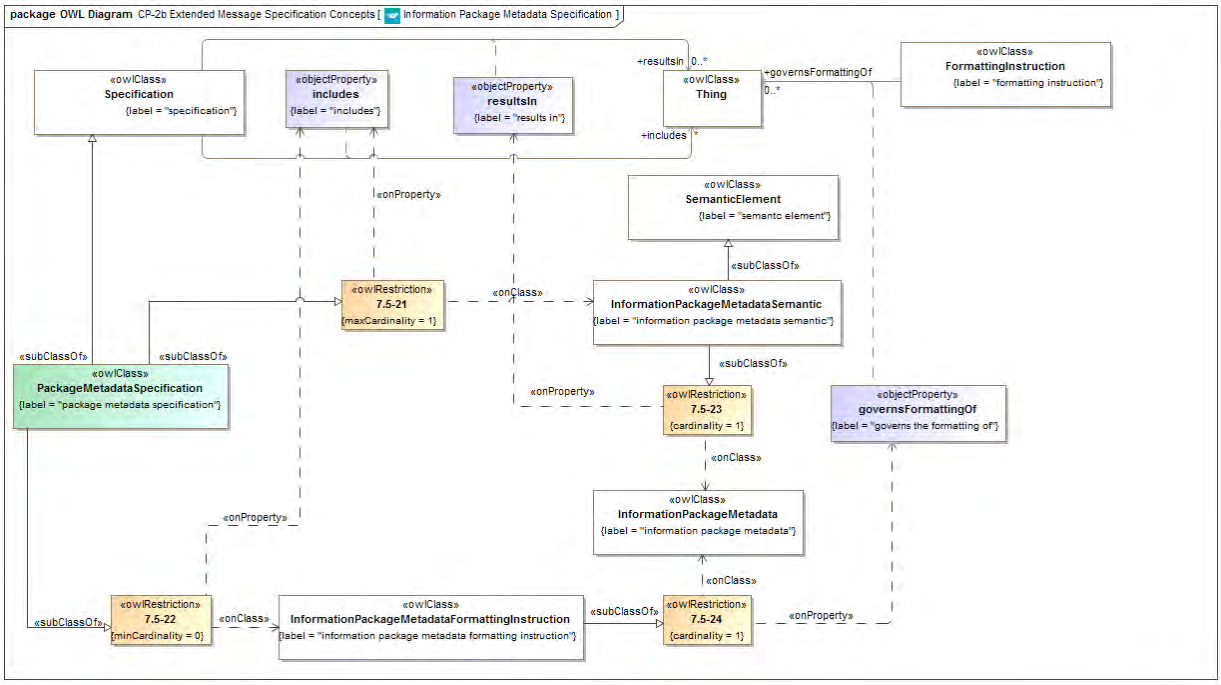

Figure 7.39 - Information Package Metadata Specification

The following rules apply to the Information Package Metadata Specification:

1. PackageMetadataSpecification includes a maximum of one InformationPackageMetadataSemantic.
2. PackageMetadataSpecification includes an optional set of InformationPackageMetadataFormattingInstruction.
3. InformationPackageMetadataSemantic resultsIn one and only one InformationPackageMetadata.
4. InformationPackageMetadataFormattingInstruction governsFormattingOf one and only one InformationPackageMetadata.

* The name used to identify the rules is derived from the restriction encompassed by the rule.

7.5.10 Information Payload Specification Concepts

The Information Package Specification identifies concepts within the Vocabulary that combine to express the rules governing the assembly and processing of an InformationPayload.

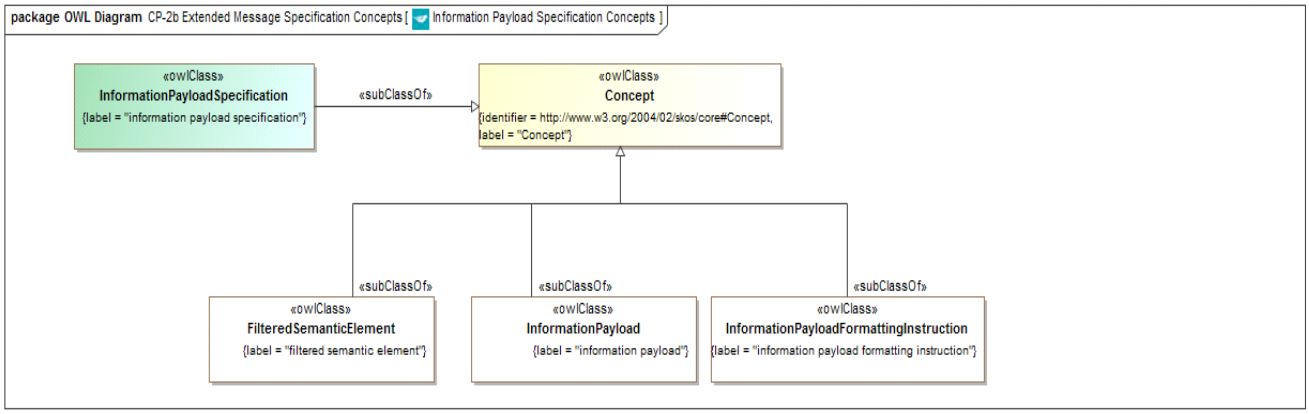

Figure 7.40 - Information Payload Specification Concepts

7.5.11 Information Payload Specification

The following figure illustrates the relationships between concepts used in the expression of rules for the assembly and processing of an InformationPayload.

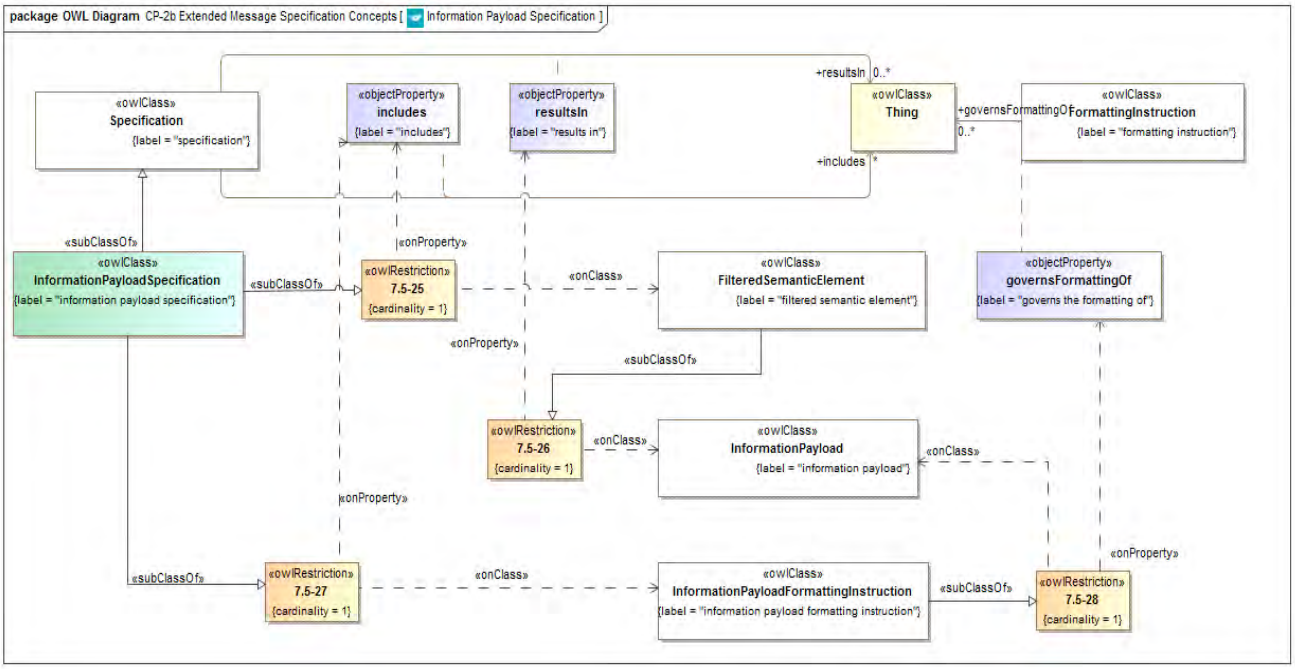

Figure 7.41 - Information Payload Specification

The following rules apply to the Information Payload Specification:

1. InformationPayloadSpecification includes one and only one FilteredSemanticElement.
2. FilteredSemanticElement resultsIn one and only one InformationPayload.
3. InformationPayloadSpecification includes one and only one InformationPayloadFormattingInstruction.
4. InformationPayloadFormattingInstruction governsFormattingOf one and only one InformationPayload.

* The name used to identify the rules is derived from the restriction encompassed by the rule.

7.5.12 Attachment Specification Concepts

The Attachment Specification identifies concepts within the Vocabulary that combine to express the rules used to specify the attachment to be included with a Message. CP-2b extends the concepts provided in CP-2a by permitting multiple attachments to the Message structure.

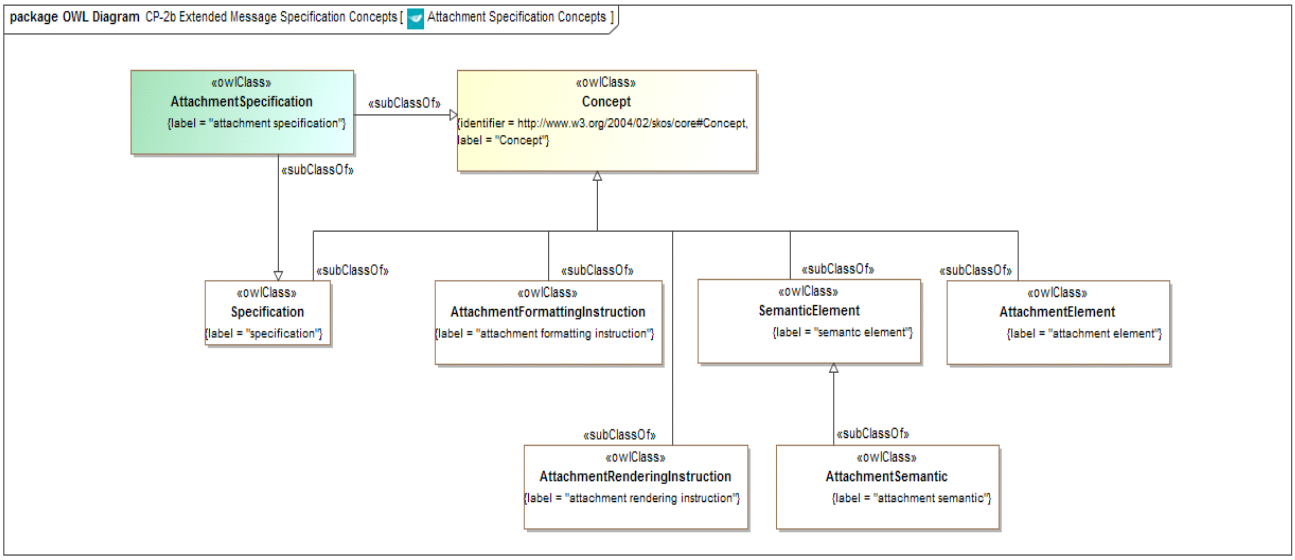

Figure 7.42 - Attachment Specification Concepts

7.5.13 Attachment Specification

The following figure illustrates the relationships between concepts used in the expression of rules for attaching unstructured data/information Attachments to a Message.

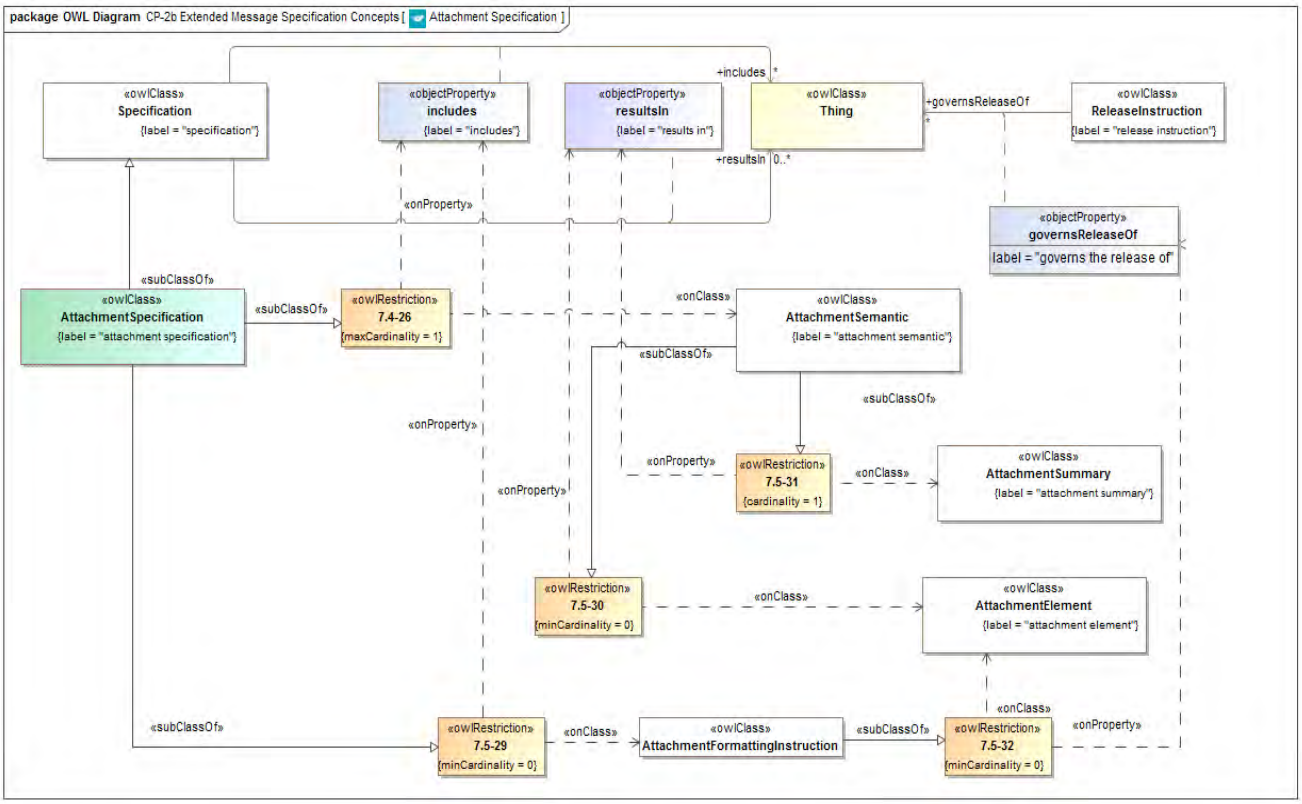

Figure 7.43 - Attachment Specification

The following rules apply to the Attachment Specification:

1. AttachmentSpecification includes a maximum one AttachmentSemantic.
2. AttachmentSpecification includes an optional set of AttachmentFormattingInstruction.
3. AttachmentSemantic resultsIn an optional set of AttachmentElement.
4. AttachmentSemantic resultsIn one and only one AttachmentSummary.
5. AttachmentFormattingInstruction governsReleaseOf an optional set of AttachmentElement.

* The name used to identify the rules is derived from the restriction encompassed by the rule.

7.6 CP-2c Full Information Specification Concepts

The following sub clauses define the concepts, properties, and restrictions for a vocabulary that describes the rules governing the processing and assembly of information exchange agreements that includes the rules to construct and format the message being exchanged. These sub clauses extend CP-2a&b and CP-1, which form an inherent part of CP-2c; described in 2.3.2.1. CP-2c extends the Message structure of CP-2a by adding structures to the basic message.

7.6.1 Information Package Specification Concepts

The Information Package Specification identifies concepts within the Vocabulary that combine to express the rules governing the assembly and processing of an InformationPackage. CP-2c extends the InformationPackage concepts presented in CP-2b.

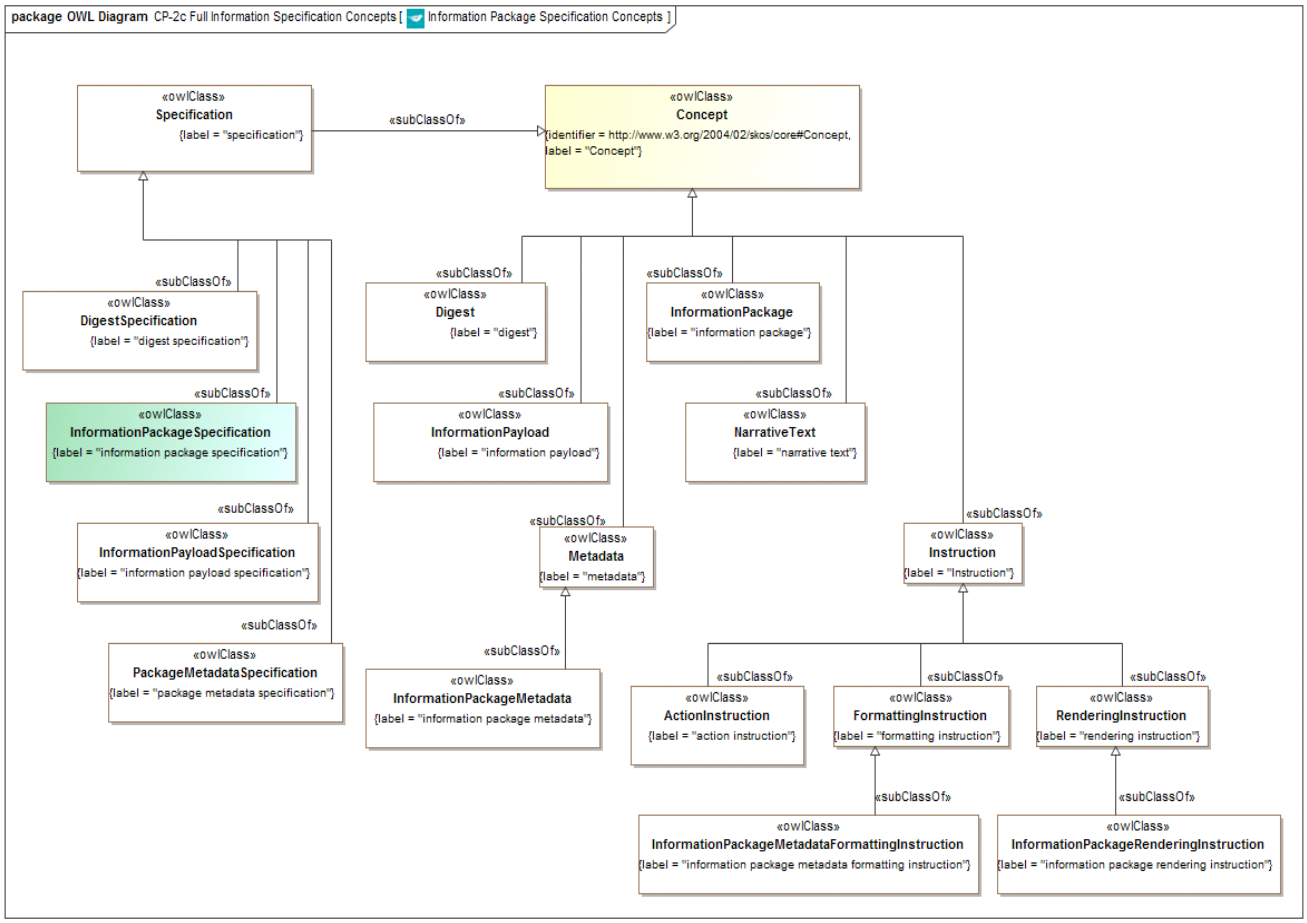

Figure 7.44 - Information Package Specification Concepts

7.6.2 Information Package Specification

The following figure illustrates the relationships between concepts used in the expression of rules governing the assembly and processing of an InformationPackage.

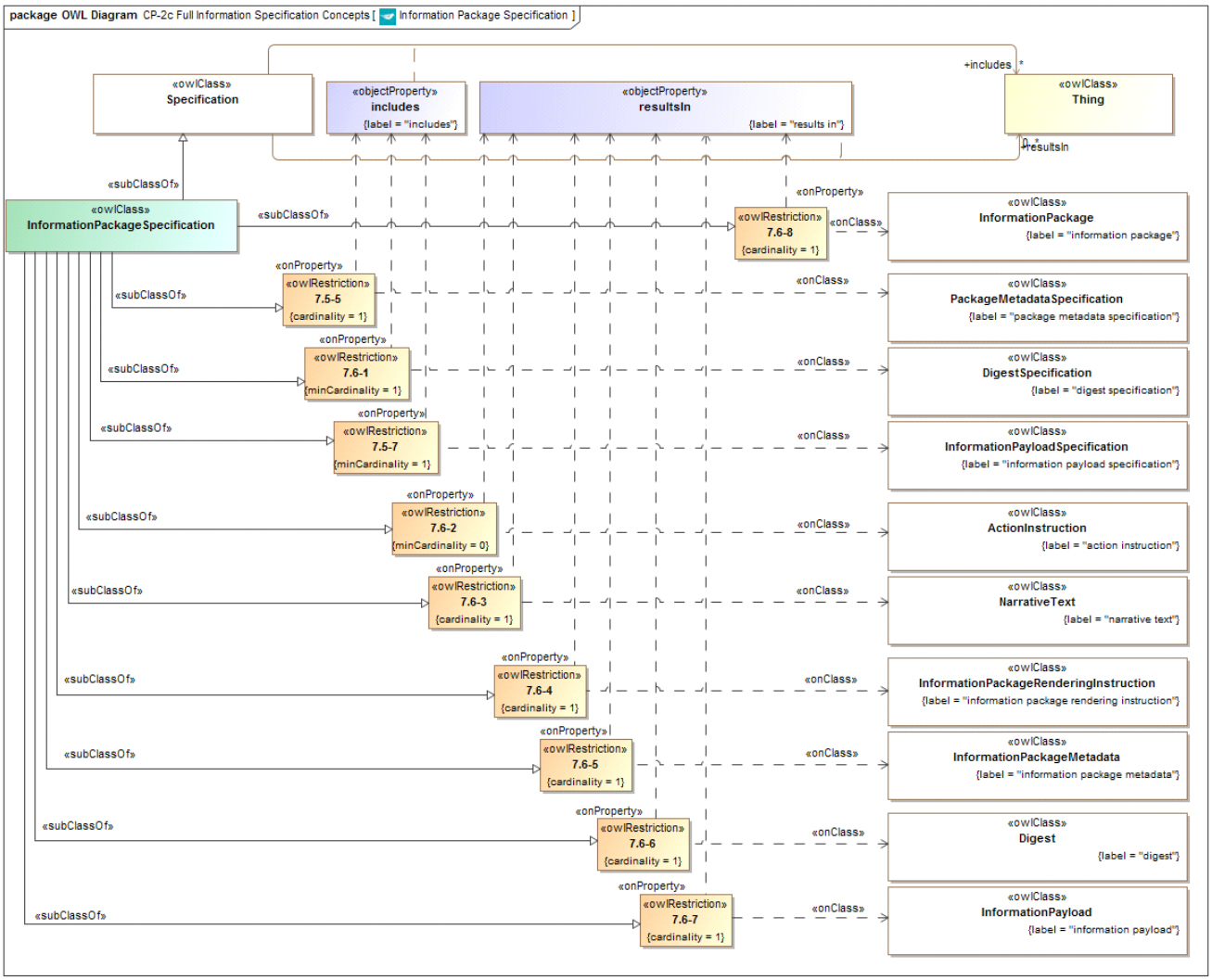

Figure 7.45 - Information Package Specification

The following rules apply to the Information Package Specification:

1. InformationPackageSpecification includes one and only one PackageMetadataSpecification.
2. InformationPackageSpecification includes at least one InformationPayloadSpecification.
3. InformationPackageSpecification includes at least one DigestSpecification.
4. InformationPackageSpecification resultsIn an optional set of ActionInstruction.
5. InformationPackageSpecification resultsIn one and only one NarrativeText.
6. InformationPackageSpecification resultsIn one and only one InformationPackageRenderingInstruction.
7. InformationPackageSpecification resultsIn one and only one InformationPackageMetadata.

8. InformationPackageSpecification resultsIn one and only one Digest.
9. InformationPackageSpecification resultsIn one and only one InformationPayload.
10. InformationPackageSpecification resultsIn one and only one InformationPackage.

* The name used to identify the rules is derived from the restriction encompassed by the rule.

7.6.3 Information Package Specification Results

The following figure illustrates the relationships between concepts used in the expression of rules governing the assembly and processing of an InformationPackage.

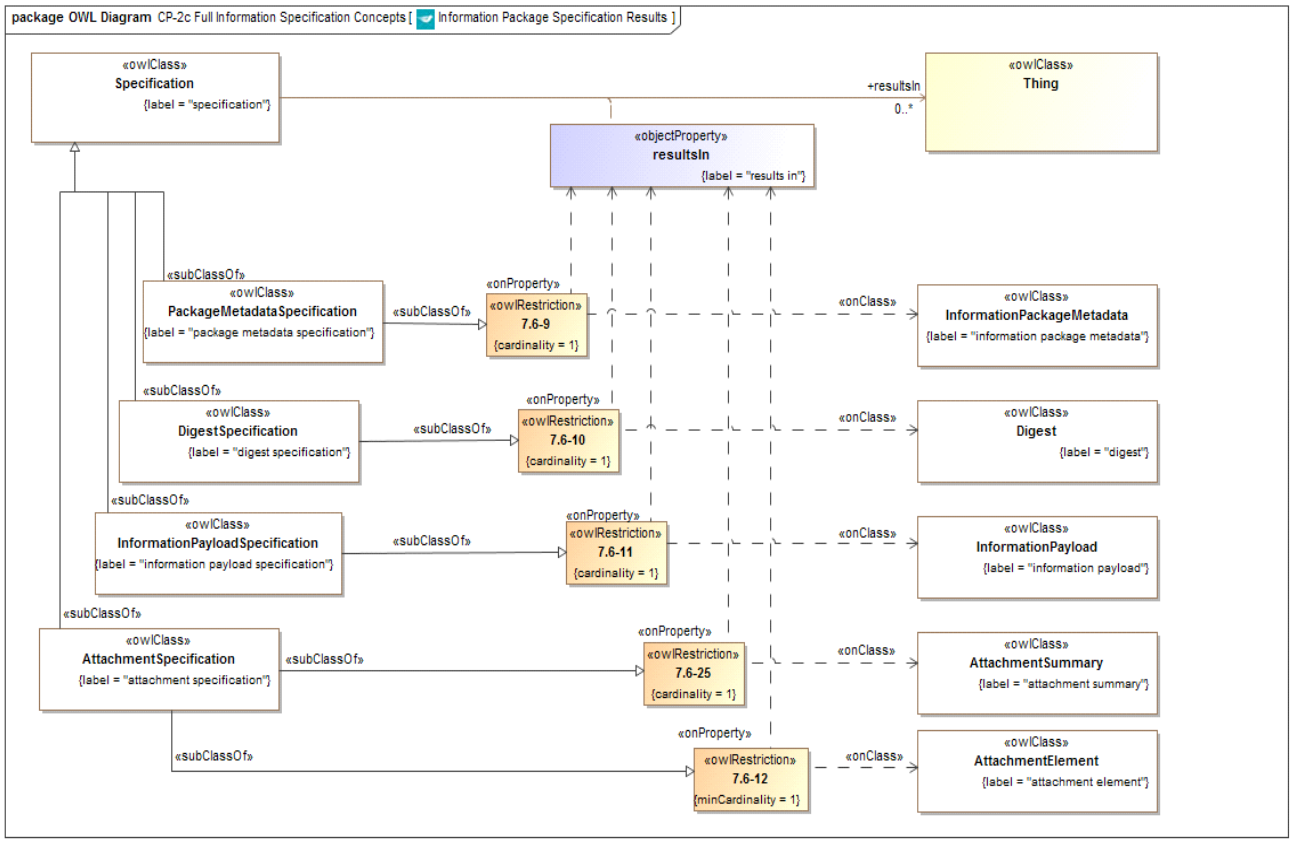

Figure 7.46 - Information Package Specification Results

The following rules apply to the Information Package Specification Results:

1. DigestSpecification resultsIn one and only one Digest.
2. InformationPayloadSpecification resultsIn one and only one InformationPayload.
3. AttachmentSpecification resultsIn at least one AttachmentElement.

4. AttachmentSpecification resultsIn one and only one AttachmentSummary.
5. PackageMetadataSpecification resultsIn one and only one InformationPackageMetadata.

* The name used to identify the rules is derived from the restriction encompassed by the rule.

7.6.4 Digest Specification Concepts

The Attachment Specification identifies concepts within the Vocabulary that combine to express the rules governing the assembly and processing of a Digest.

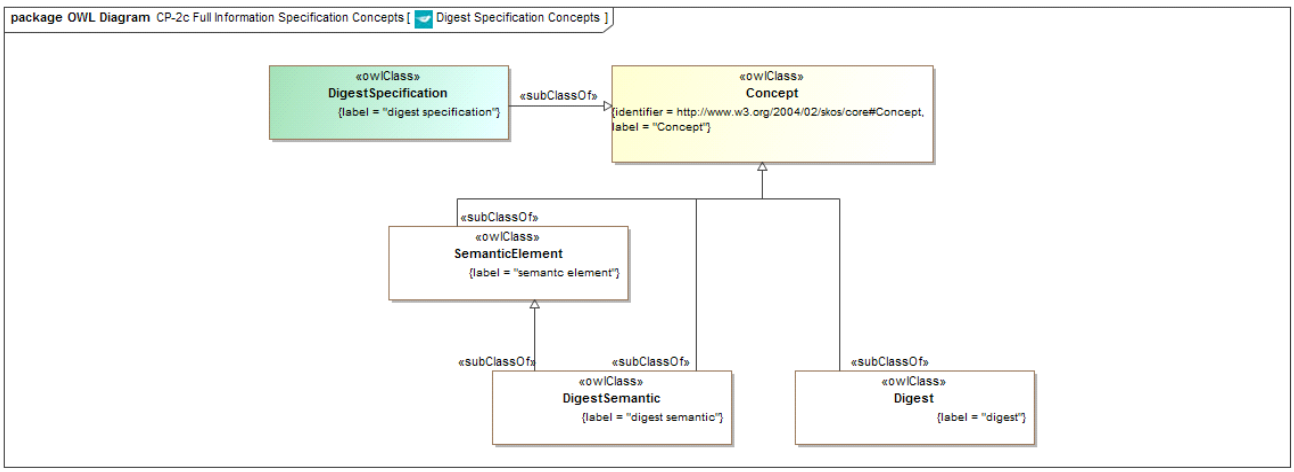

Figure 7.47 - Digest Specification Concepts

7.6.5 Digest Specification

The following figure illustrates the relationships between concepts used in the expression of rules governing the assembly and processing of a Digest.

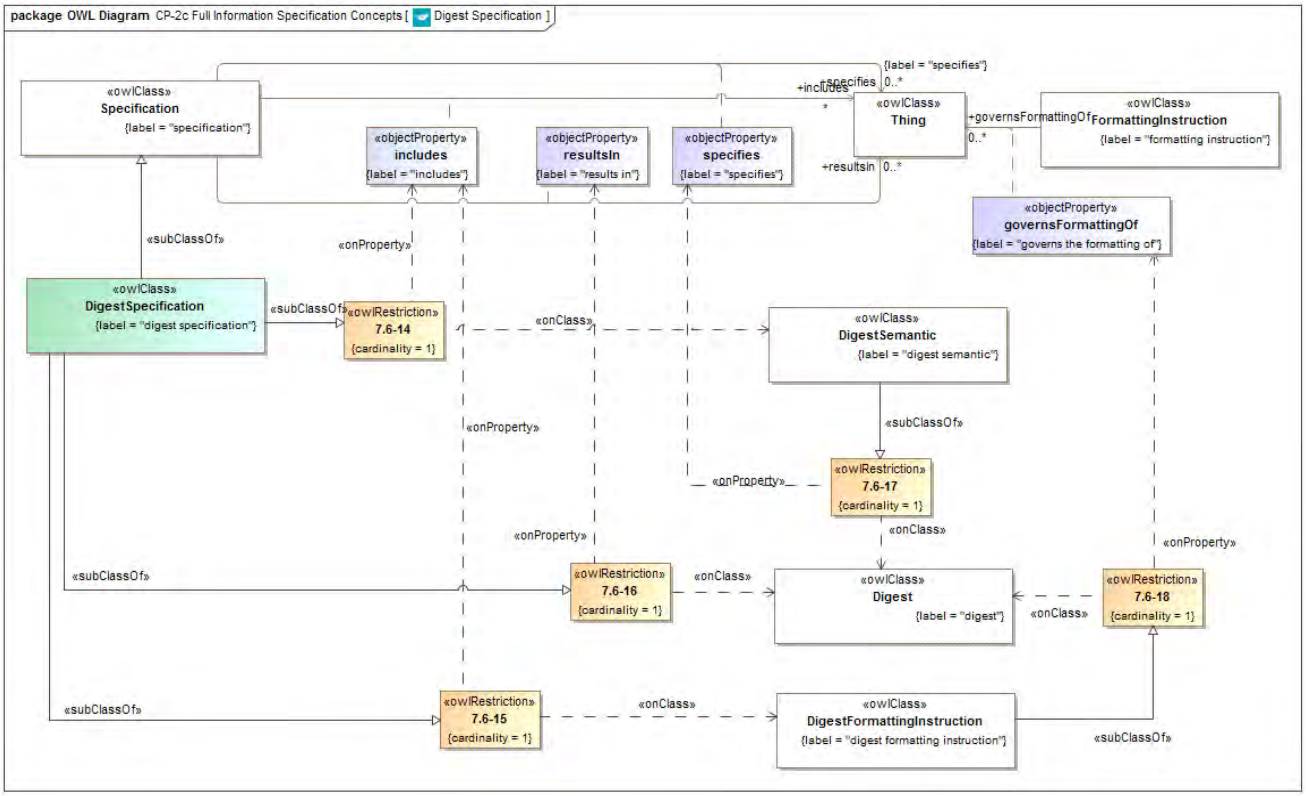

Figure 7.48 - Digest Specification

The following rules apply to the Digest Specification:

1. DigestSpecification includes one and only one DigestSemantic.
2. DigestSpecification includes one and only one DigestFormattingInstruction.
3. DigestSpecification resultsIn one and only one Digest.
4. DigestSemantic specifies one and only one Digest.
5. DigestFormattingInstruction governsFormattingOf one and only one Digest.

* The name used to identify the rules is derived from the restriction encompassed by the rule.

7.6.6 Attachment Specification Concepts

The Attachment Specification identifies concepts within the Vocabulary that combine to express the rules used to specify the attachment to be included with a Message. CP-2c extends the concepts provided in CP-2a and CP-2b by permitting multiple attachments to the Message structure and the assembly and inclusion of the attachment summary within each of the InformationPackages. The AttachmentSummaries identify which Attachments are associated with the InformationPackage.

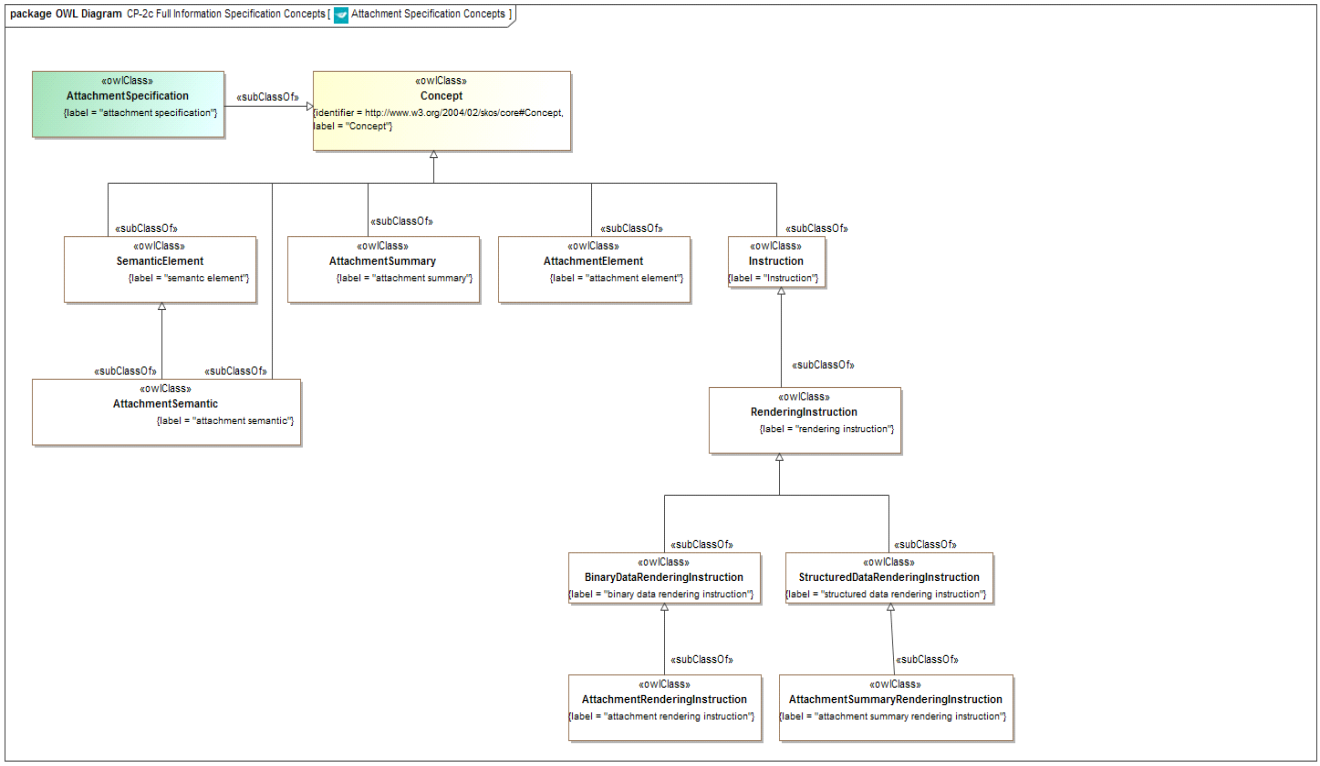

Figure 7.49 - Attachment Specification Concepts

7.6.7 Attachment Specification

The following figure illustrates the relationships between concepts used in the expression of rules for attaching unstructured data/information Attachments to a Message.

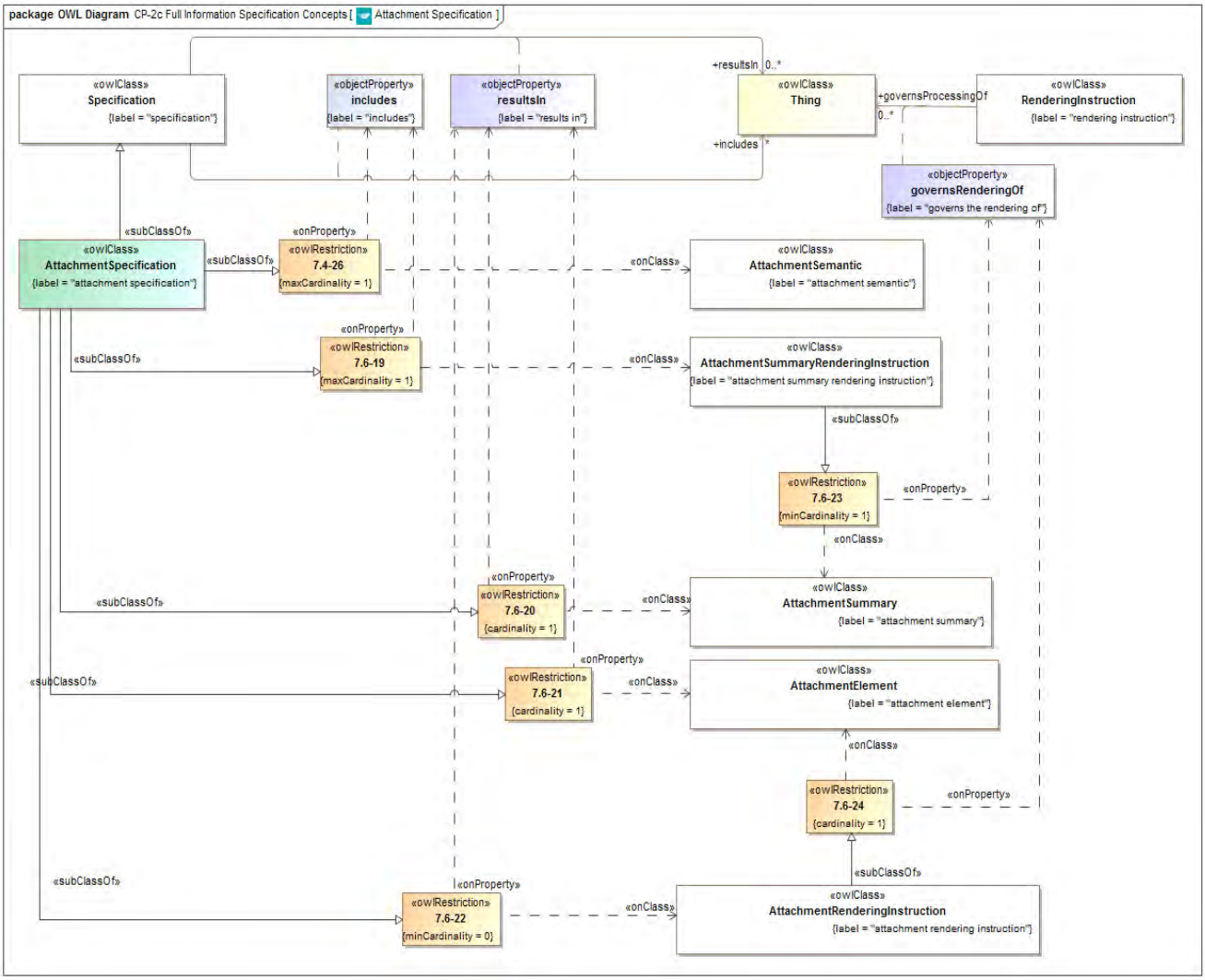

Figure 7.50 - Attachment Specification

The following rules apply to the Attachment Specification:

1. AttachmentSpecification includes a maximum one AttachmentSemantic.
2. AttachmentSpecification includes a maximum one AttachmentSummaryRenderingInstruction.
3. AttachmentSpecification resultsIn one and only one AttachmentSummary.
4. AttachmentSpecification resultsIn one and only one AttachmentElement.
5. AttachmentSpecification resultsIn an optional set of AttachmentRenderingInstruction.
6. AttachmentSummaryRenderingInstruction governsRenderingOf at least one AttachmentSummary.
7. AttachmentRenderingInstruction governsRenderingOf one and only one AttachmentElement.

* The name used to identify the rules is derived from the restriction encompassed by the rule.

7.7 CP-3 Distribution Specification Concepts

Compliance Point 3 is optional. It provides the concepts needed to assign an InformationElement (ReleasableDataSet or Message) to the service specified to distribute or disseminate the information.

7.7.1 Distribution Specification Concepts

The Distribution Specification identifies the concepts within the Vocabulary that enable the expression of rules for specifying the distribution or dissemination services to be used.

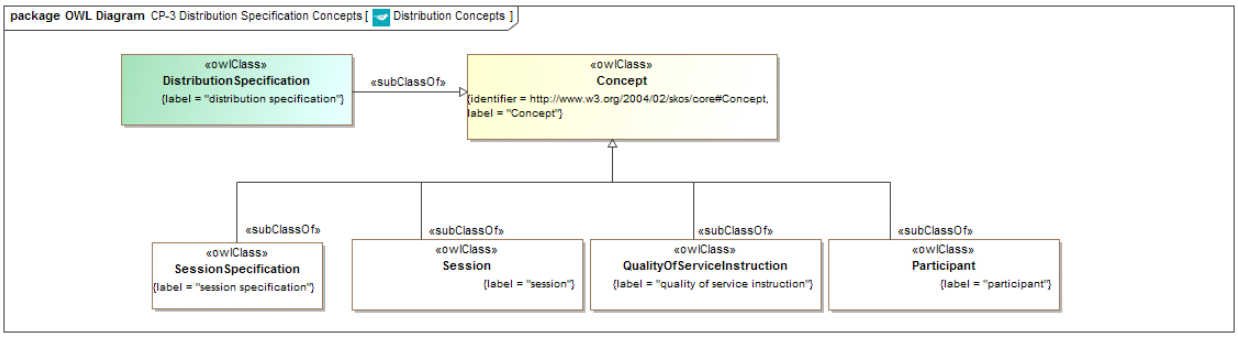

Figure 7.51 - Distribution Concepts

7.7.2 Distribution Specification

The following figure illustrates the relationships between concepts used in the expression of rules for specifying the distribution or dissemination services to be used.

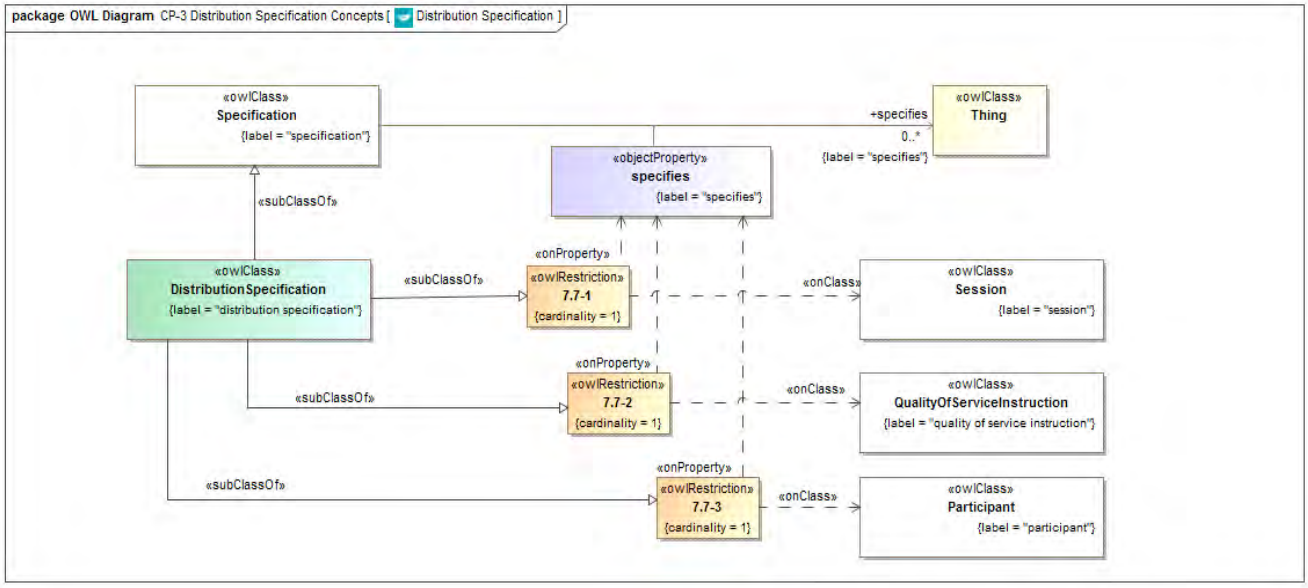

Figure 7.52 - Distribution Specification

The following rules apply to the Distribution Specification:

1. DistributionSpecification specifies one and only one Session.
2. DistributionSpecification specifies one and only one QualityOfServiceInstruction.
3. DistributionSpecification specifies one and only one Participant.

* The name used to identify the rules is derived from the restriction encompassed by the rule.

Annex A IEPPV Taxonomy (Normative)

The following table provides an alphabetical presentation of the IEPPV concepts and repeats the information presented in Clause 7:

Table A.1 - IEPPV Taxonomy

Name	Is A	Has Specializations	Definition
AcknowledgeInstruction	ActionInstruction ReceiptInstruction Concept		An instruction to the recipient of an information exchange directing the issuance of an acknowledgment to the receipt of the information to the provider of the information.
ActionInstruction	Concept Instruction	ValidateInstruction DiscardInstruction ReleaseInstruction ForwardInstruction HandlingInstruction Acknowledge Instruction ReceiptInstruction PersistenceInstruction RetentionInstruction	An instruction directing the producer or receiver of a message to take a specific action, (1) message specific rules governing the release of the information, or (2) message specific actions to be taken upon receipt of the message.
AttachmentElement	Concept MessageElement		A binary file or (e.g., PDF file, image or video) or document, and information about the binary or document, such as the size and type and description. Source: Logical Entity Exchange Specification (LEXS): Attachment (N): A binary, such as an image or PDF file or video, as well as information about the binary, such as the size and type and description.
AttachmentFormatting Instruction	Concept FormattingInstruction Instruction		An instruction to the provider of information defining the rules for formatting the data set in accordance with the agreed protocol for the exchange.
AttachmentRendering Instruction	Concept RenderingInstruction BinaryDataRendering Instruction Instruction		An instruction to the recipient of an information exchange defining the rules for rendering or displaying an attachment or set of attachments.
AttachmentSemantic	Concept SemanticElement		A Semantic that specifies the rules for assembling the attachments to a message. It also provides the rules for generating an attachment summary and linkages.

Table A.1 - IEPPV Taxonomy

AttachmentSpecification	Concept Specification		A specification of the rules governing attachment of binary information elements to an information exchange or message.
AttachmentSummary	Concept MessageElement		A summary or list of attachments for a specific data package.
AttachmentSummary RenderingInstruction	StructuredData RenderingInstruction RenderingInstruction		An instruction to the recipient of an information exchange defining the rules for rendering or displaying an attachment summary.
Attribute	Concept	SemanticAttribute SubtendedElement Attribute Transformation ResultingAttribute Transactional Attribute WrapperAttribute	A defined property of an entity, object, triple, schema, etc. Source: A Dictionary of Computing. Oxford University Press, 2008. Oxford Reference Online. Oxford University Press.
BinaryDataRendering Instruction	RenderingInstruction Concept	AttachmentRendering Instruction	An instruction to the recipient of an information exchange defining the rules for rendering or displaying binary data.
Container	Concept	Digest InformationPackage Message InformationPayload	A receptical for results of an aggregation of data and information elements. Derived from http:// www.merriam-webster.com/dictionary/container, a receptacle (as a box or jar) for holding goods.
DataCreatorMetadata	InformationPackage Metadata MessageMetadata Concept		Metadata tags and markings that identify the creator of data or information elements.
DataElement	Concept		Representation of information (data) in a formalized manner suitable for communication, interpretation, or processing by humans or by automated means. In the context of IEPPV, data elements are atomic facts. Derived from UPDM.
DataOwnerMetadata	Metadata Concept		Tags and markings that identify the owner or steward of the data or information elements.

Table A.1 - IEPPV Taxonomy

Digest	Container Concept MessageElement		An information structure, format, and syntax common to all communities. It provides the ability for systems to handle heterogeneous data without having to understand the specific context and or semantics of the source. As long as the entities relevant to the packaged data items are represented in the Digest, users will be able to discover, link, map, etc. the information within. Source: the concept for digest is derived from and intended to support the Logical Entity eXchange Specification (LEXS). http://130.207.211.107/content/lexs-overview. The Digest provides the common level of understanding, it does not mean that all sources have to populate all elements, or that all consumers have to use all elements; merely that at a schema level all applications understand the Digest. Implementers only need to build one module in order to produce or consume a basic set of data understandable by many. It also means that implementers do not have to develop large applications for each exchange, but rather build one that handles the basics and then additional smaller modules in order to produce or consume more complex exchanges. The objective of the Digest is to present the most common characteristics of real-world objects that can be supported by any data source or data consumer. Digest-level data objects may be further augmented or described with additional details in included packages or narrative text integrated into the message. The information in the digest must be semantically complete for both the data source or data consumer; the information package contents may rely on the digest to complete its semantics. The enforcement of a “Digest Semantic” by a software service will result in the generation of the digest for the instance of the Information Package. In other applications, where the digest is not used, the “Payload” comprises the entire data portion of the message content.
DigestFormatting Instruction	Concept FormattingInstruction		An instruction to the provider of information specifying the rules for formatting the data set for a Digest in accordance with the agreed protocol for the exchange.
DigestSemantic	Concept SemanticElement		A SemanticElement that specifies the rules for assembling data and information elements for a Digest.
DigestSpecification	Concept Specification		A specification and set of rules governing the preparation (generation) of a digest.
DiscardInstruction	ActionInstruction Concept		An instruction to the recipient of an information exchange specifying the rules for destruction or discarding of data included within an information package or message.

Table A.1 - IEPPV Taxonomy

Distribution Specification	Specification Concept		A specification of the rules governing the assignment of InformationElements to a specific information dissemination service (e.g., User Application, Service Interface, and Middleware).
DoNotForward Instruction	Concept ReceiptInstruction		An instruction to the recipient of an information exchange specifying that the information must not be forwarded to any other recipient or destination.
DoNotPersist Instruction	Concept ReceiptInstruction		An instruction to the recipient of an information exchange directing the recipient not to persist any of the information or data in a payload or message.
DynamicFilter	Concept Filter		Rules for a data or domain filter whose parameters may be configured at run-time.
EnclosedTransactional Element	Concept TransactionalElement		A TransactionalElement included as part of the build pattern of a TransactionalElement or SemanticElement.
EnclosingTransactional Element	Concept TransactionalElement		A TransactionalElement that includes one or more TransactionalElements or WrapperElements.
EncryptInstruction	ReleaseInstruction Concept Safeguard		An instruction or set of instructions to the producer of the information directing that the message or elements of the message need to be encrypted prior to release.
Entity	Concept SourceData		Independent, separate, or self-contained existence. Source: Merriam-Webster Dictionary
File	Concept SourceData		A collection of information, referred to by file name; for example, a user-created document, program data, or the program itself. With a program, the information is held on backing store (i.e., usually on magnetic disk) in order (a) to enable it to persist beyond the time of execution of a single job and/or (b) to overcome space limitations in main memory. Files with a very brief existence (i.e., in case (b) above, or where they simply carry information between one job and the next in sequence) are called work files. See also master file, data file. Source: A Dictionary of Computing. Ed John Daintith and Edmund Wright. Oxford University Press, 2008. Oxford Reference Online.
Filter	Concept	StaticFilter DynamicFilter SecurityFilter	A profile or script containing the rules to restrict the assembly of data or information elements. Source: Defined for the Information Exchange Policy Vocabulary. Derived from “A general term used to describe software which examines some content and prevents it from reaching its destination, based on a number of rules stored in a script or a profile.” A Dictionary of the Internet. Darrel Ince. Oxford University Press, 2009. Oxford Reference Online.
FilteredSemantic Element	Specification Concept InformationElement		Specifies rules for the assignment of one or more DynamicFilters to a specified SemanticElement. Source: Derived from SOPES IEDM V1

Table A.1 - IEPPV Taxonomy

FilteredTransactional Element	InformationElement Concept		Rules specifying the WrapperAttributes that are filterable at runtime.
FilterRule	Concept		A rule or rules governing the inclusion of or rejection of data or information elements based on the value of a specified attribute, or values of specified attributes.
FormattingInstruction	Concept ReleaseInstruction Instruction	InformationPackage FormattingInstruction InformationPayload FormattingInstruction Attachment FormattingInstruction DigestFormatting Instruction InformationPackage MetadataFormatting Instruction MessageFormatting Instruction MessageMetadata FormattingInstruction	An instruction to the provider of information defining the rules for formatting a generated data set.
ForwardInstruction	ReceiptInstruction ActionInstruction Concept		An instruction to the recipient of an information exchange to forward the information to authorized recipients in accordance with any provided list, or in accordance with specified information sharing agreements.
HandlingInstruction	ReceiptInstruction ActionInstruction MessageMetadata Concept InformationPackage Metadata		An instruction to the recipient of an information exchange specifying how this information must be handled.
Identifier	Concept		Identifies the element (TransactionalElement or WrapperElement) that holds a unique identifier or key needed for the construction of a data set. This subtended class would contain, as a minimum, the base global unique identifier (e.g., database key, foreign keys, or unique identifier) that would differentiate which Transactional or Wrapper instance (information element instances) is included in the construction of the composite. (e.g., foreign key relationships) There exists one and only one identifier for each SemanticElement or TransactionalElement.
Identifier			Source: Derived from UML Profile for DODAF and MODAF (UPDM) V2.0, formal/2012-01-03 and Shared Operational Picture Exchange Services (SOPES) Information Exchange Data Model (IEDM) Version 1.0, formal/2011-05-04

Table A.1 - IEPPV Taxonomy

InformationElement	Concept	TransactionalElement FilteredTransactional Element SubtendedElement SemanticElement FilteredSemantic Element WrapperElement	An item of information that flows between operational activities and nodes. For IEPPV, an information element refers to a grouping of data elements (including other information elements) providing meaning within the context of an operation or situation. Derived from: MODAF: A formalized representation of information subject to an operational process. DoDAF: Information that is passed from one operational node to another. Associated with an information element are such performance attributes as timeliness, quality, and quantity values. (DoDAF) Information Exchange: The collection of information elements and their performance attributes such as timeliness, quality, and quantity values. (DoDAF) Note: Within the architectural context of the UPDM, SOPES, and IEPPV, the Information element provides a description of, or specification for, the data or information processed or exchanged. The Information element does not refer to the instance data or information being processed or exchanged, as this can only be determined at run-time.
InformationExchange Specification	Concept Specification		Specifies the information elements shared as part of a specific information sharing agreement and the information dissemination services to be used.
InformationPackage	Container MessageElement Concept		A standard representation of structured, semi-structured, and binary information applicable to an information sharing agreement. Packages may contain metadata, a Digest, a Structured Payload, Rendering Instructions, and optional linkages depending on the established agreements.
InformationPackage FormattingInstruction	Concept FormattingInstruction		An instruction to the provider of information defining rules for formatting the elements of a Data Package in accordance with the agreed protocol for the exchange.
InformationPackage Metadata	Concept MessageElement Metadata	MessageType MessageSensitivity DataCreatorMetadata MessageTimeStamp HandlingInstruction	Tags and markings that identify and describe the contents of an information package.
InformationPackage MetadataFormatting Instruction	Concept FormattingInstruction		An instruction to the provider of information defining the rules for formatting the Data Package Metadata in accordance with the agreed protocol for the exchange.
InformationPackage MetadataSemantic	Concept SemanticElement		A SemanticElement that specifies the rules for assembling the data elements to be included within Information Package Metadata.

Table A.1 - IEPPV Taxonomy

InformationPackageReleaseInstruction	ReleaseInstruction Concept		An instruction to the producer of an information exchange specifying instructions (e.g., Encrypt) pertaining to the release of the information package or message.
InformationPackage RenderingInstruction	Concept RenderingInstruction StructuredDataRendering Instruction		An instruction to the recipient of an information exchange defining the rules for rendering or displaying an Information Package.
InformationPackage Specification	Specification Concept		The rules and constraints governing the construction preparation of an information or data package.
InformationPayload	Container Concept MessageElement		A formatted dataset without protocols and metadata required for an information exchange. Derived from: Body (payload) The part of a cell or packet in a network that holds the information supplied by the end-user for transmission from the sender to the receiver. A Dictionary of Computing. Ed John Daintith and Edmund Wright. Oxford University Press, 2008. Oxford Reference Online. Data Payload: Refers to the “actual data” in a packet or file minus all headers attached for transport and minus all descriptive meta-data. In a network packet, headers are appended to the payload for transport and then discarded at their destination. In a key-length-value structure, the key and length are descriptive data about the value (the payload) http://www.pcmag.com/encyclopedia_term0,1237,t=payload&i=48909,00.asp
InformationPayload FormattingInstruction	FormattingInstruction Concept		An instruction to the provider of information defining the rules for formatting the information payload in accordance with the agreed protocol for the information exchange.
InformationPayload Specification	Concept Specification		The rules governing the assembly and processing of a structured dataset for an information exchange.
Information Specification	Concept Specification		Specifies the InformationElements that are included as part of the Information Exchange Agreement. Source: Defined for the Information Exchange Packaging Policy Vocabulary.

Table A.1 - IEPPV Taxonomy

Instruction	Concept	ReceiptInstruction MessageFormatting Instruction ActionInstruction FormattingInstruction AttachmentRendering Instruction Attachment FormattingInstruction RenderingInstruction QualityOfService Instruction	The description of an operation that is to be performed by a computer or human operator. Derived from: "The description of an operation that is to be performed by a computer. It consists of a statement of an operation to be performed and some method of specifying the operands (or their locations) and the disposition of the result of the operation." A Dictionary of Computing. Ed John Daintith and Edmund Wright. Oxford University Press, 2008.
Message	Container Concept		A formatted InformationElement transferred by a message switching system (or Network). Messages may be of any length, from a few bits to a complete file, and no part of a message is released to its final recipient until all of the message has been received at the network node adjacent to the destination. Source: A Dictionary of Computing. Ed John Daintith and Edmund Wright. Oxford University Press, 2008. Oxford Reference Online.
MessageElement	Concept	InformationPackage Metadata ReceiptInstruction MessageMetadata AttachmentElement AttachmentSummary Digest InformationPackage InformationPayload	An identifiable part of a message structure containing contextually relevant data or information elements. Message elements are integrated and formatted in accordance with contract or information exchange specification rules and instructions prior to release.
MessageFormatting Instruction	Concept Instruction FormattingInstruction		An instruction to the provider of information defining the rules for formatting the elements of a Message in accordance with the agreed protocol for the exchange.
MessageMetadata	Metadata MessageElement Concept	MessageSensitivity MessageTimeStamp DataCreatorMetadata PublisherMetadata MessageType RetrievalMetadata SearchMetadata HandlingInstruction	Set of tags and markings (including their established Values) that describe the content of a message.

Table A.1 - IEPPV Taxonomy

MessageMetadata FormattingInstruction	Concept FormattingInstruction		An instruction to the provider of information defining the rules for formatting the elements of MessageMetadata in accordance with the agreed protocol for the exchange.
MessageMetadata RenderingInstruction	StructuredData RenderingInstruction MetadataRendering Instruction Concept RenderingInstruction		An instruction to the recipient of an information exchange defining the rules for rendering or displaying message metadata.
MessageMetadata Specification	Specification MetadataSpecification Concept		The rules governing the assembly of message metadata.
MessageRendering Instruction	Concept RenderingInstruction StructuredData RenderingInstruction		An instruction to the recipient of an information exchange defining the rules for rendering or displaying a message.
MessageSensitivity	MessageMetadata InformationPackage Metadata Concept		Metadata Tag or marking that provides an indication of the sensitivity of the information with reference to privacy, confidentiality, or security.
MessageSpecification	Concept Specification		Specifies the rules and constraints governing the assembly of a community compliant structured or semi-structured message in accordance with a specified message protocol. (e.g., LEXS, EDXL-DE and ATOM)
MessageTimeStamp	Concept MessageMetadata TimeStamp InformationPackage Metadata		Metadata Tag indicating when the Message was created.
MessageType	MessageMetadata InformationPackage Metadata Concept		Metadata tag that identifies the type of message being exchanged.
Metadata	Concept	InformationPackage Metadata PrivacyMetadata SecurityMetadata DataOwnerMetadata SubmitterMetadata MessageMetadata	Data (tags and markings) which describes other data. Source: A Dictionary of the Internet. Darrel Ince. Oxford University Press, 2009. Oxford Reference Online. Oxford University Press.

Table A.1 - IEPPV Taxonomy

MetadataRendering Instruction	RenderingInstruction Concept StructuredData RenderingInstruction	MessageMetadata RenderingInstruction	An instruction to the recipient of an information exchange defining the rules for rendering or displaying metadata.
MetadataSemantic	SemanticElement Concept		A SemanticElement that specifies the rules for assembling the metadata elements.
MetadataSpecification	Concept Specification	MessageMetadata Specification PackageMetadata Specification	The rules governing the assembly of metadata to be attached to a message, package, or information elements of an exchange covered by the contract.
NarrativeText	Concept		Identifies the location and rules for attaching a narrative of free text field to a message or package of information elements.
PackageMetadata Specification	Concept Specification MetadataSpecification		The rules governing the assembly of metadata and tags for an information package.
Participant	Concept		A List of entities to produce or receive the information or message. DODAF: Any entity - human, automated, or any aggregation of human and or automated - that participates in an information exchange agreement.
PersistenceInstruction	ReceiptInstruction Concept ActionInstruction	RetentionInstruction	An instruction to the recipient of an information exchange indicating that the information may be persisted in local stores.
PrivacyMetadata	Metadata Concept		Tags and or markings that support the enforcement of privacy policy.
PublisherMetadata	Concept MessageMetadata		Tags and markings that support the publishing of sharable information to a data registry, repository, or publication-subscription middleware infrastructure. This metadata provides the structures required to represent the data as well as that associated with publishing and storage of data. The data registry, repository or middleware receives and records the published metadata in a manner for users and systems to discover the associated information elements. Derived from: Logical Entity Exchange Specifications 4.0 (LEXS) User Guide (http://130.207.211.107/sites/all/lexs/docs/lexs-4.0/LEXS_4_UserGuide%209-27-2011.pdf)
QualityOfService Instruction	Concept Instruction		An instruction or set of instructions to the producer or publisher of the information specifying the quality of services requirements for the exchange of the information.

Table A.1 - IEPPV Taxonomy

ReceiptInstruction	Instruction MessageElement Concept ActionInstruction	PersistenceInstruction RenderingInstruction ForwardInstruction HandlingInstruction DoNotForward Instruction ValidateInstruction Acknowledge Instruction DoNotPersist Instruction RetentionInstruction	An instruction to the recipient of an information exchange to perform a particular operation, or multiple operations upon the receipt of that information.
ReleasableDataSet	Concept		The assembly of data elements resulting from the enforcement of rules enclosed by a SemanticElement or FilteredSemanticElement.
ReleaseInstruction	Concept ActionInstruction	EncryptInstruction InformationPackage ReleaseInstruction FormattingInstruction	An instruction or set of instructions to the producer or publisher of the information specifying actions to be taken prior to the release of the information. (e.g., Encryption requirements).
RenderingInstruction	ReceiptInstruction Concept Instruction	MetadataRendering Instruction StructuredData RenderingInstruction InformationPackage RenderingInstruction AttachmentSummary RenderingInstruction AttachmentRendering Instruction MessageMetadata RenderingInstruction BinaryDataRendering Instruction MessageRendering Instruction	An instruction or set of instructions to the receiver of information describing the rules for rendering or displaying the information.
RetentionInstruction	PersistenceInstruction ReceiptInstruction Concept ActionInstruction		An instruction to the recipient of an information exchange defining the rules regarding the allowable persistence of the information.
RetrievalMetadata	MessageMetadata Concept		Tags and markings included in a message or information package that assists in the retrieval of that information.

Table A.1 - IEPPV Taxonomy

Safeguard	Concept	SecurityPolicy SecurityMetadata SecurityFilter EncryptInstruction	Policies, rules, services and technologies that serve to guard or protect data and information elements from malicious or inadvertent release of sensitive or protected information. Derived from http://www.thefreedictionary.com/ safeguard, one that serves as protection or a guard
SearchMetadata	Concept MessageMetadata		Refers to metadata that broadly identifies the information elements being sought and results in a response that returns possible candidates for the user to examine further. This metadata provides the characteristics of a query to the registry, repository or publication-subscription infrastructure that responds with information pertaining to the sharable information elements, topics or channels that can be accessed. The response provides the information needed to request specific information elements, topic or channel subscription. The intent is that the requesting entity can narrow the search by reviewing the search response and then request more detailed information on a specific information element, topic, or channel. Depending on the implementation, metadata could include a text-string and request for a text search on unstructured data in a registry or repository (e.g., report), or on structured data, such as a name, attachment or narrative element. A data item metadata search looks for one or more information elements containing information matching the criteria described in the SearchMetadata. Derived from Logical Entity Exchange Specifications 4.0 (LEXS) User Guide (http://130.207.211.107/sites/ all/lexs/docs/lexs-4.0/LEXS_4_UserGuide%209-27-2011.pdf).
SecurityFilter	Concept Safeguard Filter		A specialization of a filter that provides the rules that restrict the assembly of data and information elements based on the values of a security tag or label.
SecurityMetadata	Concept Metadata Safeguard		Tags and Makings that assist in the enforcement of security policy and malicious or inadvertent release of classified information to unauthorized recipients.

Table A.1 - IEPPV Taxonomy

SecurityPolicy	Concept Safeguard		A set of objectives, rules of behavior for users and administrators, and requirements for the configuration, operation and management of computer systems to enhance the security of organization or enterprise people, operations and systems. Note: This specification is focused on the specification of policies and rules for the packaging and release of information for authorized recipients. A Security Policy might include requirements of processes for: 1. Virus detection and prevention; 2. Firewall use and configuration; 3. Password strength and management; 4. Host System administration practices; 5. Access Control rules; 6. Use of Access Logs; 7. Use of screen locking software; 8. Logging out of unattended workstations; 9. Physical security; 10. Account termination; and 11. Procedures for granting and revoking system access.
SemanticAttribute	Attribute Concept		An attribute assigned to a semantic element. Derived from UPDM.
SemanticElement	InformationElement Concept	AttachmentSemantic MetadataSemantic InformationPackage MetadataSemantic DigestSemantic	Composite of rules governing the assembly of data elements in accordance with commitments defined by an information exchange agreement and policies pertaining to the safeguarding of sensitive information. Derived from SOPES IEDM V1: Semantic.
Session	Concept		The software connection to the information dissemination services to be used for the exchange of information under the informationExchangeSpecification. Derived from the Seven Layer Reference Model: 1. Session Layer - Identifies the service of binding two presentation service entities together logically and controls the dialogue between them as far as message synchronization is concerned. 2. Presentation Layer - Provides a set of services that may be selected by the application to enable it to interpret the meaning of the data exchanges. Such services include management of the entity exchange, display and control of the structured data. The presentation layer is the heart of the seven layer proposal, enabling disparate terminal and computer equipment to intercommunicate. A Dictionary of Computing. Ed John Daintith and Edmund Wright. Oxford University Press, 2008. Oxford Reference Online. Oxford University Press.

Table A.1 - IEPPV Taxonomy

SessionSpecification	Concept		Specifies the rules governing communications between the data services and information distribution services (or middleware).
SourceData	Concept	Table Tuple Entity File Triple	Raw data (sometimes called source data or atomic data) is data that has not been processed for use. A distinction is sometimes made between data and information to the effect that information is the end product of data processing. Source: http://searchdatamanagement.techtarget.com/definition/raw-data
Specification	Concept	InformationPackage Specification InformationPayload Specification PackageMetadata Specification MessageMetadata Specification FilteredSemantic Element Attachment Specification DigestSpecification Metadata Specification InformationExchange Specification Distribution Specification Information Specification MessageSpecification	A detailed precise presentation of something. Within the context of the IEPPV, a detailed and precise presentation of rules governing the assembly or processing of information elements. Derived from http://www.merriam-webster.com/dictionary/specification: a detailed precise presentation of something or of a plan or proposal for something.
StaticFilter	Concept Filter		A filter created at design-time that cannot be modified at run-time.
StructuredData Rendering Instruction	RenderingInstruction Concept	MessageMetadata RenderingInstruction AttachmentSummary RenderingInstruction MetadataRendering Instruction MessageRendering Instruction InformationPackage RenderingInstruction	An instruction to the recipient of an information exchange defining the rules for rendering or displaying structured data.

Table A.1 - IEPPV Taxonomy

SubmitterMetadata	Metadata Concept		Tags and markings identifying the submitter of the information.
SubtendedElement	Concept InformationElement		An Element (TransactionalElement or WrapperElement) forming part of another element (TransactionalElement or SemanticElement). Wrapper is always a subtended information element since it cannot exist outside of a TransactionalElement definition.
SubtendedElementAttribute	Attribute Concept		An attribute assigned to a SubtendedElement.
SubtendedTransactional	TransactionalElement Concept		A TransactionalElement included as part of another TransactionalElement or SemanticElement.(aka Supporting Transactional).
Table	SourceData Concept		A collection of records. Each record may store information associated with a key by which specific records are found, or the records may be arranged in an array so that the index is the key. In commercial applications the word table is often used as a synonym for matrix or array. Source: A Dictionary of Computing. Ed John Daintith and Edmund Wright. Oxford University Press, 2008. Oxford Reference Online. Oxford University Press.
TimeStamp	Concept	MessageTimeStamp	A tag or mark indicating the time when the message was created.
TransactionalAttribute	Attribute Concept		An attribute assigned to a TransactionalElement. Derived from UPDM.
TransactionalElement	InformationElement Concept	Enclosing TransactionalElement WatchPoint TransactionalElement Subtended Transactional Enclosed TransactionalElement	Specifies a reusable pattern comprising rules governing the assembly and processing of data and information elements. Derived from SOPES IEDM V1: Transactional.
Transformation	Concept		The conversion of data from one form to another. In this instance the specification of rules governing the conversion or transformation of data. Source: A Dictionary of Computing. Ed John Daintith and Edmund Wright. Oxford University Press, 2008. Oxford Reference Online. Oxford University Press.
Transformation ResultingAttribute	Attribute Concept		An attribute resulting from a transformationElement.

Table A.1 - IEPPV Taxonomy

Triple	Concept SourceData		An RDF triple consists of three components: - the subject, which is an IRI or a blank node; - the predicate, which is an IRI; and - the object, which is an IRI, a literal or a blank node. An RDF triple is conventionally written in the order subject, predicate, object. Source: http://www.w3.org/TR/2013/CR-rdf11-concepts-20131105/#section-triples
Tuple	Concept SourceData		An ordered set with an unspecified but finite number (n) of elements. Source: A Dictionary of Computing. Ed John Daintith and Edmund Wright. Oxford University Press, 2008. Oxford Reference Online.
ValidateInstruction	ActionInstruction Concept ReceiptInstruction		An instruction to the recipient of an information exchange containing criteria for the validation of the content and semantics of the message or information payload.
WatchPoint	Concept		A trigger mechanism used by an application to commence the assembly of a TransactionalElements. A data model assigns this tagged value to a WrapperElement aggregation arc in the Transactional pattern. Additions to the underlying data store for this WrapperElement triggers the application to start building the composite. Derived from SOPES IEDM V1: Wrapper
WatchPoint TransactionalElement	TransactionalElement Concept		A TransactionalElement with an associated Watchpoint data event that triggers the assembly of an enclosing TransactionalElements and SemanticElements. Source: Derived from SOPES IEDM V1.
WrapperAttribute	Concept Attribute		An attribute assigned to a WrapperElement. Source: derived from UPDM.
WrapperElement	Concept InformationElement		A logical construct that wraps or encapsulates the definition of a data set, table entity, triple, file, etc. A Wrapper directly maps to a data instance (e.g., row of data in a database application) in the logical data model and the physical data model. Derived from Derived from SOPES IEDM V1: Wrapper

Annex B IEPPV UML Profile (Normative)

B.1 Model Elements

B.1.1 Overview

This profile employs the concepts provided by the Information Exchange Packaging Policy Vocabulary (IEPPV) to customize UML for the expression of rules for assembling and processing information and message elements utilized in an information exchange. It provides users with the ability to develop policy models that align business policies (e.g., information sharing, security, and privacy) with specific information domains in a manner that the policy model can be integrated into a user's broader Enterprise Architecture.

The model elements comprising the IEPPV Profile define general-purpose constructs for specifying, designing, and implementing the data patterns, business rules and constraints (e.g., data or domain filters) for the packaging of shareable information or datasets. The information packages can then be assigned to specific peer-to-peer or community information sharing agreements.

The concepts expressed in the IEPPV are an extension of those developed for the Shared Operational Picture Exchange Services (SOPES) Information Exchange Data Model (IEDM) later as an extension to the UML Profile for DODAF and MODAF (UPDM v2.1). The following table outlines the changes in the terms used for the individual concepts. However, the concepts themselves have not changed.

Table B.1 - IEPPV to SOPES IEDM Concept Mapping

#	IEPPV Concept	SOPES and UPDM Concept
1	SemanticElement	Semantic
2	TransactionalElement	Transactional
3	WrapperElement	Wrapper
4	FilteredSematicElement	FilteredSemantic
5	FilteredTransactionalElement	FilteredTransactional
6	Filter	DynamicFilter
	Filter	StaticFilter
7	InformationExchangeSpecification	Contract

B.1.2 Representing Stereotype Constraints

The IEPPV has adopted the same approach as the Unified Profile for DODAF and MODAF to develop this profile. This approach will facilitate the integration of the IEPPV into the UPDM V3.0 to replace the SOPES profile integrated into UPDM 2.1. The following material was extracted from sub clause 7.4 and sub clause 7.5 of the UPDM Specification.

B.1.3 Representing Stereotype Constraints

The profile uses a non-standard notation to represent stereotype constraints in the profile to improve readability of the profile.

"metaconstraint" dependency

"metaconstraint" is a stereotype that extends the Dependency metaclass. It is used to specify constrained elements within the profile. A sample of the "metaconstraint" dependency is a diagram for stereotype extending the Dependency metaclass. See the following example:

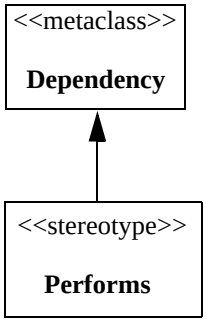

Figure B.1 - metaconstraint

Performs is a stereotype that extends Dependency. The constraint on this stereotype is that its client end must be stereotyped by a Performer and its supplier end must be stereotyped by Activity. But as this constraint is not visible, the diagram does not communicate the needed information. We are using the "metaconstraint" dependency to visualize the constraint.

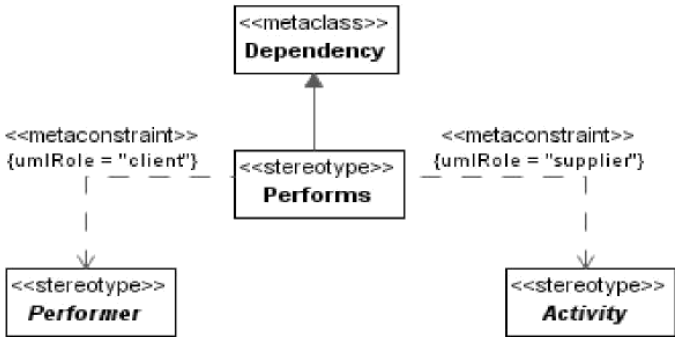

Figure B.2 - Performs Hierarchy

This diagram should be read as follows:

> *Performs is a stereotype extending the Dependency metaclass and is used for modeling a relationship between a Performer (or its specializations) and an Activity (or its specializations). A Dependency stereotyped Performs must have its values for the client property stereotyped as Performer and its values for the supplier property must be stereotyped Activity.*

The «metaconstraint» dependency will appear only in the specification diagrams, but not the profile XMI.

Note – When stereotype extends Connector, the stereotype property umlRole has values “end[0].role” and “end[1].role.” For example:

This is done because Connector has no direct “linkage” to the connected element; it links to the Connector Ends, which references the linked element. So, end[n] gives the reference to the ConnectorEnd, and role gives the reference to the linked element.

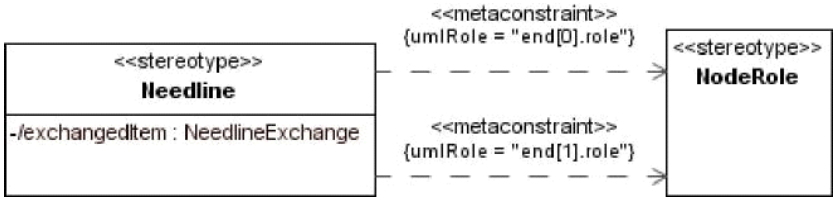

Figure B.3 - Connector Extension

“metarelationship” dependency “metarelationship” is a stereotype for dependency, showing that certain domain concepts will be implemented using regular UML relationships.

For example: A Capability may depend on other Capabilities, but this concept cannot be visualized on the diagram:

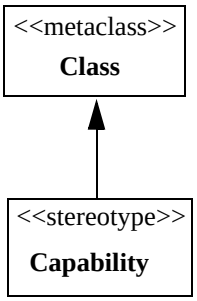

Figure B.4 - Capabilities Generalization

We are using the “metarelationship” dependency to visualize the dependency concept.

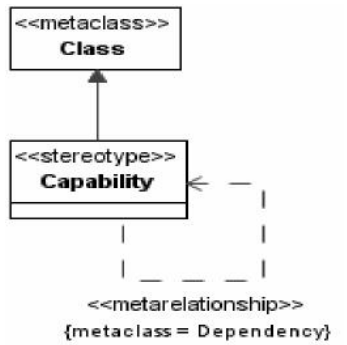

Figure B.5 - Visualizing “metarelationship”

This diagram should be read as follows:

- Capability may have other Capabilities related to it, using the UML Dependency metaclass.
- The “metarelationship” dependency will appear only in the specification diagrams, but not the profile XMI.

“stereotyped relationship” dependency

Although the “metaconstraint” dependency creates a good way to show the constrained ends of the stereotyped relationship, it also creates some overhead when showing the relationship between two stereotypes. For example, Figure 7.6 shows that one of the set of elements that are representative of the abstract element CapableElement Exhibits a Capability. A «stereotyped relation» is specified and then applied to express the constraint. First, the necessary «Exhibits» stereotype is specified.

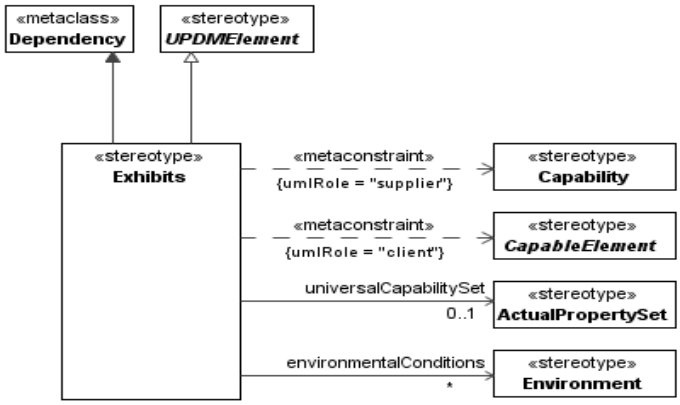

Figure B.6 - “Exhibits” extends the UML Dependency metaclass

Then, the “stereotyped relationship” dependency can then be used as follows:

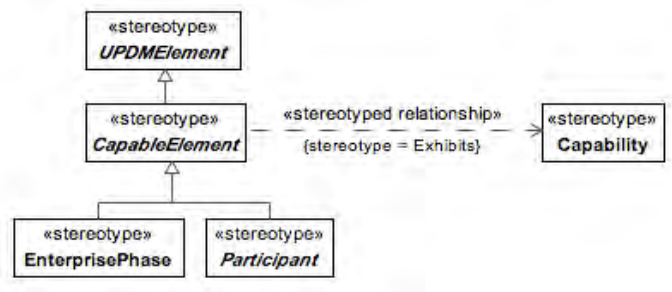

Figure B.7 - Use of the Exhibits “stereotyped relationship” dependency

The “stereotyped relationship” dependency appears only in the specification diagrams and not within the profile XMI.

B.2 IEPPV Profile

The following set of diagrams defines the IEPPV Profile for UML. It forms one in a series of Platform Specific uses for the IEPPV.

B.2.1 InformationExchangeSpecification - CP-1

The following figure identifies the modeling element used to define an InformationExchangeSpecification conforming to CP-1.

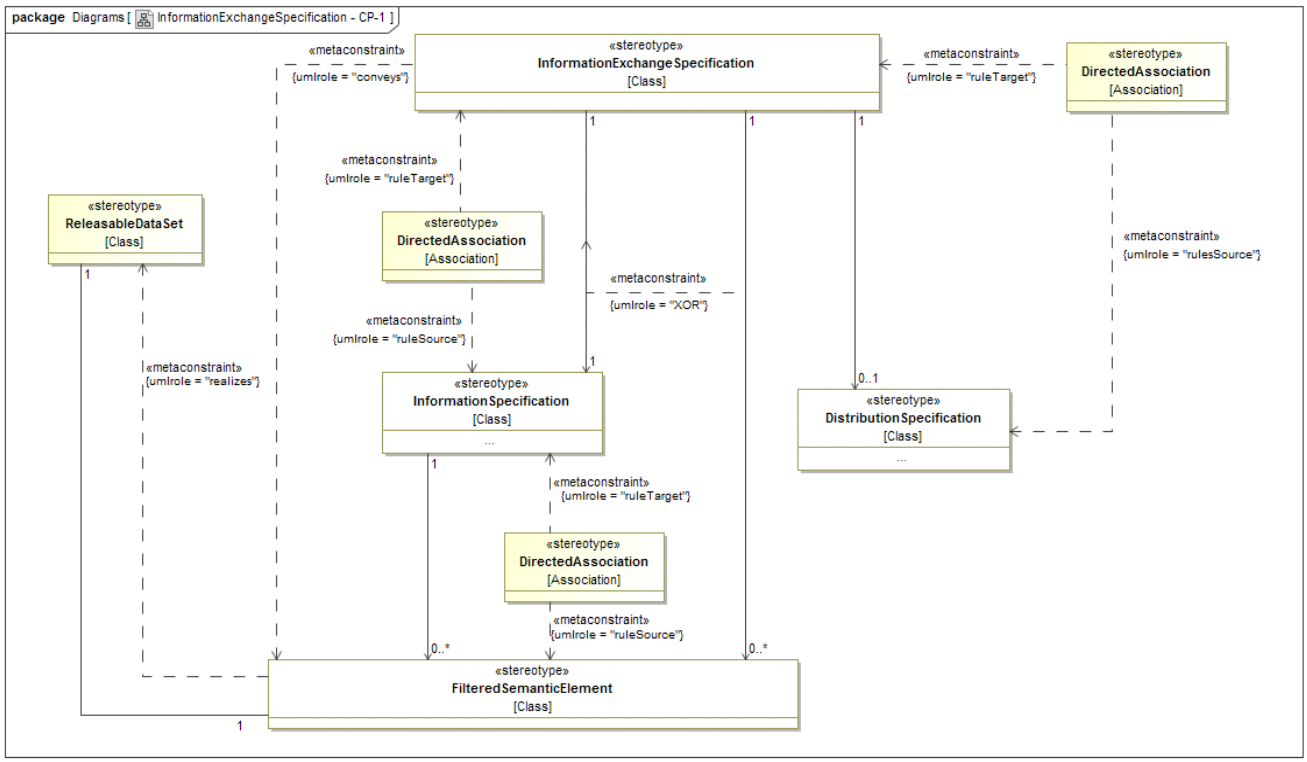

Figure B.8 - InformationExchangeSpecification - CP-1

«Class» Extensions

The IEPPV Profile includes the following extensions to stereotype «Class»:

DistributionSpecification	A Class that encloses the rules governing the distribution of an InformationElement. Element of an InformationExchangeSpecification that links the InformationSpecification to the information dissemination services (e.g., User Application, Service Interface and Middleware).
FilteredSemanticElement	A Class that encloses rules for the assignment of one or more DynamicFilters to a specified SemanticElement.
InformationExchange Specification	A Class that encloses the rules governing the assembly, processing, and dissemination of information.
InformationSpecification	A Class enclosing the set of Messages or FilteredSemantics permitted under the InformationExchangeSpecification.
ReleasableDataSet	The realization of a FilteredSemanticElement, SemanticElement. An assembled set of data elements conforming to policy.

Constraints

The following constraints are illustrated in the InformationExchangeSpecification - CP-1:

DirectedAssociation.ruleTarget:	[A directed association between the rule Target and the ruleSource. Values for the patternTarget must be stereotyped with «Specification» or its specializations. Identifies that the ruleTarget includes the policies, rules and constraints included in the ruleSource.]
DirectedAssociation.ruleSource:	[A directed association between the rule Target and the ruleSource. Values for the patternSource must be stereotyped with «Specification» or its specializations. Identifies that the ruleSource's policies, rules and constraints included in the ruleTarget.]

B.2.2 InformationExchangeSpecification - CP-2a,b&c

The following figure identifies the modeling element used to define an InformationExchangeSpecification conforming to CP-2a,b&c.

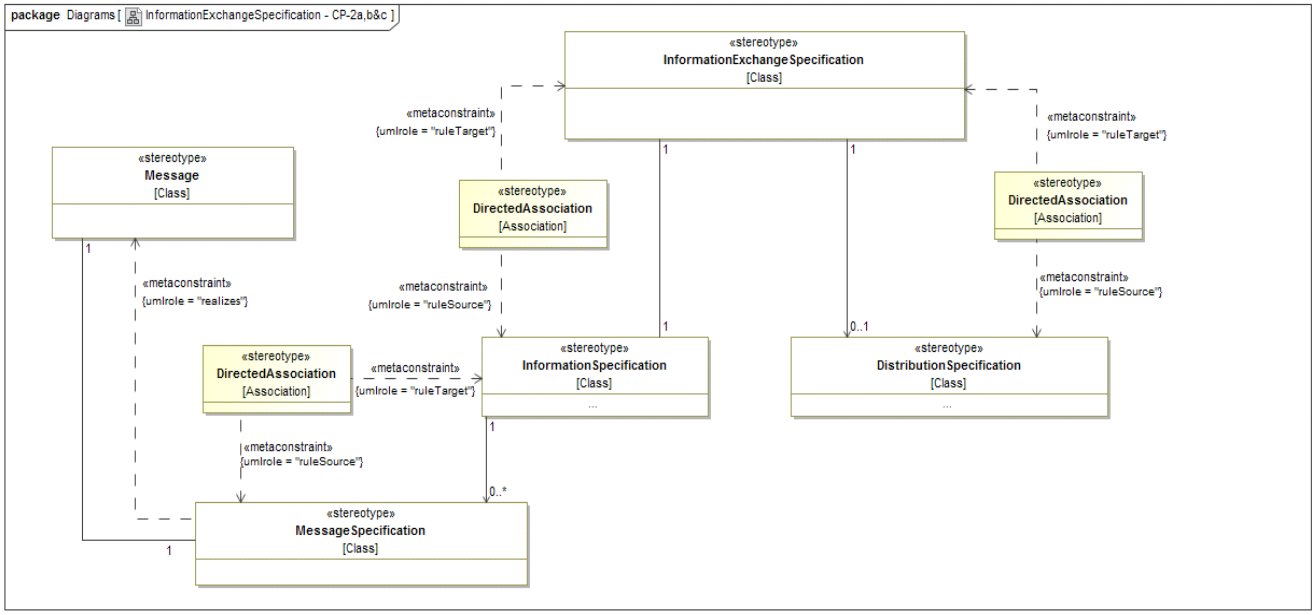

Figure B.9 - InformationExchangeSpecification - CP-2a,b&c

«Class» Extensions

The IEPPV Profile includes the following extensions to stereotype «Class»:

DistributionSpecification	A Class that encloses the rules governing the distribution of an InformationElement. Element of an InformationExchangeSpecification that links the InformationSpecification to the information dissemination services (e.g., User Application, Service Interface and Middleware).
InformationExchange Specification	A Class that encloses the rules governing the assembly, processing and dissemination of information.
InformationSpecification	A Class enclosing the set of Messages or FilteredSemantics permitted under the InformationExchangeSpecification.
Message	A Realization of a MessageSpecification. The unit of information transferred by a message switching system (or Network). Messages may be of any length, from a few bits to a complete file, and no part of a message is released to its final recipient until all of the message has been received at the network node adjacent to the destination.
MessageSpecification	A Class enclosing the rules governing the assembly and processing of a community compliant structured or semi-structured message in accordance with a specified packaging profile (e.g., LEXS, EDXL-DE, and ATOM).

Constraints

The following constraints are illustrated in the InformationExchangeSpecification - CP-2a,b&c:

DirectedAssociation.ruleTarget:	[A directed association between the rule Target and the ruleSource. Values for the patternTarget must be stereotyped with «Specification» or its specializations. Identifies that the ruleTarget includes the policies, rules and constraints included in the ruleSource.]
DirectedAssociation.ruleSource:	[A directed association between the rule Target and the ruleSource. Values for the patternSource must be stereotyped with «Specification» or its specializations. Identifies that the ruleSource's policies, rules and constraints included in the ruleTarget.]

B.2.3 Message Specification - CP-2a

The following figure illustrates the modeling relationships for a MessageSpecification under CP-2a.

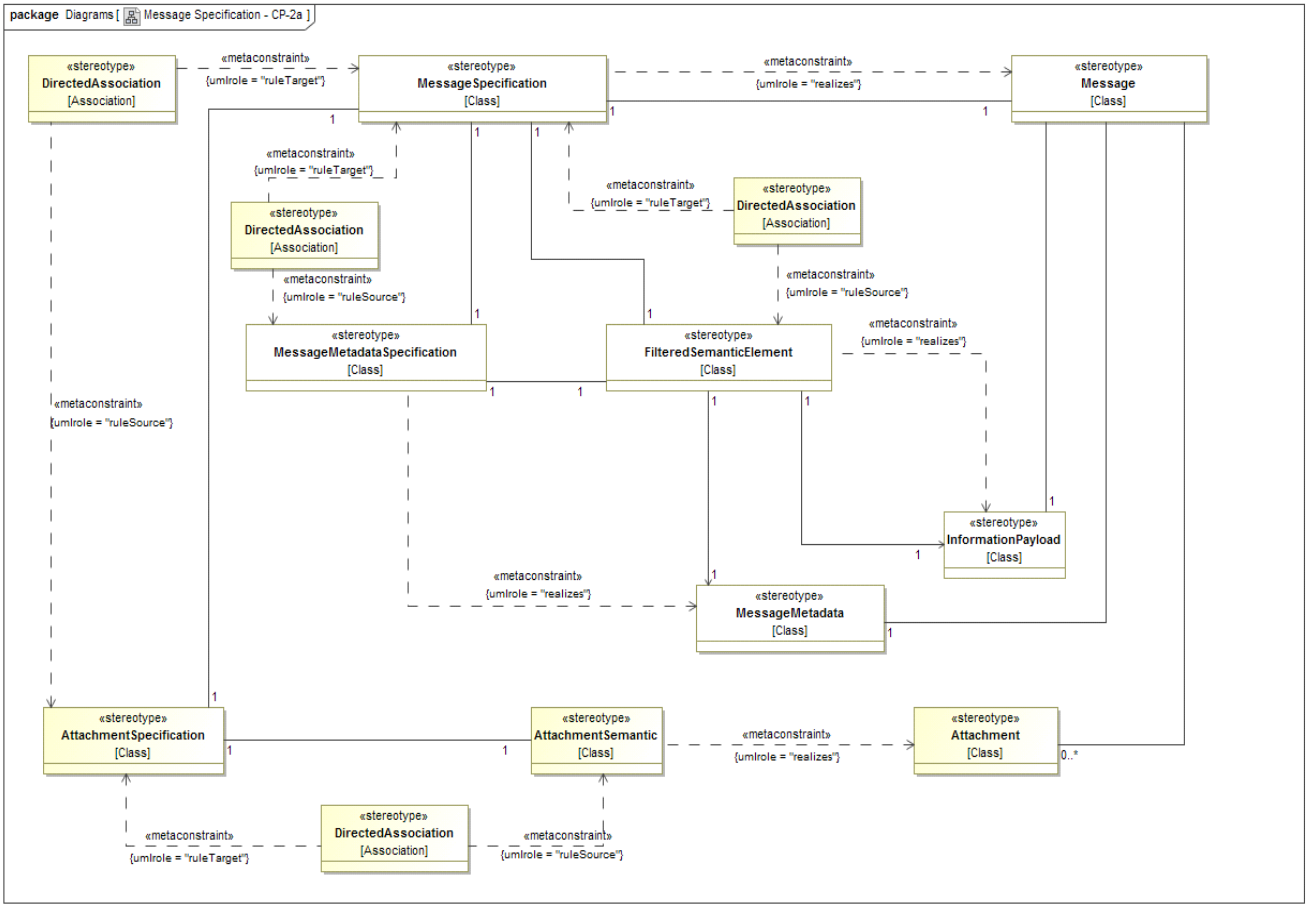

Figure B.10 - Message Specification - CP-2a

«Class» Extensions

The IEPPV Profile includes the following extensions to stereotype «Class»:

Attachment	A Class used to specify a binary file (e.g., PDF file, image or video) or document, and information about the binary or document, such as the size and type and description.
AttachmentSemantic	A Class that encloses the rules governing the assembling and inclusion of attachments to a message. It also provides the rules for generating an attachment summary and linkages.
FilteredSemanticElement	A Class that encloses rules for the assignment of one or more DynamicFilters to a specified SemanticElement.
InformationPayload	The Realization of a FilteredSemantic. A formatted dataset without protocols and metadata required for an information exchange.

Message	A Realization of a MessageSpecification. The unit of information transferred by a message switching system (or Network). Messages may be of any length, from a few bits to a complete file, and no part of a message is released to its final recipient until all of the message has been received at the network node adjacent to the destination.
MessageMetadata	The Realization of a MessageMetadataSpecification. Set of tags and markings (including their established Values) that describes the content of a message.
MessageMetadata Specification	A Class that encloses the rules governing the assembly of MessageMetadata.
MessageSpecification	A Class enclosing the rules governing the assembly and processing of a community compliant structured or semi-structured message in accordance with a specified packaging profile (e.g., LEXS, EDXL-DE, and ATOM).

Constraints

The following constraints are illustrated in the Message Specification - CP-2a:

DirectedAssociation.ruleTarget:	[A directed association between the rule Target and the ruleSource. Values for the patternTarget must be stereotyped with «Specification» or its specializations. Identifies that the ruleTarget includes the policies, rules and constraints included in the ruleSource.]
DirectedAssociation.ruleSource:	[A directed association between the rule Target and the ruleSource. Values for the patternSource must be stereotyped with «Specification» or its specializations. Identifies that the ruleSource's policies, rules and constraints included in the ruleTarget.]

B.2.4 Message Specification - CP-2b&c

The following figure illustrates the modeling relationships for a MessageSpecification under CP-2b&c.

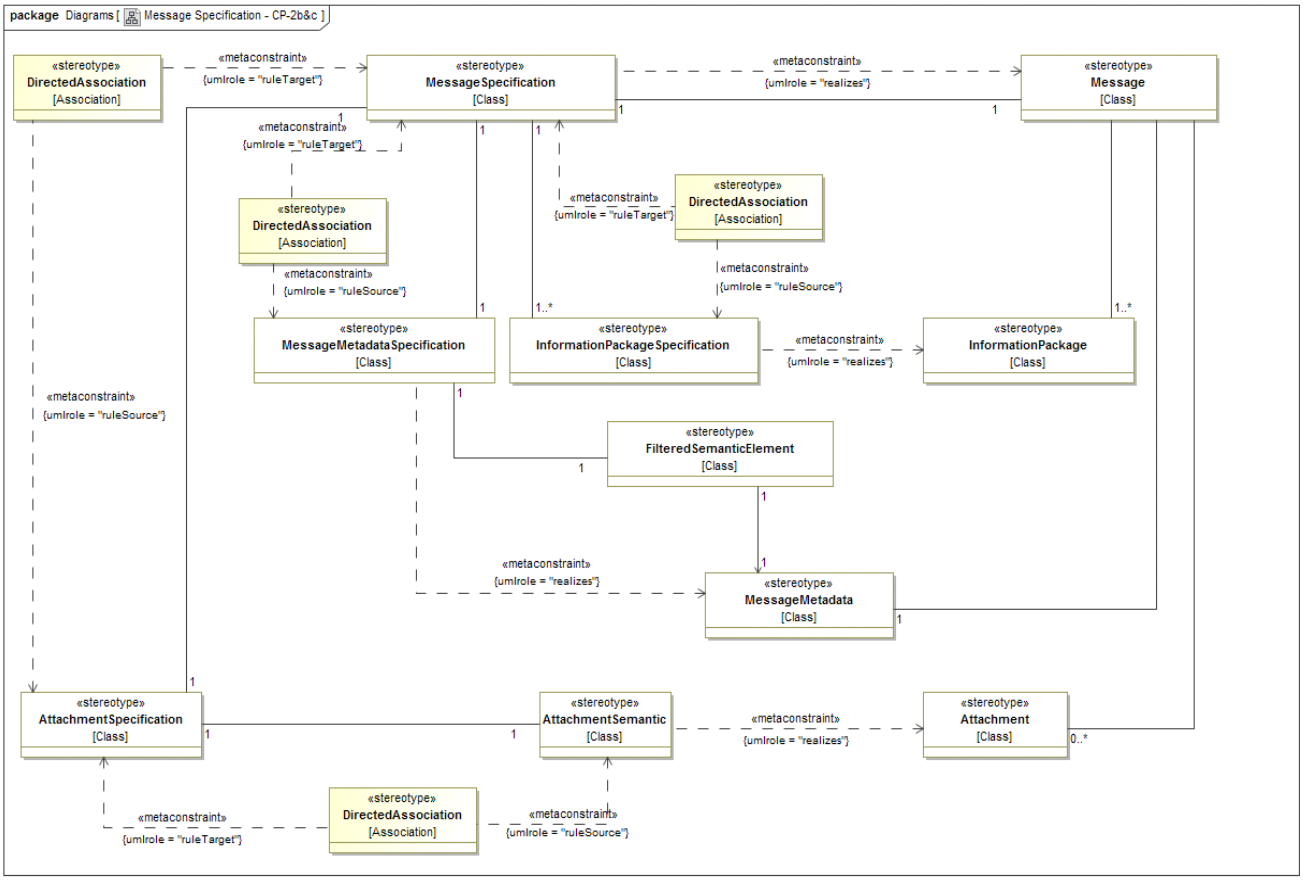

Figure B.11 - Message Specification - CP-2c&c

«Class» Extensions

The IEPPV Profile includes the following extensions to stereotype «Class»:

Attachment	A Class used to specify a binary file (e.g., PDF file, image or video) or document, and information about the binary or document, such as the size and type and description.
AttachmentSemantic	A Class that encloses the rules governing the assembling and inclusion of attachments to a message. It also provides the rules for generating an attachment summary and linkages.
AttachmentSpecification	A Class that encloses rules (AttachmentSemantics) governing attachment of binary information elements to an information exchange or message.
FilteredSemanticElement	A Class that encloses rules for the assignment of one or more DynamicFilters to a specified SemanticElement.

InformationPackage	A Class enclosing the rules governing the assembly and processing of an Information Package. A standard representation of structured, semi-structured and binary information applicable to an information sharing agreement. Packages may contain metadata, a Digest, a Structured Payload, Rendering Instructions, and optional linkages depending on the established agreements.
InformationPackage Specification	A Class enclosing the rules governing the construction preparation of an InformationPackage.
Message	A Realization of a MessageSpecification. The unit of information transferred by a message switching system (or Network). Messages may be of any length, from a few bits to a complete file, and no part of a message is released to its final recipient until all of the message has been received at the network node adjacent to the destination.
MessageMetadata	The Realization of a MessageMetadataSpecification. Set of tags and markings (including their established Values) that describes the content of a message.
MessageMetadata Specification	A Class that encloses the rules governing the assembly of MessageMetadata.
MessageSpecification	A Class enclosing the rules governing the assembly and processing of a community compliant structured or semi-structured message in accordance with a specified packaging profile (e.g., LEXS, EDXL-DE, and ATOM).

Constraints

The following constraints are illustrated in the Message Specification - CP-2b&c:

DirectedAssociation.ruleTarget:	[A directed association between the rule Target and the ruleSource. Values for the patternTarget must be stereotyped with «Specification» or its specializations. Identifies that the ruleTarget includes the policies, rules and constraints included in the ruleSource.]
DirectedAssociation.ruleSource:	[A directed association between the rule Target and the ruleSource. Values for the patternSource must be stereotyped with «Specification» or its specializations. Identifies that the ruleSource's policies, rules and constraints included in the ruleTarget.]

B.2.5 Information Package Specification - CP-2b

The following figure identifies the modeling element used to define an InformationPackageSpecification conforming to CP-2b.

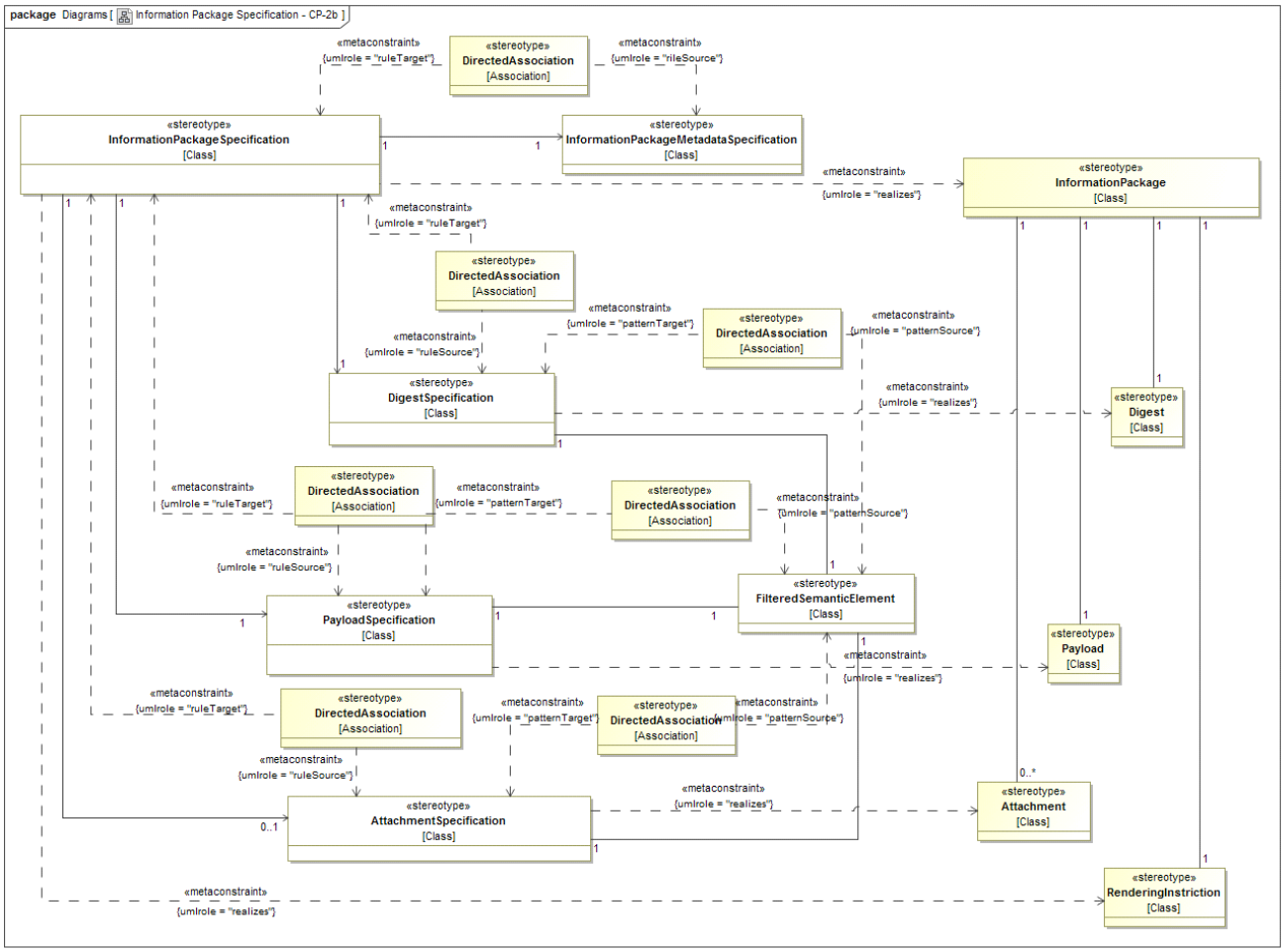

Figure B.12 - Information Package Specification - CP-2b

«Class» Extensions

The IEPPV Profile includes the following extensions to stereotype «Class»:

Attachment	A Class used to specify a binary file (e.g., PDF file, image or video) or document, and information about the binary or document, such as the size and type and description.
AttachmentSpecification	A Class that encloses rules (AttachmentSemantics) governing attachment of binary information elements to an information exchange or message.

Digest	A Realization of a DigestSpecification. An information structure, format and syntax common to all communities. It provides the ability for systems to handle heterogeneous data without having to understand the specific context and or semantics of the source. As long as the entities relevant to the packaged data items are represented in the Digest, users will be able to discover, link, map, etc. the information within.
DigestSpecification	A Class enclosing the rules governing the assembly or processing of a digest.
FilteredSemanticElement	A Class that encloses rules for the assignment of one or more DynamicFilters to a specified SemanticElement.
InformationPackage	A Class enclosing the rules governing the assembly and processing of an Information Package. A standard representation of structured, semi-structured and binary information applicable to an information sharing agreement. Packages may contain metadata, a Digest, a Structured Payload, Rendering Instructions, and optional linkages depending on the established agreements.
InformationPackage MetadataSpecification	A Class enclosing the rules governing the assembly and processing of tags and markings that identify and describe the contents of an information package.
InformationPackage Specification	A Class enclosing the rules governing the construction preparation of an InformationPackage.
Payload	Realization of a semantic or filtered Semantic.
PayloadSpecification	A class enclosing the rules governing the assembly and processing of a structured dataset for an information exchange.
RenderingInstruction	A Class containing the location of an instruction or set of instructions to the receiver of information describing the rules for rendering or displaying the information.

Constraints

The following constraints are illustrated in the Information Package Specification - CP-2b:

DirectedAssociation.ruleTarget:	[A directed association between the rule Target and the ruleSource. Values for the patternTarget must be stereotyped with «Specification» or its specializations. Identifies that the ruleTarget includes the policies, rules and constraints included in the ruleSource.]
DirectedAssociation.ruleSource:	[A directed association between the rule Target and the ruleSource. Values for the patternSource must be stereotyped with «Specification» or its specializations. Identifies that the ruleSource's policies, rules and constraints included in the ruleTarget.]

***DirectedAssociation.pattern Source**:*	[A directed association between the patternTarget and the patternSource. Values for the patternSource must be stereotyped with «Pattern» or its specializations. The association identifies the pattern defined by the patternSource (Transactional or Semantic) is used by the patternTarget.]
***DirectedAssociation.pattern Target**:*	[A directed association between the patternTarget and the patternSource. Values for the patternTarget must be stereotyped with «Specification» or its specializations. The association identifies that the patternTarget uses by the pattern defined by the patternSource.]

B.2.6 Information Package Specification - CP-2c

The following figure identifies the modeling element used to define and InformationPackageSpecification conforming to CP-2b&c.

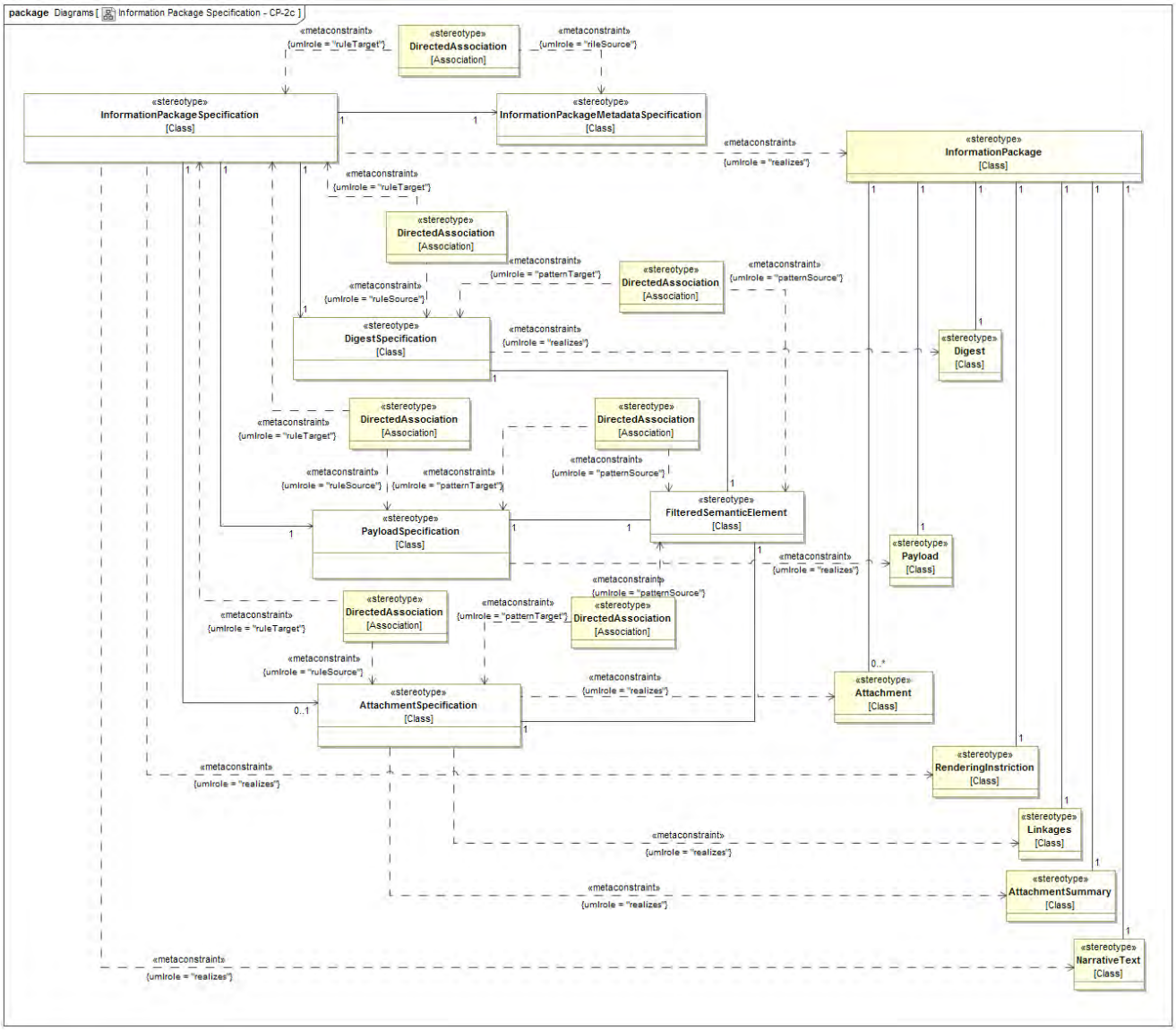

Figure B.13 - Information Package Specification - CP-2c

«Class» Extensions

The IEPPV Profile includes the following extensions to stereotype «Class»:

Attachment	A Class used to specify a binary file (e.g., PDF file, image or video) or document, and information about the binary or document, such as the size and type and description.
AttachmentSpecification	A Class that encloses rules (AttachmentSemantics) governing attachment of binary information elements to an information exchange or message.
AttachmentSummary	Realization of an AttachmentSpecification that provides a list of attachments associated to a specific data package.
Digest	A Realization of a DigestSpecification. An information structure, format and syntax common to all communities. It provides the ability for systems to handle heterogeneous data without having to understand the specific context and or semantics of the source. As long as the entities relevant to the packaged data items are represented in the Digest, users will be able to discover, link, map, etc. the information within.
DigestSpecification	A Class enclosing the rules governing the assembly or processing of a digest.
FilteredSemanticElement	A Class that encloses rules for the assignment of one or more DynamicFilters to a specified SemanticElement.
InformationPackage	A Class enclosing the rules governing the assembly and processing of an Information Package. A standard representation of structured, semi-structured and binary information applicable to an information sharing agreement. Packages may contain metadata, a Digest, a Structured Payload, Rendering Instructions, and optional linkages depending on the established agreements.
InformationPackage MetadataSpecification	A Class enclosing the rules governing the assembly and processing of tags and markings that identify and describe the contents of an information package.
InformationPackage Specification	A Class enclosing the rules governing the construction preparation of an InformationPackage.
Linkages	The realization of an AttachmentSpecification that provides References from an information package to related Attachments.
NarrativeText	A Class holding the location and rules for attaching a narrative of free text field to a message or package of information elements.
Payload	Realization of a semantic or filtered Semantic.
PayloadSpecification	A class enclosing the rules governing the assembly and processing of a structured dataset for an information exchange.
RenderingInstruction	A Class containing the location of an instruction or set of instructions to the receiver of information describing the rules for rendering or displaying the information.

Constraints

The following constraints are illustrated in the Information Package Specification - CP-2b:

DirectedAssociation.ruleTarget:	[A directed association between the rule Target and the ruleSource. Values for the patternTarget must be stereotyped with «Specification» or its specializations. Identifies that the ruleTarget includes the policies, rules and constraints included in the ruleSource.]
DirectedAssociation.ruleSource:	[A directed association between the rule Target and the ruleSource. Values for the patternSource must be stereotyped with «Specification» or its specializations. Identifies that the ruleSource's policies, rules and constraints included in the ruleTarget.]
DirectedAssociation.pattern Source:	[A directed association between the patternTarget and the patternSource. Values for the patternSource must be stereotyped with «Pattern» or its specializations. The association identifies the pattern defined by the patternSource (Transactional or Semantic) is used by the patternTarget.]
DirectedAssociation.pattern Target:	[A directed association between the patternTarget and the patternSource. Values for the patternTarget must be stereotyped with «Specification» or its specializations. The association identifies that the patternTarget uses by the pattern defined by the patternSource.]

B.2.7 FilteredSemanticElement

The following figure illustrates the modeling relationships for a FilteredSemanticElement.The FilteredSemantic is modeled as a Class Diagram and overlays a set of run-time configurable domain filters on a SemanticElement. The FilteredSemanticElement encloses the FilteredTransactionalElements that define the specific filters.

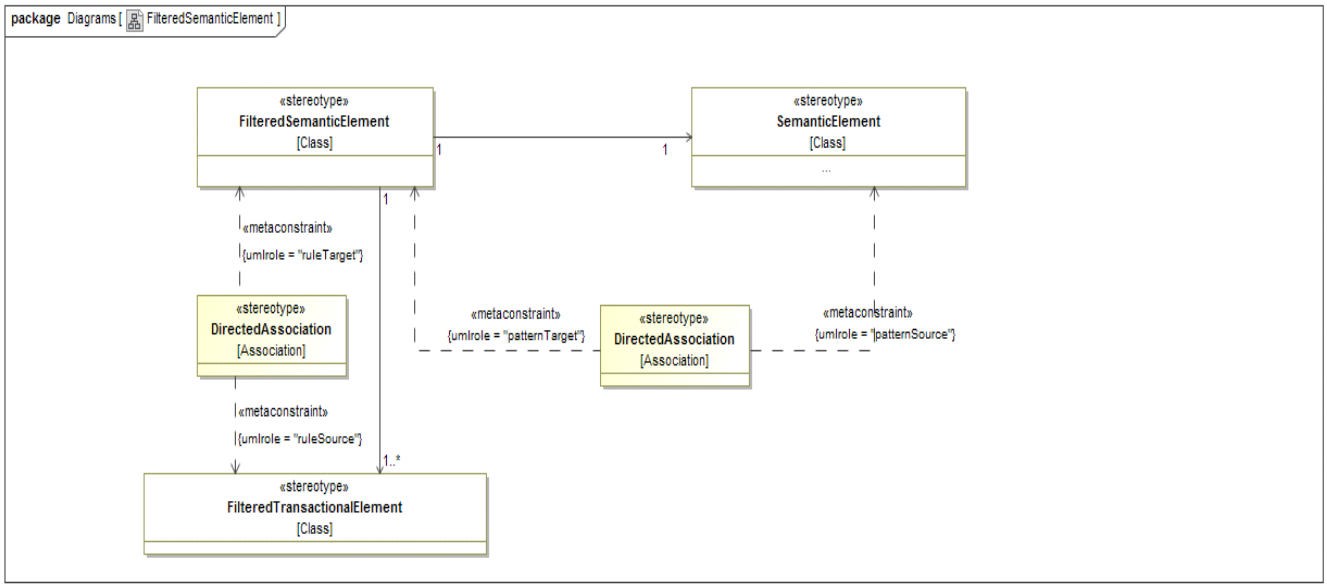

Figure B.14 - FilteredSemanticElement

«Class» Extensions

The IEPPV Profile includes the following extensions to stereotype «Class»:

FilteredSemanticElement	A Class that encloses rules for the assignment of one or more DynamicFilters to a specified SemanticElement.
FilteredTransactionalElement	A Class that encloses the specification of rules for setting which WrapperAttributes (enclosed by the SemanticElement) are filterable at runtime.
SemanticElement	A Class enclosing the rules governing the assembly of data elements in accordance with policy.

Constraints

The following constraints are illustrated in the FilteredSemanticElement:

DirectedAssociation.ruleTarget:	[A directed association between the rule Target and the ruleSource. Values for the patternTarget must be stereotyped with «Specification» or its specializations. Identifies that the ruleTarget includes the policies, rules and constraints included in the ruleSource.]

DirectedAssociation.ruleSource:	[A directed association between the rule Target and the ruleSource. Values for the patternSource must be stereotyped with «Specification» or its specializations. Identifies that the ruleSource's policies, rules and constraints included in the ruleTarget.]
DirectedAssociation.patternSource:	[A directed association between the patternTarget and the patternSource. Values for the patternSource must be stereotyped with «Pattern» or its specializations. The association identifies the pattern defined by the patternSource (Transactional or Semantic) is used by the patternTarget.]
DirectedAssociation.patternTarget:	[A directed association between the patternTarget and the patternSource. Values for the patternTarget must be stereotyped with «Specification» or its specializations. The association identifies that the patternTarget uses by the pattern defined by the patternSource.]

B.2.8 FilteredTransactionalElement

The following figure illustrates the modeling relationships for a FilteredTransactionalElement.

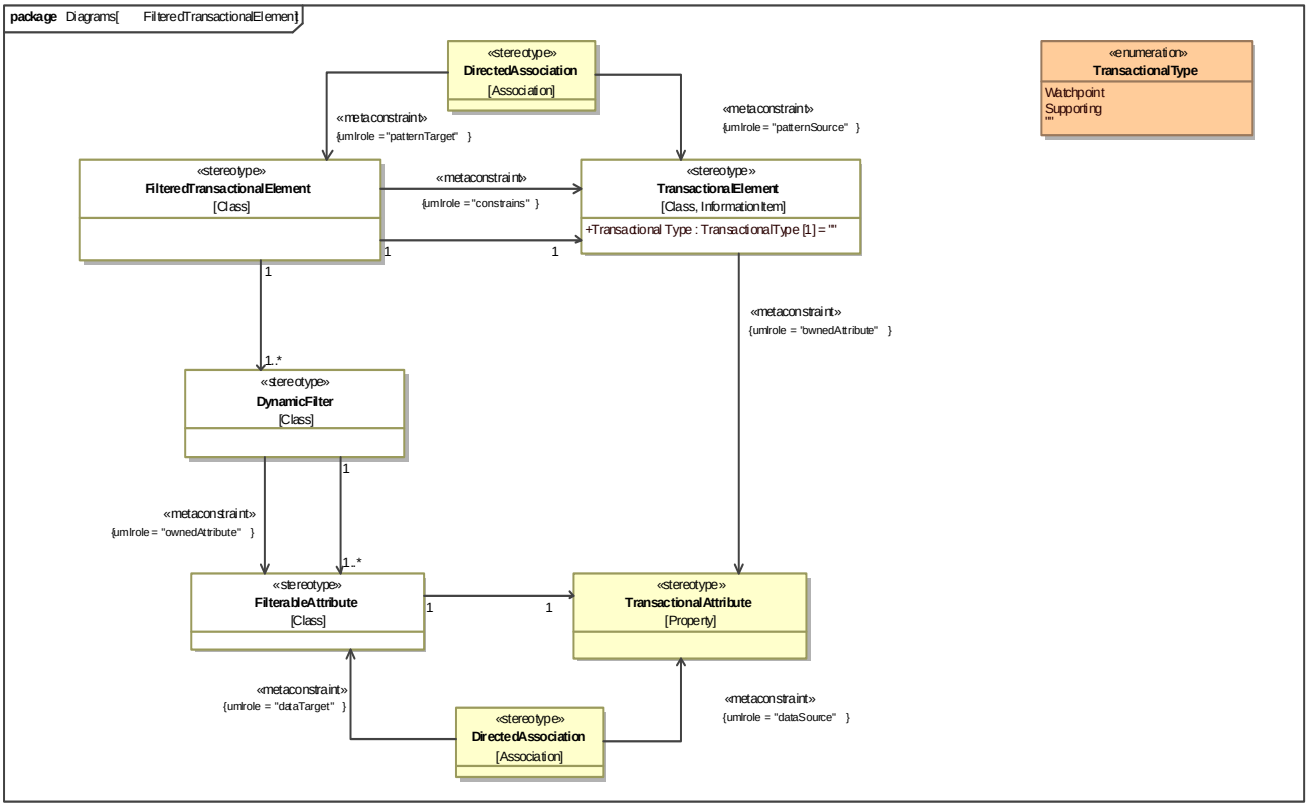

Figure B.15 - FilteredTransactionalElement

«Class» Extensions

The IEPPV Profile includes the following extensions to stereotype «Class»:

DynamicFilter	The Class enclosing the rules for domain filters whose parameters may be configured at run-time.
FilterableAttribute	An Attribute that is used to constrain the assembly of data and information elements.
FilteredTransactionalElement	A Class that encloses the specification of rules for setting which WrapperAttributes (enclosed by the SemanticElement) are filterable at runtime.
TransactionalElement	A Class that encloses the rules governing the assembly and processing of data and information elements.

Constraints

The following constraints are illustrated in the FilteredTransactionalElement:

DirectedAssociation.patternSource:	[A directed association between the patternTarget and the patternSource. Values for the patternSource must be stereotyped with «Pattern» or its specializations. The association identifies the pattern defined by the patternSource (Transactional or Semantic) is used by the patternTarget.]
DirectedAssociation.patternTarget:	[A directed association between the patternTarget and the patternSource. Values for the patternTarget must be stereotyped with «Specification» or its specializations. The association identifies that the patternTarget uses by the pattern defined by the patternSource.]

B.2.9 SemanticElement

The following figure illustrates the modeling relationships for a SemanticElement.

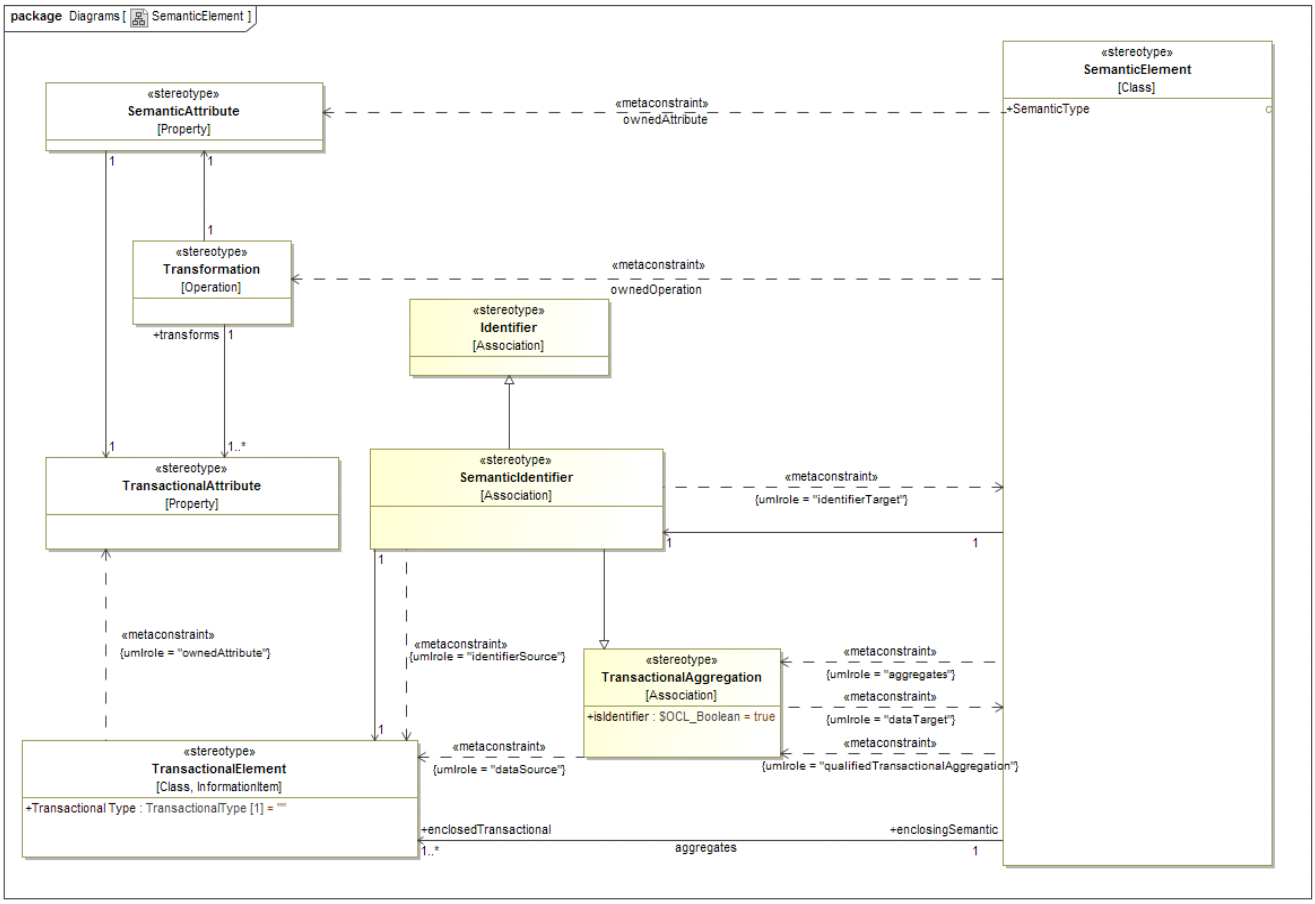

Figure B.16 - Semantic Element

«Class» Extensions

The IEPPV Profile includes the following extensions to stereotype «Class»:

SemanticElement	A Class enclosing the rules governing the assembly of data elements in accordance with policy.
TransactionalElement	A Class that encloses the rules governing the assembly and processing of data and information elements.

Constraints

The following constraints are illustrated in the SemanticElement:

SemanticIdentifier.identifier Source:	[A SemanticIdentifier aggregation will provide one Identifier Transactional (source) at its part end.]
SemanticIdentifier.identifier Target:	[A SemanticIdentifier aggregation will provide one Semantic(target) at its whole end.]
TransactionalAggregation. dataTarget:	[The information aggregate at the whole end of the aggregation must be stereotyped «Semantic» or «Transactional».]
TransactionalAggregation. dataSource:	[The information item at the part end will be stereotyped with «Transactional».]

B.2.10 TransactionalElement

The following figure illustrates the modeling relationships for a TransactionalElement.

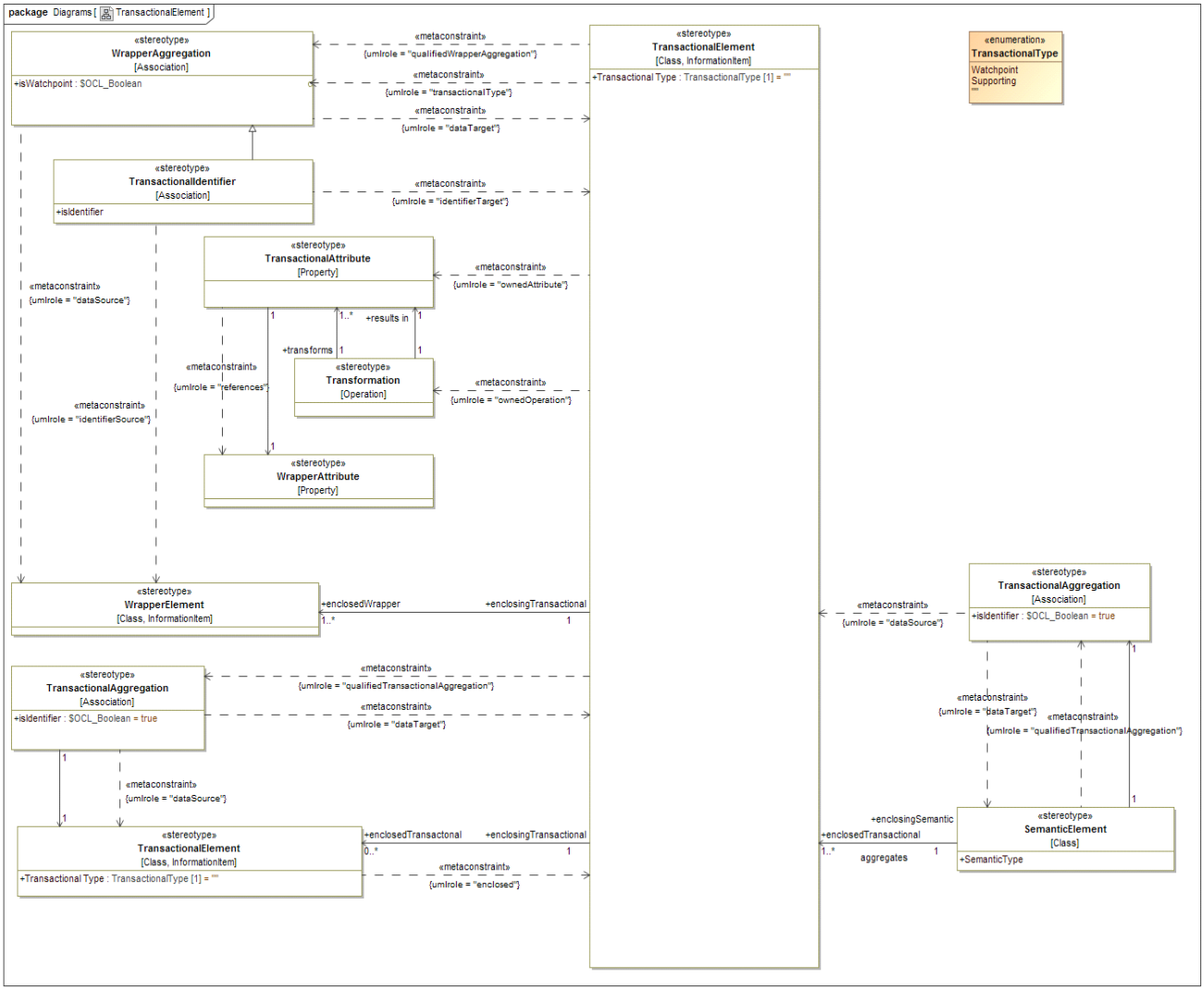

Figure B.17 - TransactionalElement

«Class» Extensions

The IEPPV Profile includes the following extensions to stereotype «Class»:

SemanticElement	A Class enclosing the rules governing the assembly of data elements in accordance with policy.

TransactionalElement	A Class that encloses the rules governing the assembly and processing of data and information elements.
WrapperElement	A Class that contains the based DataElements within the environment. A logical construct that wraps or encapsulates a data set, table entry, triple, file, etc. A Wrapper directly maps to a data instance (e.g., row of data in a database application) in the logical data model and the physical data model.

Constraints

The following constraints are illustrated in the TransactionalElement:

TransactionalIdentifier.identifierSource:	[Values for the identifierSource property must be stereotyped with «Wrapper» or its specializations. The identifier source provides the unique identifier (UI), database key or global unique identifier (GUID) for an instance of the transactional pattern.]
TransactionalIdentifier.identifierTarget:	[Values for the identifierTarget property must be stereotyped with «Transactional» or its specializations; provided with the unique identifier (UI), database key or global unique identifier (GUID) for the build of a transactional pattern.]
TransactionalAggregation.dataTarget:	[The information aggregate at the whole end of the aggregation must be stereotyped «Semantic» or «Transactional».]
TransactionalAggregation.dataSource:	[The information item at the part end will be stereotyped with «Transactional».]
WrapperAggregation.dataSource:	[Values for the dataSource property must be stereotyped with «Wrapper» or its specializations; linking a transactional pattern to its source data elements.]
WrapperAggregation.dataTarget:	[Values for the dataTarget property must be stereotyped with «Transactional» or its specializations; identifying the transactional pattern associated with the aggregation.]
TransactionalAttribute.references:	[Transactional attribute references an EnclosedElementAttribute or the result of a Transformation.]

B.2.11 WrapperElement

The following figure illustrates the modeling relationships for a WrapperElement.

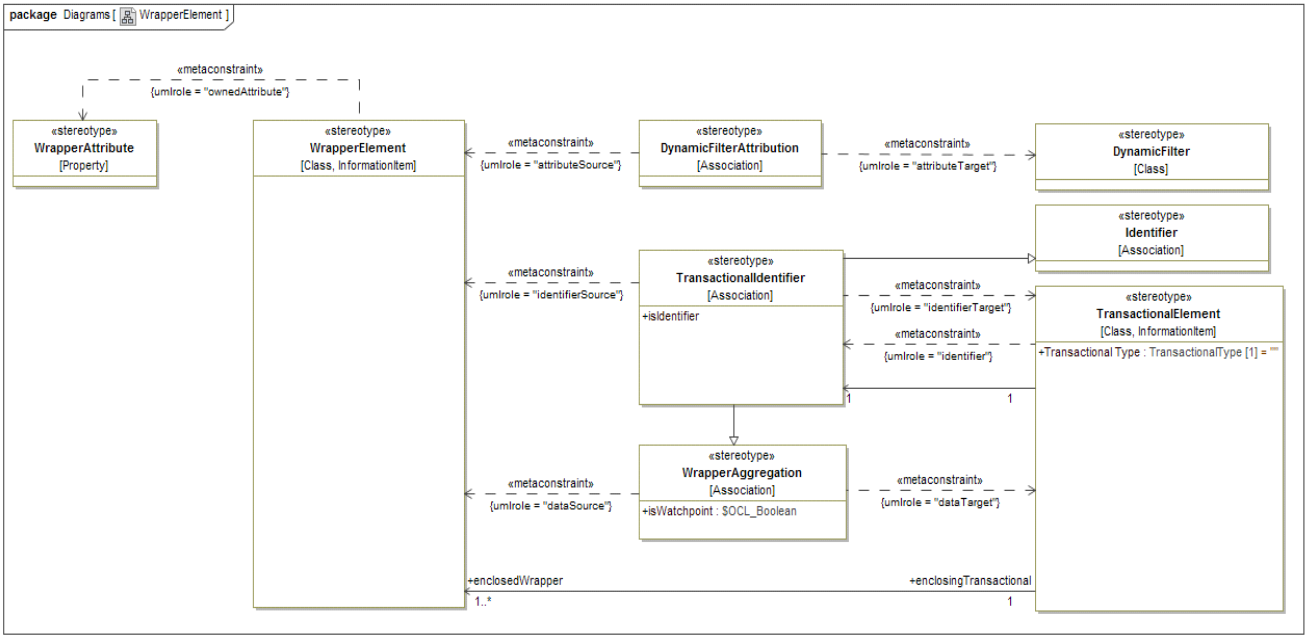

Figure B.18 - WrapperElement

«Class» Extensions

The IEPPV Profile includes the following extensions to stereotype «Class»:

DynamicFilter	The Class enclosing the rules for domain filters whose parameters may be configured at run-time.
TransactionalElement	A Class that encloses the rules governing the assembly and processing of data and information elements.
WrapperElement	A Class that contains the based DataElements within the environment. A logical construct that wraps or encapsulates a data set, table entry, triple, file, etc. A Wrapper directly maps to a data instance (e.g., row of data in a database application) in the logical data model and the physical data model.

Constraints

The following constraints are illustrated in the WrapperElement:

TransactionalIdentifier.identifierSource:	[Values for the identifierSource property must be stereotyped with «Wrapper» or its specializations. The identifier source provides the unique identifier (UI), database key or global unique identifier (GUID) for an instance of the transactional pattern.]
TransactionalIdentifier.identifierTarget:	[Values for the identifierTarget property must be stereotyped with «Transactional» or its specializations; provided with the unique identifier (UI), database key or global unique identifier (GUID) for the build of a transactional pattern.]
DynamicFilterAttribution.attributeSource:	[Values for the attributeSource property must be stereotyped with «Wrapper» or its specializations.]
DynamicFilterAttribution.attributeTarget:	[Values for the attributeTarget property must be stereotyped with «DynamicFilter» or its specializations.]
WrapperAggregation.dataSource:	[Values for the dataSource property must be stereotyped with «Wrapper» or its specializations; linking a transactional pattern to its source data elements.]
WrapperAggregation.dataTarget:	[Values for the dataTarget property must be stereotyped with «Transactional» or its specializations; identifying the transactional pattern associated with the aggregation.]

B.2.12 DistributionSpecification

The Distribution Specification defines the rules that connect the Information Specification to the distribution Services specified to disseminate that information content.

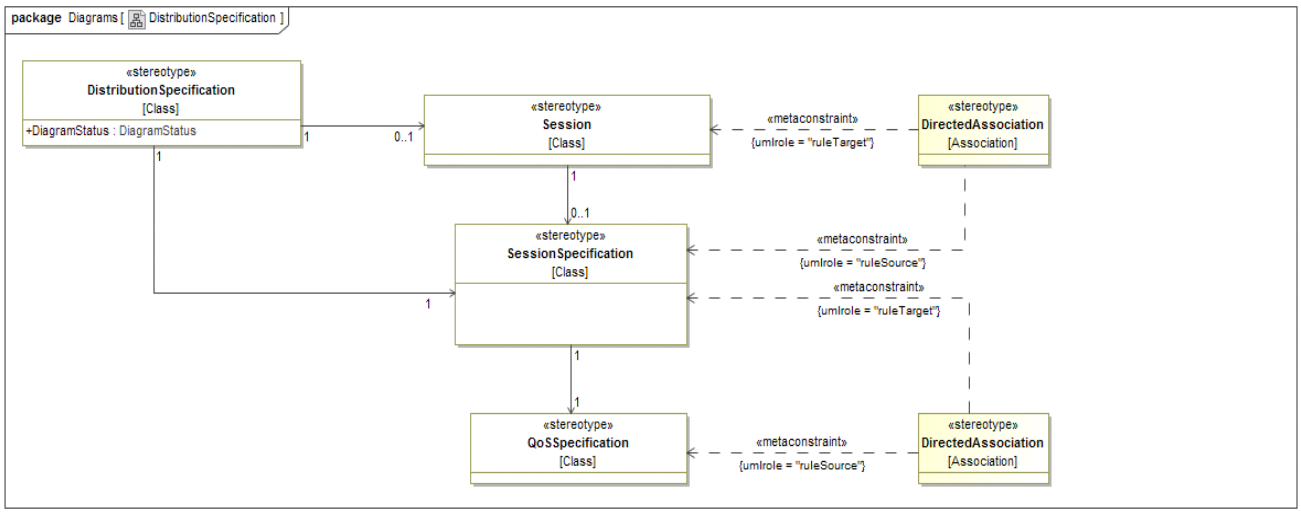

Figure B.19 - DistributionSpecification

«Class» Extensions

The IEPPV Profile includes the following extensions to stereotype «Class»:

DistributionSpecification	A Class that encloses the rules governing the distribution of an InformationElement. Element of an InformationExchangeSpecification that links the InformationSpecification to the information dissemination services (e.g., User Application, Service Interface and Middleware).
QoSSpecification	A Class enclosing the set of those quantitative and qualitative characteristics of a distributed multimedia system, which are necessary in order to achieve the required functionality of an application.
Session	A Class pointing to the services to be used to exchange information.
SessionSpecification	A Class that encloses the governing alignment of Information Elements and data services and information distribution services (e.g., user application, service interface and middleware).

Constraints

The following constraints are illustrated in the DistributionSpecification:

DirectedAssociation.ruleTarget:	[A directed association between the rule Target and the ruleSource. Values for the patternTarget must be stereotyped with «Specification» or its specializations. Identifies that the ruleTarget includes the policies, rules and constraints included in the ruleSource.]
DirectedAssociation.ruleSource:	[A directed association between the rule Target and the ruleSource. Values for the patternSource must be stereotyped with «Specification» or its specializations. Identifies that the ruleSource's policies, rules, and constraints included in the ruleTarget.]

Annex C IEPPV Domain Model (Informative)

C.1 Overview

The IEPPV Domain Model (DMM) is provided as information to tools and infrastructure developers that may implement decision and/or enforcement points for the automation of information exchange packaging policies. Core elements of this model were implemented and demonstrated as part of concept exploration prototypes on projects such as: Army Tactical Command and Control Sytem (ATCCIS, Multilateral Interoperability Programme (MIP), UK MOD BOWMAN and SOPES Test Harness. For each or these projects, the model was populated using a proprietary serialization of the SOPES IEDM modem and the metadata for the JC3IEDM.

The model is provided as an informational part of the specification because it is more applicable to the decision and enforcement points than the automation of a user defined policy model. The IEPPV specification does not provide the transformations or serialization needed to exploit this model. However, the model does provide some insight into the potential use of the IEPPV.

C.2 Attributes

The following domain model identifies platform independent attributes and attribute types. Examples:

1. Identifier: Each of the elements includes an identifier that will uniquely identify an instance of the element in the operational environment. The scope of the uniqueness of the identifier (e.g., policy/rule set, enterprise, community of interest or global) is dependent on the implementation.

2. Sting: Generic type for attributes where there are options on how the attribute is implemented.

C.3 Domain Model

The domain model is divided along the compliance points for the IEPPV:

- CP1: Information Payload Specification
- CP2a: Basic Message Specification (single Information Payload)
- CP2b: Extended Message Specification (single Information Package)
- CP2c: Full Message Specification (multiple Information Packages)
- CP3: Information Exchange Specification

C.3.1 Common Element

The following diagrams are common to CP-1, CP2, and CP-3.

Information Exchange Specification

The following figure illustrates the core elements of the Information Exchange Specification, which is divided into the information characteristics of an exchange, and the distribution characteristics of the exchange. For compliance point 1 (CP-1), the specification focuses on a simple model where the policies describe:

1. One or more messages
2. The optional identification of the Distribution services to be used for the exchange.

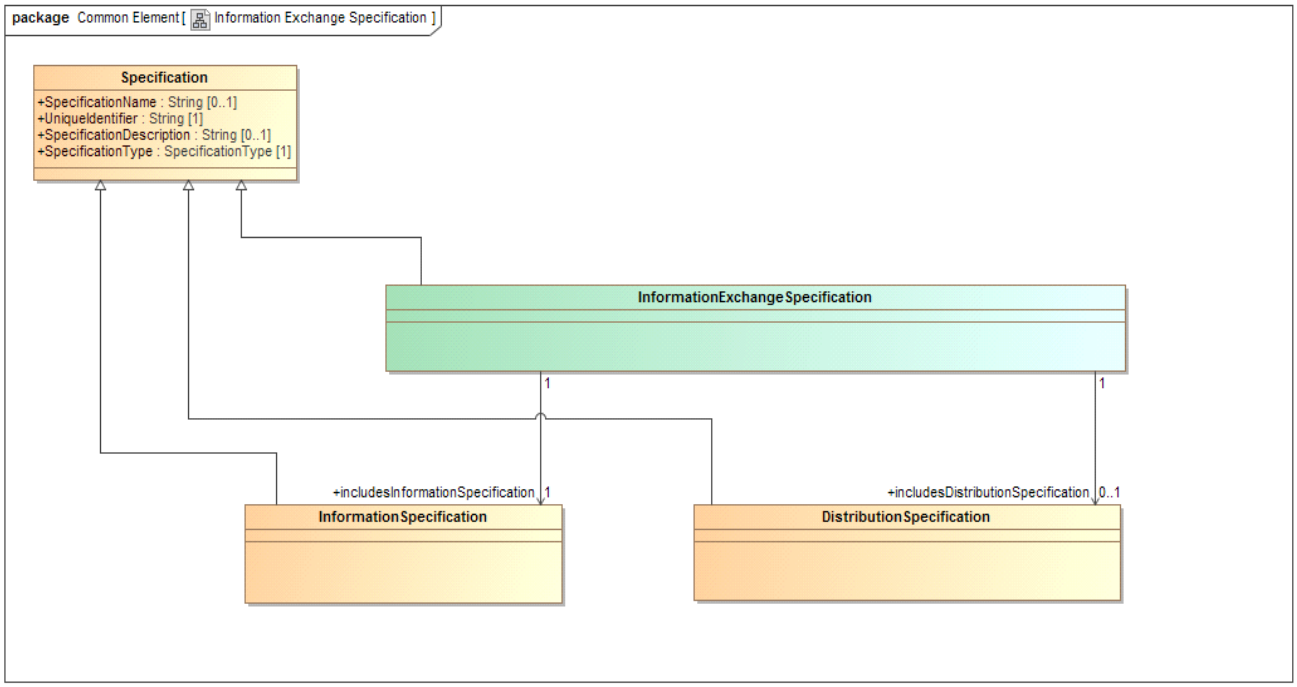

Figure C.1 - Information Exchange Specification

Specification: A detailed precise presentation of something or of a plan or proposal for something.

Attributes defined for Specification include:

- SpecificationName: Optional human readable name provided to a unique instance of a specification to aid discussions.
- UniqueIdentifier: A mandatory unique identifier for an instance of the specification. The uniqueness to the identifier is implementation specific and may provide uniqueness amongst common elements, all elements in the domain, or globally.
- SpecificationDescription: An optional description of the specification to aid in discussions and development.
- SpecificationType: Identifies the type of specification.

InformationSpecification: Specifies the information content (semantics and/or filtered semantics) permitted under the Information Exchange Specification.

Attributes inherited from its generalizations include:

- **SpecificationName:** (String,[0..1]): Optional human readable name provided to a unique instance of a specification to aid discussions.
- **UniqueIdentifier:** (String,[1]): A mandatory unique identifier for an instance of the specification. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.

- **SpecificationDescription:** (String,[0..1]): An optional description of the specification to aid in discussions and development.
- **SpecificationType:** (SpecificationType,[1]): Identifies the type of specification.

DistributionSpecification: Stores information pertaining to the distribution specification.

Attributes inherited from its generalizations include:

- **SpecificationName:** (String,[0..1]): Optional human readable name provided to a unique instance of a specification to aid discussions.
- **UniqueIdentifier:** (String,[1]): A mandatory unique identifier for an instance of the specification. The uniqueness to the identifier is implementation specific and may provide uniqueness amongst common elements, all elements in the domain, or globally.
- **SpecificationDescription:** (String,[0..1]): An optional description of the specification to aid in discussions and development.
- **SpecificationType:** (SpecificationType,[1]): Identifies the type of specification.

InformationExchangeSpecification: Stores information pertaining to a specific Information Exchange Agreement. It aligns an InformationSpecification to its Distribution Specification.

Attributes inherited from its generalizations include:

- **SpecificationName:** (String,[0..1]): Optional human readable name provided to a unique instance of a specification to aid discussions.
- **UniqueIdentifier:** (String,[1]): A mandatory unique identifier for an instance of the specification. The uniqueness to the identifier is implementation specific and may provide uniqueness amongst common elements, all elements in the domain, or globally.
- **SpecificationDescription:** (String,[0..1]): An optional description of the specification to aid in discussions and development.
- **SpecificationType:** (SpecificationType,[1]): Identifies the type of specification.

C.3.2 Compliance Point 1 Information Payload Specification

This sub clause provides various diagrams that document the Domain Metamodel (DMM) for the IEPPV Compliance Point 1 (CP-1). CP-1 is the most basic exchange comprising messages that consist of binary or structured data where the formatting of the data is performed by a separate interface.

C.3.2.1 InformationSpecification (Basic)

The InformationSpecification for CP-1 is satisfied through a single FilteredSemantic. The Filtered Semantic encompasses the following constructs:

- The FilteredSemantic includes:
  - 1 reference to a Semantic, and
  - at least one FilteredTransactional (each FilteredTransactional references a Transactional that must be part of the referenced Semantic).

- The Filtered Transactional includes:
  - one or more DynamicFilters, and
  - 1 reference to a Transactional.
- The DynamicFilter includes:
  - 1 or more filterable attributes that reference WrapperAttributes contained within a Wrapper enclosed by the referenced Transactional or by a subtended Transactional, and
  - Rules about the filter on the attributes.

The FilteredSemantic describes the set of Filters applied to a Semantic (construction or aggregation pattern) for a releasable dataset under an information sharing agreement. The FilteredSemantic enables the reuse of a base Semantic using multiple filtersets corresponding to restrictions imposed by the context of the exchange.

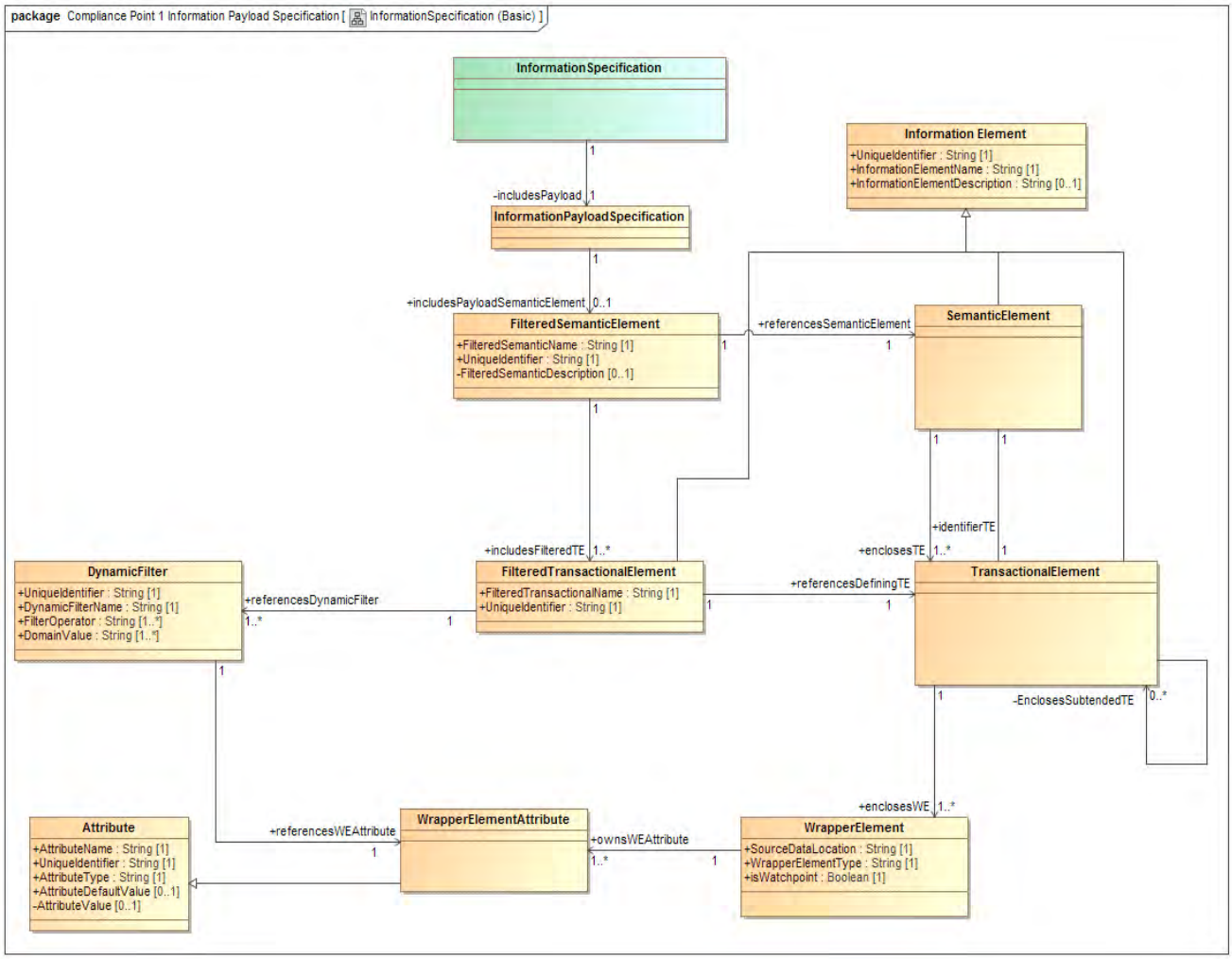

Figure C.2 - InformationSpecification (Basic)

DynamicFilter: Stored information about a Dynamic Filter.

Attributes defined for DynamicFilter include:

- UniqueIdentifier: Unique Identifier for a Dynamic Filter. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- DynamicFilterName: Name of a Dynamic Filter.
- FilterOperator: Filter Operator used as part of the filter rule.
- DomainValue: Values of the attribute used to filter the data build.

TransactionalElement: Information about a transactional element.

Attributes inherited from its generalizations include:

- **UniqueIdentifier:** (String,[1]): A unique identifier for each information element. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- **InformationElementName:** (String,[1]): Name of the information element.
- **InformationElementDescription:** (String,[0..1]): Short description of the Information element.

FilteredSemanticElement: Information about the alignment between a Semantic element and its runtime filters.

Attributes defined for FilteredSemanticElement include:

- FilteredSemanticName: The name given to the filteredSemantic.
- UniqueIdentifier: Unique identifier assigned to the filteredSemantic. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- FilteredSemanticDescription: Short Description of the Filtered Semantic Element.

Attributes inherited from its generalizations include:

- **UniqueIdentifier:** (String,[1]): A unique identifier for each information element. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- **InformationElementName:** (String,[1]): Name of the information element.
- **InformationElementDescription:** (String,[0..1]): Short description of the Information element.

WrapperElement: Information about a wrapper element.

Attributes defined for WrapperElement include:

- SourceDataLocation: Reference to, Location of, the data for the wrapper element.
- WrapperElementType: Type of Wrapper.
- isWatchpoint: Identifies a wrapper element as a watchpoint.

Attributes inherited from its generalizations include:

- **UniqueIdentifier:** (String,[1]): A unique identifier for each information element. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.

- **InformationElementName:** (String,[1]): Name of the information element.
- **InformationElementDescription:** (String,[0..1]): Short description of the Information element.

InformationPayloadSpecification: The rules governing the assembly and processing of a structured dataset for an information exchange.

Attributes inherited from its generalizations include:

- **SpecificationName:** (String,[0..1]): Optional human readable name provided to a unique instance of a specification to aid discussions.
- **UniqueIdentifier:** (String,[1]): A mandatory unique identifier for an instance of the specification. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- **SpecificationDescription:** (String,[0..1]): An optional description of the specification to aid in discussions and development.
- **SpecificationType:** (SpecificationType,[1]): Identifies the type of specification.

Attribute: Stores information about an information element attributes.

Attributes defined for Attribute include:

- AttributeName: Name of the Attribute.
- UniqueIdentifier: Unique Identifier for the Attribute. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- AttributeType: Type of Attribute.
- AttributeDefaultValue: Default Value for the Attribute.
- AttributeValue: The actual Value Attribute. Its type will depend on the value of the AttributeType.

Attributes inherited from its generalizations include:

- **FilteredSemanticName:** (String,[1]): The name given to the filteredSemantic.
- **UniqueIdentifier:** (String,[1]): Unique identifier assigned to the filteredSemantic. The uniqueness to the identifier is implementation specific and may provide uniqueness amongst common elements, all elements in the domain, or globally.
- **referencesSemanticElement:** (SemanticElement,[1]): Reference to the Semantic.
- **includesFilteredTE:** (FilteredTransactionalElement,[1..*]): Reference to the FilteredTransactionalElements.
- **FilteredSemanticDescription:** (,[0..1]): Short Description of the Filtered Semantic Element.
- **UniqueIdentifier:** (String,[1]): A unique identifier for each information element. The uniqueness to the identifier is implementation specific and may provide uniqueness amongst common elements, all elements in the domain, or globally.
- **InformationElementName:** (String,[1]): Name of the information element.
- **InformationElementDescription:** (String,[0..1]): Short description of the Information element.
- **SpecificationName:** (String,[0..1]): Optional human readable name provided to a unique instance of a specification to aid discussions.
- **UniqueIdentifier:** (String,[1]): A mandatory unique identifier for an instance of the specification. The uniqueness to the identifier is implementation specific and may provide uniqueness amongst common elements, all elements in the domain, or globally.

- **SpecificationDescription:** (String,[0..1]): An optional description of the specification to aid in discussions and development.
- **SpecificationType:** (SpecificationType,[1]): Identifies the type of specification.

WrapperElementAttribute: Information about An attribute assigned to a Wrapper.

Attributes inherited from its generalizations include:

- **AttributeName:** (String,[1]): Name of the Attribute.
- **UniqueIdentifier:** (String,[1]): Unique Identifier for the Attribute. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- **AttributeType:** (String,[1]): Type of Attribute.
- **AttributeDefaultValue:** (,[0..1]): Default Value for the Attribute.
- **AttributeValue:** (,[0..1]): The actual Value Attribute. Its type will depend on the value of the AttributeType.
- **FilteredSemanticName:** (String,[1]): The name given to the filteredSemantic.
- **UniqueIdentifier:** (String,[1]): Unique identifier assigned to the filteredSemantic. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- **ReferencesSemanticElement:** (SemanticElement,[1]): Reference to the Semantic.
- **includesFilteredTE:** (FilteredTransactionalElement,[1..*]): Reference to the FilteredTransactionalElements.
- **FilteredSemanticDescription:** (,[0..1]): Short Description of the Filtered Semantic Element.
- **UniqueIdentifier:** (String,[1]): A unique identifier for each information element. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- **InformationElementName:** (String,[1]): Name of the information element.
- **InformationElementDescription:** (String,[0..1]): Short description of the Information element.
- **SpecificationName:** (String,[0..1]): Optional human readable name provided to a unique instance of a specification to aid discussions.
- **UniqueIdentifier:** (String,[1]): A mandatory unique identifier for an instance of the specification. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- **SpecificationDescription:** (String,[0..1]): An optional description of the specification to aid in discussions and development.
- **SpecificationType:** (SpecificationType,[1]): Identifies the type of specification.

Information Element: Stores information describing an information element.

Attributes defined for Information Element include:

- UniqueIdentifier: A unique identifier for each information element. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- InformationElementName: Name of the information element.
- InformationElementDescription: Short description of the Information element.

FilteredTransactionalElement: Information about the configuration of dynamic runtime filters.

Attributes defined for FilteredTransactionalElement include:

- FilteredTransactionalName: Name of the Filtered Transactional Element.
- UniqueIdentifier: The unique identifier for the Filtered Transactional Element. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.

Attributes inherited from its generalizations include:

- **UniqueIdentifier:** (String,[1]): A unique identifier for each information element. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally;
- **InformationElementName:** (String,[1]): Name of the information element.
- **InformationElementDescription:** (String,[0..1]): Short description of the Information element.

InformationSpecification: Specifies the information content (semantics and/or filtered semantics) permitted under the Information Exchange Specification or Information Exchange Contract.

Attributes inherited from its generalizations include:

- **SpecificationName:** (String,[0..1]): Optional human readable name provided to a unique instance of a specification to aid discussions.
- **UniqueIdentifier:** (String,[1]): A mandatory unique identifier for an instance of the specification. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- **SpecificationDescription:** (String,[0..1]): An optional description of the specification to aid in discussions and development.
- **SpecificationType:** (SpecificationType,[1]): Identifies the type of specification.

SemanticElement: Composite of rules governing the assembly of data elements in accordance with a semantic commitment.

Attributes inherited from its generalizations include:

- **UniqueIdentifier:** (String,[1]): A unique identifier for each information element. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- **InformationElementName:** (String,[1]): Name of the information element.
- **InformationElementDescription:** (String,[0..1]): Short description of the Information element.

C.3.2.2 Semantic

A sematic represents a build pattern for an information exchange that conforms to the semantic specification of an exchange agreement (e.g., Information Exchange Data Package (IEPD) as specified by the National Information Exchange Model (NIEM) Program Office).

A Semantic comprises one or more Transactionals that may be statically filtered (e.g., define security or privacy filters operating with specific metadata at runtime).

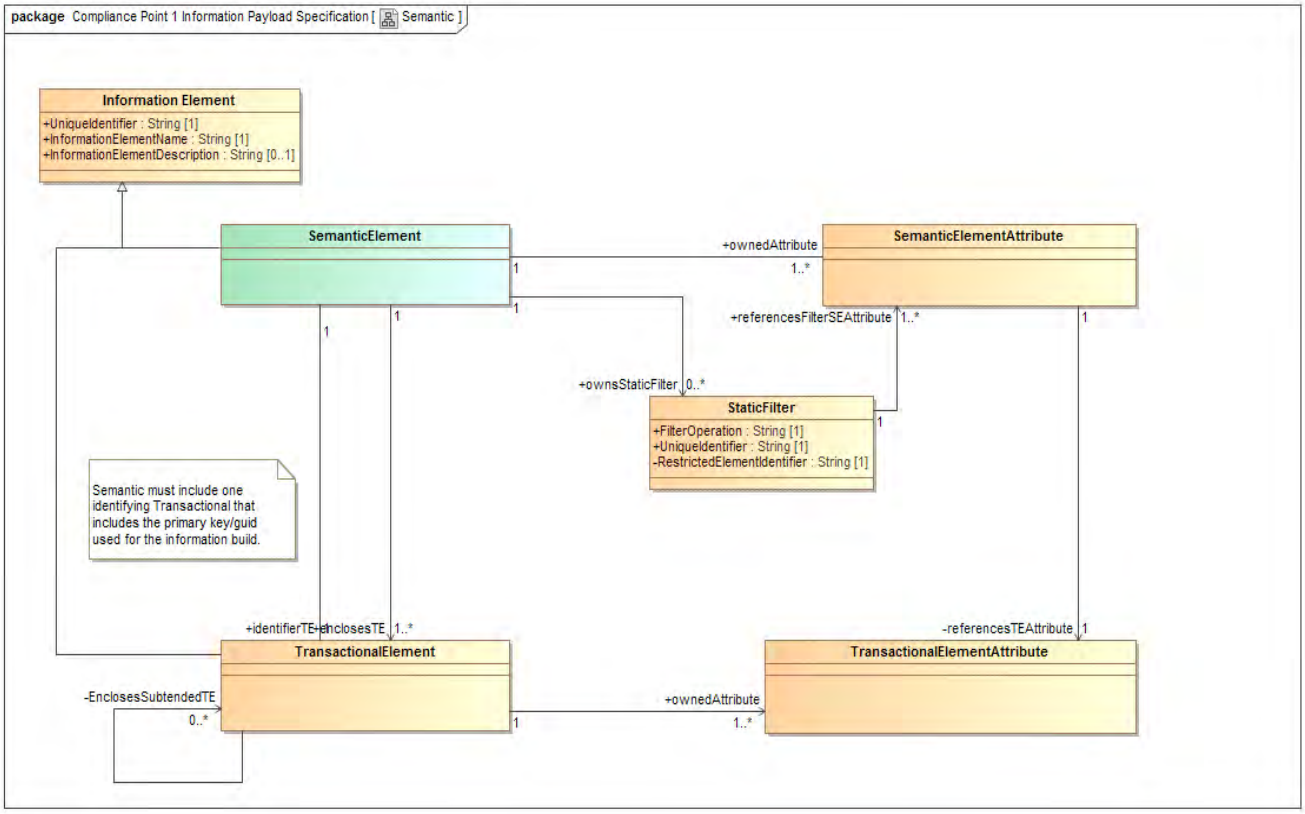

Figure C.3 - Semantic

Information Element: Stores information describing and information element.

Attributes defined for Information Element include:

- UniqueIdentifier: A unique identifier for each information element. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- InformationElementName: Name of the information element.
- InformationElementDescription: Short description of the Information element.

TransactionalElement: Information about a transactional element.

Attributes inherited from its generalizations include:

- **UniqueIdentifier:** (String,[1]): A unique identifier for each information element. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- **InformationElementName:** (String,[1]): Name of the information element.
- **InformationElementDescription:** (String,[0..1]): Short description of the Information element.

SemanticElementAttribute: An attribute assigned to a semantic.

Attributes inherited from its generalizations include:

- **AttributeName:** (String,[1]): Name of the Attribute.
- **UniqueIdentifier:** (String,[1]): Unique Identifier for the Attribute. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- **AttributeType:** (String,[1]): Type of Attribute.
- **AttributeDefaultValue:** (,[0..1]): Default Value for the Attribute.
- **AttributeValue:** (,[0..1]): The actual Value Attribute. Its type will depend on the value of the AttributeType.
- **FilteredSemanticName:** (String,[1]): The name given to the filteredSemantic.
- **UniqueIdentifier:** (String,[1]): Unique identifier assigned to the filteredSemantic. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- **referencesSemanticElement:** (SemanticElement,[1]): Reference to the Semantic.
- **includesFilteredTE:** (FilteredTransactionalElement,[1..*]): Reference to the FilteredTransactionalElements.
- **FilteredSemanticDescription:** (,[0..1]): Short Description of the Filtered Semantic Element.
- **UniqueIdentifier:** (String,[1]): A unique identifier for each information element. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- **InformationElementName:** (String,[1]): Name of the information element.
- **InformationElementDescription:** (String,[0..1]): Short description of the Information element.
- **SpecificationName:** (String,[0..1]): Optional human readable name provided to a unique instance of a specification to aid discussions.
- **UniqueIdentifier:** (String,[1]): A mandatory unique identifier for an instance of the specification. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- **SpecificationDescription:** (String,[0..1]): An optional description of the specification to aid in discussions and development.
- **SpecificationType:** (SpecificationType,[1]): Identifies the type of specification.

StaticFilter: A filter to restrict the aggregation of data and information elements that cannot be modified at run-time.

Attributes defined for StaticFilter include:

- FilterOperation: String describing the filter characteristics or a reference to an operation.
- UniqueIdentifier: Unique identifier for the static filter. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- RestrictedElementIdentifier: Unique identifier for the element restricted by the filter.

TransactionalElementAttribute: An attribute assigned to a Transactional.

Attributes inherited from its generalizations include:

- **AttributeName:** (String,[1]): Name of the Attribute.

- **UniqueIdentifier:** (String,[1]): Unique Identifier for the Attribute. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- **AttributeType:** (String,[1]): Type of Attribute.
- **AttributeDefaultValue:** (,[0..1]): Default Value for the Attribute.
- **AttributeValue:** (,[0..1]): The actual Value Attribute. Its type will depend on the value of the AttributeType.
- **FilteredSemanticName:** (String,[1]): The name given to the filteredSemantic.
- **UniqueIdentifier:** (String,[1]): Unique identifier assigned to the filteredSemantic. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- **ReferencesSemanticElement:** (SemanticElement,[1]): Reference to the Semantic.
- **IncludesFilteredTE:** (FilteredTransactionalElement,[1..*]): Reference to the FilteredTransactionalElements.
- **FilteredSemanticDescription:** (,[0..1]): Short Description of the Filtered Semantic Element.
- **UniqueIdentifier:** (String,[1]): A unique identifier for each information element. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- **InformationElementName:** (String,[1]): Name of the information element.
- **InformationElementDescription:** (String,[0..1]): Short description of the Information element.
- **SpecificationName:** (String,[0..1]): Optional human readable name provided to a unique instance of a specification to aid discussions.
- **UniqueIdentifier:** (String,[1]): A mandatory unique identifier for an instance of the specification. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- **SpecificationDescription:** (String,[0..1]): An optional description of the specification to aid in discussions and development.
- **SpecificationType:** (SpecificationType,[1]): Identifies the type of specification.

SemanticElement: Composite of rules governing the assembly of data elements in accordance with a semantic commitment.

Attributes inherited from its generalizations include:

- **UniqueIdentifier:** (String,[1]): A unique identifier for each information element. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- **InformationElementName:** (String,[1]): Name of the information element.
- **InformationElementDescription:** (String,[0..1]): Short description of the Information element.

C.3.2.3 Transactional

The Transactional represents the build policy (or pattern) for reusable information building blocks, often realized as business objects comprising the community logical data model, for which there is likely also an underlying information or data store; they maintain the referential and data integrity of that store.

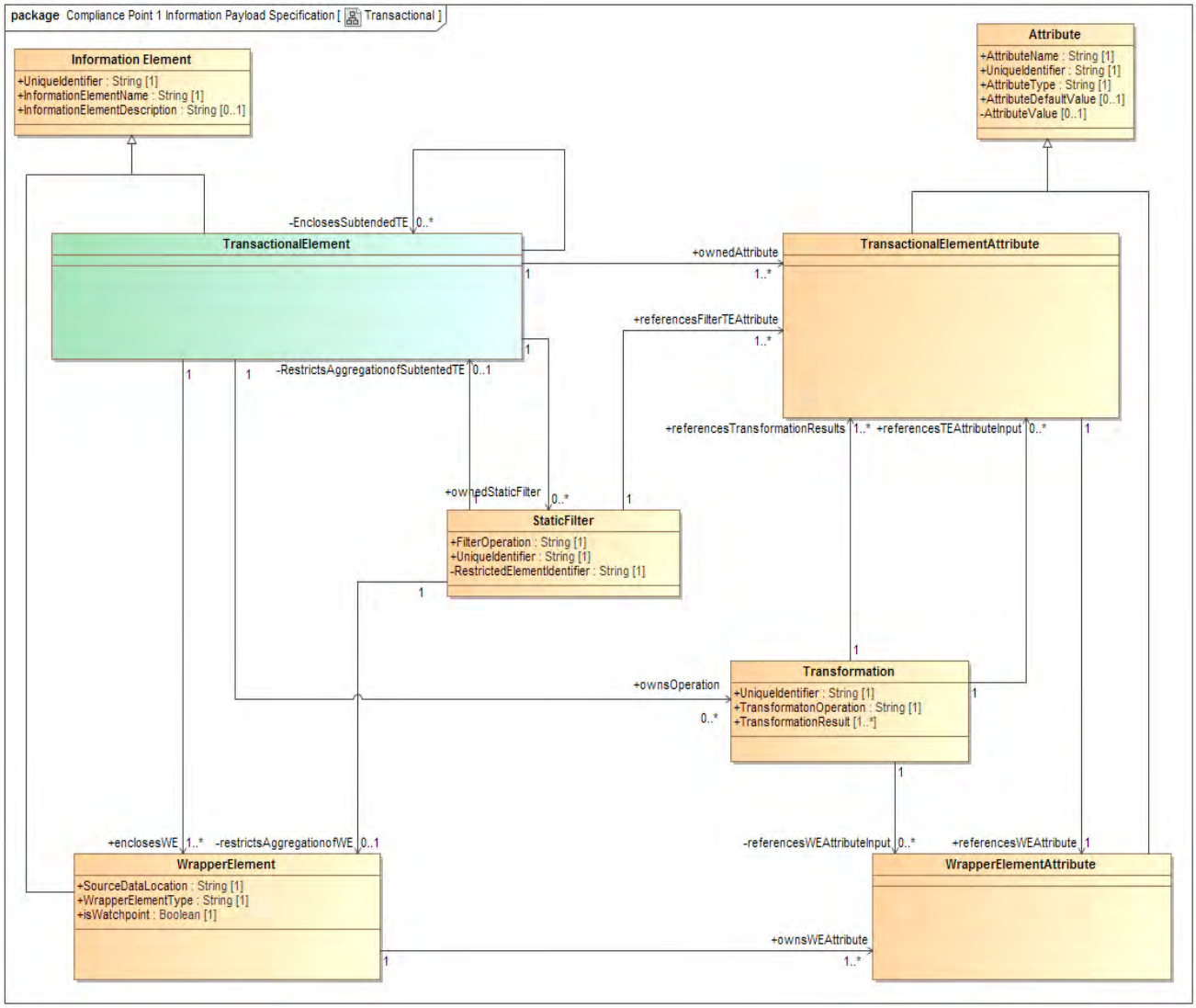

Figure C.4 - Transactional

TransactionalElement: Information about a transactional element.

Attributes inherited from its generalizations include:

- **UniqueIdentifier:** (String,[1]): A unique identifier for each information element. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- **InformationElementName:** (String,[1]): Name of the information element.
- **InformationElementDescription:** (String,[0..1]): Short description of the Information element.

WrapperElement: Information about a wrapper element.

Attributes defined for WrapperElement include:

- SourceDataLocation: Reference to, Location of, the data for the wrapper element.
- WrapperElementType: Type of Wrapper.
- isWatchpoint: Identifies a wrapper element as a watchpoint.

Attributes inherited from its generalizations include:

- **UniqueIdentifier:** (String,[1]): A unique identifier for each information element. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- **InformationElementName:** (String,[1]): Name of the information element.
- **InformationElementDescription:** (String,[0..1]): Short description of the Information element.

Attribute: Stores information about information element attributes.

Attributes defined for Attribute include:

- AttributeName: Name of the Attribute.
- UniqueIdentifier: Unique Identifier for the Attribute. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- AttributeType: Type of Attribute.
- AttributeDefaultValue: Default Value for the Attribute.
- AttributeValue: The actual Value Attribute. Its type will depend on the value of the AttributeType.

Attributes inherited from its generalizations include:

- **FilteredSemanticName:** (String,[1]): The name given to the filteredSemantic.
- **UniqueIdentifier:** (String,[1]): Unique identifier assigned to the filteredSemantic. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- **referencesSemanticElement:** (SemanticElement,[1]): Reference to the Semantic.
- **includesFilteredTE:** (FilteredTransactionalElement,[1..*]): Reference to the FilteredTransactionalElements.
- **FilteredSemanticDescription:** (,[0..1]): Short Description of the Filtered Semantic Element.
- **UniqueIdentifier:** (String,[1]): A unique identifier for each information element. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- **InformationElementName:** (String,[1]): Name of the information element.
- **InformationElementDescription:** (String,[0..1]): Short description of the Information element.
- **SpecificationName:** (String,[0..1]): Optional human readable name provided to a unique instance of a specification to aid discussions.
- **UniqueIdentifier:** (String,[1]): A mandatory unique identifier for an instance of the specification. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- **SpecificationDescription:** (String,[0..1]): An optional description of the specification to aid in discussions and development.
- **SpecificationType:** (SpecificationType,[1]): Identifies the type of specification.

WrapperElementAttribute: Information about an attribute assigned to a Wrapper.

Attributes inherited from its generalizations include:

- **AttributeName:** (String,[1]): Name of the Attribute.
- **UniqueIdentifier:** (String,[1]): Unique Identifier for the Attribute. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- **AttributeType:** (String,[1]): Type of Attribute.
- **AttributeDefaultValue:** (,[0..1]): Default Value for the Attribute.
- **AttributeValue:** (,[0..1]): The actual Value Attribute. Its type will depend on the value of the AttributeType.
- **FilteredSemanticName:** (String,[1]): The name given to the filteredSemantic.
- **UniqueIdentifier:** (String,[1]): Unique identifier assigned to the filteredSemantic. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- **referencesSemanticElement:** (SemanticElement,[1]): Reference to the Semantic.
- **includesFilteredTE:** (FilteredTransactionalElement,[1..*]): Reference to the FilteredTransactionalElements.
- **FilteredSemanticDescription:** (,[0..1]): Short Description of the Filtered Semantic Element.
- **UniqueIdentifier:** (String,[1]): A unique identifier for each information element. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- **InformationElementName:** (String,[1]): Name of the information element.
- **InformationElementDescription:** (String,[0..1]): Short description of the Information element.
- **SpecificationName:** (String,[0..1]): Optional human readable name provided to a unique instance of a specification to aid discussions.
- **UniqueIdentifier:** (String,[1]): A mandatory unique identifier for an instance of the specification. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- **SpecificationDescription:** (String,[0..1]): An optional description of the specification to aid in discussions and development.
- **SpecificationType:** (SpecificationType,[1]): Identifies the type of specification.

Information Element: Stores information describing an information element.

Attributes defined for Information Element include:

- UniqueIdentifier: A unique identifier for each information element. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- InformationElementName: Name of the information element.
- InformationElementDescription: Short description of the Information element.

Transformation: Information about a data transformation.

Attributes defined for Transformation include:

- UniqueIdentifier: Unique Identifier for the operation (transformation Algorithm). The uniqueness to the identifier

is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.

- TransformatonOperation: String describing the filter characteristics or a reference to an operation or a service (e.g., encryption).
- TransformationResult: Attribute containing the transformation result.

StaticFilter: A filter to restrict the aggregation of data and information elements that cannot be modified at run-time.

Attributes defined for StaticFilter include:

- FilterOperation: String describing the filter characteristics or a reference to an operation.
- UniqueIdentifier: Unique identifier for the static filter. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- RestrictedElementIdentifier: Unique identifier for the element restricted by the filter.

TransactionalElementAttribute: An attribute assigned to a Transactional.

Attributes inherited from its generalizations include:

- **AttributeName:** (String,[1]): Name of the Attribute.
- **UniqueIdentifier:** (String,[1]): Unique Identifier for the Attribute. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- **AttributeType:** (String,[1]): Type of Attribute.
- **AttributeDefaultValue:** (,[0..1]): Default Value for the Attribute.
- **AttributeValue:** (,[0..1]): The actual Value Attribute. Its type will depend on the value of the AttributeType.
- **FilteredSemanticName:** (String,[1]): The name given to the filteredSemantic.
- **UniqueIdentifier:** (String,[1]): Unique identifier assigned to the filteredSemantic. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- **referencesSemanticElement:** (SemanticElement,[1]): Reference to the Semantic.
- **includesFilteredTE:** (FilteredTransactionalElement,[1..*]): Reference to the FilteredTransactionalElements.
- **FilteredSemanticDescription:** (,[0..1]): Short Description of the Filtered Semantic Element.
- **UniqueIdentifier:** (String,[1]): A unique identifier for each information element. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- **InformationElementName:** (String,[1]): Name of the information element.
- **InformationElementDescription:** (String,[0..1]): Short description of the Information element;
- **SpecificationName:** (String,[0..1]): Optional human readable name provided to a unique instance of a specification to aid discussions.
- **UniqueIdentifier:** (String,[1]): A mandatory unique identifier for an instance of the specification. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- **SpecificationDescription:** (String,[0..1]): An optional description of the specification to aid in discussions and development.

• **SpecificationType:** (SpecificationType,[1]): Identifies the type of specification.

C.4 Compliance Point 2 Message Specification

The following Sub-clause provides a set of models that support the capture of information needed to support Compliance Point 2 (CP-2) of the IEPP V V1.0. CP-2 extends the concepts expressed in CP-1 and allows for the specification of complex message structures that may include Digests, multiple structured payloads and multiple binary (or unstructured) attachments. This is achieved through three separate sub-compliance points (2a,b&c). However, the domain model is structured to address all three sub-compliance points.

C.4.1 Information Message Specification

The following figure illustrates the elements included in a basic message specification.

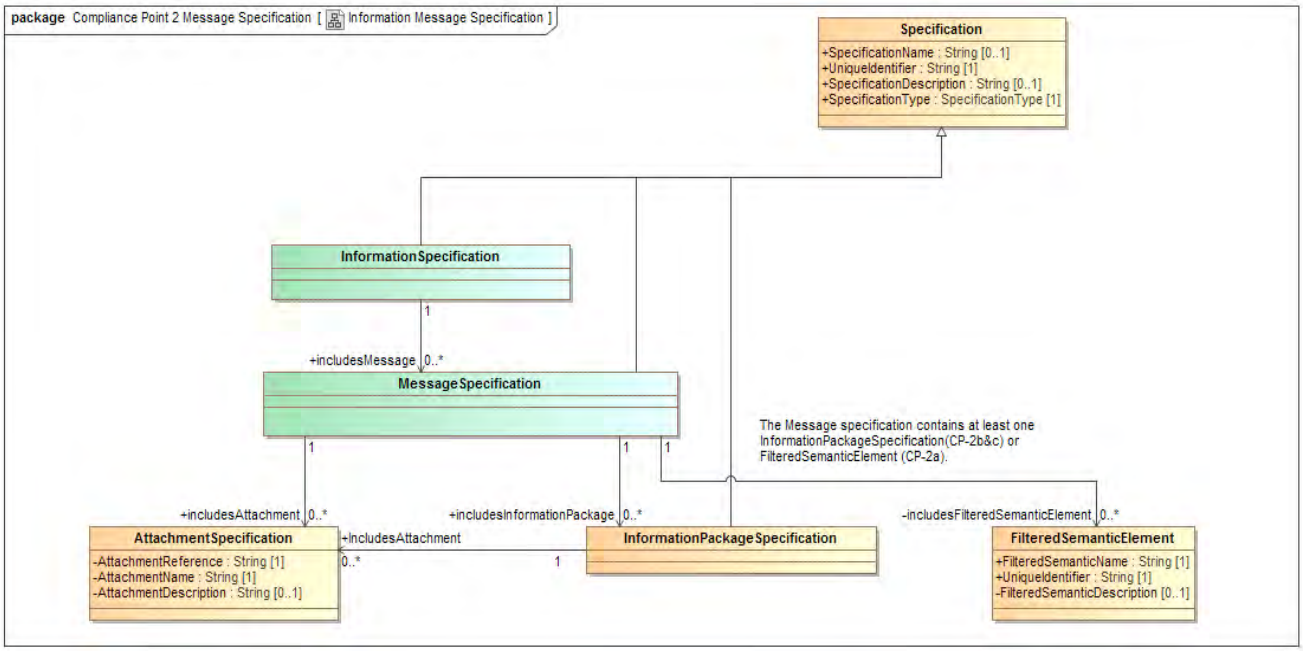

Figure C.5 - Information Message Specification

InformationPackageSpecification: Stores information about information packages.

Attributes inherited from its generalizations include:

- **SpecificationName:** (String,[0..1]): Optional human readable name provided to a unique instance of a specification to aid discussions.
- **UniqueIdentifier:** (String,[1]): A mandatory unique identifier for an instance of the specification. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- **SpecificationDescription:** (String,[0..1]): An optional description of the specification to aid in discussions and development.

- **SpecificationType:** (SpecificationType,[1]): Identifies the type of specification.

AttachmentSpecification: Stores information about attachments to a massage.

Attributes defined for AttachmentSpecification include:

- AttachmentReference: Reference to, location of, the binary element to be attached.
- AttachmentName: Name of the attachment.
- AttachmentDescription: Short description of the Attachment.

FilteredSemanticElement: Information about the alignment between a Semantic element and its runtime filters.

Attributes defined for FilteredSemanticElement include:

- FilteredSemanticName: The name given to the filteredSemantic.
- UniqueIdentifier: Unique identifier assigned to the filteredSemantic. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- FilteredSemanticDescription: Short Description of the Filtered Semantic Element.

Attributes inherited from its generalizations include:

- **UniqueIdentifier:** (String,[1]): A unique identifier for each information element. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- **InformationElementName:** (String,[1]): Name of the information element.
- **InformationElementDescription:** (String,[0..1]): Short description of the Information element.

MessageSpecification: Specifies the rules and constraints governing the assembly of a community compliant structured or semi-structured message in accordance with a specified packaging profile (e.g., LEXS, EDXL-DE and ATOM).

Attributes inherited from its generalizations include:

- **SpecificationName:** (String,[0..1]): Optional human readable name provided to a unique instance of a specification to aid discussions.
- **UniqueIdentifier:** (String, [1]): A mandatory unique identifier for an instance of the specification. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- **SpecificationDescription:** (String, [0..1]): An optional description of the specification to aid in discussions and development.
- **SpecificationType:** (SpecificationType, [1]): Identifies the type of specification.

InformationSpecification: Specifies the information content (semantics and/or filtered semantics) permitted under the Information Exchange Specification or Information Exchange Contract.

Attributes inherited from its generalizations include:

- **SpecificationName:** (String, [0..1]): Optional human readable name provided to a unique instance of a specification to aid discussions.
- **UniqueIdentifier:** (String, [1]): A mandatory unique identifier for an instance of the specification. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.

- **SpecificationDescription:** (String, [0..1]): An optional description of the specification to aid in discussions and development.
- **SpecificationType:** (SpecificationType, [1]): Identifies the type of specification.

Specification: A detailed precise presentation of something or of a plan or proposal for something.

Attributes defined for Specification include:

- SpecificationName: Optional human readable name provided to a unique instance of a specification to aid discussions.
- UniqueIdentifier: A mandatory unique identifier for an instance of the specification. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- SpecificationDescription: An optional description of the specification to aid in discussions and development.
- SpecificationType: Identifies the type of specification.

C.4.2 Information Package Specification

Reference to an attachment specification.

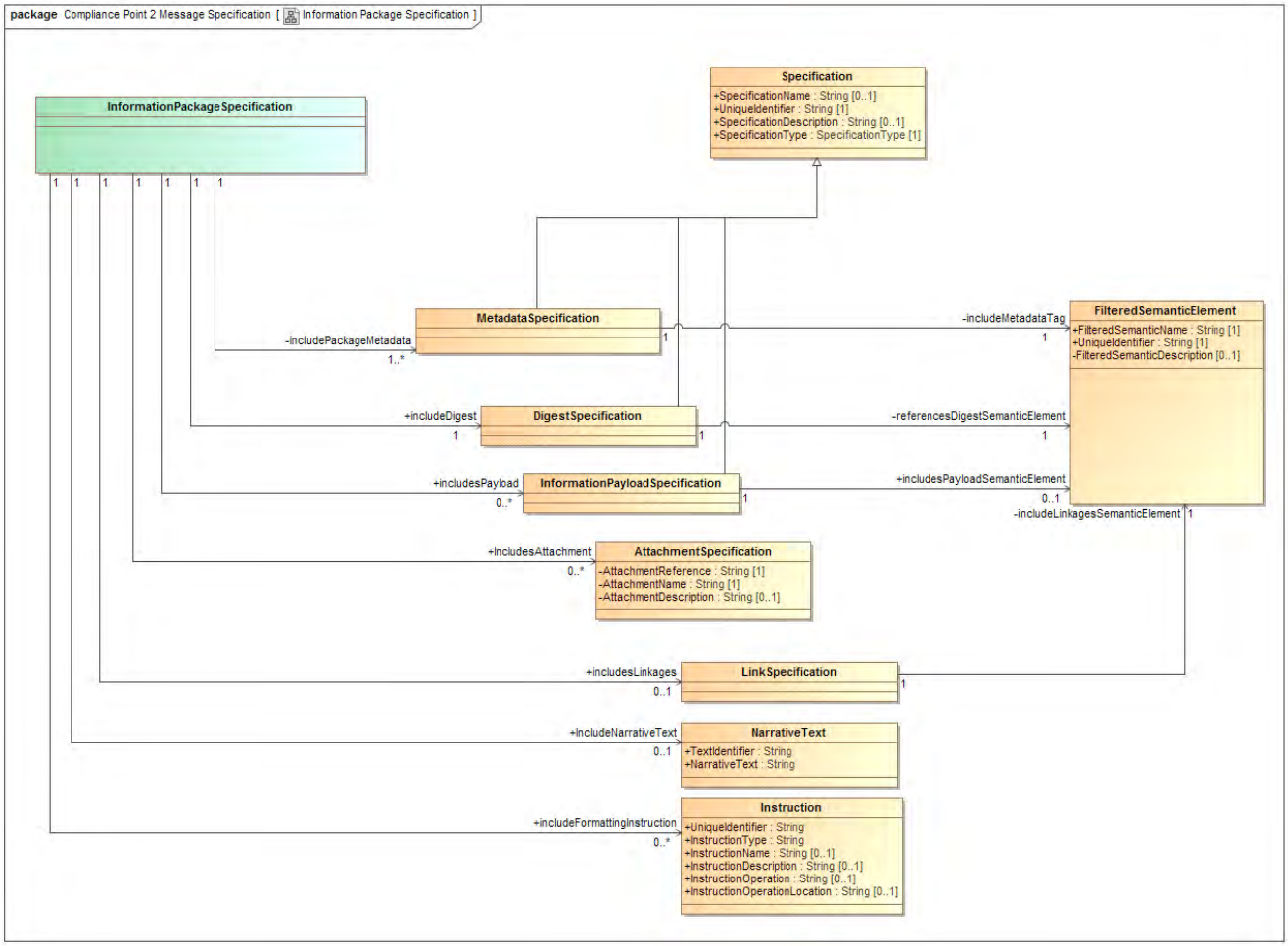

Figure C.6 - Information Package Specification

Instruction: The description of an operation that is to be performed by a computer or human operator.

Attributes defined for Instruction include:

- UniqueIdentifier: Unique Identifier for the instance of the instruction. The uniqueness to the identifier is implementation specific and may provide uniqueness amongst common elements, all elements in the domain, or globally.
- InstructionType: Type of instruction to be applied.
- InstructionName: Name of the instruction to be applied;
- InstructionDescription: Brief description of the instruction to be applied.
- InstructionOperation: Operating instruction to be applied.
- InstructionOperationLocation: identifies the location of a file containing the instructions to be applied.

InformationPackageSpecification: Stores information about information packages.

Attributes inherited from its generalizations include:

- **SpecificationName:** (String,[0..1]): Optional human readable name provided to a unique instance of a specification to aid discussions.
- **UniqueIdentifier:** (String,[1]): A mandatory unique identifier for an instance of the specification. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- **SpecificationDescription:** (String,[0..1]): An optional description of the specification to aid in discussions and development.
- **SpecificationType:** (SpecificationType,[1]): Identifies the type of specification.

Specification: A detailed precise presentation of something or of a plan or proposal for something.

Attributes defined for Specification include:

- SpecificationName: Optional human readable name provided to a unique instance of a specification to aid discussions.
- UniqueIdentifier: A mandatory unique identifier for an instance of the specification. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- SpecificationDescription: An optional description of the specification to aid in discussions and development.
- SpecificationType: Identifies the type of specification.

FilteredSemanticElement: Information about the alignment between a Semantic element and its runtime filters.

Attributes defined for FilteredSemanticElement include:

- FilteredSemanticName: The name given to the filteredSemantic.
- UniqueIdentifier: Unique identifier assigned to the filteredSemantic. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- FilteredSemanticDescription: Short Description of the Filtered Semantic Element.

Attributes inherited from its generalizations include:

- **UniqueIdentifier:** (String,[1]): A unique identifier for each information element. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- **InformationElementName:** (String,[1]): Name of the information element.
- **InformationElementDescription:** (String,[0..1]): Short description of the Information element.

InformationPayloadSpecification: The rules governing the assembly and processing of a structured dataset for an information exchange.

Attributes inherited from its generalizations include:

- **SpecificationName:** (String,[0..1]): Optional human readable name provided to a unique instance of a specification to aid discussions.
- **UniqueIdentifier:** (String,[1]): A mandatory unique identifier for an instance of the specification. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- **SpecificationDescription:** (String,[0..1]): An optional description of the specification to aid in discussions and development.

- **SpecificationType:** (SpecificationType,[1]): Identifies the type of specification.

LinkSpecification: Container for the policies or rules governing the preparation (generation) of linkage information for a specific package of data within an information exchange. Linkages describe relationships between information elements in different sections of a message.

Attributes inherited from its generalizations include:

- **SpecificationName:** (String,[0..1]): Optional human readable name provided to a unique instance of a specification to aid discussions.
- **UniqueIdentifier:** (String,[1]): A mandatory unique identifier for an instance of the specification. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- **SpecificationDescription:** (String,[0..1]): An optional description of the specification to aid in discussions and development.
- **SpecificationType:** (SpecificationType,[1]): Identifies the type of specification.

DigestSpecification: The rules governing the preparation (generation) of a digest.

Attributes inherited from its generalizations include:

- **SpecificationName:** (String,[0..1]): Optional human readable name provided to a unique instance of a specification to aid discussions.
- **UniqueIdentifier:** (String,[1]): A mandatory unique identifier for an instance of the specification. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- **SpecificationDescription:** (String,[0..1]): An optional description of the specification to aid in discussions and development.
- **SpecificationType:** (SpecificationType,[1]): Identifies the type of specification.

NarrativeText: Identifies the location and rules for attaching a narrative of free text field to a message or package of information elements.

Attributes defined for NarrativeText include:

- TextIdentifier: Unique identifier for the block of text.
- NarrativeText: Block of free text to be added to a message.

AttachmentSpecification: Stores information about attachments to a message.

Attributes defined for AttachmentSpecification include:

- AttachmentReference: Reference to, location of, the binary element to be attached.
- AttachmentName: Name of the attachment.
- AttachmentDescription: Short description of the Attachment.

MetadataSpecification: The rules governing the assembly of metadata to be attached to a message, package, information element of an exchange covered by the contract.

Attributes inherited from its generalizations include:

- **SpecificationName:** (String,[0..1]): Optional human readable name provided to a unique instance of a specification to aid discussions.

- **UniqueIdentifier:** (String,[1]): A mandatory unique identifier for an instance of the specification. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- **SpecificationDescription:** (String,[0..1]): An optional description of the specification to aid in discussions and development.
- **SpecificationType:** (SpecificationType,[1]): Identifies the type of specification.

C.5 Compliance Point 3 Distribution Specification

This set of models supports the capture of information needed to support Compliance Point 3 of the IEPPV V1.0. CP-3 provides for the specification of the release and distribution patterns for the payload or message. The distribution specification can be as simple as handing a dataset to the specific software session (application or service) or full processing of individual communities or recipients for each payload or message.

Version 1 of the IEPPV focuses on the simple release of data to a session uncontrolled by the data packaging services. Later versions of the IEPV will address more complex policies on access, release-ability and distribution. Definitional work on these services is already underway, but not ready for inclusion in this version of the specification.

C.5.1 Distribution Specification Domain Model

The DistributionSpecification comprises a set of rules and instructions needed to define a basic message sharing function.

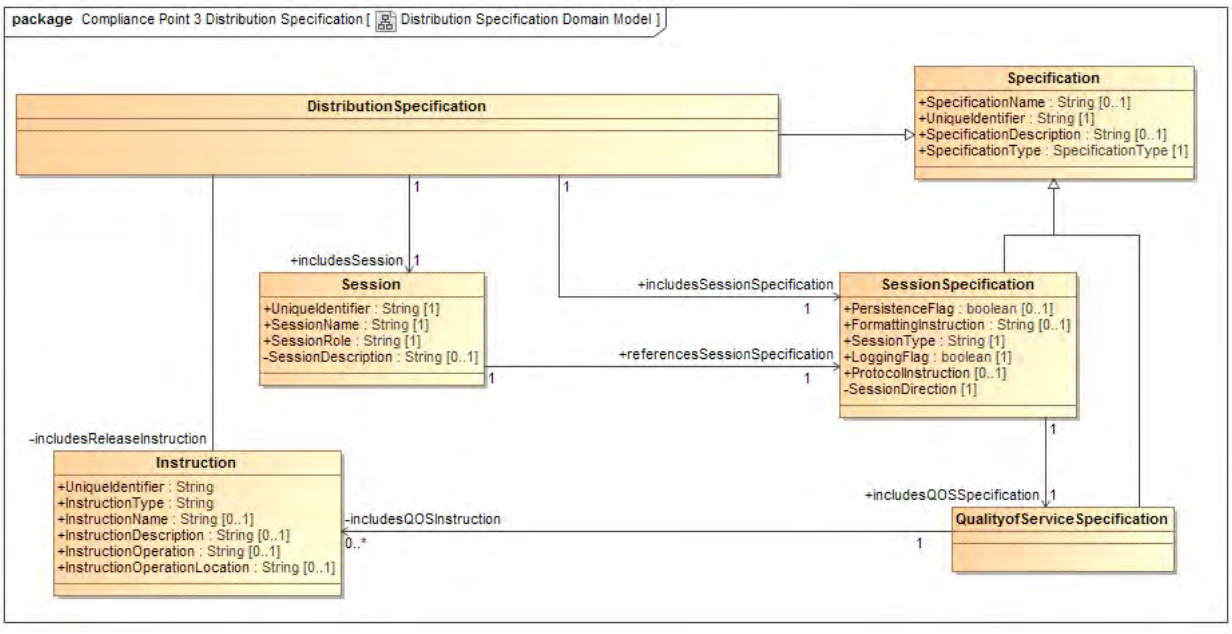

Figure C.7 - Distribution Specification Domain Model

SessionSpecification: Specifies the rules governing communications between the data services and information distribution services (or middleware).

Attributes defined for SessionSpecification include:

- PersistenceFlag: Flag that indicates that the information received on the session should be persisted.
- FormattingInstruction: Instruction or pointer to an instruction guiding the formatting of the information on the session. In conjunction with CP-2 messages - this is not used.
- SessionType: Identifies the type of session being employed (notification service, DDS, etc.).
- LoggingFlag: Flag indicating whether or not activity on the session should be logged.
- ProtocolInstruction: Identifies the message or network protocol to be applied.
- SessionDirection: Sets the direction of the session (Producer, Receiver or Both).

Attributes inherited from its generalizations include:

- **SpecificationName:** (String,[0..1]): Optional human readable name provided to a unique instance of a specification to aid discussions.
- **UniqueIdentifier:** (String,[1]): A mandatory unique identifier for an instance of the specification. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- **SpecificationDescription:** (String,[0..1]): An optional description of the specification to aid in discussions and development.
- **SpecificationType:** (SpecificationType,[1]): Identifies the type of specification.

Instruction: The description of an operation that is to be performed by a computer or human operator.

Attributes defined for Instruction include:

- UniqueIdentifier: Unique Identifier for the instance of the instruction. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- InstructionType: Type of instruction to be applied.
- InstructionName: Name of the instruction to be applied.
- InstructionDescription: Brief description of the instruction to be applied.
- InstructionOperation: Operating instruction to be applied.
- InstructionOperationLocation: identifies the location of a file containing the instructions to be applied.

Specification: A detailed precise presentation of something or of a plan or proposal for something.

Attributes defined for Specification include:

- SpecificationName: Optional human readable name provided to a unique instance of a specification to aid discussions.
- UniqueIdentifier: A mandatory unique identifier for an instance of the specification. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- SpecificationDescription: An optional description of the specification to aid in discussions and development.
- SpecificationType: Identifies the type of specification.

DistributionSpecification: Stores information pertaining to the distribution specification.

Attributes inherited from its generalizations include:

- **SpecificationName:** (String,[0..1]): Optional human readable name provided to a unique instance of a specification to aid discussions.
- **UniqueIdentifier:** (String,[1]): A mandatory unique identifier for an instance of the specification. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- **SpecificationDescription:** (String,[0..1]): An optional description of the specification to aid in discussions and development.
- **SpecificationType:** (SpecificationType,[1]): Identifies the type of specification.

Session: Information about the service used to distribute information.

Attributes defined for Session include:

- UniqueIdentifier: Unique identifier for the session. The uniqueness to the identifier is implementation specific and may provide uniqueness amongst common elements, all elements in the domain, or globally.
- SessionName: Name of the session.
- SessionRole: Identifies the role of the session (e.g., producer, receiver or both).
- SessionDescription: A brief description of the session.

QualityofServiceSpecification: Collection of quality of service instructions.

Attributes inherited from its generalizations include:

- **SpecificationName:** (String,[0..1]): Optional human readable name provided to a unique instance of a specification to aid discussions.
- **UniqueIdentifier:** (String,[1]): A mandatory unique identifier for an instance of the specification. The uniqueness to the identifier is implementation specific and may provide uniqueness among common elements, all elements in the domain, or globally.
- **SpecificationDescription:** (String,[0..1]): An optional description of the specification to aid in discussions and development.
- **SpecificationType:** (SpecificationType,[1]): Identifies the type of specification.

Annex D Example Model (Informative)

D.1 Introduction

The following example illustrates the patterns that would be used to develop a policy model using the UML profile provided in Annex B. The examples build on several of the models delivered as part of the Shared Operational Picture Exchange Services Information Exchange Data Model (SOPES IEDM). The SOPES IEDM defined 192 Transactional (Data) Patterns that derive from the business rules of the Joint Consultation, Command, and Control Information Exchange Data Model (JC3IEDM). The SOPES IEDM elements provide the TransactionalElements used to form the SemanticElements illustrated in the example.

The SOPES IEDM and JC3IEDM materials can be found at:

- SOPES IEDM (formal/2011-05-04): http://www.omg.org/spec/SOPES/
- JC3IEDM:
  Https://mipsite.lsec.dnd.ca/Public%20Document%20Library/Forms/AllItems.aspx?RootFolder=%2FPublic%20Document%20Library%2F04-Baseline_3.1&FolderCTID=0x012000CDEC559A618DF74781A1E0AE00DB1626&View={1DE80D78-9CC7-43F2-BDA0-08741E0F35E7}

D.2 Scope

The following example illustrates the modeling patterns that can be used to develop policy models that translate policy instruments into machine readable and executable rules. These rules can be enforced (/automated) by policy decision and enforcement points as illustrated in Figure D.2.

The combination of patterns presented in the example aligns an InformationExchangeAgreement, or information Exchange Requirement, to a specific data domain; in this instance the JC3IEDM. The JC3IEDM was selected because the TransactionalElements already exist as part of the SOPES IEDM specification.

There has been a vocabulary (stereotype names) change since the adoption of the SOPES IEDM. The following table provided the differences between IEPPV and the terms used in the SOPES IEDM.

Note – The terms (stereotypes) userd in the two models are equivalent and provide for the one-to-one mapping presented below.

Table D.1 - IEPPV to SOPES IEDM Concept Mapping

#	IEPPV Concept	SOPES and UPDM Concept
1	SemanticElement	Semantic
2	TransactionalElement	Transactional
3	WrapperElement	Wrapper
4	FilteredSematicElement	FilteredSemantic
5	FilteredTransactionalElement	FilteredTransactional

Table D.1 - IEPPV to SOPES IEDM Concept Mapping

6	Filter	DynamicFilter
7	InformationExchangeSpecification	Contract

The SOPES IEDM specification did not define a set of SemanticElements (specific message content) for the MIP community. It was limited to the definition of the transactional patterns (data transaction patterns) from which the SemanticElements could be defined. The example (below) illustrates the modeling patterns for the specification/design of semantic elements (e.g., FilteredSemanticElement, Semantic Element, DataPayload, and Digest).

The example also illustrates modeling patterns:

1. That enables the assignment of specific InformationElements to the services used to disseminate them.
2. That enable the specification of reusable patterns that group information elements that service a specific information requirement (e.g., status reporting a set of units (e.g., Organizations and platforms)) that may be used as elements in multiple InformationExchangeAgreements.
3. That enables the specification of filters, on an InformationElement, that can be configured by users at runtime.

D.3 SOPES IEDM

The Shared Operational Picture Exchange Services Information Exchange Data Model defines a set of nearly 200 TransactionalElements in 16 subject areas, and reflects the business rules encoded in the JC3IEDM. The SOPES Model has been transformed into a set of serialized rules that were ingested by a rules engine to successfully execute all the Multilateral Interoperability Program (MIP) test cases (exchange messages) for the operation of the JC3IDEM data environment. When executed, the serialized rules enable the assembly and processing of all Information/Elements transiting from and to the JC3IEDM.

D.4 JC3IEDM

The Joint Consultation, Command, Control Information Exchange Data Model is under the governance of the Multilateral Interoperability Programme (MIP). The JC3IEDM is based on twenty or more years of development in support coalition interoperability requirements for a community of more than 25 nations. It is a complex normalized database. Rules for the assembly and processing of data exchanges were embedded in more than 40 information systems using a Data Exchange Mechanism (DEM) that is proprietary to the MIP community. Many in the community wanted to exploit commercial infrastructure such as SOA, DDS and WEB. The SOPES IEDM and subsequently the IEPPV are efforts to develop a framework to enable the development of Platform Independent Information Exchange Models that would enable portability and the exploitation of these evolving capabilities.

D.5 Scenario Overview

The Example models were drawn from a policy automation demonstration built on the JC3IEDM data patterns. The demonstration scenario, Figure D.1, addressed the information exchanges between four operations centres during a maritime emergency operation. The Operations centers included:

- Government Operating Centre with Public safety (PSC_OPCentre)
- Maritime Operating Centre (MaritimeOPCentre)

- Royal Canadian Mounted Police Operating Centre (RCMP_OPCentre)
- National Defence Operating Centre (NationalDefenceOPCentre).

The following figure was developed using the Unified Profile for DODAF and MODAF (UPDM) Version 1. In UPDM version 2.x, the model would illustrate an Information Exchange between two Performers, rather than OperationalNode. It provides a partial Operational View (OV-2) Operational Resource Flow Description. When linked to UPDM, an InformationExchangeAgreement (e.g., StatusReportingAgreement) is assigned to an Information Exchange. Through the InformationExchangeAgreement, a user can increase the fidelity of the InformationElement specification; including the specification of business rules, data transformations and filters tailored to the specific receiving OperationalNode or Performer. The patterns included in the InformationExchangeAgreement also provide a direct mapping from the Information Exchange to the Logical Data Model (UPDM 1 – OV-7 and UPDM 2 – Data and Information View 2 (DIV-2)).

The example provides modeling patterns used in the “StatusReportingAgreement” between the MaritimeOpCentre and the RCMP_OPCentre.

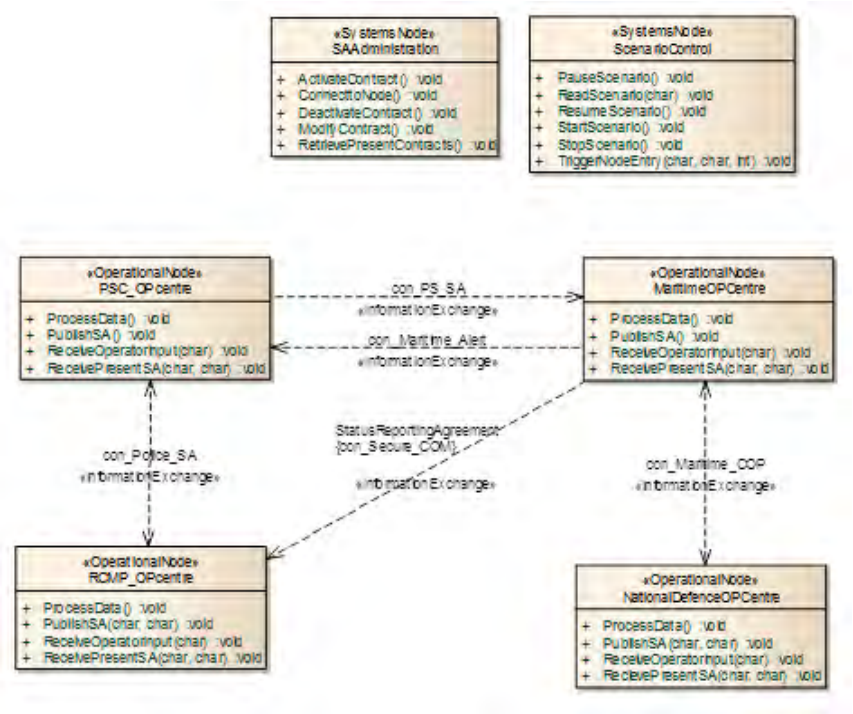

Figure D.1 - Example Scenario

The figure (above) is included for context only. It does not form part of the IEPPV Example.

D.6 CP-1 Policy Model Examples

The following example illustrates the modeling patterns used to develop information assembly and processing models conforming to Compliance Point 1. The following sub clauses illustrate and describe the modeling patterns for

1. The CP-1 version of the InformationExchangeSpecification

2. The CP-1 version of the InformationSpecification
3. The FilteredSemanticElement
4. The FilteredTransactionalElement
5. The SemanticElement
6. The TransactionalElement

Note – The concepts illustrated in CP-1 patterns are reused by CP-2 concepts as well.

D.6.1 InformationExchangeSpecification

The following figure is the modeling pattern for a CP-1 InformationExchangeSpecification. In its simplest form it assigns one InformationElement (FilteredSemanticElement) to a SessionSpecification. The SessionSpecification routes the InformationElement to the release or dissemination services (e.g., DDS, Web Service, and User Application).

When executed by the decision and enforcement points comprising a data/information packaging service, user application or Extract, Transform and Load (ETL) tool – the FilteredSemantic (Navy_SA) will execute the subtended rules for the assembly (aggregation, transformation, Tagging/labeling, and filtering) of data and information elements describing a Navy Unit status. This is derived from the combination of the naming of the InformationExchangeAgreement (StatusReportingAgreement) and the naming of the FilteredSemanticElement (NavyUnit_SA).

The SessionSpecification identifies that the NavyUnit status must be disseminated using DDS, Using MIP_XML Messaging Protocol and no logging is required.

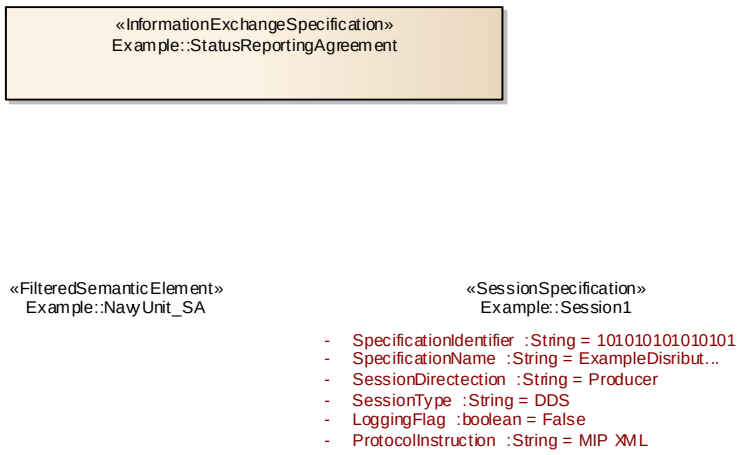

Figure D.2 - Information Exchange Specification (Simple)

The simple pattern from the previous figure can be extended. Multiple InformationElements (FilteredSemanticElement) can be attached to the InformationExchangeSpecification or grouped into a separate InformationSpecification as illustrated below.

The use of the InformationSpecification to group InformationElements provides the ability to reuse the pattern to define multiple InformationExchangeSpecifications using different distribution services, message protocols, dissemination services and quality of Service (QoS) characteristic. For example:

1. IES 1: NavyUnit_SA, MilitaryAircraft_SA & CommercialAircraft_SA exchanged over DDS using MIP XML Protocol.
2. IES 2: NavyUnit_SA echanged over Web Service using NIEM XML Protocol; and so on.

Building reusable patterns

- Helps to reduce the complexity of operational information environments;
- Facilitates the analysis of operational requirements;
- Facilitates communication with stakeholders; and
- Enables the rapid generation of information exchange patterns for new operations.

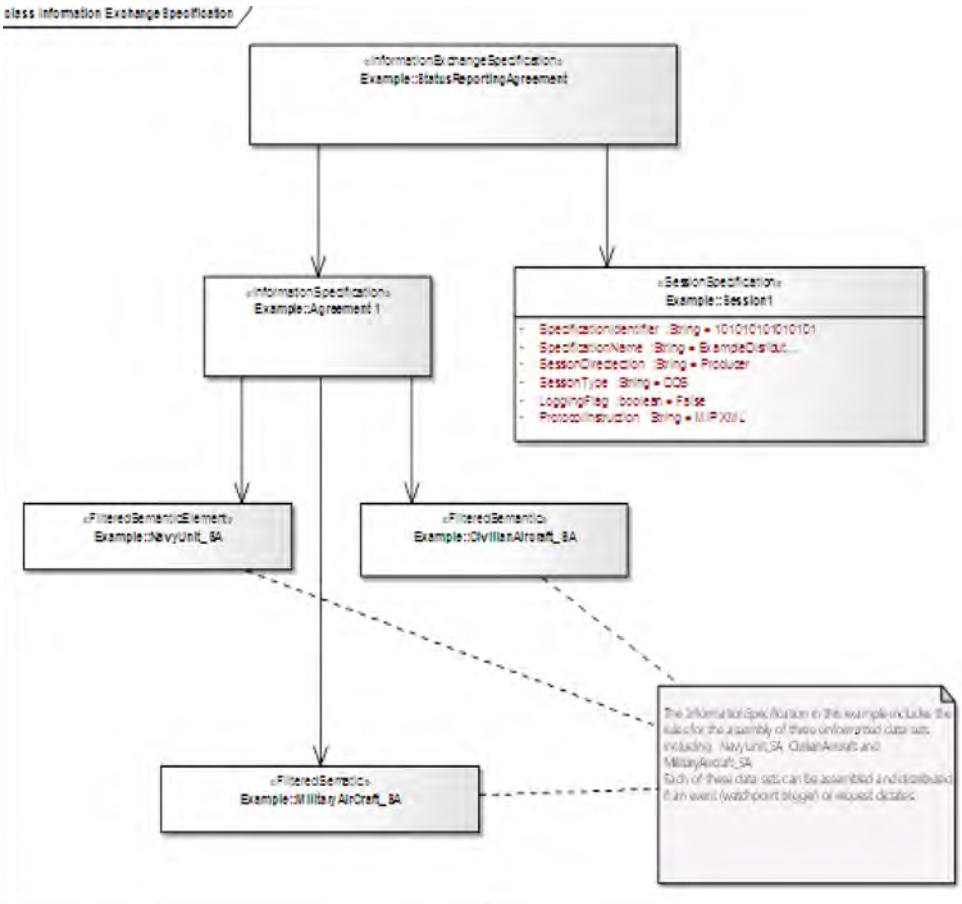

Figure D.3 - Information Exchange Specification

D.6.2 FilteredSemantic & FilteredTransactional

The FilteredSemantic groups or encloses a set of run-time configurable filters for a Semantic Element. As illustrated below, the FilteredTransactionalElement references a single SemanticElement from which it draws its internal patterns. The filters are assigned to attributes within the TransactionalElements in the EnclosingSematicElement (e.g., NavyUnit_SA). It is the subtended FilteredTransactionalElement that assigns the filters to the attributes within the Semantic Pattern.

The FilteredTransactionalElement assigns runtime (user configurable) filters to a specific TransactionalElement enclosed by the SemanticElement. In this case, NavyUnit data is derived from the SemanticElement (Organization). An "Organization" is a generic Semantic Pattern used to assemble data pertaining to an organization contained within an instance of a JC3IEDM database. To limit (filter/redact) the assembly process to specific "units," a type of organization, and further restrict that to a Navy Unity), one needs ability to configure two specific domain filters:

1. cat-code in Wrapper Element "Organization," and
2. object-type-name-text in WrapperElement "OrganizationType."

In order to restrict the reports to only those from NAVY UNITS:

1. The object-type-name-text must be set to "NAVY," and
2. The cat-code must be set to "UNIT."

Both of these WrapperElements are contained within one TransactionalElement, "Organization_Item_Type." Thus only one FilteredTransactionalElement is needed.

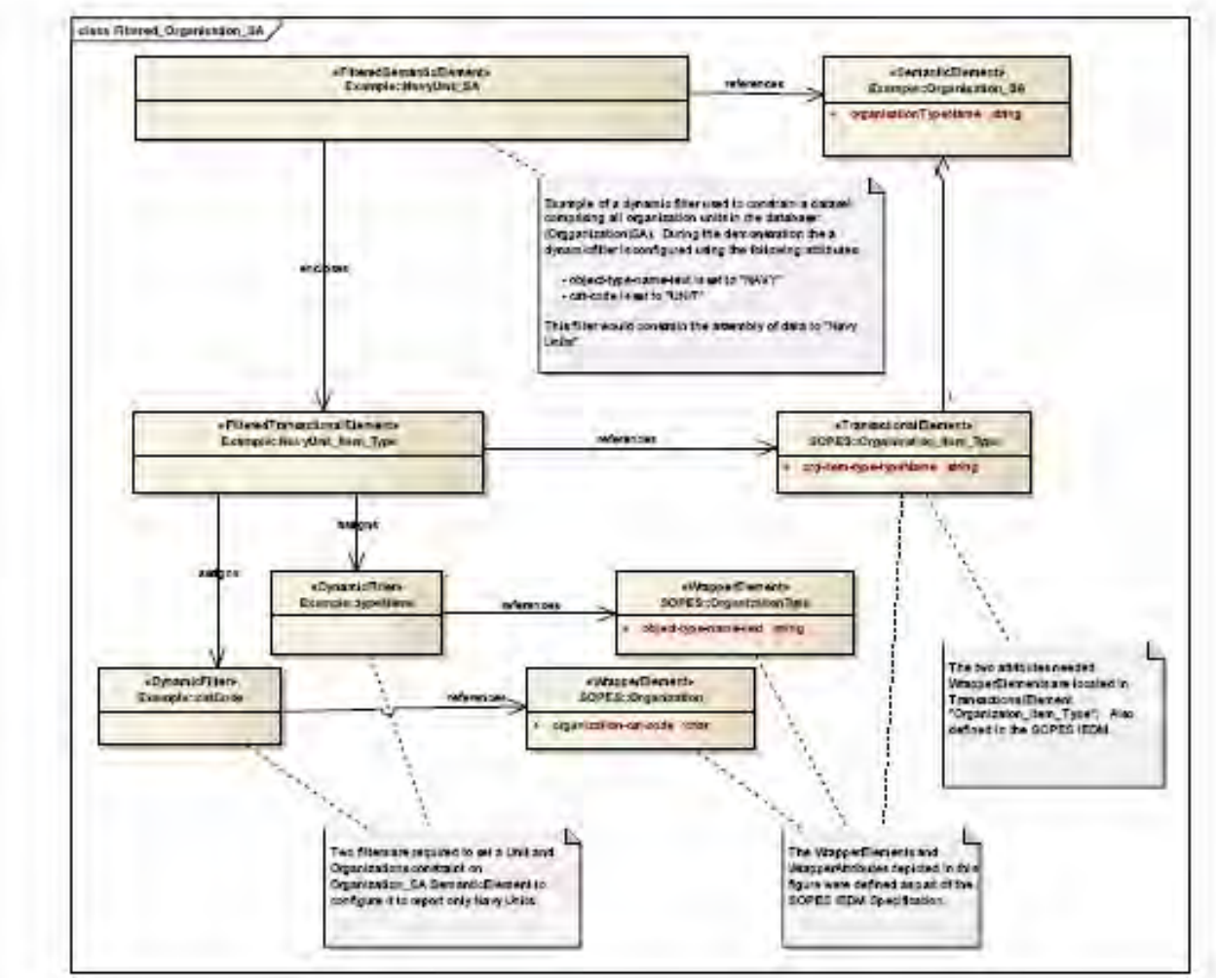

Figure D.4 - FilteredSemantic / FilteredTransactional

D.6.3 SemanticElement

A Semantic Element groups or encloses a set of TransactionalElements (Data Patterns) that in combination define a set of rules for assembling a complete and meaning dataset for the stakeholder (user or community); e.g., Organization: rules for assembling data pertaining to all organizations maintained in an instance of a JC3IEDM database. Within the context of the JC3IEDM, a Unit is a type of Organization. The types of information reported on any organization is specified or defined by the stakeholders. For the purpose of this example, only tombstone data, status, and position are reported or exchanged.

The Transactionals needed to assemble organization information are drawn from the SOPES IEDM specification:

- OrganizationalItem (SOPES IEDM sub clause 10.14.7)
- Organization Item_Type (SOPES IEDM sub clause 10.14.8)
- Organizational_Status (SOPES IEDM sub clause 10.14.14)

- Organizational Position (SOPES IEDM sub clause 10.14.12)

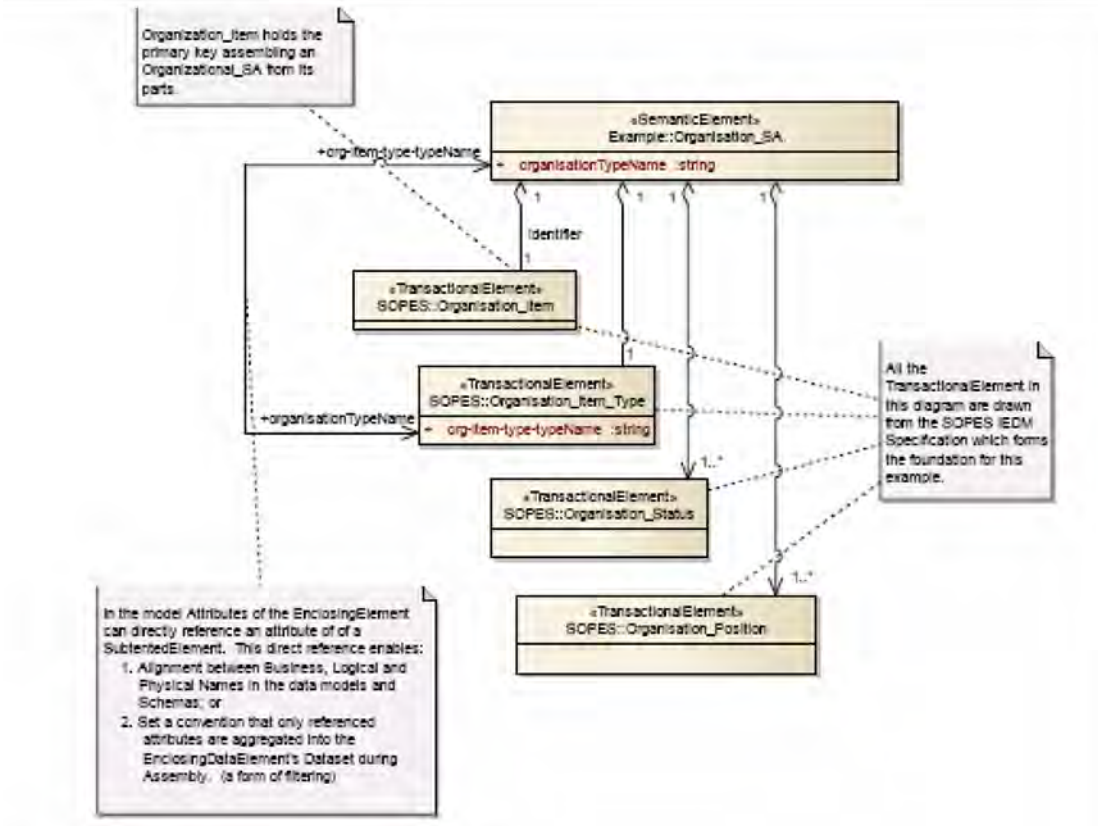

Figure D.5 - Organization_SA

D.6.4 SemanticElement (staticFilters)

In the event that a user wants a NavyUnit Reporting element that cannot be configured at run-time, the following modeling pattern is used. The aggregations from the SubtendedElement (Organizational_Item and Organizational_Item_Type) have been qualified to only assemble elements that have a cat-code of “UNIT” and OrganizationTypeName of “NAVY.” This form of filtering would yield the same results and the FilterSemanticElement (NavyUnit_Item_Type in Figure D.6).

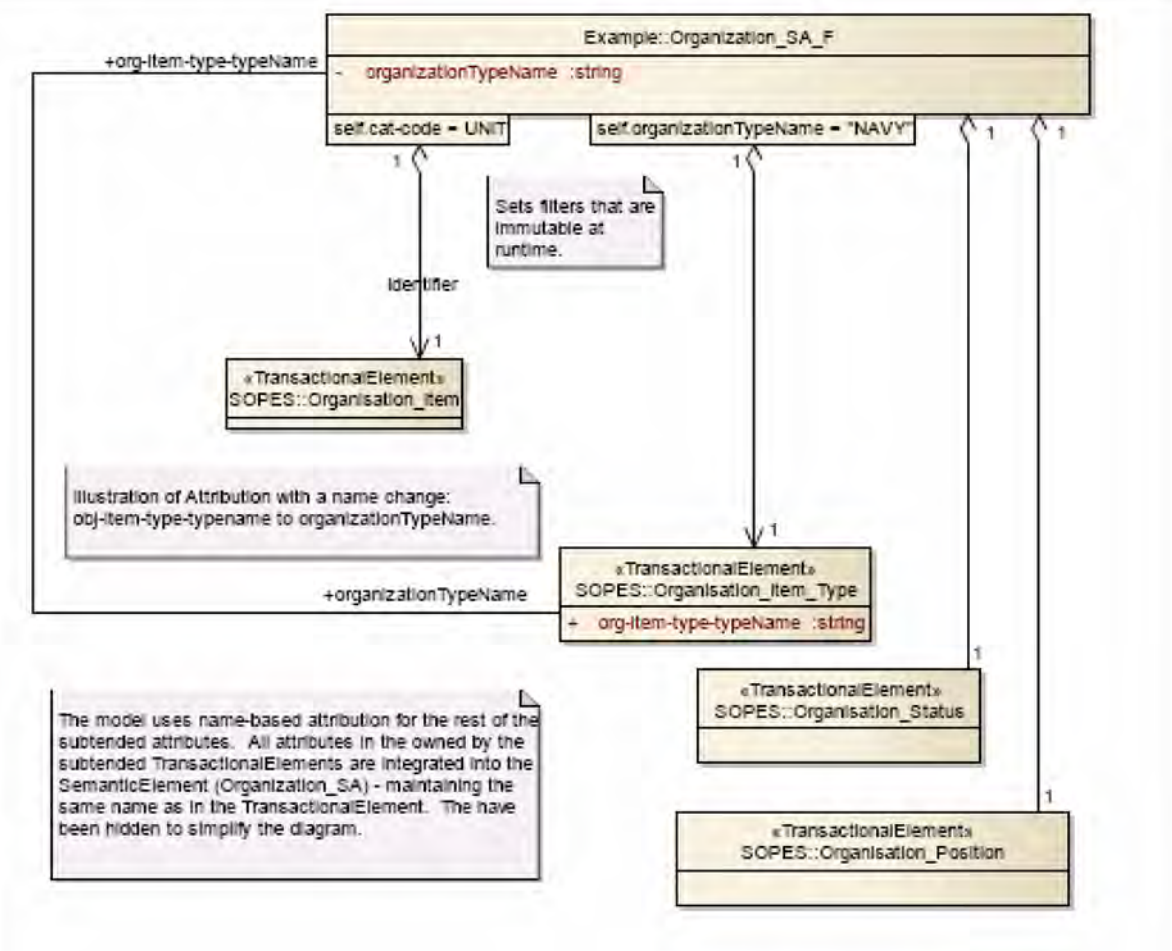

Figure D.6 - Semantic with Static Filters

Note – The exclusive use of static filters would require a separate SemanticElement to be developed and deployed for each type of UNIT and provide no flexibility for the User. By using a FilteredSemanticElement, only one pattern needs to be deployed and the specific reporting pattern can be established at runtime. The latter provides more flexibility and agility in the operational environment. The selection of pattern to use is the choice of the stakeholders.

Static filters can be applied to the TransactionalElement aggregation arc. It performs the same function during the aggregation of the subtended elements in the semantic pattern.

D.6.5 SemanticElement (with Markings and Transformations)

In practice, all transformations are performed in the assembly of the transactional Elements. In the example we have the requirement to convert the “reportedDateTime” attribute in the SOPES IEDM Organization_Status TransactionalElement to a ReportDate and a ReportTime. In addition, there is a requirement to determine and generate a ReportSensitivity Tag based on:

- OrganizationType (org-item-type-typeName)

- Reporting Date(reportedDataTime)
- Reporting Time(reportedDataTime)
- Organization Status (org-stat-oper-stat-code)

To enable these transformations to the assembled data set, the Organization_SA (Organization_SA_Transform) is extended. The ReportingSensitivity TransactionalElement has been added. This TransactionalElement includes three operations to satisfy the preparation of the three required data elements (i.e., ReportSensitivity, ReportDate, and ReportTime). The modeling pattern for the inclusion of Transformation is illustrated below.

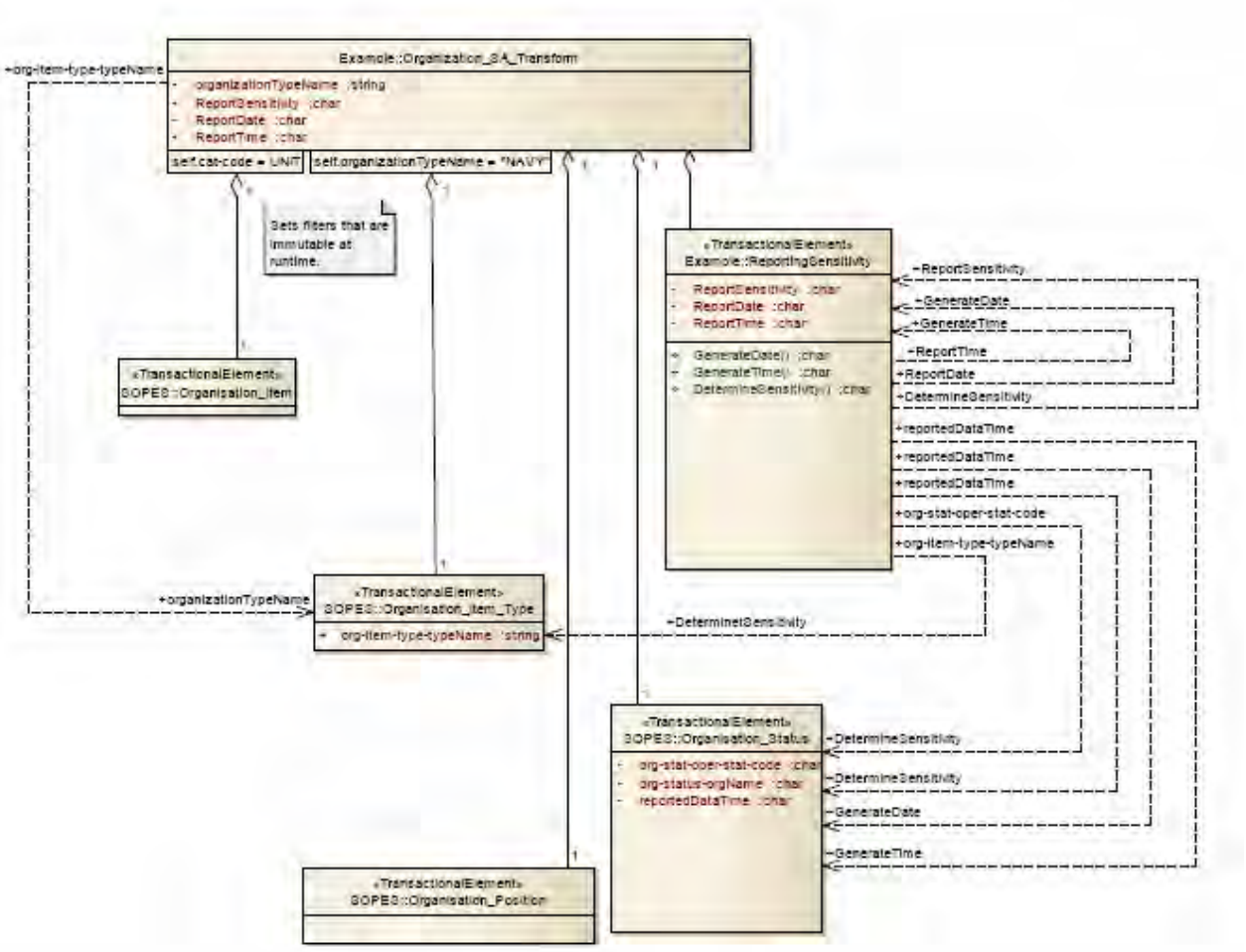

Figure D.7 - Modeling Transformations

D.6.6 TransactionalElement

The following models were extracted from the SOPES IEDM Specification. Note that the Stereotyping is not consistent with the IEPPV profile as the SOPES IEDM specification predates the Information Exchange Framework (IEF) effort and this specification. The following table represents the change in terminology (stereotype names) from SOPES IEDM to the IEPPV.

Table D.2 - IEPPV to SOPES IEDM Concept Mapping

#	IEPPV Concept	SOPES and UPDM Concept
1	SemanticElement	Semantic
2	TransactionalElement	Transactional
3	WrapperElement	Wrapper
4	FilteredSematicElement	FilteredSemantic
5	FilteredTransactionalElement	FilteredTransactional
6	Filter	DynamicFilter
	Filter	StaticFilter
7	InformationExchangeSpecification	Contract

The IEPPV formalized the core modeling concepts used in the SOPES IEDM and many of the modeling extensions described in Annex A to that specification. The only changes to the SOPES IEDM models were the addition of notes highlighting several of the modeling concepts.

D.6.6.1 Organization_Item

The “Organization_Item” represents one of 192 reusable TransactionalElements in 16 subject areas defined by the SOPES IEDM for the JC3IEDM. The Specific model has been replicated in this example to illustrate the hierarchy in the IEPPV modeling patterns.

The Organization_Item illustrates differences between the SOPES IEDM models and the IEPPV. The SOPES IEDM applies constraints to the AggregationArcs. These Constraints were used to address the JC3IEDM’s many uses of subtypes. The addition of constraints assisted in the generation of an extended set of rules that aided in the processing of rules at runtime and enhance performance. This use of constraints was not carried forward to the formal IEPPV profile. It does however illustrate that the modeling patterns in the IEPPV can be extended, using standard UML constructs, to address specific domain requirements.

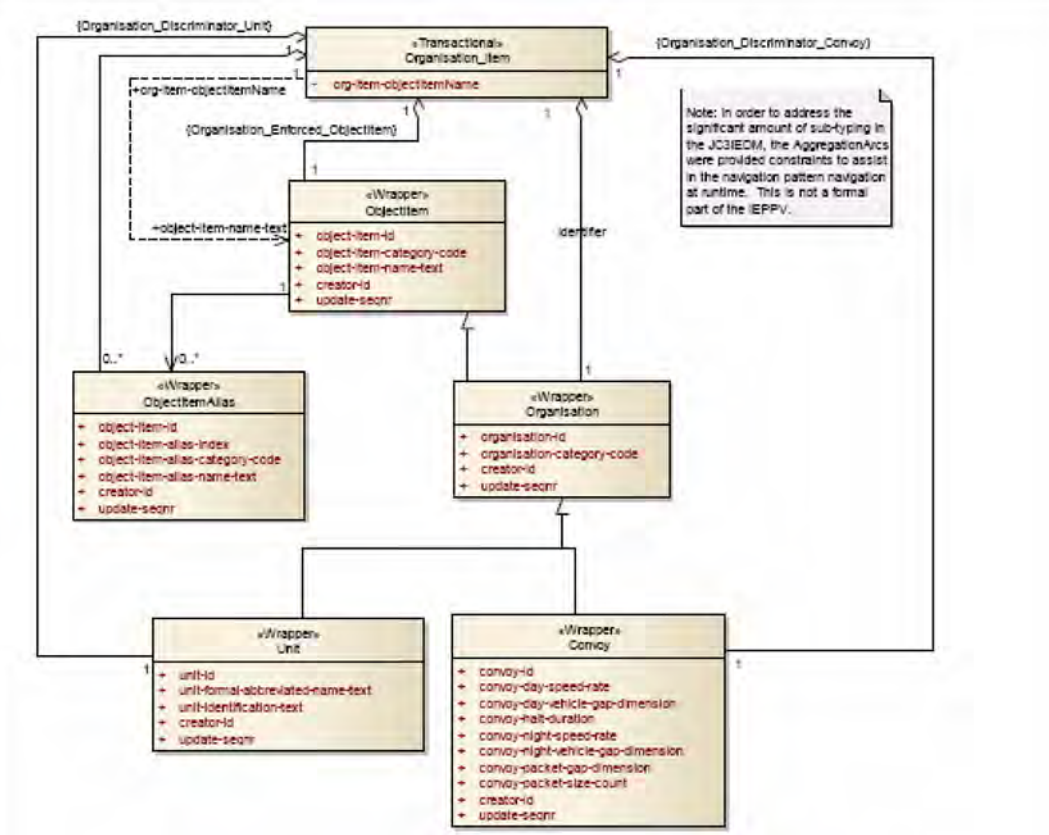

Figure D.8 - Organization_Item

D.6.7 Organization Position

As with the Organization_Item, the Organization_Position has been taken from the JC3IEDM. It is added for completeness and to highlight modeling concepts. The Organization_Item model illustrates the hierarchical nature of the IEPPV modeling patterns and that the Wrapper or WrapperElements provide the linkage of the rules to the data to which they apply.

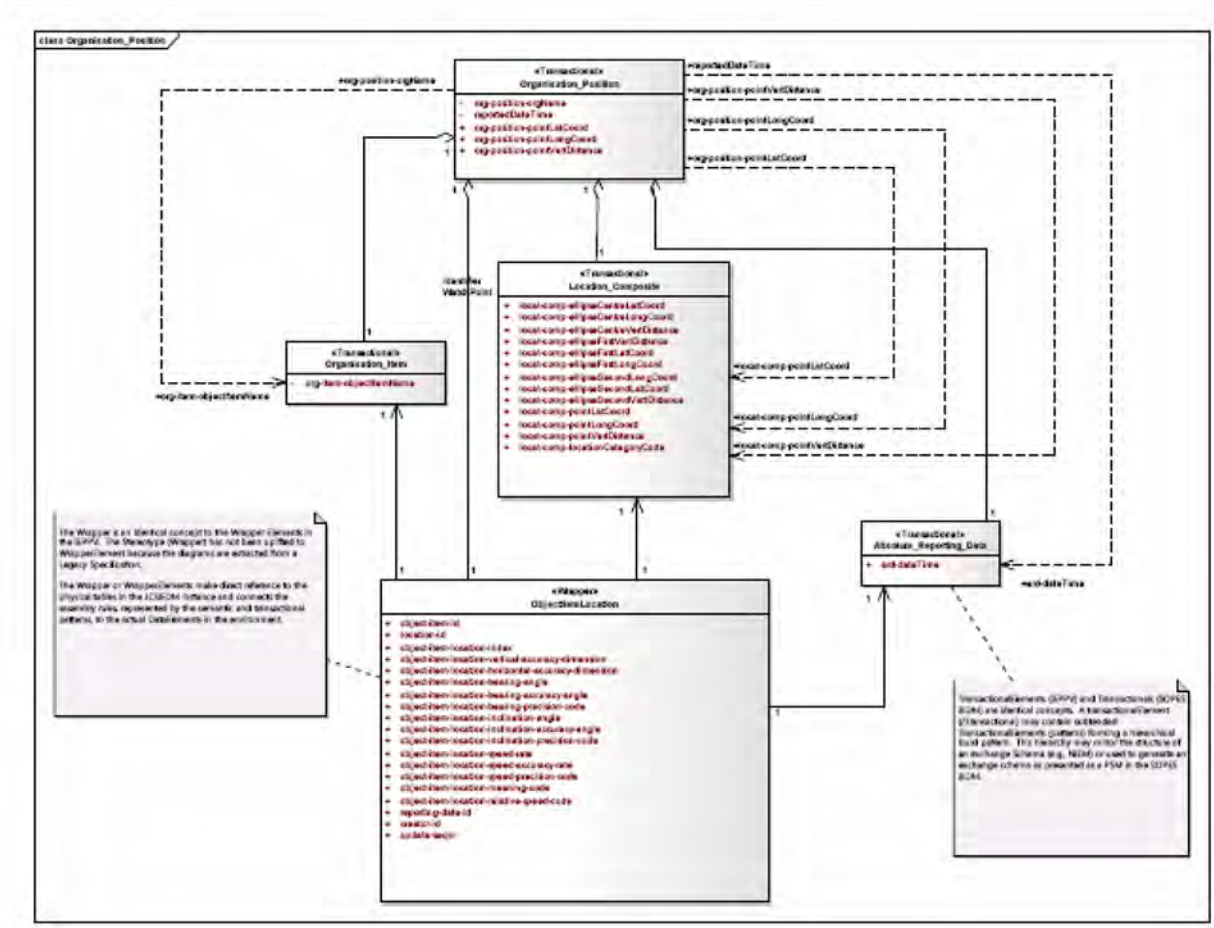

Figure D.9 - Organization_Position

D.6.8 WrapperElement

Figure D.10 illustrates the mapping of a WrapperElement (Wrapper in SOPES) and the physical table definition of the JC3IEDM. As illustrated in this model there is a transformation of physical into the logical naming conventions. The WrapperElement sole function is the mapping of a policy model to its operational information stores.

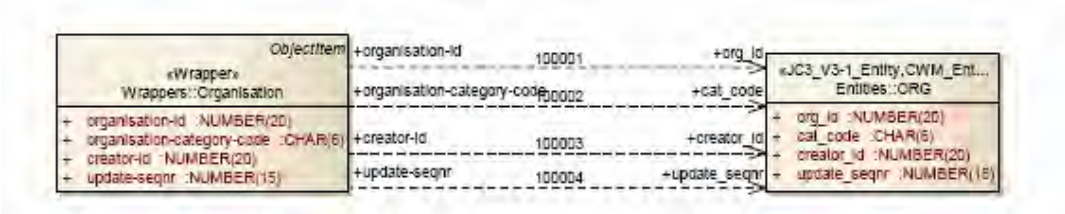

Figure D.10 - WrapperElement

D.7 CP-2 Policy Model Examples

Compliance Point 2 (CP-2) extends the CP-1 focus on the expression of rules for assembly and processing of data and information elements. CP-2 modeling patterns provide the ability to express rules governing the assembly of formatted messages. As illustrated in the following table, each of three CP-2 compliance points extends the number of message elements supported.

Table D.3 - CP-2 Elements

Message Element	Sub-element	CP-2a	CP-2b	CP-2c	Type of FilteredSemantic
Message		1	1	1	
Message Metadata		1	1	1	Yes
Submitter Metadata			1	1	Yes
Information Payload		1	0	0	Yes
Information Package		0	1	1..n	
	Information Package Metadata		1	1	Yes
	Information Payload		1	1	Yes
	Digest		1	1	Yes
	Attachment Summary			1	
	Linkages		1		
	Narrative Text			1	
	Rendering Instruction		1	1	
Attachment		0..1	0..n	0..n	

D.7.1 CP-2a Examples

The following sub clauses illustrate and describe the modeling patterns for the CP-2 Message Structure.

D.7.2 CP2 InformationExchangeSpecification & Information Specification

The informationExchangeSpecification for CP-2 is similar to those presented in CP-1. The CP-2 InformationSpecification replaces the FilteredSemanticElement (resulting in an unformatted dataset) with a Message (resulting in a formatted message).

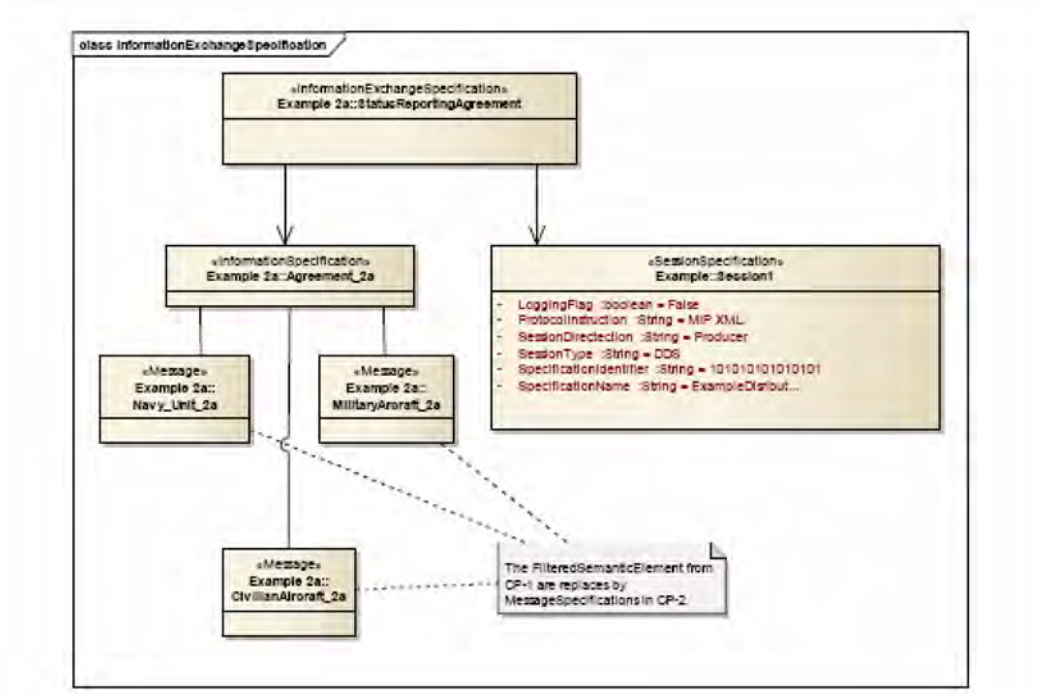

Figure D.11 - CP-2 InformationExchangeSpecification & InformationSpecification

D.7.3 CP-2a MessageSpecification

The following figure illustrates the CP-2a Message Structure. The CP-2a massage supports a single InformationPayload that references a single filtered semantic. The payload (NavyUnit_SA) contains the resulting data after the enforcement of the referenced FilteredSemantic. The filters limit the resulting organization information that refers to a Unit belonging to the Navy and where the report times match metadata filter constraints.

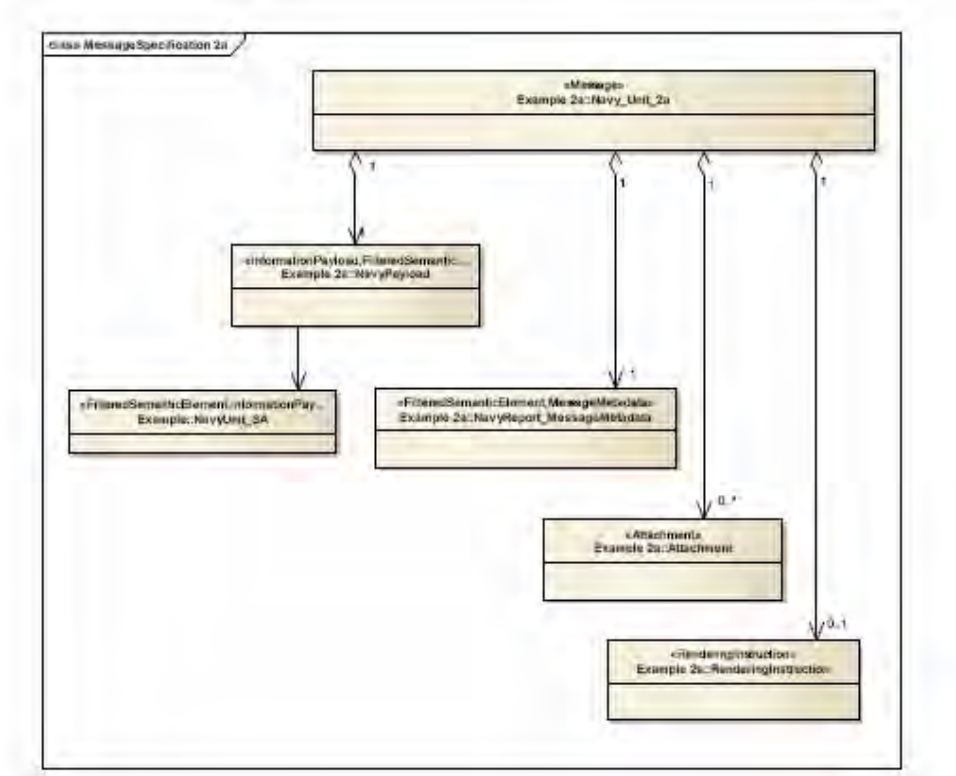

Figure D.12 - CP-2a Message (Single Payload)

D.7.4 MessageMetadata

The assembly of MessageMetadata is performed by a single FilteredSemanticElement. As illustrated Message Metadata as the references MessageMetaDataSematic, which in turn aggregates DataSubmitterMetadata (TransactionalElement) and PublishMessageMetadata (TransactionalElement). These combine to assemble the metadata needed for the message.

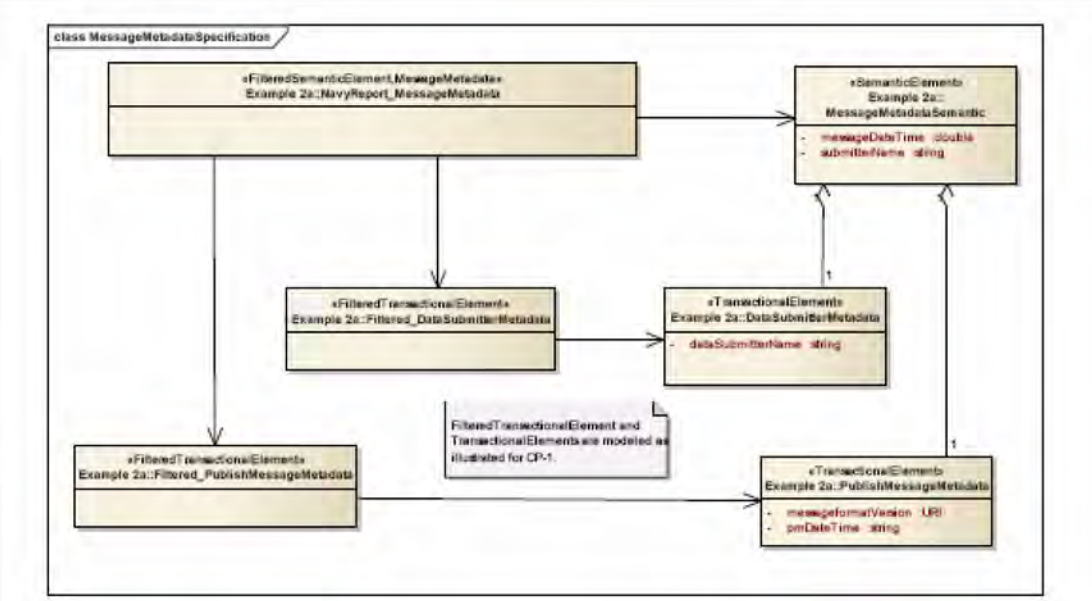

Figure D.13 – Message Metadata

The MessageMetadata is another use of the SemanticElement. It is used to assemble (aggregate, transform, Filter) metadata elements for a message.

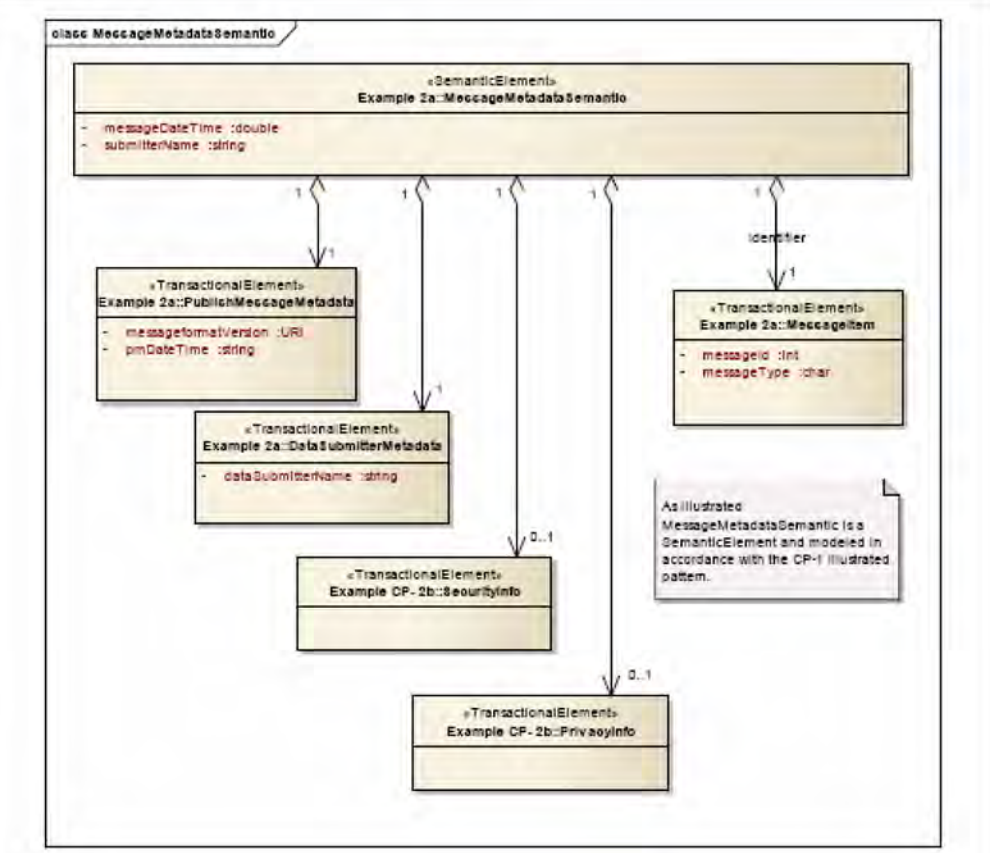

Figure D.14 - Message-Metadata: SemanticElement

D.8 CP-2b Examples

CP-2b adds the InformationPackageStructure to the Message.

D.8.1 Example MessageSpecification_2b

The CP-2b Message Specification further extends the message structure by replacing the InformationPayload used in CP-2a and replacing it with an InformationPackage. The InformationPackage adds several features to the overall message structure:

- Package Specific Metadata
- A Digest
- Rendering Instructions
- Payload

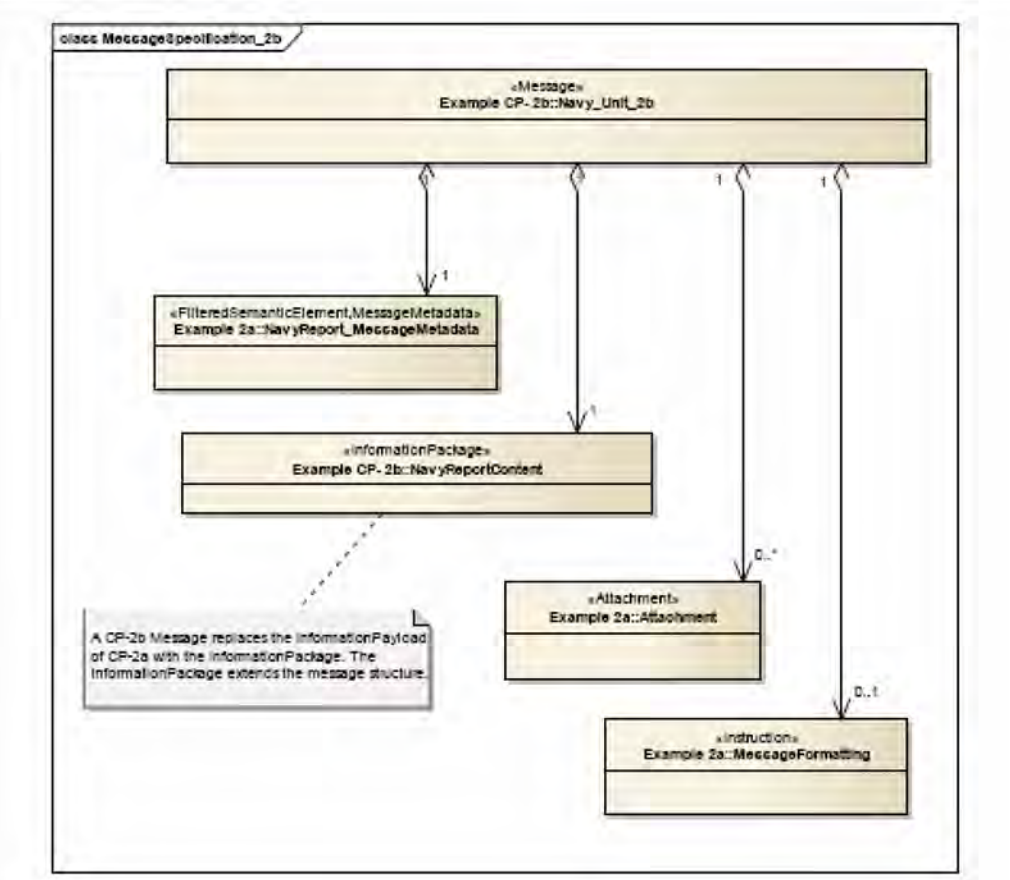

Figure D.15 - Example Message Specification 2b

D.8.1.1 InformationPackage

The following figure depicts the elements of the NavyReportContent InformationPackage.

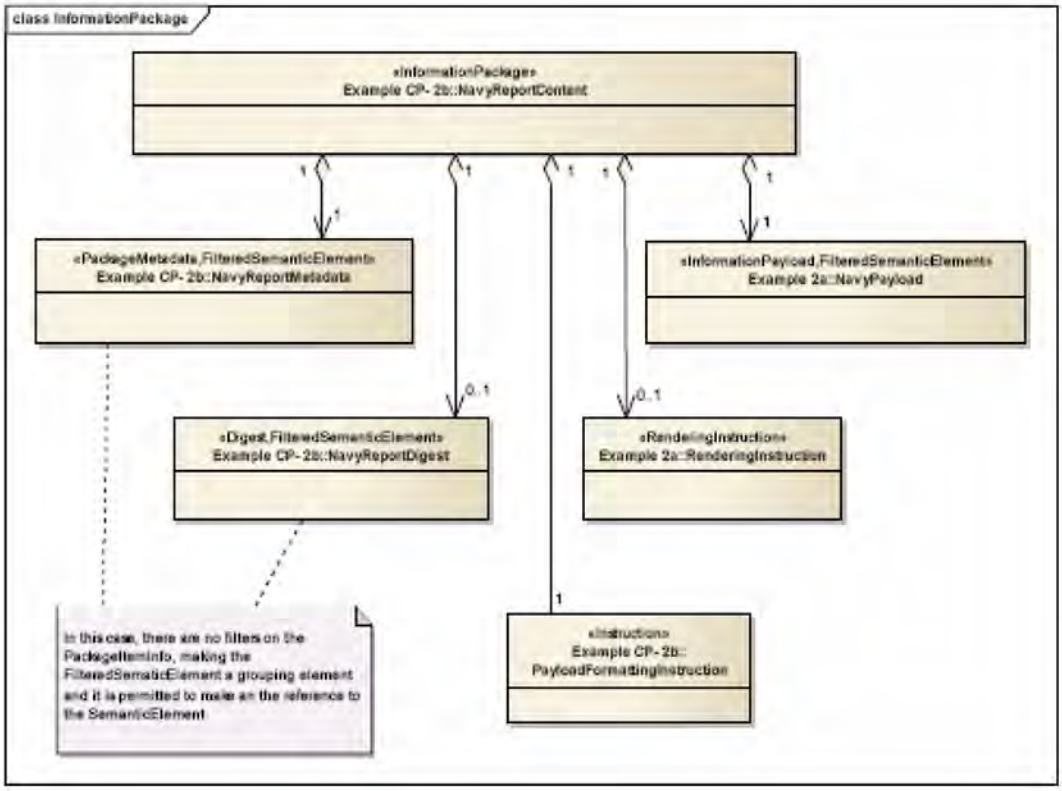

Figure D.16 - Example InformationPackage

D.8.1.2 InformationPackageMetadata

The InformationPackageMetadata is another use of the SemanticElement. It is used to assemble (aggregate, transform, Filter) metadata elements.

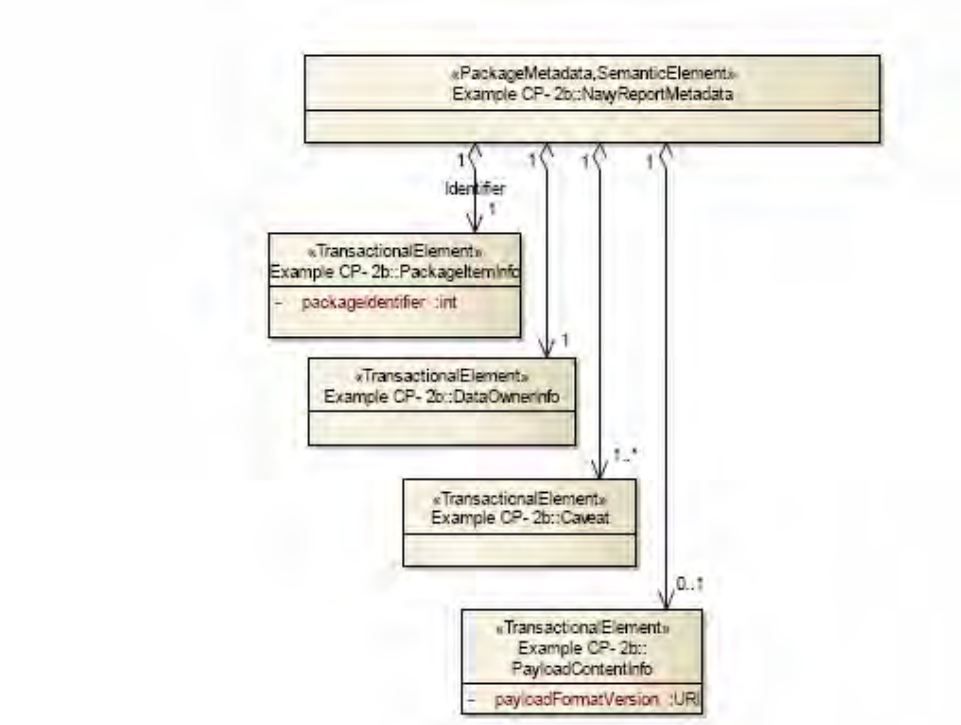

Figure D.17 - InformationPackageMetadata

D.8.1.3 InformationPayload

The informationPayload is the same as the pattern used in CP-2a. It has been enclosed within the InformationPackage to align it with its associated digest and rendering instructions.

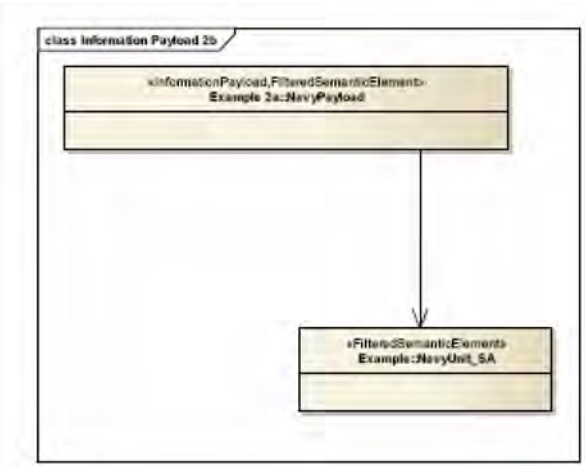

Figure D.18 - Example InformationPayload

D.8.1.4 Digest

The Digest references a FilteredSemanticElement in the same manner as the InformationPayload.

D.9 CP-2c Examples

The following diagrams illustrate policy model elements for CP-2c. CP-2c expands on the modeling construct from CP-2b.

D.9.1 Example Message Specification_2c

The CP-2c Message extends the message structure of CP-2a by permitting multiple InformationPackages. It also includes additional information within the InformationPackage: AttachmentSummary, linkages, and NarrativeText.

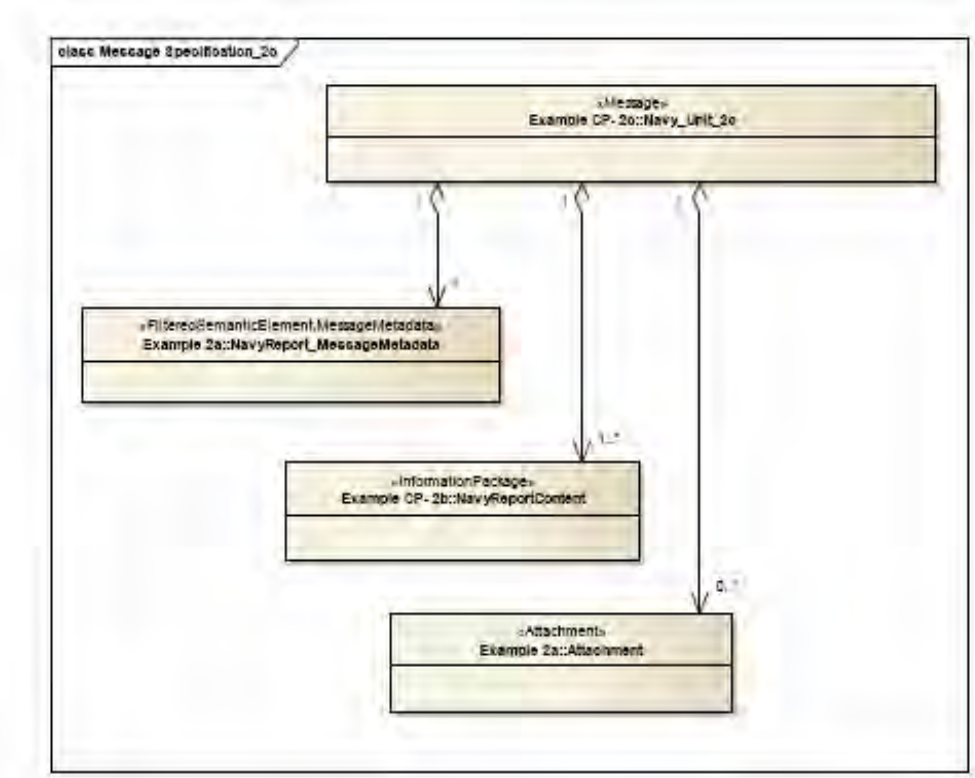

Figure D.19 - CP-2c Message

D.9.2 InformationPackage

The following figure depicts the elements for a CP-3 Information package. Most of the element types have been addressed in previous sub clauses.

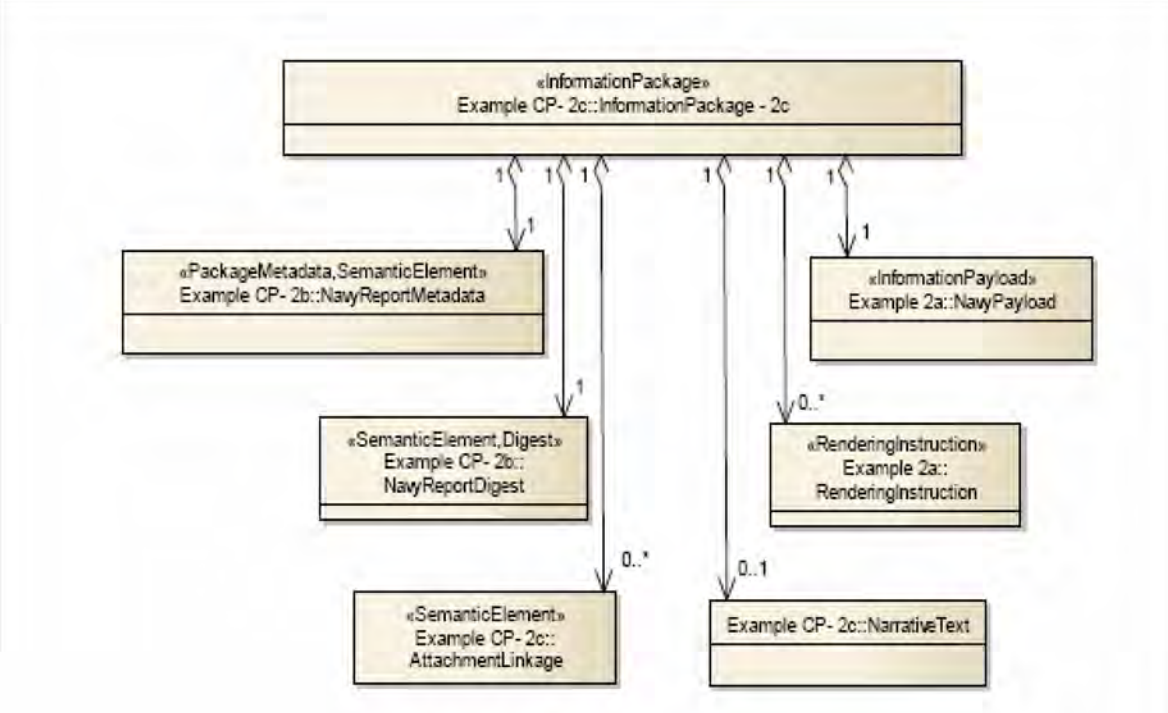

Figure D.20 - Example Information Package 2c

D.9.3 Attachment

The attachment enables the inclusion of references to binary files that can be embedded within a message structure.

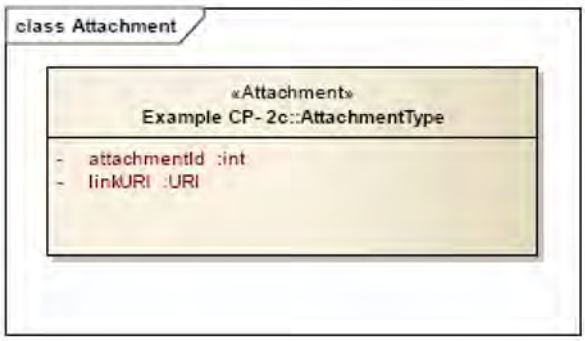

Figure D.21 - Example Information Package 2c

Annex E Bibliography (Informational)

These are informational Elements referenced in this specification:

- Pellet: OWL 2 Reasoner for Java; http://clarkparsia.com/pellet/
- Ponder; http://ponder2.net/cgi-bin/moin.cgi/PonderPublications?action=AttachFile&do=view&target=2001ponderlanguage.pdf
- Protégé; http://protege.stanford.edu/overview/index.html
- Resource Description Framework (RDF): Concepts and Abstract Syntax, W3C Recommendation 10 February 2004, available at http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/
- RDF Vocabulary Description Language 1.0: RDF Schema, W3C Recommendation 10 February 2004, available at http://www.w3.org/TR/2004/REC-rdf-schema-20040210/
- Security Assertion Markup Language (SAML); from OASIS; http://docs.oasis-open.org/security/saml/v2.0/saml-2.0-os.zip
- The Dublin Core® Metadata Initiative: http://www.dublincore.org/
- SKOS Simple Knowledge Organization System Reference: http://www.w3.org/standards/techs/skos#w3c_all
- ISO/IEC JTC1 SC32 WG2 is the Working Group that develops international standards for metadata and related technologies: http://www.metadata-standards.org/ -- home page of ISO JTC 1 SC32 WG 2, where the ISO standard documents
- eXtensible Access Control Markup Language (XACML); from OASIS; http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.pdf

Annex F Terms and Acronyms (Informational)

F.1 General Terms and Definitions

The following represent more general terms used in this specification:

Accurate: Information that exactly, precisely, and correctly presents availability, usability and deploy-ability of C4ISR capability, systems and services.

Aggregation: Defines the process through which data elements are combined to referentially and semantically complete data sets.

Caveat Separation: The process for selective exchange of information based on security policy and security profiles of the information and consumer of the information. Caveat separation may apply to data elements with the information or the aggregation of information.

Communication Channel: A means of communication or access. For the purposes of this specification communication channels will be limited to the middleware used to move information between suppliers (/publishers) and consumers (/subscribers).

Confidential Information: Privileged communication shared with only a few people for furthering certain purposes, such as with an attorney for a legal matter, or with a doctor for treatment of a disease. Receiver of confidential information is generally prohibited from using it to take advantage of the supplier of that information.

Contract: (source: SOPES and UPDM) A contract represents a grouping of Semantic construction rules and information flow controls which specify a formal information sharing agreement between two or more operational nodes or participants in a domain or community. Equivalent terms in this specification are Information Exchange Agreement, Information Exchange Specification and Information Exchange Contract.

Challenged Networks or Communication: Under operational conditions most front line communications are provided by radio (HF, VHF, or HCDR). These forms of communications are inherently less robust than the Wi-Fi and wired networks realized by most organizations. Challenged refers to the reality that these networks:

- Have limited bandwidth capability (as low as 1Kb/Sec)
- Are prone to outages (e.g., range limitations, jamming, and voice override)
- Large node count
- Packet loss

Classified Information: Classified information is sensitive information to which access is restricted by law or regulation to particular classes of persons. A formal security clearance is required to handle classified documents or access classified data.

Conceptual Interoperability: The assumptions and constraints of the meaningful abstraction of reality – are aligned, the highest level of interoperability is reached. This requires that conceptual models are documented based on engineering methods enabling their interpretation and evaluation by other engineers.

Common Operating Picture (COP): A collaborative set of technologies that provide the user(s) with a shared understanding of the operational environment including: Threats; Opportunities; Resources; Situational Awareness and other relevant information. The technologies combine to integrate perspectives; deliver actionable knowledge and structure information to the specific User(s) needs.

Common Representational Operating Picture (CROP): Is equivalent to the COP but limits access to that information required to exercise the role or function of the user.

Community: A community of interest or community of practice.

Community of Interest (CoI): A collaborative group of users that must exchange information in pursuit of its shared goals, interests, missions, or business processes and therefore must have shared vocabulary for the information exchanges. DoD 8320.2, December 2, 2004.

Community of Practice: Informal, self-organized, network of peers with diverse skills and experience in an area of practice or profession. Such groups are held together by the members' desire to help others (by sharing information) and the need to advance their own knowledge.

Crisis Management: Coordinated actions taken to diffuse crises, prevent their escalation into armed conflict and/or contain resulting hostilities. The crisis management machinery provides decision-makers with the necessary information and arrangements to use appropriate instruments (political, diplomatic, economic, and military) in a timely and coordinated manner. (MC 400/1).

Data: Facts used usually to calculate, analyze, or plan.

Data Composite: A data set resulting from the aggregation of data elements.

Data Integrity: Compliance to the allowable types, ranges or domain values for each data element (or attribute).

Data Integration: The process of combining two or more data elements from separate sources into a single semantically and referentially complete piece of information (or business object).

Data ownership: Identification that the data or information is controlled by the entity in such a way that only that entity is allowed to modify the data or information elements.

Data Packaging: see Information Packaging.

Data Pattern: A plan, diagram, or model to aggregate data elements.

Data Stewardship: Accountable for **integrity and quality of data.**

Deadline: A QoS attribute describing the latest acceptable time for the occurrence of certain events.

Definition: A representation of a concept by a descriptive statement which serves to differentiate it from related concepts.

Domain: A sphere of knowledge or information identified by a name.

Dynamic Interoperability: As a system operates on data over time, the state of that system will change, and this includes **the assumptions and constraints** that affect its data interchange. The systems are able to identify the state changes in the assumptions and constraints and they can adjust or be adjusted to address changes in context or situation. The effect of the information exchange within the participating systems is unambiguously defined.

Information: Facts or details about a subject (Data in Context; composite of data elements used to inform a decision).

Information Artifact: A composite of data elements that satisfy the Semantic construction rules for an agreement to exchange information between a supplier and a consumer.

Information Consumer: Any User, System Application, Channel or Node using information managed by the IEPPS.

Information Contract: An agreement between an information supplier and information consumer to exchange selected information, based on a specified format, protocol and communication link.

Information Quality: Describes the ability of organizations, systems and persons to provide information that is:

- **Trustworthy**: Information quality and content can be trusted by stakeholders, decision makers and users.
- **Relevant**: Information content tailored to specific needs of the decision maker.
- **Timely**: Information provided when and where it is needed to support the decision making process.
- **Usable**: Information is presented in a common functional format, easily understood by the decision makers and their supporting applications.
- **Complete**: Information that provides all necessary and relevant data (where available) to facilitate a decision.
- **Concise**: Information is provided in a form that is brief and succinct, yet including all important information.
- **Trusted**: Information that is accepted as authoritative by stakeholders, decision makers and users.
- **Secure:** Information is protected from inadvertent or Malicious Release to unauthorized persons, systems or organizations.
- **Protected:** Information is protected from inadvertent or malicious release.

Information Packaging: The process of assembling (aggregating, transforming, tagging/marking and redacting/filtering) data and information elements and formatting them to service a specific information exchange requirement.

Information Processing: The parsing, transformation and marshaling of information and data elements to information or data store(s).

Information Supplier: This includes any user, application or system providing information to the environment.

Marshaling: Defines the process through which data sets are divided and put into the data elements described by the underlying data store(s).

Memorandum of Understanding (MOU): A bilateral or multilateral agreement between parties.

Messaging Protocol: The rules, formats and functions for exchanging messages between the components of a messaging system.

Middleware: Software that serves as an intermediary between systems software and an application.

Ontology: "In the context of knowledge sharing, the term ontology means a specification of a conceptualization. Ontology is a description (like a formal specification of a program) of the concepts and relationships that can exist for an agent or a community of agents. This definition is consistent with the usage of ontology as set-of-concept-definitions, but more general. And it is certainly a different sense of the word than its use in philosophy." DOI:10.1006/knac.1993.1008 DOI:10.1006/ijhc.1995.1081

Operation: For the purpose of this RFP the term operation is restricted to events and activities describing a Crisis Response Action including Military.

Operational Context: A set of network, node, system, application or user characteristics that define the current state of dynamically evolving operational conditions.

Operational Domain: The sphere of knowledge, influence, or activity for a specific mission or operation.

Pattern: A plan, diagram, or model to be followed in making things (in this instance – dataset conforming to information sharing and safeguarding agreement).

Planned Incident: An incident for which there exists standard operating procedures or safeguards to mitigate or recover from the impact of the incident.

Planned Threat: A threat for which there exists standard operating procedures or safeguards to prevent or mitigate the impact of the threat.

Policy: "a definite course or method of action selected from among alternatives and in light of given conditions to guide and determine present and future decisions." http://www.merriam-webster.com/dictionary/policy. Within this document it refers to: "a defined course or method of action in response to a request for or change in information or data." Within the context of this specification: "specification of a method of action for aggregating, transforming and filtering data and information elements to conform to stipulated Semantic construction rules for an information sharing agreement or Community of Interest."

Pragmatic Interoperability: The systems are aware of the methods and procedures that each system is using. The use of the data – or the context of its application – is understood by the participating systems; the context in which the information is exchanged is unambiguously defined. This layer puts the (word) meaning into context.

Private Information: Information about behavior that occurs in a context in which an individual can reasonably expect that no observation or recording is taking place, and information which has been provided for specific purposes by an individual, which the individual can reasonably expect will not be made public.

Proprietary Information: Privately owned knowledge or data, such as that protected by a registered patent, copyright, or trademark.

Protocol Data Unit (PDU): Binary variable length messaging protocol used by the MIP Data Exchange Mechanism.

QoS History: A record of past information generated by the system that is kept around for the benefit of applications that are late joining the network.

QoS: Quality of Service - A set of attributes that can be used to define the middleware's capabilities to meet the requirements of the application for the purpose of data-delivery or management such as reliability, ownership policy, history size, time-to-keep, etc.

Real-time: Refers to the event-triggered (e.g. data change) global update of information across all nodes, systems and applications requiring access to the information.

Redact: To obscure or remove (text or data) from a document prior to publication or release. This function is typically performed by data filters.

Releasable Dataset: A collection of data elements that can be provided to the recipient(s) as defined by policy.

Releasable Message: A message where the content can be provided to the recipient(s) as defined by policy.

Reliability: A QoS attribute describing the guarantees and feedback provided to the application regarding the delivery of the information supplied to the middleware.

Responsible Information Sharing: Compliant with law, regulation and policy; consistent with community and agency strategy and direction, to include protection of sources and methods, and civil liberties and privacy; and accountable through governance and oversight while maximizing the quantity and quality of information that is discoverable and accessible to users and partners.

Semantic Integrity: Compliance to the structure, format and content (mandatory or optional) for information sets (or business objects).

Semantic Interoperability: Semantics concerns the study of meanings. Semantic interoperability refers to the ability of information systems to exchange information/data with unambiguous, shared meaning. It is a requirement to enable information integration, machine analytics, inferencing, knowledge discovery, and data federation. Semantic interoperability is not only concerned with the packaging of data (structure and syntax), but the simultaneous provision of intent and meaning (semantics).

Semantic Pattern: A plan, diagram, or model to aggregate Transactional patterns that conform to an information sharing and safeguarding agreement.

Service Level Agreement (SLA): An agreement between two or more parties where the level of service is formally defined.

Specialized Data Set: A collection of data that is specifically tailored to a specific context and recipient.

Specialized Message: A message for which the content is specifically tailored to a specific context and recipient.

Stakeholder: a person with an interest or concern in the effective application of ISS Policy.

Stage: To gather and prepare information for release to a community in accordance with established policy, memorandum of understanding or service level agreements.

Syntactic Interoperability: A common structure to exchange information; i.e., a common data format is applied. On this level, a common protocol to structure the data is used; the format of the information exchange is unambiguously defined.

Tearline: A physical line on a message or document separating categories of information that have been approved for disclosure and release.

Technical Interoperability: An agreed communication protocol exists for exchanging data between participating systems. The protocol operates over an agreed and established communication infrastructure allowing systems to exchange bits and bytes, and the underlying networks and protocols are unambiguously defined.

Trust: Within the scope of this RFP – Trust refers to the level of confidence an information supplier has relating to the release of selected information to a specific consumer of that information.

Unplanned Incidents: An occurrence of an action or situation that is not addressed by plans or operating procedures.

Unplanned Threat: An expression of intention to inflict evil, injury, or damage that is not accounted for in the threat risk assessment or mitigation plans.

Vocabulary: A representation of a set of concepts by formal, descriptive statements which serves to differentiate those concepts from related concepts within a given domain or area of expertise. Terminological dictionary (3.7.1) which contains designations (3.4.1) and definitions (3.3.1) from one or more specific subject fields (3.1.2). NOTE: The vocabulary may be monolingual, bilingual, or Multilingual. ISO 1087-1:2000.

F.2 Acronyms

The following acronyms are used as part of this specification.

C4I	Consultation, Command, Control, Communications and Intelligence
COP	Common Operational Picture
CP	Compliance Point
CRO	Crisis Response Operation

CROP	Common Representative Operational Picture
DEM	Data Exchange Mechanism
DHS	Department of Homeland Security
DNDAF	Department of National Defence Architecture Framework
DODAF	Department of Defense Architecture Framework
DTF	Domain Task Force
EDXL	*Emergency Data Exchange Language*
EDXL-DE	*Emergency Data Exchange Language Distribution Element*
HCDR	High Capacity Digital Radio
HF	High Frequency
ICAM	Identity, Credentials and Access Management
IE	Information Exchange
IEA	Information Exchange Agreement
IEAPV	Information Exchange Access Policy Vocabulary
IECPV	Information Exchange Credential Policy Vocabulary
IEDM	Information Exchange Data Model
IEDPV	Information Exchange Dissemination Policy Vocabulary
IEF	Information Exchange Framework
IEIPV	Information Exchange Identity Policy Vocabulary
IEM	Information Exchange Mechanism
IEP	Information Exchange Policy
IEPAS	Information Exchange Policy-based Authorization Service(s)
IEPL	Information Exchange Policy Language
IEPMS	Information Exchange Policy Management Service(s)
IEPV	Information Exchange Policy Vocabulary
IEPPS	Information Exchange Policy-based Packaging Service
IEPPV	Information Exchange Packaging Policy Vocabulary
IEQPV	Information Exchange Quality of Service (QoS) Policy
ISA	Information System Application
ISE	Information Sharing Environment
LEXS	Logical Entity eXchange Specification
MDA	Model Driven Architecture
MEM	Message Exchange Mechanism
MIP	Multilateral Interoperability Programme
MLS	Multi-level Security
MODAF	Ministry of Defence Architecture Framework
MOF	Meta-Object Facility
MOU	Memorandum of Understanding

NAF	NATO Architecture Framework
NATO	North Atlantic Treaty Organization
NGO	Non-Government Organization
OCL	Object Constraint Language
ODM	Ontology Definition Metamodel
OODBMS	Object Oriented Database Management System
ORDBMS	Object-Relational Database Management System
PDU	Protocol Data Unit
PIM	Platform Independent Model
PM-ISE	Project Manager Information Sharing Environment
PSM	Platform Specific Model
PVO	Private Volunteer Organization
SLA	Service Level Agreement
SOPES	Shared Operational Picture Exchange Services
UPDM	Unified Profile for DODAF and MODAF
UML	Unified Modeling Language
XMI	XML Metadata Interchange

Annex G Addressing RFP Requirements (Informational)

G.1 RFP Required Discussions

G.1.1 Existing Policy Languages

This specification is focusing on the generation of a UML Profile for the IEPPV and through the Unified Profile for DODAF and MODAF (UPDM) an alignment for architecture frameworks such as DODAF, MODAF, and NAF. The specification is seeking an architectural basis for the specification and design of semantic interoperability solutions; providing the institutional knowledge retention needed to sustain and maintain interoperability in response to dynamic real world events.

G.1.2 Relationship to Other Specifications and Standards

The following table outlines the relationship between the IEPPV and other related specifications.

Table G.1 - Related Specifications and Standards

Specification	Reference	Relationship
UPDM	http://www.omg.org/spec/UPDM	The UPDM provides one set of architectural contexts for the semantic and business rules encompassed by the IEPPV. The IEPPV addresses a gap in the DODAF, MODAF, and NAF; this gap involves the specification and design of business rules (aggregation, transformation, redaction, and formatting) between the Information Exchange Requirements (IERs) and Logical Data Models.
SOPES	http://www.omg.org/cgi-bin/doc?formal/2011-05-04.pdf	The IEPPV integrates the modeling profile provided in Annex A to the SOPES IEDM Version 1.0 specification. The “Concept” Models illustrate the relationships between the IEPPV Concepts and UPDM Concepts. Several terms have been generalized but have a one-to-one relationship with a term in the SOPES specification.
DDS	http://www.omg.org/DDS	The IEPPV Distribution Specification is intended to provide a linkage to a UML Profile for DDS.
LEXS	http://lexs.codeplex.com/	The IEPPV “Information Specification” provides the concepts needed to specify delivery concepts in architecture and policies needed to support the LEXS.

Table G.1 - Related Specifications and Standards

NIEM	https://www.niem.gov/Pages/default.aspx	The IEPPV provides vocabulary to specify the rules for aggregating and processing datasets published and received in XML format. The XSD needed to publish and process the XML documents are specified in the Formatting and rendering instructions that may be embedded in the messages.
XML	http://www.w3.org/REC-xml/	The IEPPV provides vocabulary to specify the rules for aggregating and processing datasets published and received in XML format as specified in a NIEM IEPD. The XSD in the IEPD needed to publish and process the NIEM documents are specified in the Formatting and rendering instructions that may be embedded in the messages.
Unified Modeling Language	http://www.omg.org/UML	The IEPPV Modeling Profile for UML: Annex B.
Ontology Definition Metamodel (ODM)	http://www.omg.org/spec/ODM/1.0/	Integral part of the MDA transformation used to generate the OWL language implementation provided as a separate machine readable file - see specification Manifest.
OWL 2 Web Ontology Language	http://www.w3.org/TR/2009/REC-owl2-syntax-20091027/	OWL Expression of the vocabulary provided in OWL, see Machine readable files to this specification.
Joint Consultation Command and Control Information Exchange Data Model	https://mipsite.lsec.dnd.ca/Public%20Document%20Library/Forms/AllItems.aspx?RootFolder=%2fPublic%20Document%20Library%2f04-Baseline_3.1%2fInterface-Specification%2fJC3IEDM&FolderCTID=0x012000CDEC559A618DF74781A1E0AE00DB1626	Inherent part of SOPES, which is used as a foundation for the example model in Annex E.
Information Exchange Policy-based Packaging Service (IEPPS)	mars/2011-12-12	The IEPPV represents the Policy Vocabulary for the IEPPS.
SKOS		

G.1.3 Supporting the “ilities”

The Information Exchange Framework (IEF) is intended to specify an agile, flexible, extensible, supportable, and maintainable platform for semantic interoperability; as characterized by;

- Agility: The quality or state of being able to move or adapt with quick easy grace.
- Flexibility: Characterized by a ready capability to adapt to new, different, or changing requirements.

- Extensibility: Characterized by a capability to be extended.
- Supportability: Inherent characteristics of design that enables the effective and efficient maintenance and support of a system throughout the life cycle.
- Serviceability: Degree to which the servicing of an item can be accomplished with given resources and within a specified timeframe.
- Maintainability: Characteristic of design and installation, which determines the probability that a failed system can be restored to its normal operable state within a given timeframe, using the prescribed practices and procedures.

The Information Exchange Policy Vocabulary supports these goals by supporting these objectives in the following manner.

Table G.2 - IEPPV Supports to the "ilities"

Objective	Description
Agility	The IEPPV is specifically designed to separate policies and rules governing the exchange of information from operating systems and managing them independently. The ability to load policy sets at runtime will allow users to change policies, rules, and constraints to adapt to changes in operational context rapidly, without the need for recoding applications. It is anticipated that the IEF supporting services (e.g., IEPPS, IEPA, and IEPMS) will be able to ingest new or multiple policy sets at runtime and selectively activate these policies, rules, and constraints to address operational context.
Flexibility	The IEPPV being tied to architecture and architecture frameworks will support the analysis and design capabilities needed by organizations to adapt to new, different, or changing requirements.
Extensibility	As demonstrated in the expansion of concepts between the Exchange Vocabulary concepts expressed in the SOPES and UPDM Modeling Profiles, the IEPPV demonstrates the capacity to extend concepts. In later versions, it is anticipated that the concepts expressed will be extended in the domains of Privacy, Identity, and Credentialing.
Supportability	The IEPPV applies several of the MDA concepts that will enable enterprises to share and reuse the defined policies and rules across multiple interoperability requirements within and between organizations. In addition the separation of policy and rules from the enforcing application simplifies the release and deployment of new capability.
Maintainability	The separation of the policies and rules from the executing applications and services means the new policies and rules can be deployed and enforced without the need for the deployment of a new application or service.
Serviceability	Policies can be developed, tested, and deployed by operational and business analysts without the requirement for software development teams. This will reduce the resource requirements need to correct issues or enhance capability.

G.1.4 Model Driven Architecture (MDA)

The IEPPV defines a set for concepts for the expression of rules governing the packaging and processing of information elements (datasets and messages) shared across information system interfaces. To enable the use of Model Driven Architecture, the IEPPV was integrated into a UML Profile (Annex B). The profiles will enable users to model the packaging and processing patterns needed to align their data environments:

1. To the exchange and information protocols agreed to in the information sharing agreement; and

2. to the information sharing and safeguarding (e.g., Security, Privacy and confidentiality).

The use of UML to develop the policy/rules models provides the platform for exploiting MDA to transform the models into machine consumable policy languages (e.g., SAML and XACML), middleware scripting language or interface code (e.g., JAVA and C++) .

The application of MDA to the generation of runtime environments will expedite the development, testing, and deployment of capability.

G.1.5 Policy Model Validation

Policy, or more specifically rules, validation should be addressed during specification/design (Analytics / Modeling & Simulation), testing, and post mission.

- Design: during design, the provision of common vocabulary and concept restrictions provides the opportunity for the development of reasoning and analytic applications that can assess the policy sets against user defined criteria.
- Design: Modeling and Simulation (M&S) can be used to test evolving policy models against simulated runtime environments.
- Testing: development of formal test-cases will enable policy sets to be tested. The adoption of MDA and policy automation will enable rapid error correction and regression testing. The separation of policy (serialization to rules to a machine readable form) from the services that automate them will further increase the validation process.
- Post Mission: Enhanced policy sets can be validated against operational logs. This will enable the tuning of operation policies/rules to unforeseen differences in operational context from those specified during design.

The provision of an OWL implementation of the vocabulary will enable the development of machine reasoning applications that can for example:

- Assist in the identification of complex relationships in policy models that may affect the worthiness of the model against security and privacy policies.
- Assist in the identification of conflicts between policy models.
- Assist in the identification tampering in deployed policy model.
- Assist in the assessment of a partners policies and their conformance to a MOU or SLA.

G.1.6 Use with Current Interoperability Specifications

The IEPPV is a vocabulary that can be used to describe data aggregation, information protection, and information processing policies in a manner that can be translated into any number of policy and rules languages used by a wide range of interoperability solutions. At present we are targeting IEF, DDS, OWL families of specifications. We are confident that additional Language Implementation and serializations will be developed to support additional families of interoperability specifications.

G.1.7 System and Software Platforms

The IEPPV defines a set of concepts that combine to specify the packaging and processing patterns for information shared between information systems in a clear, consistent, and platform independent manner. The use of policy models in UML provides for the integration aspects of the policy models into broader enterprise architecture constructs (e.g., platform and

system views (interfaces), operational deployment views; information and data views, and security views. MDA can be used to translate the policy model to the policy, rules and scripting languages or code required by the platform specific implementations.

G.1.8 Users of IEPPV

The IEPPV specification is targeted at the following categories of users:

- Information Analysts and Architects
- User, Operator
- System Integrators
- Stakeholders
- Security / Privacy Specialists
- Tool Vendors

Table G.3 shows how SOPES IEDM benefits these various types of users.

Table G.3 - IEPPV Use Cases

User Category	Use Case	Problem Statement	Required Capability	IEPPV Delivers
1. Information Analysts and Architects	Interface Specification and Design	Lack of a clear and consistent vocabulary and language for documenting and communicating information sharing and safeguarding (ISS) packaging specifications and designs, independent of target platform and services. Inability to concisely and consistently communicate ISS specifications and designs to stakeholders, users, and developers. Inability to simultaneously define ISS rules within standards architecture views and viewpoints.	A common vocabulary for precisely and accurately specifying ISS rules and constraints. This specification addresses the rules governing the packaging of information payloads and/or messages. ISS Rules and constraints include: • Aggregation of data and information elements; • Transformation of data and information elements; • Insertion of Metadata Tags and Markings; • Filtering, guarding, and redacting data and information elements; • packaging and formatting payloads and/or messages; and • Application of release and receipt instructions. The ability to communicate, validate, and verify ISS specifications with stakeholders and users.	IEPPV provides a common vocabulary to clearly and concisely specify information and message packaging rules in a manner that is independent of target platform and services. IEPPV provides a language independent Vocabulary. UML and OWL language representations are provided, however other representations are possible (e.g., RulesML, SAML, and XACML). Provides a UML Profile that enables the integration of the IEPPV into standard architecture frameworks and tools. This integration would align ISS rules within the context of enterprise and system architectures. The UML profiles provide a specialized Class Diagram to define core elements of an ISS specification, providing a clear and concise communication vehicle for stakeholders, users, and developers.

User Category	Use Case	Problem Statement	Required Capability	IEPPV Delivers
			The ability to transform ISS rules into multiple Rules and Policy machine readable and enforceable Languages.	The IEPPV is specified in a manner that is independent of the community or enterprise information domain vocabulary. This enables the use of the IEPPV for the specification and design of most IIS requirements. The UML profile provides a platform independent method for expressing ISS rules. The profile provides the opportunity to exploit QVT tools to transform UML models into platform specific platform and service implementations.
2. User and Operator	Business and/or Operational Analysis	Inability to communicate ISS rules and in a clear, consistent manner. Inability to define ISS rules independent of the target platforms and services. Inability to capture and reuse ISS rules from previous missions and operations. Inability to share ISS rules and constraints with business and operational partners. Inability to retain institutional memory and knowledge about business and operational interfaces. Inability to rapidly adapt interfaces to changing business and operational context. Inability to trace ISS rules to initiating legislation, regulation, and policy.	A common vocabulary for precisely and accurately describing ISS rules. The ability for operational analysts to rapidly define, develop, test, certify, and deploy ISS rules to address dynamic changes in operating conditions or business opportunities. The ability to support the evolutionary development, testing, and deployment of ISS rules. The ability to capture, maintain, and reuse ISS rules.	IEPPV provides a common vocabulary to clearly and concisely specify information and message packaging rules. Provides a UML Profile that enables the integration of the IEPPV into UPDM. The common vocabulary and pattern based approach provided by the IEPPV facilitates the sharing and reuse of models and serialized rules. The IEF separation of policy/rules from service implementations enables greater flexibility in the runtime environment and increases the ability of users to develop and deploy ISS rules that accommodate changes in the operating or business environment.

User Category	Use Case	Problem Statement	Required Capability	IEPPV Delivers
				The common vocabulary and UML profile will provide users with the ability to develop and exploit Architecture and tool environments. Architecture Frameworks and tools provide the opportunity to trace ISS rules to initiating legislation, regulation, and policy.
3. Stakeholders and Business Owners	Life-cycle	Inability to control the spiraling life-cycle costs for information sharing and safeguarding solutions. Inability to modernize rigid and brittle point-to-point system interfaces that are unable to adapt to changing business and operational requirements. Inability to adapting ISS rules to new or modified legislative, regulatory, and policy mandates.	A common vocabulary for precisely and accurately describing ISS rules. Practices and tools that shorten the development cycles to the translation of legislative, regulatory, and policy mandates to certified and deployed ISS rules. Practices and tools that enable the certification of ISS rules for operation. Practices and tools that provide the ability to retain institutional memory and knowledge pertaining to ISS rules applied for each individual, organization, and external partner. Practices and tools that enable the auditing and analysis of ISS rules. The ability to increase the number of SMEs available to develop and test ISS rules.	IEPPV provides a common vocabulary to clearly and concisely specify information and message packaging rules. Additional IEPV specifications (Figure 2) will address other ISS rules types. IEPPV (UML Profile) provides the opportunity for the IEPPV's integration into standard UML tools, supporting the capture and reuse of artifacts (data and information patterns). This will aid in both the retention of institutional knowledge and the reduction in life-cycle costs. The IEPPV (UML Profile) provides for the definition of platform independent packaging specifications - meaning they can be transformed into the rules and configurations for multiple platforms and service configurations; further helping to control life-cycle costs. IEPPV (UML Profile) can be used as part of an MDA process that translates models (data and information patterns) into machine executable rules. MDA assists in the shortening of development cycles and the control of life-cycle costs.

User Category	Use Case	Problem Statement	Required Capability	IEPPV Delivers
4. System Integrators	Life-cycle	See Stakeholders and Business Owners. Also see Tool Vendors.	See Stakeholders and Business Owners. Also see Tool Vendors.	See Stakeholders and Business Owners..Also see Tool Vendors.
5. Security/ Privacy Specialists	Certification and Accreditation	Inability to validate, verify and certify ISS specifications, designs, and solutions for operation. Inability to generate objective ISS evidence to support analysis and generation of: • ISS Threat Risk Analysis; • ISS Statements of Sensitivity; • ISS Certification and Accreditation. Inability to support and accommodate rapid environmental change.	Practices and tools that enable the validation, verification, and certification of ISS rules. Practices and tools that generate the ISS objective evidence at all phases of development. Practices and tools that support the generation of materials and documentation for: • ISS Threat Risk Analysis; • ISS Statements of Sensitivity; and • ISS Certification and Accreditation.	The IEPPV provides a formal vocabulary specifying information safeguarding as part of enterprise, segment, and system architectures. The IEPPV (UML Profile) provides an opportunity for the development of a tool and services ecosystem (Figure 4) to support the ISS Policy life-cycle. Applications of reusable patterns to reduce complexity in the information sharing and safeguarding specification, design, and implementation. The IEF separation of policy/rules from platform and service implementations will improve an organizations ability to manage policies/ rules within their architectures with assistance in the generation of objective evidence and support the generation of material and documentation supporting: • ISS Threat Risk Analysis; • ISS Statements of Sensitivity; and • ISS Certification and Accreditation.
6. Tool Vendors	Policy Life Cycle Support	The lack of products and services for growing ISS development and testing market. The lack of platform independent practices and standards for ISS Development.	Specification for broad range of tools and services addressing customer needs.	The IEPPV is first in a series of specifications that underpin the development and management of ISS services. The IEPPV provides the opportunity for vendors to develop tools and services to support the Policy life-cycle (Figure 3) and Conceptual Architecture and ecosystem (Figure 4).