MOF to RDF Mapping

Version 1.0

OMG Document Number:	formal/2021-09-01
Date:	September 2021
Specification URL:	https://www.omg.org/spec/MOF2RDF/1.0/
Machine-readable Files:	https://www.omg.org/spec/MOF2RDF/20181201
Normative Files:	https://www.omg.org/spec/MOF2RDF/20181201/MOFSupport.ttl https://www.omg.org/spec/MOF2RDF/20181201/MOFtoRDFmapping.xmi https://www.omg.org/spec/MOF2RDF/20181201/MOFtoRDFmapping.ttl https://www.omg.org/spec/MOF2RDF/20181201/MOFtoRDFprofileMapping.xmi https://www.omg.org/spec/MOF2RDF/20181201/MOFtoRDFprofileMapping.ttl

An Object Management Group Specification

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.

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, under Documents, Report a Bug/Issue (https://www.omg.org/report_issue.)

This page intentionally left blank

Contents

This page intentionally left blank

Preface

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 Meta-model); and industry-specific standards for dozens of vertical markets. More information on the OMG is available at https://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 categories; listing selected major categories below:

Fundamental Information Modeling Technologies

- Unified Modeling Langue (UML)
- Systems Engineering Modeling Language (SysML)
- Interface Definition Language (IDL)

Domain-Specific Technologies

- Business and Enterprise Modeling
- Industrial Engineering and Manufacturing
- Space, Control and Transportation Technologies
- Robotics
- Software and Systems Modernization
- Information Security

- Healthcare
- Retail

Middleware and Real-Time Technologies

- Common Object Request Broker (CORBA)
- Data Distribution Service (DDS)
- Middleware Services
- Real-Time and Embedded Systems
- Signal Processing and Communication

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 Specifications Catalog cited above or by contacting the Object Management Group, Inc. at:

OMG Headquarters
9C Medway Road, PMB 274
Milford MA 01757
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

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 https://www.omg.org, under Documents, Report a Bug/Issue.

1 Scope

RDF and Linked Open Data (LOD) have become important technologies for exposing, managing, analyzing, linking and federating data and metadata. This set of RDF based technologies, sometimes known as the "Semantic Web" or "Web 3.0", are emerging as the lingua franca of data on the web. Industry and governments worldwide are moving to the use of RDF and Linked Open Data as the foundation for publishing their data and metadata in support of transparency. OMG & MOF based models should be a part of the LOD and Web 3.0 data cloud.

More generally, OWL and semantics-based approaches are seen as appropriate technologies for implementing, analyzing and integrating OMG's modeling technologies.

The objectives of this specification are:

- to define a structural mapping between OMG-MOF models and RDF/OWL to provide for better integration of MDA and LOD
- to enable the ability to apply semantic capabilities to MOF compliant models
- to make the information available in MOF compliant models available as LOD web resources. Any MOF based model should be able to become a LOD resource.

The specification defines a set of transformations between OMG-MOF models and RDF/OWL. The scope of the specification includes:

- mapping any MOF compliant model and metamodel to RDF and OWL
- mapping all elements within a model to RDF and OWL
- preservation of original model semantics

The MOF2RDF specification is not intended to

- provide formal logic or model-theoretic semantics addressed by the ODM specification.
- provide facilities for manipulation of either MOF models or RDF models.
- provide facilities for versioning or provenance related to either MOF models or RDF models.
- provide facilities for adding assertions to models, relationships between models, or any specific capability related to inference or query usage of the RDF model.
- constrain the platform-specific rendering of either a MOF model or RDF model.

This page intentionally left blank

2 Conformance

2.1 Basic Conformance

All conformant products must comply with these conformance points:

- *MOF-compliant model to Ontology*. Compliant products map a MOF-compliant (meta) model to an OWL2 ontology, consistent with the MOF to RDF Transformation specifications listed in Clause 9, except for the transformation of AssociationClasses (Clause 9.2.7), since AssociationClass is not yet part of MOF.
- *MOF Semantics*. Compliant products provide the ability to enforce the MOF constraints and semantics
- *Model to Document*. Compliant products map a model (M1) compliant with a MOF-compliant metamodel to a RDF document which is consistent with the ontology created from the metamodel.
- *Model Semantics*. Compliant products enforce the metamodel constraints on RDF documents.
- *Retrieval*. Compliant products provide access to ontologies and documents by dereferencing their URIs

A compliant product is not required to use the transformation specifications directly, but the result of mapping must be consistent with the respective transformation specification.

2.2 Extended Conformance

Products compliant with Extended Conformance must comply to all points of Basic Conformance, and in addition provide complying support for AssociationClass (Clause 9.2.7).

2.3 Profile Conformance

All compliant products must support either Basic Conformance or Extended Conformance, and also provide complying support for MOF Profiles as specified in Clause 10.

This page intentionally left blank

3 References

3.1 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.

[MOF] Meta Object Facility (MOF) Core, version 2.5.1, OMG Specification
https://www.omg.org/spec/MOF/2.5.1

[OWL2] OWL 2 Web Ontology Language Document Overview (Second Edition).
W3C Recommendation, 2012
https://www.w3.org/TR/owl2-overview

[RDF Concepts] Resource Description Framework (RDF) 1.1: Concepts and Abstract Syntax.
W3C Recommendation, 2014,
https://www.w3.org/TR/rdf11-concepts

[RDF Semantics] RDF 1.1 Semantics, W3C Recommendation, 2014,
https://www.w3.org/TR/rdf11-mt

[RDF Syntax] RDF 1.1 XML Syntax Specification, W3C Recommendation, 2014,
https://www.w3.org/TR/rdf-syntax-grammar

[SMOF] MOF Support for Semantic Structures (SMOF), version 1.0, OMG Specification
https://www.omg.org/spec/SMOF/1.0

[SPARQL] SPARQL 1.1 Query Language for RDF, W3 Recommendation 2013,
https://www.w3.org/TR/sparql11-overview

[UML] Unified Modeling Language, version 2.5.1, OMG Specification
https://www.omg.org/spec/UML/2.5.1

[XMI] XML Metadata Interchange (XMI), version 2.5.1, OMG Specification
https://www.omg.org/spec/XMI/2.5.1

[XML Schema] XML Schema Part 2: Datatypes. W3C Recommendation, 2004.
https://www.w3.org/TR/xmlschema-2.

3.2 Non-normative References

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

[Linked Data Book] Linked Data: Evolving the Web into a Global Data Space, Tom Heath and Christian Bizer, Morgan and Claypool 2011. Available online at http://linkeddatabook.com/editions/1.0

[LOD] Linked Data Design Issues. http://www.w3.org/DesignIssues/LinkedData.html .

This page intentionally left blank

4 Terms and Definitions

For the purposes of this specification, the following terms and definitions apply.

Complete MOF (CMOF)
The CMOF, or Complete MOF, Model is the model used to specify other metamodels such as UML2. The CMOF package does not define any classes of its own. Rather, it merges packages with its extensions that together define basic metamodeling capabilities.

Content Negotiation
A pattern used on the web to return different representations of a web resource based on the type of data requested. For example, a model's URI could return RDF to a modeling tool and a web page to a browser.

Entailment
The property that one vocabulary (model) is derived from another by the application of a set of inferencing rules.

Essential MOF (EMOF)
Essential MOF is the subset of MOF that most closely corresponds to the facilities found in object-oriented programming languages and in XML. It provides a straightforward framework for mapping MOF models to implementations such as JMI and XMI for simple metamodels. A primary goal of EMOF is to allow simple metamodels to be defined using simple concepts while supporting extensions (by the usual class extension mechanism in MOF) for more sophisticated metamodeling using CMOF.

Linked Open Data (LOD)
The conventions for using RDF as a backbone for federated internet data. More information can be found at: http://www.Linkeddata.org.

Mapping
Specification of a mechanism for transforming the elements of a model conforming to a particular metamodel into elements of another model that conforms to another (possibly the same) metamodel.

Metamodel
A model that acts as the schema for a family of models.

Meta-Object Facility (MOF)
The Meta Object Facility (MOF), an OMG specification, provides a metadata management framework, and a set of metadata services to enable the development and interoperability of model and metadata driven systems. Examples of these systems that use MOF include modeling and development tools, data warehouse systems, metadata repositories etc.

Model
A formal specification of the function, structure and/or behavior of an application or system.

Object Constraint Language (OCL)
The Object Constraint Language (OCL), an adopted OMG standard, is a formal language used to describe expressions on MOF models. These expressions typically specify invariant conditions that must hold for the system being modeled or queries over objects described in a model. Note that when the OCL expressions are evaluated, they do not have side effects; i.e., their evaluation cannot alter the state of the corresponding executing system. For the purpose of this MOF to RDF specification, references to OCL should be considered references to the Object Constraint Language Specification, cited in Normative References, above.

Ontology Definition Metamodel (ODM)
The Ontology Definition Metamodel (ODM) is a family of MOF metamodels, mappings between those metamodels as well as mappings to and from UML, and a set of profiles that enable ontology modeling through the use of UML-based

tools. The metamodels that comprise the ODM reflect the abstract syntax of several standard knowledge representation and conceptual modeling languages.

Resource Description Framework (RDF)
The Resource Description Framework (RDF) is a framework for representing information in the Web. RDF has an abstract syntax that reflects a simple graph-based data model, and formal semantics with a rigorously defined notion of entailment providing a basis for well founded deductions in RDF data. The vocabulary is fully extensible, being based on URIs with optional fragment identifiers (URI references, or URIrefs). For the purpose of this MOF to RDF specification, references to RDF should be considered references to the set of RDF recommendations available from the World Wide Web Consortium, and in particular, the RDF Concepts and Abstract Syntax recommendation, cited in Normative References, above.

RDF Schema (RDFS)
RDF's vocabulary description language, RDF Schema, is a semantic extension of RDF. It provides mechanisms for describing groups of related resources and the relationships between these resources. These resources are used to determine characteristics of other resources, such as the domains and ranges of properties. The RDF vocabulary description language class and property system is similar to the type systems of object-oriented programming languages such as Java. RDF differs from many such systems in that instead of defining a class in terms of the properties its instances may have, the RDF vocabulary description language describes properties in terms of the classes of resource to which they apply. For the purpose of this MOF to RDF specification, references to RDF Schema should be considered references to the set of RDF recommendations cited in Normative References, above.

RDF Vocabulary
An RDF Vocabulary comprises a set of resources that define classes and properties in terms of the RDF model. A RDF vocabulary is the language used to define a RDF model. A RDF vocabulary may be at any or multiple MOF "meta levels".

Semantic MOF (SMOF)
The "Semantic MOF", or "SMOF" specification adds flexibility to MOF such as are already found in the semantic web, such as the ability to have multiple classes for an element and that may that change over time.

Semantic Web
Semantic web is an overarching term for the set of RDF and OWL related standards from W3C. Cited in Normative References, above.

Web Ontology Language (OWL)
The OWL Web Ontology Language is designed for use by applications that need to process the content of information instead of just presenting information to humans. OWL can be used to explicitly represent the meaning of terms in vocabularies and the relationships between those terms. This representation of terms and their interrelationships is called an ontology. OWL has more facilities for expressing meaning and semantics than XML, RDF, and RDF-S, and thus OWL goes beyond these languages in its ability to represent machine interpretable content on the Web.

XML Metadata Interchange (XMI)
XMI is a widely used interchange format for sharing objects using XML. Sharing objects in XML is a comprehensive solution that build on sharing data with XML. XMI is applicable to a wide variety of objects: analysis (UML), software (Java, C++), components (EJB, IDL, CORBA Component Model), and databases (CWM).

5 Symbols

The following symbols and/or abbreviations are used throughout this specification.

None.

This page intentionally left blank

6 Additional Information

(informative)

6.1 How to read this Specification

This specification presents a method to transform MOF-compliant models into OWL2. While MOF shares its metamodel with the UML metamodel (since MOF/UML version 2.4), MOF compliance implies that the set of modeling constructs used is limited to the subset specified in the MOF specification. Clauses 1 to 6 provide compliance rules, terms definitions and reference information. Clause 7 brovides an introductory background on Linked Open Data, on the provided transformation method, and concludes with some representative use cases. Clause 8 introduces and describes the MOF Support Ontology, which bridges semantical differences between MOF and OWL and provides traceability. Clause 9 Specifies the transformation rules for models. Clause 10 adresses the handling of Profiles and Stereotypes. Annex A provides the complete MOF Support Ontology, Annex B introduces the associated machine-readable files, Annex C and Annex D provide examples for model and profile transformations using a test models developed by the OMG Model Interchange Working Group.

All clauses of this document are normative unless explicitly marked “(informative)”. The marking “(informative)” of a particular clause applies also to all contained sub-clauses of that clause.

6.2 Acknowledgments

The following organizations submitted this specification:

- Adaptive
- 88solutions
- Model-Driven Solutions
- Thematix
- International Business Machines

This page intentionally left blank

7 MOF to RDF Mapping and Linked Open Data

(informative)

7.1 Introduction

Considered as data, models can be queried, transformed, linked and managed. Considered as information, models can be verified and reasoned about in order to leverage data to achieve business intent. A change to one piece of model data can have profound effects on other systems and models - those effects can be managed and tracked by software. That software can "project" model data into viewpoints that are better suited to particular stakeholders. Additional value can be derived from models as data - such as a system implementation created from a model using MDA. Software can help to better understand what the models represent, to analyze the models, and how sets of models are the same, different or related. In summary - models as data are fundamental to being able to architect, create, communicate, manage and leverage complex systems and enterprises using software.

For models to be data there has to be some format, structure and semantic integrity to that model data - these data structures represent the vocabularies and semantics of modeling, things like classes, processes and business rules. The OMG has defined a way to represent these modeling language vocabularies called the "Meta Object Facility" (MOF). The MOF defines a "model of models" and an XML exchange format for those models called XMI. MOF and XMI are the language of modeling languages in OMG.

The W3C in the semantic web and linked open data communities has defined a general model for web data. This allows any kind of data to be represented on the web; used directly by web-savvy tools and then linked, repurposed and analyzed by software tools. This Linked Open Data web is growing in scope and capability - as more data is added, the more interesting it becomes. Since the data can be "mashed up" - each new piece of data added can be combined with and leverage other data - the value of this web of data is greater than the value of the sum of its parts. Linked open data is also very malleable - the same data can be repurposed or "classified" in many ways, depending on the needs and viewpoints of the user. Web data goes beyond web pages in that web data can be used and understood by software.

The goal is to have OMG based models as a part of this web of data - to be able to be viewed, analyzed, linked and re-purposed as web data. What this requires is the connection between the MOF and RDF standards, the purpose of this specification.

7.2 Problem Statement

Both OMG-MOF and W3C RDF support the ability to define the abstract syntax of modeling languages and provide for the capability to manage and serialize models expressed in those languages. The abstract syntax is the logical structure of a model's data. The serialization format is the way data can be stored or exchanged. These independently developed standards have both complementary and overlapping capabilities such that defining the relationship between them will provide value to the associated communities.

MOF and RDF are among the leading integrating frameworks. Integration of these frameworks provides for interoperability between the "MDA" and "Linked Open Data" communities, to the benefit of users and vendors alike.

Another specific benefit that can be achieved is to utilize the semantic web and associated semantic technologies to analyze, federate, compare, query and rationalize models expressed in MOF compliant languages and notations. This would be accomplished by mapping these models to RDF using the standards developed herein, and then using semantic tools and technologies to process resultant RDF derived from those models.

The Ontology Definition Meta Model (ODM) specification also serves to unify the MDA and Semantic web communities, but in a different way. ODM specifies ontology language metamodels in MOF and a partial semantic mapping of UML to OWL, whereas this RFP specifies how to map an arbitrary MOF model into an RDF vocabulary. ODM provides the ability to view RDF and OWL models using UML notation while this specification does the opposite by specifying a means of expressing any MOF metamodel in its equivalent RDF. These are symmetric and complementary approaches to integrating OMG and semantic web standards.

7.3 Nature of Mapping

This specification provides a "structural mapping" and a limited "semantic mapping" between MOF and RDF/OWL. A structural mapping preserves the original (MOF) model's vocabulary and semantics but expresses that vocabulary and semantics in another structure (RDF). This kind of structural mapping has no information loss and can be fully isomorphic (bi directional).

This specification also provides a "semantic mapping" to OWL for those MOF constructs that have an equivalent OWL representation. For other aspects, this specification documents constraints and restrictions that OWL implementations will be required to enforce.

The semantic mapping is at the MOF level and does not attempt to encompass semantics specific to individual metamodels such as UML or BPMN. Nor does it attempt an automated translation of metamodel constraints expressed in OCL.

7.4 Linked Open Data

This specification provides a mapping between MOF and RDF supporting "Linked Open Data" (LOD). While RDF is a standard of the 3C, LOD is a set of conventions for using the web and RDF to provide a set of benefits, as such there are no specific LOD standards, just guidelines summarized in [LOD] and more specifically in [RDF Recipes]. This section defines LOD for the purposes of this specification.

Linked Data

The Semantic Web isn't just about putting data on the web. It is about making links, so that a person or machine can explore the web of data. With linked data, when you have some of it, you can find other, related, data.

Like the web of hypertext, the web of data is constructed with documents on the web. However, unlike the web of hypertext, where links are relationships anchors in hypertext documents written in HTML, for data that links between arbitrary things described by RDF. The URIs identify any kind of object or concept. But for HTML or RDF, the same expectations apply to make the web grow:

- Use URIs as names for things
- Use HTTP URIs so that people can look up those names.
- When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL)
- Include links to other URIs, so that they can discover more things.

Applied to MOF models, the following principles apply:

- Each MOF model and identifiable model element represented in RDF will have a unique URI that names a resource logically represented in the RDF data model.
- The URI for the MOF model in RDF will use the HTTP protocol. When dereferenced, the URI for a model will access a web accessible resource.
- The URI for the model and model elements will, based on the content type, return the model or a query across that model in a standard RDF syntax. The query language(s) supported will include [SPARQL].
- Links to model elements within the model or in other models will use URIs as identified above. Links to non-model information may or may not return RDF.

The vision and intent behind these principles is that LOD represents a global data cloud where we can traverse from RDF data resource (models) to other RDF data resource via web URIs. This enables models that are mapped to RDF and exposed on web servers to be directly referenced, linked and queried using standard web tools.

While the core data format is RDF, it is the intent and expectation that web pages and other tools will provide user-friendly views of the RDF data. Content negotiation can be used to select the format and view of the data that is returned. Users will see web pages while programs can see data. This enables the data to be linked, queried, analyzed and repurposed in ways the original publisher of the information never intended.

7.5 Notional Architecture

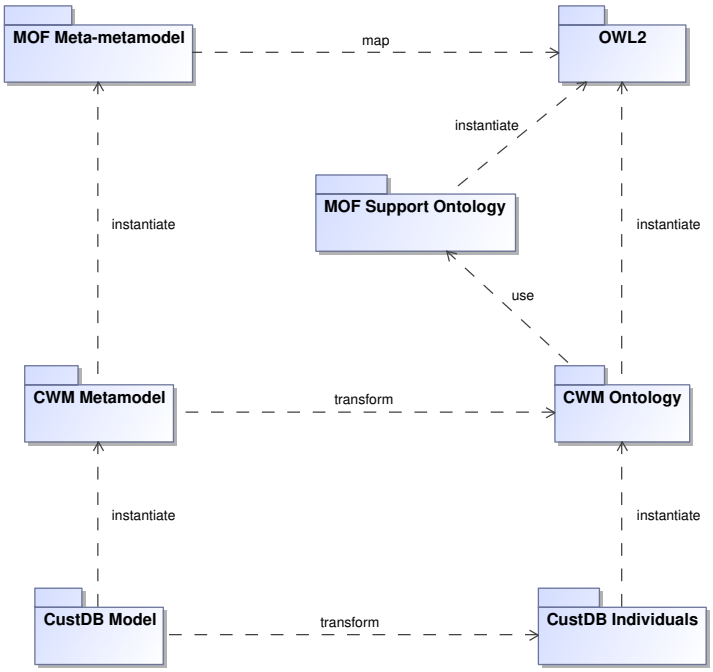


Figure 7.1: Notional Architecture

The generic diagram above defines the major components of this specification and the resulting capability. The component provided by this specification is the map and the transforms. A model represented in a MOF language, such as a "Cust DB" model represented in CWM, could be mapped to an RDF representation described by the CWM ontology. Likewise the CWM metamodel would be mapped to a CWM Ontology defined in terms of OWL2 supplemented by a MOFSupport ontology. The MOF metamodel would be mapped to OWL as described in this specification.

Note that the M0 or data level is not in scope of this specification, just as it is not for MOF specifications. However, depending on the nature of the metamodel it would be possible to use OWL2 punning to achieve that level of mapping. Specifically an Individual from the database model e.g. an Individual of class CWM/Relational/Table such as http://ex/CustDB/CustTable could also be declared to be of type owl:Class and further individuals created to represent real data such as Customer1, Customer2 etc.

This page intentionally left blank

8 MOF Support Ontology

8.1 Introduction

This section describes the isomorphic mapping from MOF to RDF, actually to OWL. The descriptions show the mapping from MOF to OWL; most, but not all, mappings are reversible, providing a possible path from OWL to MOF. The reason for "not all" is some discrepancies between the two modeling systems. For example, the concept of "abstractness" with all its implications is foreign to OWL and retained through annotations. However, since there is no force in OWL preventing direct instantiation of abstract elements, these instances have no legal place back in the MOF model.

The actual transformation technology underlying the descriptions in this clause is based on XSLT, reading the XMI file representing the MOF model and producing the equivalent OWL file in RDFXML syntax. For improved clarity, the results have then be converted into Turtle syntax. The descriptions in this clause adopt the graphical transformation notation from QVT.

8.2 MOF Support Ontology

The specification includes a normative OWL Ontology that is used to represent the MOF core concepts that are not directly representable in standard OWL.

There are actually surprisingly few of these and they include:

- isAbstract
- composition
- isOrdered and the sequence of links
- isUnique
- isDerived
- defaultValue
- redefinition

The MOF Support Ontology is defined as:

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix mof2rdf: <http://www.omg.org/spec/MOF2RDF/MOFSupport/> .

<http://www.omg.org/spec/MOF2RDF/MOFSupport/>
    a owl:Ontology ;
    rdfs:comment """Provides annotations for basic MOF concepts not directly
                 supported by OWL""" ;
    rdfs:label "MOF2RDF Support"@en .
```

MOF Support Ontology: Ontology definition

The following subclauses list and describe the OWL elements contained in the MOF Support Ontology and explain their respective role in bridging between established MOF concepts and their approximation in the OWL environment.

8.2.1 MOFSupport Class

```
<http://www.omg.org/spec/MOF2RDF/MOFSupport/Class>
    a owl:Class ;
    rdfs:comment """Classes instantiating MOF classes""" ;
    rdfs:label "class"@en .
```

MOF Support Ontology: Class

This is inherited by every OWL Class generated from a MOF Class and is used to trace that fact since there are many OWL Classes without that traceability.

8.2.2 MOFSupport Stereotype

```
<http://www.omg.org/spec/MOF2RDF/MOFSupport/Stereotype>
    a owl:Class ;
    rdfs:comment """A Stereotype extends MOF metaclasses""" ;
    rdfs:label "stereotype"@en .
```

MOF Support Ontology: Stereotype

This is inherited by any OWL Class generated from a MOF Stereotype indicating that the resulting owl:Class represents an extension of an existing metaclass.

8.2.3 MOFSupport DataType

```
<http://www.omg.org/spec/MOF2RDF/MOFSupport/DataType>
    a owl:Class ;
    rdfs:comment """Provides traceability for a MOF DataType represented
                 by an owl:Class""" ;
    rdfs:label "data type"@en .
```

MOF Support Ontology: DataType

Many forms of MOF DataType must be represented by an owl:Class in the OWL environment. Through inheritance from mof2rdf:DataType traceability to the DataType origin of the resulting OWL element is preserved.

8.2.4 MOFSupport Enumeration

```
<http://www.omg.org/spec/MOF2RDF/MOFSupport/Enumeration>
    a owl:Class ;
    rdfs:comment """Provides traceability for a MOF Enumeration represented
                 by an owl:Class""" ;
    rdfs:label "data type"@en .
```

MOF Support Ontology: Enumeration

MOF Enumerations must be represented by an owl:Class in the OWL environment. Through inheritance from mof2rdf:Enumeration traceability to the Enumeration origin of the resulting OWL element is preserved.

8.2.5 MOFSupport Association

```
<http://www.omg.org/spec/MOF2RDF/MOFSupport/Association>
    a owl:Class ;
    rdfs:comment """Classes instantiating associations i.e. specific Links
                 should inherit from this class""" ;
    rdfs:label "association"@en .
```

MOF Support Ontology: Association

Similar to MOF Class, this inheritance from mof2rdf:Association indicates that the resulting owl:Class represents in fact a MOF Association. Instances of that owl:Class will represent Links.

8.2.6 MOFSupport AssociationClass

```
<http://www.omg.org/spec/MOF2RDF/MOFSupport/AssociationClass>
    a owl:Class ;
    rdfs:subClassOf mof2rdf:Association, mof2rdf:Class ;
    rdfs:comment """Classes instantiating associations with own properties
                 (association classes) i.e. specific Links with property values,
                 should inherit from this class""" ;
    rdfs:label "association class"@en .
```

MOF Support Ontology: Association

Similar to MOF Association, this inheritance from mof2rdf:AssociationClass indicates that the resulting owl:Class represents in fact a MOF AssociationClass. Instances of that owl:Class will represent Links that carry additional properties resulting from MOF AssociationClass Attributes.

8.2.7 MOFSupport OrderedAssociation

```
<http://www.omg.org/spec/MOF2RDF/MOFSupport/OrderedAssociation>
    a owl:Class ;
    rdfs:comment """Classes instantiating ordered associations i.e. specific
                 Links should inherit from this class. At least one end of this
                 association must have isOrdered true.""" ;
    rdfs:label "ordered association"@en .
```

MOF Support Ontology: Ordered Association

This inheritance from mof2rdf:OrderedAssociation is equivalent to the inheritance from mof2rdf:Association, with the addition of signalling that the resulting Link instances are ordered with respect to one end. The ordering sequence is controlled through the additional linkSequence attribute.

8.2.8 MOFSupport linkSequence

```
<http://www.omg.org/spec/MOF2RDF/MOFSupport/linkSequence>
    a owl:AnnotationProperty ;
    rdfs:comment """For ordered associations the ordinal value of this link in
                 sequence. The values do not need to be contiguous - in fact
                 gaps are recommended to avoid frequent renumbering""" ;
    rdfs:domain <http://www.omg.org/spec/MOF2RDF/MOFSupport/OrderedAssociation> ;
    rdfs:label "link sequence"@en ;
    rdfs:range xsd:naturalInteger .
```

MOF Support Ontology: linkSequence annotation

This represents the sequence number of the current within a set of links. To allow for easier update it is not required that the linkSequence values are contiguous. Creating the links with gaps (e.g. 10, 20, 30) allows new links to be inserted without requiring updates to the succeeding links.

8.2.9 MOFSupport hasPart

```
<http://www.omg.org/spec/MOF2RDF/MOFSupport/hasPart>
    a owl:ObjectProperty ;
    rdfs:comment """Property representing composition/ownership. Actual
                 composition properties should inherit from this""" ;
    rdfs:label "has part"@en .
```

MOF Support Ontology: hasPart annotation

MOF Properties marked with isComposite = true (or derived from a Property marked this way) represent composition. Since OWL lacks the concept of composition, the resulting owl:ObjectProperties must inherit from the hasPart superProperty.

8.2.10 MOFSupport hasContainer

```
<http://www.omg.org/spec/MOF2RDF/MOFSupport/hasContainer>
    a owl:ObjectProperty ;
    rdfs:comment """Property representing container. MOF/UML rules state that
                 each object may have at most one.""" ;
    rdfs:label "has container"@en ;
    owl:inverseOf <http://www.omg.org/spec/MOF2RDF/MOFSupport/hasPart> .
```

MOF Support Ontology: hasContainer annotation

This is the inverse of hasPart. It is not generally explicitly asserted. It is restricted to a maxCardinality of 1.

8.2.11 MOFSupport isAbstract

```
<http://www.omg.org/spec/MOF2RDF/MOFSupport/isAbstract>
    a owl:AnnotationProperty ;
    rdfs:comment """Whether the class is permitted to have any direct
                 instances""" ;
    rdfs:domain rdfs:Class ;
    rdfs:label "is abstract"@en ;
    rdfs:range xsd:boolean .
```

MOF Support Ontology: isAbstract annotation

This captures the fact that the original MOF class may not be directly instantiated. This is hard to both represent and enforce in OWL, but several implementations have included directType and the following SPARQL can be used to determine the direct type of an individual:

```
SELECT * {
    <ind> rdf:type ?directType .
    FILTER NOT EXISTS {
      <ind> rdf:type ?type .
      ?type rdfs:subClassOf ?directType .
      FILTER NOT EXISTS {
         ?type owl:equivalentClass ?directType .
      }
    }
  }
```

SPARQL Query to determine the directType of an element

8.2.12 MOFSupport isOrdered

```
<http://www.omg.org/spec/MOF2RDF/MOFSupport/isOrdered>
    a owl:AnnotationProperty ;
    rdfs:comment """Whether the values of a property for a specific object
                 retain their sequence""" ;
    rdfs:domain rdfs:Property ;
    rdfs:label "is ordered"@en ;
    rdfs:range xsd:boolean .
```

MOF Support Ontology: isOrdered annotation

This AnnotationProperty retains the information that the values of the originating MOF Property have been ordered. Since OWL lacks the concept of ordering, direct enforcement of order-preservation is not possible, relying on the application using these data.

8.2.13 MOFSupport isUnique

```
<http://www.omg.org/spec/MOF2RDF/MOFSupport/isUnique>
    a owl:AnnotationProperty ;
    rdfs:comment """Whether the values of a property for a specific object
                 can contain the same value more than once""" ;
    rdfs:domain rdfs:Property ;
    rdfs:label "is unique"@en ;
    rdfs:range xsd:boolean .
```

MOF Support Ontology: isUnique annotation

This represents the fact that the values of the original MOF property were unique.

9 MOF to RDF Structural Mapping

9.1 Introduction

This section describes the isomorphic mapping from MOF to RDF, actually to OWL. The descriptions show the mapping from MOF to OWL; most, but not all, mappings are reversible, providing a possible path from OWL to MOF. The reason for "not all" are some discrepancies between the two modeling systems. For example, the concept of "abstractness" with all its implications is foreign to OWL and can only be retained through annotations. However, since there is no force in OWL preventing direct instantiation of abstract elements, these instances have no legal place back in the MOF model.

The transformation results defined in the following sub-clauses are based on these MOF elements:

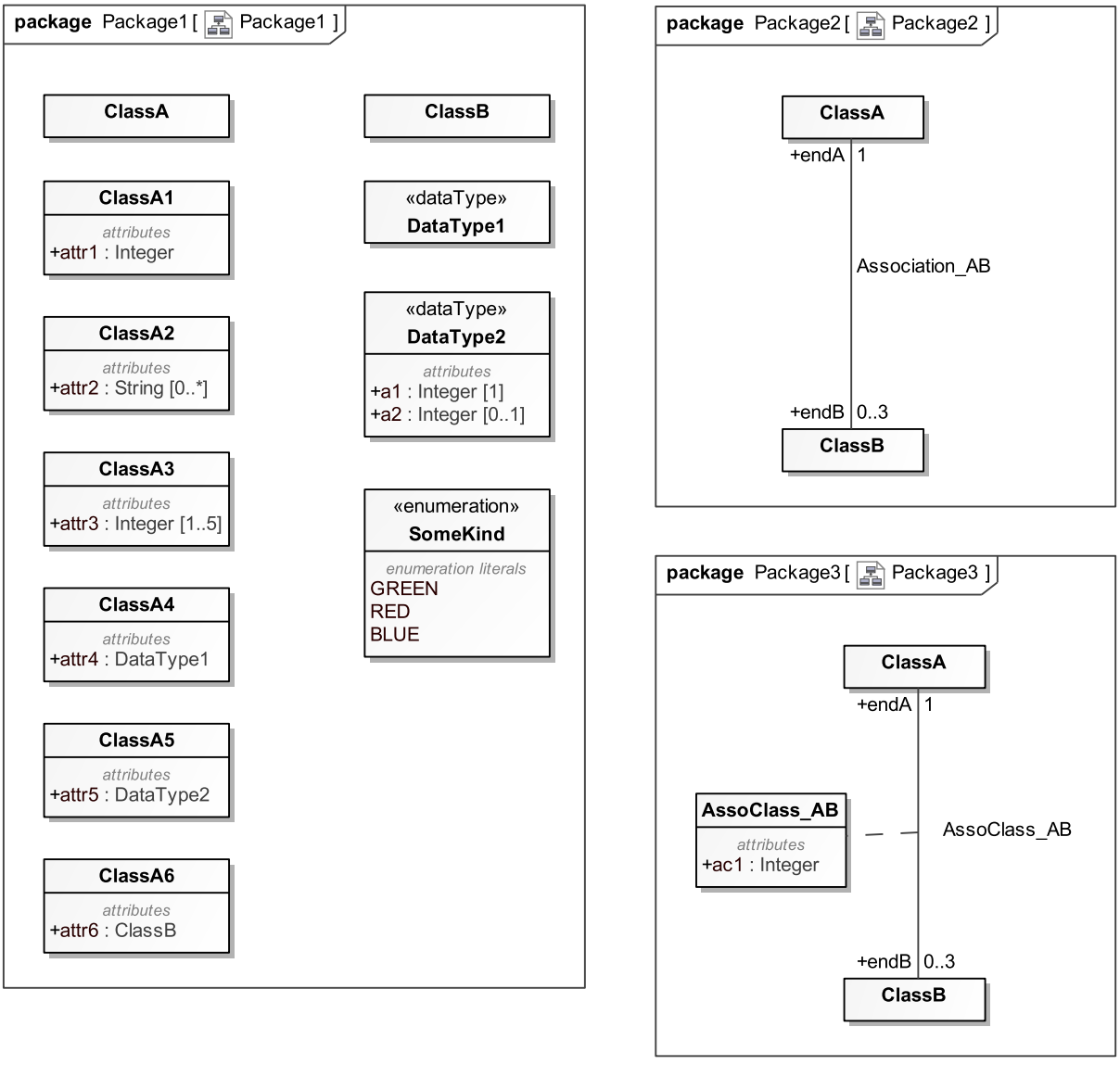


Figure 9.1: MOF elements used as input for transformation

The actual transformation technology underlying the descriptions in this clause is based on XSLT, reading the XMI

file representing the MOF model and producing the equivalent OWL file in RDFXML syntax. For improved clarity, the results have then be converted into Turtle syntax. The descriptions in this clause adopt the graphical transformation notation from QVT.

9.2 Mappings

The following clauses define mapping details between MOF (and therefor also UML) elements and OWL-2. The Turtle Syntax is used for presentation of the OWL-2 code. All these OWL-2 listings imply the following prefixes (also in Turtle Syntax):

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xs: <http://www.w3.org/2001/XMLSchema> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix mof2rdf: <http://www.omg.org/spec/MOF2RDF/MOFSupport/> .
```

9.2.1 MOF Package

Each MOF Package is mapped to an owl:Ontology. Nested Packages will result in nested Ontologies.


```
<http://example.com/Package1/>
    a owl:Ontology ;
    rdfs:label "Package1" ;
    owl:imports <http://www.omg.org/spec/MOF2RDF/MOFSupport/> .
```

Mapping of a MOF Package to an OWL Ontology

Each MOF Package is mapped to owl:Ontology. It is mandatory that the outer-most Package in a Package hierarchy has its Package URI set to a legal URI value. That Package URI will become the IRI of the resulting owl:Ontology, and the last element of this IRI wil become the Ontology name, which might be different from the MOF Package name. If the Package is a nested sub-Package, and it has no Package::URI set, then the MOF Package name will be appended to the parent Ontology IRI separated by a slash ("/"). If the nested sub-Package has its URI set, then that URI will become the IRI of the corresponding owl:Ontology, and the last element of this IRI becomes the ontology name. In this case it is the MOF modeler's responsibility to ensure that the resulting IRIs are unique.

It is recommended that only the top-Package carries an explicit URI (or that all URIs reflect the Package hierarchy). This ensures that the model structure is preserved, and in particular, that elements residing in the MOF model on different Package level, but with identical names, are correctly disambiguated in the Ontology model.

Every owl:Ontology produced through this transformation automatically imports the MOF Support Ontology with IRI: http://www.omg.org/spec/MOF2RDF/MOFSupport.

9.2.2 MOF DataType

In the MOF modeling environment, data types may range from primitive and unstructured data types to complex structured data types which may have attributes and even operations. In contrast, the OWL environment is very restrictive regarding data types, they may only be unstructured literal-kind types. Therefore the transformation distinguishes between strictly unstructured (MOF) data types that can be mapped to OWL DataTypes, and mapps any other (MOF) data types to OWL Classes.

Unstructured DataType

Unstructured MOF DataType maps to owl:DataType.

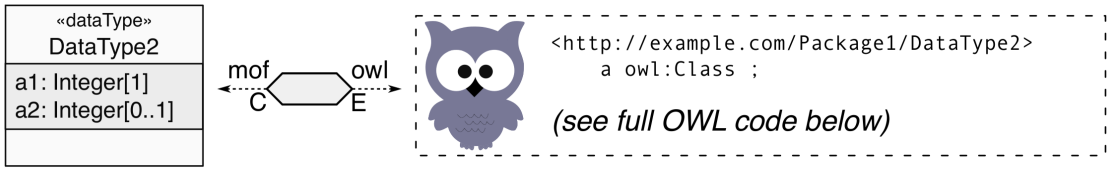

```
<http://example.com/Package1/DataType1>
    a rdfs:DataType ;
    rdfs:isDefinedBy [
        rdfs:resource "http://example.com/Package1/"
    ] ;
    rdfs:label "DataType1" .
```

Mapping of an Unstructured DataType

Mapping of MOF DataTypes to owl:DataType is restricted to unstructured and flat data type, which have data literals as their instances. This includes MOF PrimitiveTypes.

Structured DataType

Any MOF DataType that cannot be instantiated by a flat data literal must be mapped to an owl:Class. To retain traceability to the MOF DataType origin of the resulting owl:Class, the resulting owl:Class representing the MOF DataType must become a subclass of mof2rdf:DataType contained in the MOF Support Ontology.

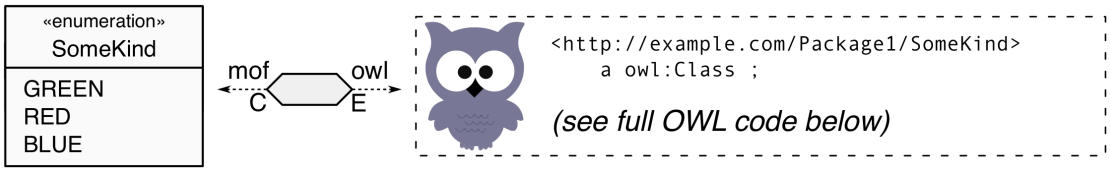



```
<http://example.com/Package1/DataType2>
    a owl:Class ;
    rdfs:isDefinedBy [
        rdfs:resource "http://example.com/Package1/"
    ] ;
    rdfs:label "DataType2" ;
    rdfs:subClassOf mof2rdf:DataType, [
        a owl:Restriction ;
        owl:allValuesFrom xsd:integer ;
        owl:onProperty <http://example.com/Package1/DataType2.a1>
    ], [
        a owl:Restriction ;
        owl:onDataRange xsd:integer ;
        owl:onProperty <http://example.com/Package1/DataType2.a1> ;
        owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger
    ], [
        a owl:Restriction ;
        owl:allValuesFrom xsd:integer ;
        owl:onProperty <http://example.com/Package1/DataType2.a2>
    ], [
        a owl:Restriction ;
        owl:onDataRange xsd:integer ;
        owl:onProperty <http://example.com/Package1/DataType2.a2> ;
        owl:maxQualifiedCardinality "1"^^xsd:nonNegativeInteger
    ] .

<http://example.com/Package1/DataType2.a1>
    a owl:DatatypeProperty ;
    rdfs:domain <http://example.com/Package1/DataType2> ;
    rdfs:isDefinedBy [
        rdfs:resource "http://example.com/Package1/"
    ] ;
    rdfs:label "a1" ;
    rdfs:range xsd:integer .

<http://example.com/Package1/DataType2.a2>
    a owl:DatatypeProperty ;
    rdfs:domain <http://example.com/Package1/DataType2> ;
    rdfs:isDefinedBy [
        rdfs:resource "http://example.com/Package1/"
    ] ;
    rdfs:label "a2" ;
    rdfs:range xsd:integer .
```

Mapping of a Structured DataType

9.2.3 MOF Enumeration

MOF Enumerations map to owl:Class.




```
<http://example.com/Package1/SomeKind>
    a owl:Class ;
    rdfs:isDefinedBy [
        rdfs:resource "http://example.com/Package1/"
    ] ;
    rdfs:label "SomeKind" ;
    rdfs:subClassOf mof2rdf:Enumeration ;
    owl:equivalentClass [
        a owl:Class ;
        owl:oneOf (<http://example.com/Package1/SomeKind.GREEN>
            <http://example.com/Package1/SomeKind.RED>
            <http://example.com/Package1/SomeKind.BLUE>
        )
    ] .

<http://example.com/Package1/SomeKind.BLUE>
    a <http://example.com/Package1/SomeKind>, owl:NamedIndividual ;
    rdfs:isDefinedBy [
        rdfs:resource "http://example.com/Package1/"
    ] ;
    rdfs:label "BLUE" .

<http://example.com/Package1/SomeKind.GREEN>
    a <http://example.com/Package1/SomeKind>, owl:NamedIndividual ;
    rdfs:isDefinedBy [
        rdfs:resource "http://example.com/Package1/"
    ] ;
    rdfs:label "GREEN" .

<http://example.com/Package1/SomeKind.RED>
    a <http://example.com/Package1/SomeKind>, owl:NamedIndividual ;
    rdfs:isDefinedBy [
        rdfs:resource "http://example.com/Package1/"
    ] ;
    rdfs:label "RED" .

[]
    a owl:AllDifferent ;
    owl:distinctMembers (<http://example.com/Package1/SomeKind.GREEN>
        <http://example.com/Package1/SomeKind.RED>
        <http://example.com/Package1/SomeKind.BLUE>
    ) .
```

Mapping of Enumeration types

MOF Enumerations map to owl:Class, which are constrained datatypes allowing only a closed set of named individuals as its instances. These named individuals correspond to the MOF EnumerationLiterals and are listed in the oneOf selection construct. All named individuals corresponding to the enumeration literals must be listed in an owl:AllDifferent statement (lines 32 to 41). This statement may be global to the ontology file and may contain more entries, unrelated to this enumeration type.

9.2.4 MOF Class

MOF Class map to owl:Class.


```
<http://example.com/Package1/ClassA>
    a owl:Class ;
    rdfs:isDefinedBy [
        rdfs:resource "http://example.com/Package1/"
    ] ;
    rdfs:label "ClassA" ;
    rdfs:subClassOf mof2rdf:Class .
```

Mapping of Class

MOF Class maps to owl:Class. The mapping is straight-forward one-to-one. Each owl:Class produced by this transformation becomes also a subclass of mof2rdf:Class, which is contained in the MOFSupport ontology.

9.2.5 MOF Property (Attribute)

In a MOF model, Properties representing Attributes (in contrast to AssociationEnds) make no difference if the type of that Attribute denotes a flat literal value or an element with structure. In OWL however, these possibilities are distinct and result in different representations. Also, different Multiplicities on the MOF side result in much more pronounced structural differences on the OWL side than just a different numeric cardinality value.

The following six mapping variations show the effect of MOF Property type and Multiplicity on the resulting OWL mapping. For emphasis, the Attribute compartment in the Class symbol shown in these mapping variations is shaded gray. These are the variations:

attr1 Property typed as literal Integer, with default multiplicity.

attr2 Optional Property typed as an unbounded set of String literals.

attr3 Property typed as a set of literal Integer, with explicit multiplicity range.

attr4 Property typed by a flat user-defined DataType, with default multiplicity.

attr5 Property typed by a structured user-defined DataType, with default multiplicity.

attr6 Property typed by a Class, with default multiplicity.

Properties typed by flat Literal Types

MOF Properties typed by a predefined flat literal type, or typed by a flat, unstructured user-defined DataType map to owl:DatatypeProperty.


```
1  <http://example.com/Package1/ClassA1>
2      a owl:Class ;
3      rdfs:isDefinedBy [
4          rdfs:resource "http://example.com/Package1/"
5      ] ;
6      rdfs:label "ClassA1" ;
7      rdfs:subClassOf mof2rdf:Class, [
8          a owl:Restriction ;
9          owl:allValuesFrom xsd:integer ;
10         owl:onProperty <http://example.com/Package1/ClassA1.attr1>
11     ], [
12         a owl:Restriction ;
13         owl:onDataRange xsd:integer ;
14         owl:onProperty <http://example.com/Package1/ClassA1.attr1> ;
15         owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger
16     ] .
17
18 <http://example.com/Package1/ClassA1.attr1>
19     a owl:DatatypeProperty ;
20     rdfs:domain <http://example.com/Package1/ClassA1> ;
21     rdfs:isDefinedBy [
22         rdfs:resource "http://example.com/Package1/"
23     ] ;
24     rdfs:label "attr1" ;
25     rdfs:range xsd:integer .
```

Mapping of a Property (Attribute) typed by a Literal Type, with default Multiplicity

MOF Properties representing Attributes that are typed by a MOF defined flat literal type (or MOF PrimitiveType), like for example Integer or String, are mapped to owl:DatatypeProperties in the OWL environment.

Variations in Multiplicities have no significant influence on the structure of a MOF model, however, variations of the MOF Multiplicity specification may cause pronounced structure variations in the OWL environment.

The Multiplicity of the MOF Property attr1 is not explicitly specified, therefore it defaults to 1. This translates into the requirement that attr1 refers to exact one literal integer value, which is expressed by two Restrictions in the resulting OWL code of the related owl:Class. The owl:allValuesFrom in line 9 is a universal quantifier requiring that all values of attr1 must be literal integer, and the second restriction limits at line 15 the number of values to exactly one.

The resulting definition of the Property ClassA.attr1 itself starts at line 18 of the listing. Since it is typed by a literal type on the MOF side, it results in a owl:DatatypeProperty, as shown in line 19. On the MOF side, attr1 is owned by ClassA and has an Integer value, this translates to the rdfs:domain of attr1 being ClassA (line 20), and the rdfs:range being xsd:integer (line 25), relating integer literals to individuals of ClassA.

In contrast to the tight relationship between namespace and ownership in MOF models, OWL has only a global names-

pace, and any (weak) ownership concept needs to be defined explicitly. This is accomplished by the rdfs:isDefinedBy constructs in lines 3-5 and 21-23 of the listing for the case of ClassA.attr1.


```
1  <http://example.com/Package1/ClassA2>
2      a owl:Class ;
3      rdfs:isDefinedBy [
4          rdfs:resource "http://example.com/Package1/"
5      ] ;
6      rdfs:label "ClassA2" ;
7      rdfs:subClassOf mof2rdf:Class, [
8          a owl:Restriction ;
9          owl:allValuesFrom xsd:string ;
10         owl:onProperty <http://example.com/Package1/ClassA2.attr2>
11     ] .
12
13 <http://example.com/Package1/ClassA2.attr2>
14     a owl:DatatypeProperty ;
15     rdfs:domain <http://example.com/Package1/ClassA2> ;
16     rdfs:isDefinedBy [
17         rdfs:resource "http://example.com/Package1/"
18     ] ;
19     rdfs:label "attr2" ;
20     rdfs:range xsd:string .
```

Mapping of an Optional Property (Attribute) typed as an unbounded set of String Literals

In this case, attr2 defines an optional MOF Property, which is also unlimitedly multi-valued. However, each value is required to be a (fixed) string literal, therefore mapping of attr2 to owl:DatatypeProperty is possible. On MOF side the Multiplicity of the Property is [0..*] for this case, which translates to a single restriction in the resulting OWL for the related class, at line 9 in the OWL code listing. The universal quantifier owl:allValuesFrom just requires that all values must be string literals, however no restriction on minimum or maximum cardinality are specified.

The resulting definition of the Property ClassA.attr2 itself starts at line 13 of the listing. Since it is typed by a literal type on the MOF side, it results in a owl:DatatypeProperty, as shown in line 14. On the MOF side, attr2 is owned by ClassA and has an optional, but multivalued, String value, this translates to the rdfs:domain of attr2 being ClassA (line 15), and the rdfs:range being xsd:string (line 20), relating a, possibly empty, set of string literals to individuals of ClassA.

In contrast to the tight relationship between namespace and ownership in MOF models, OWL has only a global namespace, and any (weak) ownership concept needs to be defined explicitly. This is accomplished by the rdfs:isDefinedBy constructs in lines 3-5 and 16-18 of the listing for the case of ClassA.attr2.


```
1  <http://example.com/Package1/ClassA3>
2      a owl:Class ;
3      rdfs:isDefinedBy [
4          rdfs:resource "http://example.com/Package1/"
5      ] ;
6      rdfs:label "ClassA3" ;
7      rdfs:subClassOf mof2rdf:Class, [
8          a owl:Restriction ;
9          owl:allValuesFrom xsd:integer ;
10         owl:onProperty <http://example.com/Package1/ClassA3.attr3>
11     ], [
12         a owl:Restriction ;
13         owl:onDataRange xsd:integer ;
14         owl:onProperty <http://example.com/Package1/ClassA3.attr3>
15         owl:minQualifiedCardinality "1"^^xsd:nonNegativeInteger
16     ], [
17         a owl:Restriction ;
18         owl:onDataRange xsd:integer ;
19         owl:onProperty <http://example.com/Package1/ClassA3.attr3> ;
20         owl:maxQualifiedCardinality "5"^^xsd:nonNegativeInteger
21     ] .
22
23 <http://example.com/Package1/ClassA3.attr3>
24     a owl:DatatypeProperty ;
25     rdfs:domain <http://example.com/Package1/ClassA3> ;
26     rdfs:isDefinedBy [
27         rdfs:resource "http://example.com/Package1/"
28     ] ;
29     rdfs:label "attr3" ;
30     rdfs:range xsd:integer .
```

Mapping of a Property (Attribute) typed by a Literal Type, with explicit Multiplicity range

This case is similar to the case of attr1. MOF Property attr3 is typed as a multivalued Integer, with at least one, and up to five literal values. This translates into three restrictions for the related owl:Class, ClassA. The first restriction expresses the the requirement that attr3 refers to literal integer values, by the universal quantifier owl:allValuesFrom at line 9. The second restriction represents the lower bound of the MOF Multiplicity through owl:minQualifiedCardinality at line 15, implying that at least one literal value must be present. The third restriction sets the upper bound of the MOF Multiplicity through owl:maxQualifiedCardinality at line 20.

The resulting definition of the Property ClassA.attr3 itself starts at line 23 of the listing. Since it is typed by a literal type on the MOF side, it results in a owl:DatatypeProperty, as shown in line 24. On the MOF side, attr3 is owned by ClassA and has an Integer value, this translates to the rdfs:domain of attr3 being ClassA (line 25), and the rdfs:range being xsd:integer (line 30), relating integer literals to individuals of ClassA.

In contrast to the tight relationship between namespace and ownership in MOF models, OWL has only a global namespace, and any (weak) ownership concept needs to be defined explicitly. This is accomplished by the rdfs:isDefinedBy constructs in lines 3-5 and 26-28 of the listing for the case of ClassA.attr3.


```
1  <http://example.com/Package1/ClassA4>
2      a owl:Class ;
3      rdfs:isDefinedBy [
4          rdfs:resource "http://example.com/Package1/"
5      ] ;
6      rdfs:label "ClassA4" ;
7      rdfs:subClassOf mof2rdf:Class, [
8          a owl:Restriction ;
9          owl:allValuesFrom <http://example.com/Package1/DataType1> ;
10         owl:onProperty <http://example.com/Package1/ClassA4.attr4>
11     ], [
12         a owl:Restriction ;
13         owl:onDataRange <http://example.com/Package1/DataType1> ;
14         owl:onProperty <http://example.com/Package1/ClassA4.attr4> ;
15         owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger
16     ] .
17
18 <http://example.com/Package1/ClassA4.attr4>
19     a owl:DatatypeProperty ;
20     rdfs:domain <http://example.com/Package1/ClassA4> ;
21     rdfs:isDefinedBy [
22         rdfs:resource "http://example.com/Package1/"
23     ] ;
24     rdfs:label "attr4" ;
25     rdfs:range <http://example.com/Package1/DataType1> .
```

Mapping of a Property (Attribute) typed by an unstructured User-Defined Type, with default Multiplicity

The case of ClassA.attr4 is identical to the case of ClassA.attr1, except that attr4 is typed by a flat and unstructured, user-defined literal type named DataType1. The transformation results again in two restrictions added to ClassA.

As with the case of attr1, the type of the property is enforced through the universal quantifier owl:allValuesFrom, requiring DataType1 (line 9). The (default) Multiplicity of 1 is enforced through the owl:qualifiedCardinality with value 1 in the second restriction (line14).

The resulting definition of the Property ClassA.attr4 itself starts at line 18 of the listing. Since it is typed by an unstructured type on the MOF side, owl:DatatypeProperty can be used on the OWL side, as shown in line 19. On the MOF side, attr4 is owned by ClassA and has a DataType1 value, this translates to the rdfs:domain of attr4 being ClassA (line 20), and the rdfs:range being DataType1 (line 25), relating DataType1 literals to individuals of ClassA.

DataType1 is a user-defined data type. It is mandatory, that DataType1 is unstructured, otherwise a mapping to owl:Class is necessary (see below). Also, DataType1 is not a member of the “datatype map” as defined in the OWL Structural Specification. While this should be no problem, this might trigger complaints from reasoners.

In contrast to the tight relationship between namespace and ownership in MOF models, OWL has only a global namespace, and any (weak) ownership concept needs to be defined explicitly. This is accomplished by the rdfs:isDefinedBy constructs in lines 3-5 and 21-23 of the listing for the case of ClassA.attr4.

Properties typed by Structured or Class Types

Any structured MOF DataType maps to owl:Class.


```
1  <http://example.com/Package1/ClassA5>
2      a owl:Class ;
3      rdfs:isDefinedBy [
4          rdfs:resource "http://example.com/Package1/"
5      ] ;
6      rdfs:label "ClassA5" ;
7      rdfs:subClassOf mof2rdf:Class, [
8          a owl:Restriction ;
9          owl:allValuesFrom <http://example.com/Package1/DataType2> ;
10         owl:onProperty <http://example.com/Package1/ClassA5.attr5>
11     ], [
12         a owl:Restriction ;
13         owl:onClass <http://example.com/Package1/DataType2> ;
14         owl:onProperty <http://example.com/Package1/ClassA5.attr5> ;
15         owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger
16     ] .
17
18 <http://example.com/Package1/ClassA5.attr5>
19     a owl:ObjectProperty ;
20     rdfs:domain <http://example.com/Package1/ClassA5> ;
21     rdfs:isDefinedBy [
22         rdfs:resource "http://example.com/Package1/"
23     ] ;
24     rdfs:label "attr5" ;
25     rdfs:range <http://example.com/Package1/DataType2> .
```

Mapping of a Property (Attribute) typed by a structured User-Defined Type, with default Multiplicity

OWL is very restrictive regarding data types: owl:DatatypeProperty can *only* be used to refer to flat, unstructured data types that result in plain, constant literals when instantiated. Anything else must be represented as owl:Class and consequently referred to using owl:ObjectProperty. By this the original intention of the MOF Property is lost, however through inheritance from mof2rdf:DataType, some traceability to the DataType origin is preserved.

Besides the definition of DataType2 as owl:Class in this case, and definition of the referring Property ClassA.attr5 as owl:ObjectProperty, there is no difference to the case of the reference to the unstructured DataType1 in ClassA.attr4.

ClassA receives two restrictions for property attr5, the first requires DataType2 through the universal quantifier owl:allValuesFrom (line 9), the second restriction enforces Multiplicity 1 through owl:qualifiedCardinality (line 14).

The resulting definition of the Property ClassA.attr5 itself starts at line 18 of the listing. Since it is typed by a structured type on the MOF side, owl:ObjectProperty must be used on the OWL side, as shown in line 19. On the MOF side, attr5 is owned by ClassA and has a DataType2 value, this translates to the rdfs:domain of attr5 being ClassA (line 20), and the rdfs:range being DataType2 (line 25), relating DataType2 literals to individuals of ClassA.

In contrast to the tight relationship between namespace and ownership in MOF models, OWL has only a global namespace, and any (weak) ownership concept needs to be defined explicitly. This is accomplished by the rdfs:isDefinedBy constructs in lines 3-5 and 21-23 of the listing for the case of ClassA.attr5.


```
1  <http://example.com/Package1/ClassA6>
2      a owl:Class ;
3      rdfs:isDefinedBy [
4          rdfs:resource "http://example.com/Package1/"
5      ] ;
6      rdfs:label "ClassA6" ;
7      rdfs:subClassOf mof2rdf:Class, [
8          a owl:Restriction ;
9          owl:allValuesFrom <http://example.com/Package1/ClassB> ;
10         owl:onProperty <http://example.com/Package1/ClassA6.attr6>
11     ], [
12         a owl:Restriction ;
13         owl:onClass <http://example.com/Package1/ClassB> ;
14         owl:onProperty <http://example.com/Package1/ClassA6.attr6> ;
15         owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger
16     ] .
17
18 <http://example.com/Package1/ClassA6.attr6>
19     a owl:ObjectProperty ;
20     rdfs:domain <http://example.com/Package1/ClassA6> ;
21     rdfs:isDefinedBy [
22         rdfs:resource "http://example.com/Package1/"
23     ] ;
24     rdfs:label "attr6" ;
25     rdfs:range <http://example.com/Package1/ClassB> .
```

Mapping of a Property (Attribute) typed by a Class, with default Multiplicity

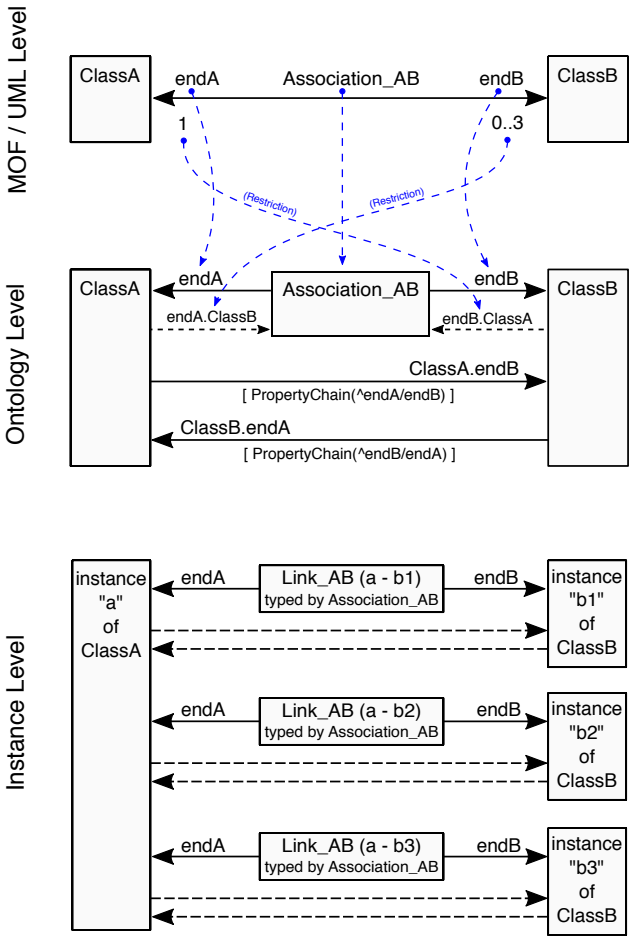
This final case handles the situation where a MOF Property (Attribute) is typed by a user-defined MOF Class. It is completely identical to the case of DataType2 typed Properties in the previous section of ClassA.attr5. Just substitute “DataType2” by “ClassB”.

9.2.6 MOF Association

The most substantial difference between the MOF (and UML) modeling methodology and the OWL modeling methodology is the way relationships, and their associated semantics, are expressed. OWL relationships are modeled using owl:ObjectProperty, resulting in a simple, directional pointer from the subject to the object. This pointer may be *named* to express a certain semantic meaning, but its type will always be owl:ObjectProperty. In contrast, a MOF Association is a classifier, and therefore providing a distinct type for the instantiating Links. MOF Associations are binary, on model-level expressed as a relationship between two MOF Classes. UML supports n-ary Associations, where a single

Association can defines a relationship between three or more UML Classes[1]. By default, MOF Associations define bi-directional relationships, but can be specified as uni-directional. This is called "navigability" but expresses a deeper semantic meaning for each direction than just navigation capabilities. Each MOF Association musst be named, this name becomes the name of the type typing the instantiating Links. Each Association end is a Property, named by the end name adornment of the Association line in the MOF Class diagram. Each Association end has also an Multiplicity adornment, which in combination specifies how many Class Instances may participate and how many Links are established between these Class Instances, and what pattern these Links form, when the Association is instantiated. Lower bounds of these Multiplicities greater then zero *guarantee* that the corresponding Links must be established at instantiation time. This is all very different form OWL relationships expressed by owl:ObjectProperty. Following the Open World Assumption, no guarantee is given that the relationship is ever instantiated, and if, how many instances of the owl:ObjectProperty may be created, unless there are specific restrictions placed on the target Class definition.

Figure 9.2 shows the MOF Association "Association_AB" between the two MOF Classes "ClasA" and "ClassB". The Association end at ClassA is named "endA" and has a Multiplicity of one (1), the end at ClassB is called "endB" and has a Multiplicity range from zero (0) to three (3). When instantiated in the MOF model, there may be no Link if no ClassB instance is participating, as a consequence of the zero Multiplicity lower bound of endB. There may be at most three (3) instances of ClassB participating in a relationship with one (1) ClassA instance.



Note: Client uses Property Chains for navigation (shown dashed at Instance Level) but Property Chains are inferred, not materialized.

Figure 9.2: MOF Association Transformation Scheme

The resulting Links form a fan pattern, with one Link reaching from the single ClassA instance to each of the ClassB instances.

The "Ontology Level" section of Figure 9.2 shows the resulting scheme on the OWL side after transforming the binary MOF Association shown in the "MOF/UML Level" from MOF to OWL. The goal of the selected mapping scheme (shown by the blue dashed arrows) was to preserve the MOF Association semantics while at the same time reduce the number of required triples at OWL/RDF instantiation time to a minimum. The "Instance Level" portion of Figure 9.2 shows the necessary instances to represent the MOF Association shown in the top section of Figure 9.2.

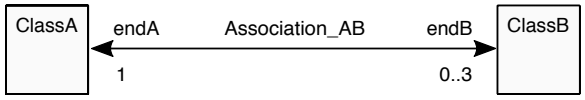
The MOF Association maps to a corresponding owl:Class, which has two properties "endA" and "endB" pointing to the participating owl:Classes "ClassA" and "ClassB". To provide the user with the illusion of a direct relationship and direct navigability, owl:PropertyChains are defined between the participating classes. See the OWL code discussion below.

Since owl:PropertyChain does not allow any restrictions in the end classes, the owl:Class representing Association_AB defines two additional properties: "Association_AB.endA.ClassB" and "Association_AB.endB.ClassA", shown with dashed arrows in the "Ontology Level" section. These properties allow the insertion of the necessary restrictions to express the relationship multiplicities in the participating end classes.

At instantiation time, instances of the "Association_AB.endA.ClassB"-style properties are used to select the right Link (Association_AB instance), then the selected Link points to the two ends, which are instances of ClassA and ClassB.

[1] MOF at this time does not support n-ary Associations.

The following OWL code listings provide additional detail on the mapping of MOF Association into the OWL environment (repeating the MOF diagram in the first listing for clarity).




```
<http://example.com/Package2/Association_AB>
    a owl:Class ;
    rdfs:isDefinedBy [
        rdfs:resource "http://example.com/Package2/"
    ] ;
    rdfs:label "Association_AB" ;
    rdfs:subClassOf mof2rdf:Association, [
        a owl:Restriction ;
        owl:onClass <http://example.com/Package2/ClassB> ;
        owl:onProperty <http://example.com/Package2/Association_AB.endB> ;
        owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger
    ], [
        a owl:Restriction ;
        owl:onClass <http://example.com/Package2/ClassA> ;
        owl:onProperty <http://example.com/Package2/Association_AB.endA> ;
        owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger
    ] .

<http://example.com/Package2/Association_AB.endA>
    a owl:ObjectProperty ;
    rdfs:domain <http://example.com/Package2/Association_AB> ;
    rdfs:label "endA" ;
    rdfs:range <http://example.com/Package2/ClassA> .

<http://example.com/Package2/Association_AB.endA.ClassB>
    a owl:ObjectProperty ;
    rdfs:domain <http://example.com/Package2/ClassA> ;
    rdfs:label "endA link" ;
    rdfs:range <http://example.com/Package2/Association_AB> .

<http://example.com/Package2/Association_AB.endB>
    a owl:ObjectProperty ;
    rdfs:domain <http://example.com/Package2/Association_AB> ;
    rdfs:label "endB" ;
    rdfs:range <http://example.com/Package2/ClassB> .

<http://example.com/Package2/Association_AB.endB.ClassA>
    a owl:ObjectProperty ;
    rdfs:domain <http://example.com/Package2/ClassB> ;
    rdfs:label "endB link" ;
    rdfs:range <http://example.com/Package2/Association_AB> .
```

Mapping of MOF Association to OWL - Part 1: Association_AB

The MOF Association, named “Association_AB” in this case, mapps directly to an owl:Class, retaining the name “Association_AB” (line 6). This class has two properties, here named “endA” and “endB”, which point to the two classes participating in the relationship defined by the MOF Association. These two properties have a cardinality of at-most one (1). This ensures that each materialized instance of Association_AB creates referes to exactly one

instance of ClassA and ClassB, each. The corresponding restrictions are at lines 8-11 and 13-16. As usual, the owl:Class representing the Association is defined within the enclosing ontology corresponding to the original MOF Package (lines 3-5).

The two properties Association_AB.endA (lines 19-23) and Association_AB.endB (lines 31-35) are identical in their design. Both are of type owl:ObjectProperty (lines 20 and 32) and have owl:Class Association_AB as their domain (lines 21 and 33). Property Association_AB.endA points to ClassA, therefore has ClassA as its range (line 23), while property Association_AB.endB points to ClassB and therefore has ClassB as its range (line 35).

The remaining two properties Association_AB.endA.ClassB (lines 19-23) and Association_AB.endB.ClassA (lines 37-41) are also identical in their design. Their purpose is to carry any restrictions in the end classes and to point to the correct Link individuals when instantiated. Therefore they have the *opposite* end class as domain (lines 27 and 39), and the Association_AB as range (lines 29 and 41). More on this in the discussion of the end classes ClassA and ClassB.

```
1  <http://example.com/Package2/ClassA>
2      a owl:Class ;
3      rdfs:isDefinedBy [
4          rdfs:resource "http://example.com/Package2/"
5      ] ;
6      rdfs:label "ClassA" ;
7      rdfs:subClassOf mof2rdf:Class, [
8          a owl:Restriction ;
9          owl:allValuesFrom <http://example.com/Package2/Association_AB> ;
10         owl:onProperty <http://example.com/Package2/Association_AB.endB.ClassA>
11     ], [
12         a owl:Restriction ;
13         owl:maxQualifiedCardinality "3"^^xsd:nonNegativeInteger ;
14         owl:onClass <http://example.com/Package2/Association_AB> ;
15         owl:onProperty <http://example.com/Package2/Association_AB.endB.ClassA>
16     ] .
17
18 <http://example.com/Package2/ClassA.endB>
19     a owl:ObjectProperty ;
20     rdfs:domain <http://example.com/Package2/ClassA> ;
21     rdfs:isDefinedBy [
22         rdfs:resource "http://example.com/Package2/"
23     ] ;
24     rdfs:label "endB" ;
25     rdfs:range <http://example.com/Package2/ClassB> ;
26     owl:inverseOf <http://example.com/Package2/ClassB.endA> ;
27     owl:propertyChainAxiom ([
28             owl:inverseOf <http://example.com/Package2/Association_AB.endA>
29         ]
30         <http://example.com/Package2/Association_AB.endB>
31     ) .
```

Mapping of MOF Association to OWL - Part 2: ClassA

For its participation in the relationship representing the MOF Association_AB between MOF ClassA and MOF ClassB, the owl:Class “ClassA” is extended by at least two restrictions and one property. The two restrictions are both related to property Association_AB.endB.ClassA (defined by Association_AB). The first restriction uses the universal quantifier owl:allValuesFrom to ensure the property points to instances of Association_AB (lines 9 and 10). The second restriction defines the property cardinality according to the corresponding MOF Association end multiplicity. Since

the MOF Multiplicity range is [0..3], only an owl:maxQualifiedCardinality value of three (3) is defined, and no lower bound.

The Property “ClassA.endB” defines the user-visible direct relationship to the opposite end of the transformed MOF Association. Therefore it has its own class, ClassA, as domain (line 20) and the opposite class, ClassB as range (line 25). The actual reference to the opposite class is constructed as PropertyChain composed from the two properties Association_AB.endA and Association_AB.endB (lines 27-30), by taking the inverse of Association_AB.endA (line 28), chained by the forward of Association_AB.endB (line 30).

```
1  <http://example.com/Package2/ClassB>
2      a owl:Class ;
3      rdfs:isDefinedBy [
4          rdfs:resource "http://example.com/Package2/"
5      ] ;
6      rdfs:label "ClassB" ;
7      rdfs:subClassOf mof2rdf:Class, [
8          a owl:Restriction ;
9          owl:allValuesFrom <http://example.com/Package2/Association_AB> ;
10         owl:onProperty <http://example.com/Package2/Association_AB.endA.ClassB>
11     ], [
12         a owl:Restriction ;
13         owl:onProperty <http://example.com/Package2/Association_AB.endA.ClassB> ;
14         owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
15         owl:someValuesFrom <http://example.com/Package2/Association_AB>
16     ] .
17
18 <http://example.com/Package2/ClassB.endA>
19     a owl:ObjectProperty ;
20     rdfs:domain <http://example.com/Package2/ClassB> ;
21     rdfs:isDefinedBy [
22         rdfs:resource "http://example.com/Package2/"
23     ] ;
24     rdfs:label "endA" ;
25     rdfs:range <http://example.com/Package2/ClassA> ;
26     owl:propertyChainAxiom ([
27             owl:inverseOf <http://example.com/Package2/Association_AB.endB>
28         ]
29         <http://example.com/Package2/Association_AB.endA>
30     ) .
```

Mapping of MOF Association to OWL - Part 3: ClassB

For its participation in the relationship representing the MOF Association_AB between MOF ClassA and MOF ClassB, the owl:Class “ClassB” is extended by at least two restrictions and one property. The two restrictions are both related to property Association_AB.endA.ClassB (defined by Association_AB). The first restriction uses the universal quantifier owl:allValuesFrom to ensure the property points to instances of Association_AB (lines 9 and 10). The second restriction defines the property cardinality according to the corresponding MOF Association end multiplicity. Since the MOF Multiplicity is exactly 1, a combination of owl:qualifiedCardinality with value of 1 (line 14) and of the existential quantifier owl:someValuesFrom (line 15) is defined, for an exact cardinality of 1.

The Property “ClassB.endA” defines the user-visible direct relationship to the opposite end of the transformed MOF Association. Therefore it has its own class, ClassB, as domain (line 20) and the opposite class, ClassA as range (line 25). The actual reference to the opposite class is constructed as PropertyChain composed from the two properties Association_AB.endA and Association_AB.endB (lines 27-30), by taking the inverse of Association_AB.endA (line

27), chained by the forward of Association_AB.endB (line 29).

Ordered Association

Ordered Associations in a MOF model are Associations where at least one end is marked {ordered}. In this case the resulting owl:Class representing the Association caries an AnnotationProperty mof2rdf:linkSequence. Links instantiating the Association will then carry integer-valued annotations determining the Link sequence. Creating these sequence numbers and enforcing the ordering relies on the application creating and managing the Link instances. The owl:Class representing the ordered Association inherits from mofowl:OrderedAssociation.

```
1  <http://example.com/Package2/Association_AB>
2      a owl:Class ;
3      rdfs:isDefinedBy [
4          rdfs:resource "http://example.com/Package2/"
5      ] ;
6      rdfs:label "Association_AB" ;
7      rdfs:subClassOf mof2rdf:OrderedAssociation, [
8          a owl:Restriction ;
9          owl:onClass <http://example.com/Package2/ClassB> ;
10         owl:onProperty <http://example.com/Package2/Association_AB.endB> ;
11         owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger
12     ], [
13         a owl:Restriction ;
14         owl:onClass <http://example.com/Package2/ClassA> ;
15         owl:onProperty <http://example.com/Package2/Association_AB.endA> ;
16         owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger
17     ] .
18
19 <http://example.com/Package2/Association_AB.linkSequence>
20     a owl:AnnotationProperty ;
21     rdfs:domain <http://example.com/Package2/Association_AB> ;
22     rdfs:label "Association_AB link sequence" ;
23     rdfs:range xsd:naturalInteger .
```

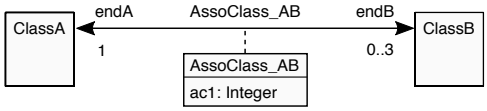
Mapping of an Ordered Association (only association definition shown)

9.2.7 MOF AssociationClass

In a MOF (and UML) model, an AssociationClass is a declaration of an Association that has a set of features of its own. An AssociationClass can therefore be considered both an Association and a Class, and preserves the static and dynamic semantics of both, however with some constraints: The name of the AssociationClass is the name of the Association; the names for AssociationEnds and AssociationClass Attributes must be distinct; AssociationEnds are semantically distinct from AssociationClass Attributes and not interchangeable.

AssocaitionClasses instantiate as Links like regular Associations, where each Link holds specific instance values for the AssociationClass Attributes.

AssociationClasses transform from MOF to OWL identical to regular Associations (see Clause 9.2.6), with the addition of Restrictions for the AssociationClass Attribute properties (Part 1, lines 18 to 26 below) in the the owl:Class representing the AssociationClass, and property definitions for the AssociationClass Attribute properties (Part 2 below). These additions are equivalent to MOF Attribute trasformations (see Clause 9.2.5).




```
<http://example.com/Package3/AssoClass_AB>
    a owl:Class ;
    rdfs:isDefinedBy [
        rdfs:resource "http://example.com/Package3/"
    ] ;
    rdfs:label "AssoClass_AB" ;
    rdfs:subClassOf mof2rdf:Association, [
        a owl:Restriction ;
        owl:onClass <http://example.com/Package3/ClassB> ;
        owl:onProperty <http://example.com/Package3/AssoClass_AB.endB> ;
        owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger
    ], [
        a owl:Restriction ;
        owl:onClass <http://example.com/Package3/ClassA> ;
        owl:onProperty <http://example.com/Package3/AssoClass_AB.endA> ;
        owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger
    ], [
        a owl:Restriction ;
        owl:allValuesFrom xsd:integer ;
        owl:onProperty <http://example.com/Package3/AssoClass_AB.ac1>
    ], [
        a owl:Restriction ;
        owl:onDataRange xsd:integer ;
        owl:onProperty <http://example.com/Package3/AssoClass_AB.ac1> ;
        owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger
    ] .

<http://example.com/Package3/AssoClass_AB.endA>
    a owl:ObjectProperty ;
    rdfs:domain <http://example.com/Package3/AssoClass_AB> ;
    rdfs:label "endA" ;
    rdfs:range <http://example.com/Package3/ClassA> .

<http://example.com/Package3/AssoClass_AB.endA.ClassB>
    a owl:ObjectProperty ;
    rdfs:domain <http://example.com/Package3/ClassA> ;
    rdfs:label "endA link" ;
    rdfs:range <http://example.com/Package3/AssoClass_AB> .

<http://example.com/Package3/AssoClass_AB.endB>
    a owl:ObjectProperty ;
    rdfs:domain <http://example.com/Package3/AssoClass_AB> ;
    rdfs:label "endB" ;
    rdfs:range <http://example.com/Package3/ClassB> .

<http://example.com/Package3/AssoClass_AB.endB.ClassA>
    a owl:ObjectProperty ;
    rdfs:domain <http://example.com/Package3/ClassB> ;
    rdfs:label "endB link" ;
    rdfs:range <http://example.com/Package3/AssoClass_AB> .
```

Mapping of MOF AssociationClass to OWL - Part 1: AssoClass_AB

```
<http://example.com/Package3/AssoClass_AB.ac1>
    a owl:DatatypeProperty ;
    rdfs:domain <http://example.com/Package3/AssoClass_AB> ;
    rdfs:isDefinedBy [
        rdfs:resource "http://example.com/Package3/"
    ] ;
    rdfs:label "ac1" ;
    rdfs:range xsd:integer .
```

Mapping of MOF AssociationClass to OWL - Part 2: AssociationClass Attribute

```
<http://example.com/Package3/ClassA>
    a owl:Class ;
    rdfs:isDefinedBy [
        rdfs:resource "http://example.com/Package3/"
    ] ;
    rdfs:label "ClassA" ;
    rdfs:subClassOf mof2rdf:Class, [
        a owl:Restriction ;
        owl:allValuesFrom <http://example.com/Package3/AssoClass_AB> ;
        owl:onProperty <http://example.com/Package3/AssoClass_AB.endB.ClassA>
    ], [
        a owl:Restriction ;
        owl:maxQualifiedCardinality "3"^^xsd:nonNegativeInteger ;
        owl:onClass <http://example.com/Package3/AssoClass_AB> ;
        owl:onProperty <http://example.com/Package3/AssoClass_AB.endB.ClassA>
    ] .

<http://example.com/Package3/ClassA.endB>
    a owl:ObjectProperty ;
    rdfs:domain <http://example.com/Package3/ClassA> ;
    rdfs:isDefinedBy [
        rdfs:resource "http://example.com/Package3/"
    ] ;
    rdfs:label "endB" ;
    rdfs:range <http://example.com/Package3/ClassB> ;
    owl:inverseOf <http://example.com/Package3/ClassB.endA> ;
    owl:propertyChainAxiom ([
            owl:inverseOf <http://example.com/Package3/AssoClass_AB.endA>
        ]
        <http://example.com/Package3/AssoClass_AB.endB>
    ) .
```

Mapping of MOF AssociationClass to OWL - Part 3: ClassA

```
<http://example.com/Package3/ClassB>
    a owl:Class ;
    rdfs:isDefinedBy [
        rdfs:resource "http://example.com/Package3/"
    ] ;
    rdfs:label "ClassB" ;
    rdfs:subClassOf mof2rdf:Class, [
        a owl:Restriction ;
        owl:allValuesFrom <http://example.com/Package3/AssoClass_AB> ;
        owl:onProperty <http://example.com/Package3/AssoClass_AB.endA.ClassB>
    ], [
        a owl:Restriction ;
        owl:onProperty <http://example.com/Package3/AssoClass_AB.endA.ClassB> ;
        owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
        owl:someValuesFrom <http://example.com/Package3/AssoClass_AB>
    ] .

<http://example.com/Package3/ClassB.endA>
    a owl:ObjectProperty ;
    rdfs:domain <http://example.com/Package3/ClassB> ;
    rdfs:isDefinedBy [
        rdfs:resource "http://example.com/Package3/"
    ] ;
    rdfs:label "endA" ;
    rdfs:range <http://example.com/Package3/ClassA> ;
    owl:propertyChainAxiom ([
            owl:inverseOf <http://example.com/Package3/AssoClass_AB.endB>
        ]
        <http://example.com/Package3/AssoClass_AB.endA>
    ) .
```

Mapping of MOF AssociationClass to OWL - Part 4: ClassB

10 MOF Profile Structural Mapping

10.1 Introduction

A MOF or UML Profile is a means to extend the MOF or UML metamodel. It is not a first-class extension mechanism since it does not allow to augment the original metamodel with *brand new* metaclasses. Profiles are restricted to *extend existing* metaclasses by *adding* new meaning, features, semantics or constraints. Since this is an additive extension, Profiles *cannot remove* existing features, semantics or constraints from those metaclasses extended by the Profile. The resulting extended metaclass is called Stereotype.

This clause describes the isomorphic mapping from MOF or UML Profiles and Stereotypes to the equivalent OWL constructs. Figure 10.1 shows the MOF or UML metamodel extension constructs for which mapping instructions are shown in the following sub-clauses.

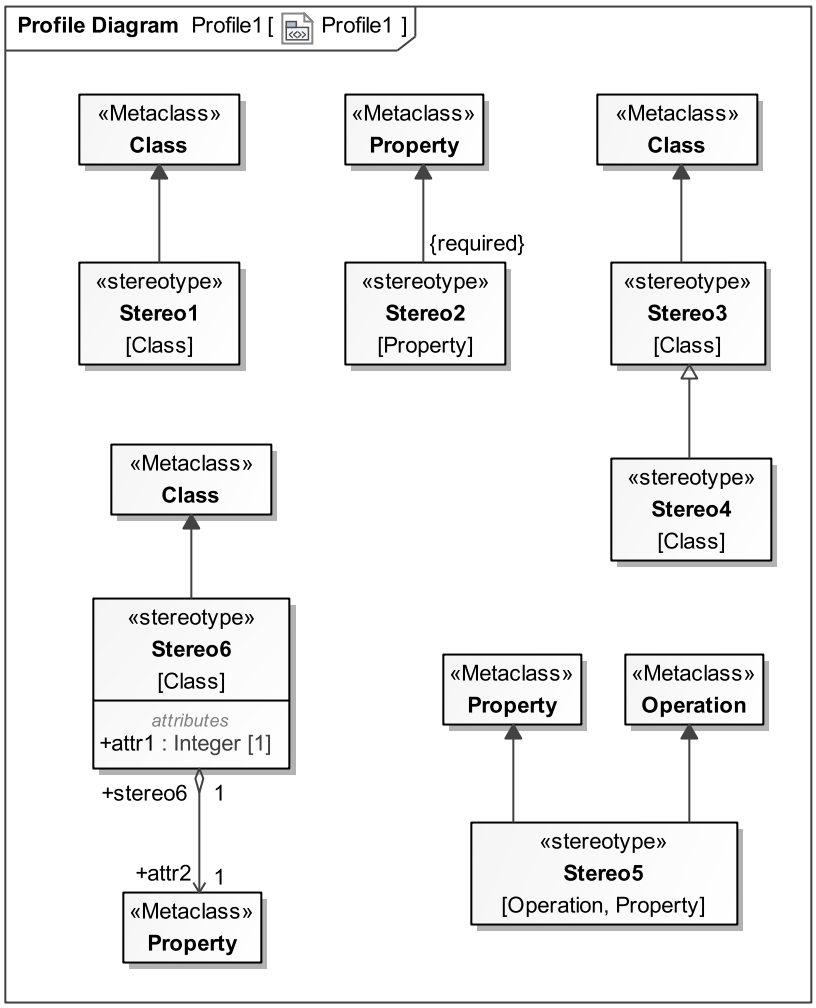


Figure 10.1: MOF Stereotypes in Profile1

10.2 Mappings

The following clauses define mapping details between MOF (and therefor also UML) Profile elements and OWL-2. The Turtle Syntax is used for presentation of the OWL-2 code. All these OWL-2 listings imply the following prefixes (also in Turtle Syntax):

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xs: <http://www.w3.org/2001/XMLSchema> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix mof2rdf: <http://www.omg.org/spec/MOF2RDF/MOFSupport/> .
```

10.2.1 MOF Profile

MOF Profile is mapped to an owl:Ontology. Nested Profiles will result in nested Ontologies.


```
<http://exampe.com/Profile1/>
    a owl:Ontology ;
    rdfs:isDefinedBy [
        rdfs:resource "http://exampe.com/Profile1/"
    ] ;
    rdfs:label "Profile1" ;
    owl:imports <http://www.omg.org/spec/MOF2RDF/MOFSupport/>,
                <http://www.omg.org/spec/UML/20131001/UML/> .
```

Mapping of a MOF Profile to an OWL Ontology

MOF Profile is an extensions of MOF Package, therefore each MOF Profile is mapped to owl:Ontology in the same way as MOF Packages. It is mandatory that the outer-most Profile in a Profile hierarchy has its Package URI set to a legal URI value. This applies also to any Packages or Package hierarchies within a Profile. That PrPackage URI will become the IRI of the resulting owl:Ontology, and the last element of this IRI wil become the Ontology name, which might be different from the MOF Package name. If the Package is a nested sub-Package, and it has no Package::URI set, then the MOF Package name will be appended to the parent Ontology IRI separated by a slash (“/”). If the nested sub-Package has its URI set, then that URI will become the IRI of the corresponding owl:Ontology, and the last element of this IRI becomes the ontology name. In this case it is the MOF modeler’s responsibility to ensure that the resulting IRIs are unique.

It is recommended that only the top-Profile carries an explicit URI (or that all URIs reflect the Profile/Package hierarchy). This ensures that the model structure is preserved, and in particular, that elements residing in the MOF model on different Package level, but with identical names, are correctly disambiguated in the Ontology model.

Every owl:Ontology produced through this transformation automatically imports the MOF Support Ontology with IRI: http://www.omg.org/spec/MOF2RDF/MOFSupport.

10.2.2 MOF Stereotype

In the MOF and UML modeling environment, Stereotypes define new metamodel elements, which are then available to the modeler like any other elements of that extended modeling language. This means, the definition of a Stereotype is situated one metalevel above the modeling level, even though Profile diagrams containing Stereotype definitions may appear in many modeling tools as if they were on model level. OWL does not provide a concept of metalevels, the transformation result is an owl:Class that adds the Stereotype-specific classification to the equivalent of the extended MOF metaclass.

Regular Stereotype

MOF Stereotype maps to an owl:Class, which adds additional classification(s) to the transformed equivalence of the extended MOF metaclass.


```
<http://exampe.com/Profile1/Stereo1>
    a owl:Class ;
    rdfs:isDefinedBy [
        rdfs:resource "http://exampe.com/Profile1/"
    ] ;
    rdfs:label "Stereo1" ;
    rdfs:subClassOf mof2rdf:Stereotype, [
        a owl:Restriction ;
        owl:onProperty rdf:type ;
        owl:someValuesFrom <http://www.omg.org/spec/UML/20131001/UML/Class>
    ] .
```

Mapping of a regular Stereotype to OWL

Regular Stereotypes may be applied at-will in the MOF environment. Therefore, the equivalent owl:Class may also used at will to add the additional classification to the original Class corresponding to the extended MOF metaclass. Both, the original Class and the extended Class may coexist.

Required Stereotype


```
<http://exampe.com/Profile1/Stereo2>
    a owl:Class ;
    rdfs:isDefinedBy [
        rdfs:resource "http://exampe.com/Profile1/"
    ] ;
    rdfs:label "Stereo2" ;
    rdfs:subClassOf mof2rdf:Stereotype, [
        a owl:Restriction ;
        owl:onProperty rdf:type ;
        owl:someValuesFrom <http://www.omg.org/spec/UML/20131001/UML/Property>
    ] .

<http://www.omg.org/spec/UML/20131001/UML/Property>
    rdfs:subClassOf [
        a owl:Restriction ;
        owl:onProperty rdf:type ;
        owl:someValuesFrom <http://exampe.com/Profile1/Stereo2>
    ] .
```

Mapping of a required Stereotype to OWL

If the {required} keyword is present on the Stereotype extension relationship in the MOF model, then the Stereotype must always be applied, masking the extended original metaclass completely. After transformation, only the owl:Class representing the Stereotype shall exist and shall mask the equivalent of the extended metaclass off the model.

The mandatory nature of the Stereotype is reflected in the OWL environment by a Restriction placed on the extended metaclass.

10.2.3 Stereotype Generalization

Stereotypes may use Generalization like regular classes.


```
<http://exampe.com/Profile1/Stereo3>
    a owl:Class ;
    rdfs:isDefinedBy [
        rdfs:resource "http://exampe.com/Profile1/"
    ] ;
    rdfs:label "Stereo3" ;
    rdfs:subClassOf mof2rdf:Stereotype, [
        a owl:Restriction ;
        owl:onProperty rdf:type ;
        owl:someValuesFrom <http://www.omg.org/spec/UML/20131001/UML/Class>
    ] .

<http://exampe.com/Profile1/Stereo4>
    a owl:Class ;
    rdfs:isDefinedBy [
        rdfs:resource "http://exampe.com/Profile1/"
    ] ;
    rdfs:label "Stereo4" ;
    rdfs:subClassOf <http://exampe.com/Profile1/Stereo3>, mof2rdf:Stereotype .
```

Mapping of Stereotype Generalization

Here Stereotype “Stereo4” inherits from Stereotype “Stereo3”, which includes the extended classification of the underlying original metaclass.

10.2.4 Stereotype extending multiple Metaclasses

A single Stereotype may extend multiple metaclasses concurrently.


```
<http://exampe.com/Profile1/Stereo5>
    a owl:Class ;
    rdfs:isDefinedBy [
        rdfs:resource "http://exampe.com/Profile1/"
    ] ;
    rdfs:label "Stereo5" ;
    rdfs:subClassOf mof2rdf:Stereotype, [
        a owl:Restriction ;
        owl:onProperty rdf:type ;
        owl:someValuesFrom [
            a owl:Class ;
            owl:unionOf (<http://www.omg.org/spec/UML/20131001/UML/Property>
                <http://www.omg.org/spec/UML/20131001/UML/Operation>
            )
        ]
    ] .
```

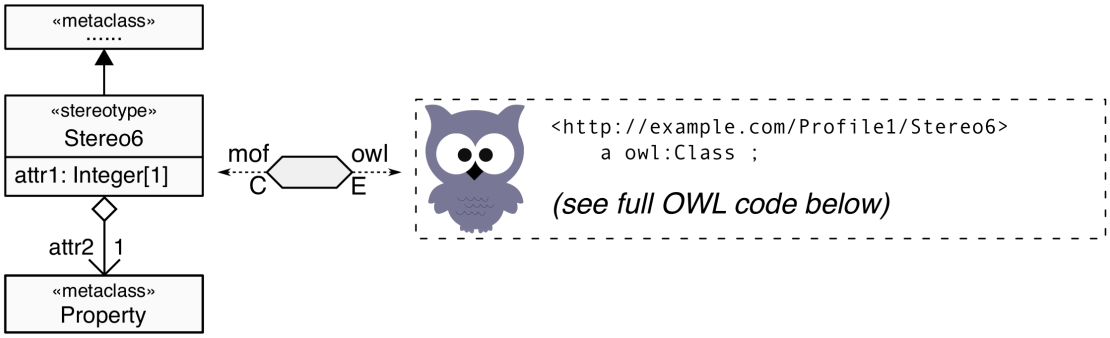
Mapping of a Stereotype extending multiple MOF Metaclasses to OWL

In this case, a single Stereotype reclassifies a combination of metaclasses. Particular attention should be given to the semantic difference of inheritance in the MOF/UML world (and most object-oriented systems), and in the OWL world (and most logic-based and functional systems): In MOF, UML and classic oo, a subclass inherits the *union* of features from all superclasses of a multiple inheritance. A single Stereotype extending multiple MOF metaclasses follows the same principle. In OWL and most logic systems, multiple inheritance results in the *intersection* of all inherited features.

The owl:Class resulting from a transformed Stereotype that extends multiple metaclasses must therefore apply its reclassification explicitly to the union of extended metaclasses.

10.2.5 Stereotypes with Metaattributes

Stereotypes define new elements of a modeling language. Therefore their definition lives one metalevel above the modeling level. Regular attributes present in a Stereotype *definition* result in value slots in the new modeling element resulting from Stereotyp application. If however the desired effect is the creation of an Attribute definiton in the model element to which the Stereotype is applied (and which will result in a value slot when that stereotyped element is instantiated), then the Stereotype definition must provide a reference to the metaclass Property. This reference is transformed identical to a regular Association (see also Clause 9.2.6).




```
<http://exampe.com/Profile1/A_stereo6_attr2>
    a owl:Class ;
    rdfs:isDefinedBy [
        rdfs:resource "http://exampe.com/Profile1/"
    ] ;
    rdfs:label "A_stereo6_attr2" ;
    rdfs:subClassOf mof2rdf:Association, [
        a owl:Restriction ;
        owl:onClass <http://exampe.com/Profile1/Stereo6> ;
        owl:onProperty <http://exampe.com/Profile1/A_stereo6_attr2.stereo6> ;
        owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger
    ], [
        a owl:Restriction ;
        owl:onClass <http://www.omg.org/spec/UML/20131001/UML/Property> ;
        owl:onProperty <http://exampe.com/Profile1/A_stereo6_attr2.attr2> ;
        owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger
    ] .

<http://exampe.com/Profile1/A_stereo6_attr2.attr2>
    a owl:ObjectProperty ;
    rdfs:domain <http://exampe.com/Profile1/A_stereo6_attr2> ;
    rdfs:label "attr2" ;
    rdfs:range <http://www.omg.org/spec/UML/20131001/UML/Property> .

<http://exampe.com/Profile1/A_stereo6_attr2.attr2.Stereo6>
    a owl:ObjectProperty ;
    rdfs:domain <http://www.omg.org/spec/UML/20131001/UML/Property> ;
    rdfs:label "attr2 link" ;
    rdfs:range <http://exampe.com/Profile1/A_stereo6_attr2> .

<http://exampe.com/Profile1/A_stereo6_attr2.stereo6>
    a owl:ObjectProperty ;
    rdfs:domain <http://exampe.com/Profile1/A_stereo6_attr2> ;
    rdfs:label "stereo6" ;
    rdfs:range <http://exampe.com/Profile1/Stereo6> .

<http://exampe.com/Profile1/A_stereo6_attr2.stereo6.Property>
    a owl:ObjectProperty ;
    rdfs:domain <http://exampe.com/Profile1/Stereo6> ;
    rdfs:label "stereo6 link" ;
    rdfs:range <http://exampe.com/Profile1/A_stereo6_attr2> .
```

Mapping of a MOF Stereotype with regular Attribute and Metaattribute - Part 1

```
<http://exampe.com/Profile1/Stereo6>
    a owl:Class ;
    rdfs:isDefinedBy [
        rdfs:resource "http://exampe.com/Profile1/"
    ] ;
    rdfs:label "Stereo6" ;
    rdfs:subClassOf mof2rdf:Stereotype, [
        a owl:Restriction ;
        owl:onProperty rdf:type ;
        owl:someValuesFrom <http://www.omg.org/spec/UML/20131001/UML/Class>
    ], [
        a owl:Restriction ;
        owl:allValuesFrom xsd:integer ;
        owl:onProperty <http://exampe.com/Profile1/Stereo6.attr1>
    ], [
        a owl:Restriction ;
        owl:onDataRange xsd:integer ;
        owl:onProperty <http://exampe.com/Profile1/Stereo6.attr1> ;
        owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger
    ], [
        a owl:Restriction ;
        owl:allValuesFrom <http://exampe.com/Profile1/A_stereo6_attr2> ;
        owl:onProperty <http://exampe.com/Profile1/A_stereo6_attr2.attr2.Stereo6>
    ], [
        a owl:Restriction ;
        owl:onProperty <http://exampe.com/Profile1/A_stereo6_attr2.attr2.Stereo6> ;
        owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
        owl:someValuesFrom <http://exampe.com/Profile1/A_stereo6_attr2>
    ] .

<http://exampe.com/Profile1/Stereo6.attr1>
    a owl:DatatypeProperty ;
    rdfs:domain <http://exampe.com/Profile1/Stereo6> ;
    rdfs:isDefinedBy [
        rdfs:resource "http://exampe.com/Profile1/"
    ] ;
    rdfs:label "attr1" ;
    rdfs:range xsd:integer .

<http://exampe.com/Profile1/Stereo6.attr2>
    a owl:ObjectProperty ;
    rdfs:domain <http://exampe.com/Profile1/Stereo6> ;
    rdfs:isDefinedBy [
        rdfs:resource "http://exampe.com/Profile1/"
    ] ;
    rdfs:label "attr2" ;
    rdfs:range <http://www.omg.org/spec/UML/20131001/UML/Property> ;
    owl:propertyChainAxiom ([
            owl:inverseOf <http://exampe.com/Profile1/A_stereo6_attr2.stereo6>
        ]
        <http://exampe.com/Profile1/A_stereo6_attr2.attr2>
    ) .
```

Mapping of a MOF Stereotype with regular Attribute and Metaattribute - Part 2

Appendix A MOF Support Ontology

(normative)

The complete MOF Support Ontology in Turtle Syntax.

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix mof2rdf: <http://www.omg.org/spec/MOF2RDF/MOFSupport/> .

<http://www.omg.org/spec/MOF2RDF/MOFSupport/>
    a owl:Ontology ;
    rdfs:comment """Provides annotations for basic MOF concepts not directly
                 supported by OWL""" ;
    rdfs:label "MOF2RDF Support"@en .

<http://www.omg.org/spec/MOF2RDF/MOFSupport/Class>
    a owl:Class ;
    rdfs:comment """Classes instantiating MOF classes""" ;
    rdfs:label "class"@en .

<http://www.omg.org/spec/MOF2RDF/MOFSupport/Stereotype>
    a owl:Class ;
    rdfs:comment """A Stereotype extends MOF metaclasses""" ;
    rdfs:label "stereotype"@en .

<http://www.omg.org/spec/MOF2RDF/MOFSupport/DataType>
    a owl:Class ;
    rdfs:comment """Provides traceability for a MOF DataType represented
                 by an owl:Class""" ;
    rdfs:label "data type"@en .

<http://www.omg.org/spec/MOF2RDF/MOFSupport/Enumeration>
    a owl:Class ;
    rdfs:comment """Provides traceability for a MOF Enumeration represented
                 by an owl:Class""" ;
    rdfs:label "data type"@en .

<http://www.omg.org/spec/MOF2RDF/MOFSupport/Association>
    a owl:Class ;
    rdfs:comment """Classes instantiating associations i.e. specific Links
                 should inherit from this class""" ;
    rdfs:label "association"@en .

<http://www.omg.org/spec/MOF2RDF/MOFSupport/AssociationClass>
    a owl:Class ;
    rdfs:subClassOf mof2rdf:Association, mof2rdf:Class ;
    rdfs:comment """Classes instantiating associations with own properties
                 (association classes) i.e. specific Links with property values,
                 should inherit from this class""" ;
    rdfs:label "association"@en .

<http://www.omg.org/spec/MOF2RDF/MOFSupport/OrderedAssociation>
    a owl:Class ;
    rdfs:comment """Classes instantiating ordered associations i.e. specific
                 Links should inherit from this class. At least one end of this
                 association must have isOrdered true.""" ;
    rdfs:label "ordered association"@en .

<http://www.omg.org/spec/MOF2RDF/MOFSupport/linkSequence>
    a owl:AnnotationProperty ;
    rdfs:comment """For ordered associations the ordinal value of this link in
                 sequence. The values do not need to be contiguous - in fact
                 gaps are recommended to avoid frequent renumbering""" ;
    rdfs:domain <http://www.omg.org/spec/MOF2RDF/MOFSupport/OrderedAssociation> ;
    rdfs:label "link sequence"@en ;
    rdfs:range xsd:naturalInteger .

<http://www.omg.org/spec/MOF2RDF/MOFSupport/hasPart>
    a owl:ObjectProperty ;
    rdfs:comment """Property representing composition/ownership. Actual
                 composition properties should inherit from this""" ;
    rdfs:label "has part"@en .

<http://www.omg.org/spec/MOF2RDF/MOFSupport/hasContainer>
```

```
    a owl:ObjectProperty ;
    rdfs:comment """Property representing container. MOF/UML rules state that
                 each object may have at most one.""" ;
    rdfs:label "has container"@en ;
    owl:inverseOf <http://www.omg.org/spec/MOF2RDF/MOFSupport/hasPart> .

<http://www.omg.org/spec/MOF2RDF/MOFSupport/isAbstract>
    a owl:AnnotationProperty ;
    rdfs:comment """Whether the class is permitted to have any direct
                 instances""" ;
    rdfs:domain rdfs:Class ;
    rdfs:label "is abstract"@en ;
    rdfs:range xsd:boolean .

<http://www.omg.org/spec/MOF2RDF/MOFSupport/isOrdered>
    a owl:AnnotationProperty ;
    rdfs:comment """Whether the values of a property for a specific object
                 retain their sequence""" ;
    rdfs:domain rdfs:Property ;
    rdfs:label "is ordered"@en ;
    rdfs:range xsd:boolean .

<http://www.omg.org/spec/MOF2RDF/MOFSupport/isUnique>
    a owl:AnnotationProperty ;
    rdfs:comment """Whether the values of a property for a specific object
                 can contain the same value more than once""" ;
    rdfs:domain rdfs:Property ;
    rdfs:label "is unique"@en ;
    rdfs:range xsd:boolean .
```

Appendix B Content of Machine-Readable Files

(normative)

This specification document is accompanied by two *normative* machine-readable files:

ad/2018-11-02 This file contains the normative MOF Support Ontology in Turtle syntax.

ad/2018-11-03 This file contains an archive that contains two pairs of files. Each pair consists of an XMI file containing MOF elements and the corresponding transformation result file showing the equivalent OWL model. The correspondence between these two model representations is normative. One pair is provided for (regular) MOF elements (as shown in Figure 9.1 at the beginning of Clause 9), the second pair is provided for MOF Profiles (as shown in Figure 10.1 at the beginning of Clause 10).

This page intentionally left blank

Appendix C Transformation Example

(informative)

C.1 Example Model

The transformation example provided in this Annex uses a test model developed by the Model Interchange Working Group of the Object Management Group. The model is shown in Figure C.1 below, the resulting OWL code listing follows in the next section.

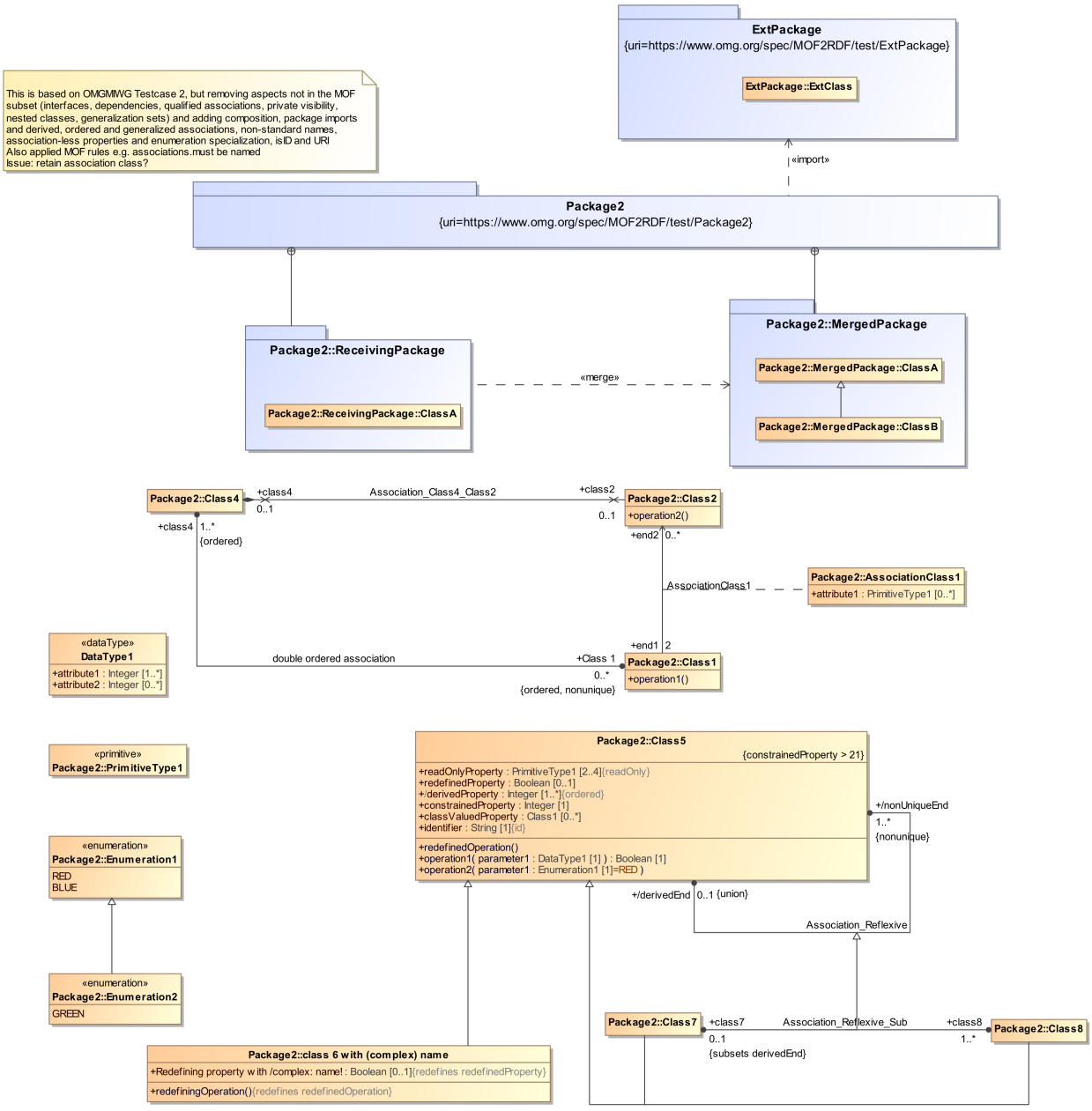


Figure C.1: MIWG Test Case 2: A Model

C.2 Resulting OWL Code

The resulting OWL code is presented below in Turtle Syntax.

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xs: <http://www.w3.org/2001/XMLSchema> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix mof2rdf: <http://www.omg.org/spec/MOF2RDF/MOFSupport/> .

<https://www.omg.org/spec/MOF2RDF/test/ExtPackage/>
    a owl:Ontology ;
    rdfs:label "ExtPackage" ;
    owl:imports <http://www.omg.org/spec/MOF2RDF/MOFSupport/> .

<https://www.omg.org/spec/MOF2RDF/test/ExtPackage/ExtClass>
    a owl:Class ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/ExtPackage/"
    ] ;
    rdfs:label "ExtClass" ;
    rdfs:subClassOf mof2rdf:Class .

<https://www.omg.org/spec/MOF2RDF/test/Package2/>
    a owl:Ontology ;
    rdfs:label "Package2" ;
    owl:imports <http://www.omg.org/spec/MOF2RDF/MOFSupport/>,
                <https://www.omg.org/spec/MOF2RDF/test/ExtPackage/>,
                <https://www.omg.org/spec/MOF2RDF/test/Package2/MergedPackage/>,
                <https://www.omg.org/spec/MOF2RDF/test/Package2/ReceivingPackage/> .

<https://www.omg.org/spec/MOF2RDF/test/Package2/AssociationClass1>
    a owl:Class ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Package2/"
    ] ;
    rdfs:label "AssociationClass1" ;
    rdfs:subClassOf mof2rdf:Association, [
        a owl:Restriction ;
        owl:onClass <https://www.omg.org/spec/MOF2RDF/test/Package2/Class2> ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Package2/AssociationClass1.end2> ;
        owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger
    ], [
        a owl:Restriction ;
        owl:onClass <https://www.omg.org/spec/MOF2RDF/test/Package2/Class1> ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Package2/AssociationClass1.end1> ;
        owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger
    ] .

<https://www.omg.org/spec/MOF2RDF/test/Package2/AssociationClass1.end1>
    a owl:ObjectProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Package2/AssociationClass1> ;
    rdfs:label "end1" ;
    rdfs:range <https://www.omg.org/spec/MOF2RDF/test/Package2/Class1> .

<https://www.omg.org/spec/MOF2RDF/test/Package2/AssociationClass1.end1.Class2>
    a owl:ObjectProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Package2/Class1> ;
    rdfs:label "end1 link" ;
    rdfs:range <https://www.omg.org/spec/MOF2RDF/test/Package2/AssociationClass1> .

<https://www.omg.org/spec/MOF2RDF/test/Package2/AssociationClass1.end2>
    a owl:ObjectProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Package2/AssociationClass1> ;
    rdfs:label "end2" ;
    rdfs:range <https://www.omg.org/spec/MOF2RDF/test/Package2/Class2> .

<https://www.omg.org/spec/MOF2RDF/test/Package2/AssociationClass1.end2.Class1>
    a owl:ObjectProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Package2/Class2> ;
    rdfs:label "end2 link" ;
    rdfs:range <https://www.omg.org/spec/MOF2RDF/test/Package2/AssociationClass1> .

<https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Class4_Class2>
    a owl:Class ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Package2/"
```

```
    ] ;
    rdfs:label "Association_Class4_Class2" ;
    rdfs:subClassOf mof2rdf:Association, [
        a owl:Restriction ;
        owl:onClass <https://www.omg.org/spec/MOF2RDF/test/Package2/Class2> ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Class4_Class2.class2> ;
        owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger
    ], [
        a owl:Restriction ;
        owl:onClass <https://www.omg.org/spec/MOF2RDF/test/Package2/Class4> ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Class4_Class2.class4> ;
        owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger
    ] .

<https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Class4_Class2.class2>
    a owl:ObjectProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Class4_Class2> ;
    rdfs:label "class2" ;
    rdfs:range <https://www.omg.org/spec/MOF2RDF/test/Package2/Class2> .

<https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Class4_Class2.class2.Class4>
    a owl:ObjectProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Package2/Class2> ;
    rdfs:label "class2 link" ;
    rdfs:range <https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Class4_Class2> .

<https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Class4_Class2.class4>
    a owl:ObjectProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Class4_Class2> ;
    rdfs:label "class4" ;
    rdfs:range <https://www.omg.org/spec/MOF2RDF/test/Package2/Class4> .

<https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Class4_Class2.class4.Class2>
    a owl:ObjectProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Package2/Class4> ;
    rdfs:label "class4 link" ;
    rdfs:range <https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Class4_Class2> .

<https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Reflexive>
    a owl:Class ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Package2/"
    ] ;
    rdfs:label "Association_Reflexive" ;
    rdfs:subClassOf mof2rdf:Association, [
        a owl:Restriction ;
        owl:onClass <https://www.omg.org/spec/MOF2RDF/test/Package2/Class5> ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Reflexive.derivedEnd> ;
        owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger
    ], [
        a owl:Restriction ;
        owl:onClass <https://www.omg.org/spec/MOF2RDF/test/Package2/Class5> ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Reflexive.nonUniqueEnd> ;
        owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger
    ] .

<https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Reflexive.derivedEnd>
    a owl:ObjectProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Reflexive> ;
    rdfs:label "derivedEnd" ;
    rdfs:range <https://www.omg.org/spec/MOF2RDF/test/Package2/Class5> .

<https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Reflexive.derivedEnd.Class5>
    a owl:ObjectProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Package2/Class5> ;
    rdfs:label "derivedEnd link" ;
    rdfs:range <https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Reflexive> .

<https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Reflexive.nonUniqueEnd>
    a owl:ObjectProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Reflexive> ;
    rdfs:label "nonUniqueEnd" ;
    rdfs:range <https://www.omg.org/spec/MOF2RDF/test/Package2/Class5> .

<https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Reflexive.nonUniqueEnd.Class5>
    a owl:ObjectProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Package2/Class5> ;
    rdfs:label "nonUniqueEnd link" ;
    rdfs:range <https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Reflexive> .

<https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Reflexive_Sub>
```

```
    a owl:Class ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Package2/"
    ] ;
    rdfs:label "Association_Reflexive_Sub" ;
    rdfs:subClassOf mof2rdf:Association, <https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Reflexive>, [
        a owl:Restriction ;
        owl:onClass <https://www.omg.org/spec/MOF2RDF/test/Package2/Class8> ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Reflexive_Sub.class8> ;
        owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger
    ], [
        a owl:Restriction ;
        owl:onClass <https://www.omg.org/spec/MOF2RDF/test/Package2/Class7> ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Reflexive_Sub.class7> ;
        owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger
    ] .

<https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Reflexive_Sub.class7>
    a owl:ObjectProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Reflexive_Sub> ;
    rdfs:label "class7" ;
    rdfs:range <https://www.omg.org/spec/MOF2RDF/test/Package2/Class7> ;
    rdfs:subPropertyOf <https://www.omg.org/spec/MOF2RDF/test/Package2/Class5.derivedEnd> .

<https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Reflexive_Sub.class7.Class8>
    a owl:ObjectProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Package2/Class7> ;
    rdfs:label "class7 link" ;
    rdfs:range <https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Reflexive_Sub> .

<https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Reflexive_Sub.class8>
    a owl:ObjectProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Reflexive_Sub> ;
    rdfs:label "class8" ;
    rdfs:range <https://www.omg.org/spec/MOF2RDF/test/Package2/Class8> .

<https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Reflexive_Sub.class8.Class7>
    a owl:ObjectProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Package2/Class8> ;
    rdfs:label "class8 link" ;
    rdfs:range <https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Reflexive_Sub> .

<https://www.omg.org/spec/MOF2RDF/test/Package2/Class1>
    a owl:Class ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Package2/"
    ] ;
    rdfs:label "Class1" ;
    rdfs:subClassOf mof2rdf:Class, [
        a owl:Restriction ;
        owl:allValuesFrom <https://www.omg.org/spec/MOF2RDF/test/Package2/AssociationClass1> ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Package2/AssociationClass1.end2.Class1>
    ], [
        a owl:Restriction ;
        owl:allValuesFrom <https://www.omg.org/spec/MOF2RDF/test/Package2/double_ordered_association> ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Package2/double_ordered_association.class4.Class1>
    ], [
        a owl:Restriction ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Package2/double_ordered_association.class4.Class1> ;
        owl:someValuesFrom <https://www.omg.org/spec/MOF2RDF/test/Package2/double_ordered_association>
    ] .

<https://www.omg.org/spec/MOF2RDF/test/Package2/Class1.class4>
    a owl:ObjectProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Package2/Class1> ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Package2/"
    ] ;
    rdfs:label "class4" ;
    rdfs:range <https://www.omg.org/spec/MOF2RDF/test/Package2/Class4> ;
    owl:propertyChainAxiom ([
            owl:inverseOf <https://www.omg.org/spec/MOF2RDF/test/Package2/double_ordered_association.Class_1>
        ]
        <https://www.omg.org/spec/MOF2RDF/test/Package2/double_ordered_association.class4>
    ) .

<https://www.omg.org/spec/MOF2RDF/test/Package2/Class2>
    a owl:Class ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Package2/"
    ] ;
```

```
    rdfs:label "Class2" ;
    rdfs:subClassOf mof2rdf:Class, [
        a owl:Restriction ;
        owl:allValuesFrom <https://www.omg.org/spec/MOF2RDF/test/Package2/AssociationClass1> ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Package2/AssociationClass1.end1.Class2>
    ], [
        a owl:Restriction ;
        owl:onClass <https://www.omg.org/spec/MOF2RDF/test/Package2/AssociationClass1> ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Package2/AssociationClass1.end1.Class2> ;
        owl:qualifiedCardinality "2"^^xsd:nonNegativeInteger
    ], [
        a owl:Restriction ;
        owl:allValuesFrom <https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Class4_Class2> ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Class4_Class2.class4.Class2>
    ], [
        a owl:Restriction ;
        owl:maxQualifiedCardinality "1"^^xsd:nonNegativeInteger ;
        owl:onClass <https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Class4_Class2> ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Class4_Class2.class4.Class2>
    ] .

<https://www.omg.org/spec/MOF2RDF/test/Package2/Class4>
    a owl:Class ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Package2/"
    ] ;
    rdfs:label "Class4" ;
    rdfs:subClassOf mof2rdf:Class, [
        a owl:Restriction ;
        owl:allValuesFrom <https://www.omg.org/spec/MOF2RDF/test/Package2/double_ordered_association> ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Package2/double_ordered_association.class1.Class4>
    ], [
        a owl:Restriction ;
        owl:allValuesFrom <https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Class4_Class2> ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Class4_Class2.class2.Class4>
    ], [
        a owl:Restriction ;
        owl:maxQualifiedCardinality "1"^^xsd:nonNegativeInteger ;
        owl:onClass <https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Class4_Class2> ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Class4_Class2.class2.Class4>
    ] .

<https://www.omg.org/spec/MOF2RDF/test/Package2/Class4.Class_1>
    a owl:ObjectProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Package2/Class4> ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Package2/"
    ] ;
    rdfs:label "Class 1" ;
    rdfs:range <https://www.omg.org/spec/MOF2RDF/test/Package2/Class1> ;
    owl:inverseOf <https://www.omg.org/spec/MOF2RDF/test/Package2/Class1.class4> ;
    owl:propertyChainAxiom ([
            owl:inverseOf <https://www.omg.org/spec/MOF2RDF/test/Package2/double_ordered_association.class4>
        ]
        <https://www.omg.org/spec/MOF2RDF/test/Package2/double_ordered_association.Class_1>
    ) .

<https://www.omg.org/spec/MOF2RDF/test/Package2/Class5>
    a owl:Class ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Package2/"
    ] ;
    rdfs:label "Class5" ;
    rdfs:subClassOf mof2rdf:Class, [
        a owl:Restriction ;
        owl:allValuesFrom <https://www.omg.org/spec/MOF2RDF/test/Package2/PrimitiveType1> ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Package2/Class5.readOnlyProperty>
    ], [
        a owl:Restriction ;
        owl:minQualifiedCardinality "2"^^xsd:nonNegativeInteger ;
        owl:onDataRange <https://www.omg.org/spec/MOF2RDF/test/Package2/PrimitiveType1> ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Package2/Class5.readOnlyProperty>
    ], [
        a owl:Restriction ;
        owl:maxQualifiedCardinality "4"^^xsd:nonNegativeInteger ;
        owl:onDataRange <https://www.omg.org/spec/MOF2RDF/test/Package2/PrimitiveType1> ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Package2/Class5.readOnlyProperty>
    ], [
        a owl:Restriction ;
        owl:allValuesFrom xsd:boolean ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Package2/Class5.redefinedProperty>
```

```
    ], [
        a owl:Restriction ;
        owl:maxQualifiedCardinality "1"^^xsd:nonNegativeInteger ;
        owl:onDataRange xsd:boolean ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Package2/Class5.redefinedProperty>
    ], [
        a owl:Restriction ;
        owl:allValuesFrom xsd:integer ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Package2/Class5.derivedProperty>
    ], [
        a owl:Restriction ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Package2/Class5.derivedProperty> ;
        owl:someValuesFrom xsd:integer
    ], [
        a owl:Restriction ;
        owl:allValuesFrom xsd:integer ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Package2/Class5.constrainedProperty>
    ], [
        a owl:Restriction ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Package2/Class5.constrainedProperty> ;
        owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
        owl:someValuesFrom xsd:integer
    ], [
        a owl:Restriction ;
        owl:allValuesFrom <https://www.omg.org/spec/MOF2RDF/test/Package2/Class1> ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Package2/Class5.classValuedProperty>
    ], [
        a owl:Restriction ;
        owl:allValuesFrom xsd:string ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Package2/Class5.identifier>
    ], [
        a owl:Restriction ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Package2/Class5.identifier> ;
        owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
        owl:someValuesFrom xsd:string
    ], [
        a owl:Restriction ;
        owl:allValuesFrom <https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Reflexive> ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Reflexive.derivedEnd.Class5>
    ], [
        a owl:Restriction ;
        owl:maxQualifiedCardinality "1"^^xsd:nonNegativeInteger ;
        owl:onClass <https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Reflexive> ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Reflexive.derivedEnd.Class5>
    ], [
        a owl:Restriction ;
        owl:allValuesFrom <https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Reflexive> ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Reflexive.nonUniqueEnd.Class5>
    ], [
        a owl:Restriction ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Reflexive.nonUniqueEnd.Class5> ;
        owl:someValuesFrom <https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Reflexive>
    ] .

<https://www.omg.org/spec/MOF2RDF/test/Package2/Class5.classValuedProperty>
    a owl:ObjectProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Package2/Class5> ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Package2/"
    ] ;
    rdfs:label "classValuedProperty" ;
    rdfs:range <https://www.omg.org/spec/MOF2RDF/test/Package2/Class1> .

<https://www.omg.org/spec/MOF2RDF/test/Package2/Class5.constrainedProperty>
    a owl:DatatypeProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Package2/Class5> ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Package2/"
    ] ;
    rdfs:label "constrainedProperty" ;
    rdfs:range xsd:integer .

<https://www.omg.org/spec/MOF2RDF/test/Package2/Class5.derivedEnd>
    a owl:ObjectProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Package2/Class5> ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Package2/"
    ] ;
    rdfs:label "derivedEnd" ;
    rdfs:range <https://www.omg.org/spec/MOF2RDF/test/Package2/Class5> ;
    owl:inverseOf <https://www.omg.org/spec/MOF2RDF/test/Package2/Class5.nonUniqueEnd> ;
```

```
    owl:propertyChainAxiom ([
            owl:inverseOf <https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Reflexive.nonUniqueEnd>
        ]
        <https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Reflexive.derivedEnd>
    ) .

<https://www.omg.org/spec/MOF2RDF/test/Package2/Class5.derivedProperty>
    a owl:DatatypeProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Package2/Class5> ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Package2/"
    ] ;
    rdfs:label "derivedProperty" ;
    rdfs:range xsd:integer .

<https://www.omg.org/spec/MOF2RDF/test/Package2/Class5.identifier>
    a owl:DatatypeProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Package2/Class5> ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Package2/"
    ] ;
    rdfs:label "identifier" ;
    rdfs:range xsd:string .

<https://www.omg.org/spec/MOF2RDF/test/Package2/Class5.nonUniqueEnd>
    a owl:ObjectProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Package2/Class5> ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Package2/"
    ] ;
    rdfs:label "nonUniqueEnd" ;
    rdfs:range <https://www.omg.org/spec/MOF2RDF/test/Package2/Class5> ;
    owl:propertyChainAxiom ([
            owl:inverseOf <https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Reflexive.derivedEnd>
        ]
        <https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Reflexive.nonUniqueEnd>
    ) .

<https://www.omg.org/spec/MOF2RDF/test/Package2/Class5.readOnlyProperty>
    a owl:DatatypeProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Package2/Class5> ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Package2/"
    ] ;
    rdfs:label "readOnlyProperty" ;
    rdfs:range <https://www.omg.org/spec/MOF2RDF/test/Package2/PrimitiveType1> .

<https://www.omg.org/spec/MOF2RDF/test/Package2/Class5.redefinedProperty>
    a owl:DatatypeProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Package2/Class5> ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Package2/"
    ] ;
    rdfs:label "redefinedProperty" ;
    rdfs:range xsd:boolean .

<https://www.omg.org/spec/MOF2RDF/test/Package2/Class7>
    a owl:Class ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Package2/"
    ] ;
    rdfs:label "Class7" ;
    rdfs:subClassOf mof2rdf:Class, <https://www.omg.org/spec/MOF2RDF/test/Package2/Class5>, [
        a owl:Restriction ;
        owl:allValuesFrom <https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Reflexive_Sub> ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Reflexive_Sub.class8.Class7>
    ], [
        a owl:Restriction ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Reflexive_Sub.class8.Class7> ;
        owl:someValuesFrom <https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Reflexive_Sub>
    ] .

<https://www.omg.org/spec/MOF2RDF/test/Package2/Class7.class8>
    a owl:ObjectProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Package2/Class7> ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Package2/"
    ] ;
    rdfs:label "class8" ;
    rdfs:range <https://www.omg.org/spec/MOF2RDF/test/Package2/Class8> ;
    owl:inverseOf <https://www.omg.org/spec/MOF2RDF/test/Package2/Class8.class7> ;
```

```
    owl:propertyChainAxiom ([
            owl:inverseOf <https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Reflexive_Sub.class7>
        ]
        <https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Reflexive_Sub.class8>
    ) .

<https://www.omg.org/spec/MOF2RDF/test/Package2/Class8>
    a owl:Class ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Package2/"
    ] ;
    rdfs:label "Class8" ;
    rdfs:subClassOf mof2rdf:Class, <https://www.omg.org/spec/MOF2RDF/test/Package2/Class5>, [
        a owl:Restriction ;
        owl:allValuesFrom <https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Reflexive_Sub> ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Reflexive_Sub.class7.Class8>
    ], [
        a owl:Restriction ;
        owl:maxQualifiedCardinality "1"^^xsd:nonNegativeInteger ;
        owl:onClass <https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Reflexive_Sub> ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Reflexive_Sub.class7.Class8>
    ] .

<https://www.omg.org/spec/MOF2RDF/test/Package2/Class8.class7>
    a owl:ObjectProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Package2/Class8> ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Package2/"
    ] ;
    rdfs:label "class7" ;
    rdfs:range <https://www.omg.org/spec/MOF2RDF/test/Package2/Class7> ;
    rdfs:subPropertyOf <https://www.omg.org/spec/MOF2RDF/test/Package2/Class5.derivedEnd> ;
    owl:propertyChainAxiom ([
            owl:inverseOf <https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Reflexive_Sub.class8>
        ]
        <https://www.omg.org/spec/MOF2RDF/test/Package2/Association_Reflexive_Sub.class7>
    ) .

<https://www.omg.org/spec/MOF2RDF/test/Package2/DataType1>
    a owl:Class ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Package2/"
    ] ;
    rdfs:label "DataType1" ;
    rdfs:subClassOf mof2rdf:DataType, [
        a owl:Restriction ;
        owl:allValuesFrom xsd:integer ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Package2/DataType1.attribute1>
    ], [
        a owl:Restriction ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Package2/DataType1.attribute1> ;
        owl:someValuesFrom xsd:integer
    ], [
        a owl:Restriction ;
        owl:allValuesFrom xsd:integer ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Package2/DataType1.attribute2>
    ] .

<https://www.omg.org/spec/MOF2RDF/test/Package2/DataType1.attribute1>
    a owl:DatatypeProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Package2/DataType1> ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Package2/"
    ] ;
    rdfs:label "attribute1" ;
    rdfs:range xsd:integer .

<https://www.omg.org/spec/MOF2RDF/test/Package2/DataType1.attribute2>
    a owl:DatatypeProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Package2/DataType1> ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Package2/"
    ] ;
    rdfs:label "attribute2" ;
    rdfs:range xsd:integer .

<https://www.omg.org/spec/MOF2RDF/test/Package2/Enumeration1>
    a owl:Class ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Package2/"
    ] ;
```

```
    rdfs:label "Enumeration1" ;
    rdfs:subClassOf mof2rdf:Enumeration ;
    owl:equivalentClass [
        a owl:Class ;
        owl:oneOf (<https://www.omg.org/spec/MOF2RDF/test/Package2/Enumeration1.RED>
            <https://www.omg.org/spec/MOF2RDF/test/Package2/Enumeration1.BLUE>
        )
    ] .

<https://www.omg.org/spec/MOF2RDF/test/Package2/Enumeration1.BLUE>
    a owl:NamedIndividual, <https://www.omg.org/spec/MOF2RDF/test/Package2/Enumeration1> ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Package2/"
    ] ;
    rdfs:label "BLUE" .

<https://www.omg.org/spec/MOF2RDF/test/Package2/Enumeration1.RED>
    a owl:NamedIndividual, <https://www.omg.org/spec/MOF2RDF/test/Package2/Enumeration1> ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Package2/"
    ] ;
    rdfs:label "RED" .

<https://www.omg.org/spec/MOF2RDF/test/Package2/Enumeration2>
    a owl:Class ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Package2/"
    ] ;
    rdfs:label "Enumeration2" ;
    rdfs:subClassOf mof2rdf:Enumeration, <https://www.omg.org/spec/MOF2RDF/test/Package2/Enumeration1> ;
    owl:equivalentClass [
        a owl:Class ;
        owl:oneOf (<https://www.omg.org/spec/MOF2RDF/test/Package2/Enumeration2.GREEN>
        )
    ] .

<https://www.omg.org/spec/MOF2RDF/test/Package2/Enumeration2.GREEN>
    a owl:NamedIndividual, <https://www.omg.org/spec/MOF2RDF/test/Package2/Enumeration2> ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Package2/"
    ] ;
    rdfs:label "GREEN" .

<https://www.omg.org/spec/MOF2RDF/test/Package2/MergedPackage/>
    a owl:Ontology ;
    rdfs:label "MergedPackage" ;
    owl:imports <http://www.omg.org/spec/MOF2RDF/MOFSupport/> .

<https://www.omg.org/spec/MOF2RDF/test/Package2/MergedPackage/ClassA>
    a owl:Class ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Package2/MergedPackage/"
    ] ;
    rdfs:label "ClassA" ;
    rdfs:subClassOf mof2rdf:Class .

<https://www.omg.org/spec/MOF2RDF/test/Package2/MergedPackage/ClassB>
    a owl:Class ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Package2/MergedPackage/"
    ] ;
    rdfs:label "ClassB" ;
    rdfs:subClassOf mof2rdf:Class, <https://www.omg.org/spec/MOF2RDF/test/Package2/MergedPackage/ClassA> .

<https://www.omg.org/spec/MOF2RDF/test/Package2/ReceivingPackage/>
    a owl:Ontology ;
    rdfs:label "ReceivingPackage" ;
    owl:imports <http://www.omg.org/spec/MOF2RDF/MOFSupport/> .

<https://www.omg.org/spec/MOF2RDF/test/Package2/ReceivingPackage/ClassA>
    a owl:Class ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Package2/ReceivingPackage/"
    ] ;
    rdfs:label "ClassA" ;
    rdfs:subClassOf mof2rdf:Class .

<https://www.omg.org/spec/MOF2RDF/test/Package2/class_6_with__complex__name>
    a owl:Class ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Package2/"
```

```
    ] ;
    rdfs:label "class 6 with (complex) name" ;
    rdfs:subClassOf mof2rdf:Class, <https://www.omg.org/spec/MOF2RDF/test/Package2/Class5>, [
        a owl:Restriction ;
        owl:allValuesFrom xsd:boolean ;
        owl:onProperty
          <https://www.omg.org/spec/MOF2RDF/test/Package2/class_6_with__complex__name.Redefining_property_with_.complex._name!>
    ], [
        a owl:Restriction ;
        owl:maxQualifiedCardinality "1"^^xsd:nonNegativeInteger ;
        owl:onDataRange xsd:boolean ;
        owl:onProperty
          <https://www.omg.org/spec/MOF2RDF/test/Package2/class_6_with__complex__name.Redefining_property_with_.complex._name!>
    ] .

<https://www.omg.org/spec/MOF2RDF/test/Package2/class_6_with__complex__name.Redefining_property_with_.complex._name!>
    a owl:DatatypeProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Package2/class_6_with__complex__name> ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Package2/"
    ] ;
    rdfs:label "Redefining property with /complex: name!" ;
    rdfs:range xsd:boolean .

<https://www.omg.org/spec/MOF2RDF/test/Package2/double_ordered_association>
    a owl:Class ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Package2/"
    ] ;
    rdfs:label "double ordered association" ;
    rdfs:subClassOf mof2rdf:OrderedAssociation, [
        a owl:Restriction ;
        owl:onClass <https://www.omg.org/spec/MOF2RDF/test/Package2/Class1> ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Package2/double_ordered_association.Class_1> ;
        owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger
    ], [
        a owl:Restriction ;
        owl:onClass <https://www.omg.org/spec/MOF2RDF/test/Package2/Class4> ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Package2/double_ordered_association.class4> ;
        owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger
    ] .

<https://www.omg.org/spec/MOF2RDF/test/Package2/double_ordered_association.Class_1>
    a owl:ObjectProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Package2/double_ordered_association> ;
    rdfs:label "Class 1" ;
    rdfs:range <https://www.omg.org/spec/MOF2RDF/test/Package2/Class1> .

<https://www.omg.org/spec/MOF2RDF/test/Package2/double_ordered_association.Class_1.Class4>
    a owl:ObjectProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Package2/Class1> ;
    rdfs:label "Class 1 link" ;
    rdfs:range <https://www.omg.org/spec/MOF2RDF/test/Package2/double_ordered_association> .

<https://www.omg.org/spec/MOF2RDF/test/Package2/double_ordered_association.class4>
    a owl:ObjectProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Package2/double_ordered_association> ;
    rdfs:label "class4" ;
    rdfs:range <https://www.omg.org/spec/MOF2RDF/test/Package2/Class4> .

<https://www.omg.org/spec/MOF2RDF/test/Package2/double_ordered_association.class4.Class1>
    a owl:ObjectProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Package2/Class4> ;
    rdfs:label "class4 link" ;
    rdfs:range <https://www.omg.org/spec/MOF2RDF/test/Package2/double_ordered_association> .
```

Appendix D A Profile Transformation Example

(informative)

D.1 Example Profile

The transformation example provided in this Annex uses a test profile developed by the Model Interchange Working Group of the Object Management Group. The model is shown in Figure D.1 below, the resulting OWL code listing follows in the next section.

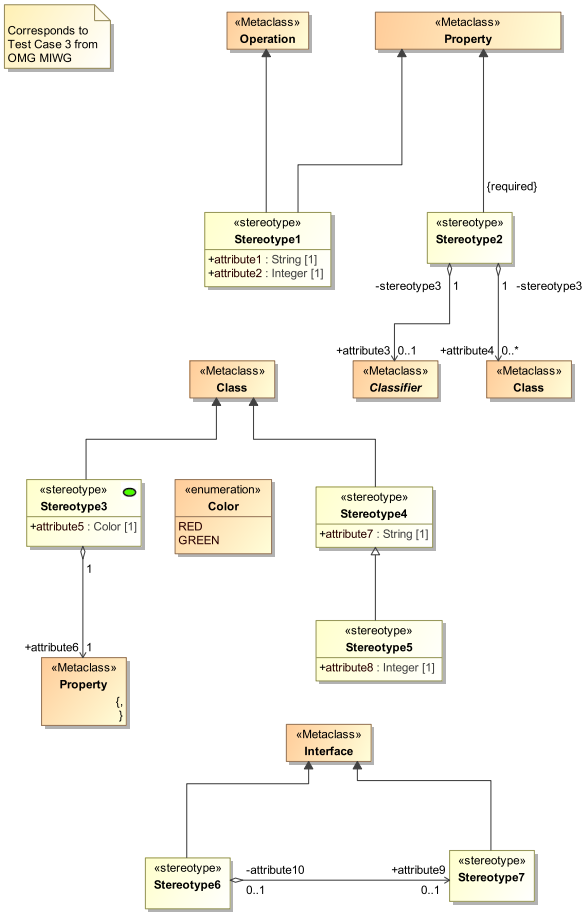


Figure D.1: MIWG Test Case 3: A Profile

D.2 Resulting OWL Code

The resulting OWL code is presented below in Turtle Syntax.

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xs: <http://www.w3.org/2001/XMLSchema> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix mof2rdf: <http://www.omg.org/spec/MOF2RDF/MOFSupport/> .

```

```
<http://www.omg.org/spec/UML/20131001/UML/Property>
    rdfs:subClassOf [
        a owl:Restriction ;
        owl:onProperty rdf:type ;
        owl:someValuesFrom <https://www.omg.org/spec/MOF2RDF/test/Profile/Stereotype2>
    ] .

<https://www.omg.org/spec/MOF2RDF/test/Profile/>
    a owl:Ontology ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Profile/"
    ] ;
    rdfs:label "TestCase3_Profile" ;
    owl:imports <http://www.omg.org/spec/MOF2RDF/MOFSupport/>, <http://www.omg.org/spec/UML/20131001/UML/> .

<https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute3_stereotype3>
    a owl:Class ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Profile/"
    ] ;
    rdfs:label "A_attribute3_stereotype3" ;
    rdfs:subClassOf mof2rdf:Association, [
        a owl:Restriction ;
        owl:onClass <http://www.omg.org/spec/UML/20131001/UML/Classifier> ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute3_stereotype3.attribute3> ;
        owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger
    ], [
        a owl:Restriction ;
        owl:onClass <https://www.omg.org/spec/MOF2RDF/test/Profile/Stereotype2> ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute3_stereotype3.stereotype3> ;
        owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger
    ] .

<https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute3_stereotype3.attribute3>
    a owl:ObjectProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute3_stereotype3> ;
    rdfs:label "attribute3" ;
    rdfs:range <http://www.omg.org/spec/UML/20131001/UML/Classifier> .

<https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute3_stereotype3.attribute3.Stereotype2>
    a owl:ObjectProperty ;
    rdfs:domain <http://www.omg.org/spec/UML/20131001/UML/Classifier> ;
    rdfs:label "attribute3 link" ;
    rdfs:range <https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute3_stereotype3> .

<https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute3_stereotype3.stereotype3>
    a owl:ObjectProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute3_stereotype3> ;
    rdfs:label "stereotype3" ;
    rdfs:range <https://www.omg.org/spec/MOF2RDF/test/Profile/Stereotype2> .

<https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute3_stereotype3.stereotype3.Classifier>
    a owl:ObjectProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Profile/Stereotype2> ;
    rdfs:label "stereotype3 link" ;
    rdfs:range <https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute3_stereotype3> .

<https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute4_stereotype3>
    a owl:Class ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Profile/"
    ] ;
    rdfs:label "A_attribute4_stereotype3" ;
    rdfs:subClassOf mof2rdf:Association, [
        a owl:Restriction ;
        owl:onClass <http://www.omg.org/spec/UML/20131001/UML/Class> ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute4_stereotype3.attribute4> ;
        owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger
    ], [
        a owl:Restriction ;
        owl:onClass <https://www.omg.org/spec/MOF2RDF/test/Profile/Stereotype2> ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute4_stereotype3.stereotype3> ;
        owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger
    ] .

<https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute4_stereotype3.attribute4>
    a owl:ObjectProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute4_stereotype3> ;
    rdfs:label "attribute4" ;
    rdfs:range <http://www.omg.org/spec/UML/20131001/UML/Class> .

```

```
<https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute4_stereotype3.attribute4.Stereotype2>
    a owl:ObjectProperty ;
    rdfs:domain <http://www.omg.org/spec/UML/20131001/UML/Class> ;
    rdfs:label "attribute4 link" ;
    rdfs:range <https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute4_stereotype3> .

<https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute4_stereotype3.stereotype3>
    a owl:ObjectProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute4_stereotype3> ;
    rdfs:label "stereotype3" ;
    rdfs:range <https://www.omg.org/spec/MOF2RDF/test/Profile/Stereotype2> .

<https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute4_stereotype3.stereotype3.Class>
    a owl:ObjectProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Profile/Stereotype2> ;
    rdfs:label "stereotype3 link" ;
    rdfs:range <https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute4_stereotype3> .

<https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute6_Stereotype3>
    a owl:Class ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Profile/"
    ] ;
    rdfs:label "A_attribute6_Stereotype3" ;
    rdfs:subClassOf mof2rdf:Association, [
        a owl:Restriction ;
        owl:onClass <http://www.omg.org/spec/UML/20131001/UML/Property> ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute6_Stereotype3.attribute6> ;
        owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger
    ], [
        a owl:Restriction ;
        owl:onClass <https://www.omg.org/spec/MOF2RDF/test/Profile/Stereotype3> ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute6_Stereotype3.> ;
        owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger
    ] .

<https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute6_Stereotype3.>
    a owl:ObjectProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute6_Stereotype3> ;
    rdfs:label "unnamed" ;
    rdfs:range <https://www.omg.org/spec/MOF2RDF/test/Profile/Stereotype3> .

<https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute6_Stereotype3.attribute6>
    a owl:ObjectProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute6_Stereotype3> ;
    rdfs:label "attribute6" ;
    rdfs:range <http://www.omg.org/spec/UML/20131001/UML/Property> .

<https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute6_Stereotype3.attribute6.Stereotype3>
    a owl:ObjectProperty ;
    rdfs:domain <http://www.omg.org/spec/UML/20131001/UML/Property> ;
    rdfs:label "attribute6 link" ;
    rdfs:range <https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute6_Stereotype3> .

<https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute6_Stereotype3.unnamed.Property>
    a owl:ObjectProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Profile/Stereotype3> ;
    rdfs:label "unnamed link" ;
    rdfs:range <https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute6_Stereotype3> .

<https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute9_attribute10>
    a owl:Class ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Profile/"
    ] ;
    rdfs:label "A_attribute9_attribute10" ;
    rdfs:subClassOf mof2rdf:Association, [
        a owl:Restriction ;
        owl:onClass <https://www.omg.org/spec/MOF2RDF/test/Profile/Stereotype7> ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute9_attribute10.attribute9> ;
        owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger
    ], [
        a owl:Restriction ;
        owl:onClass <https://www.omg.org/spec/MOF2RDF/test/Profile/Stereotype6> ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute9_attribute10.attribute10> ;
        owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger
    ] .

<https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute9_attribute10.attribute10>
    a owl:ObjectProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute9_attribute10> ;
```

```
    rdfs:label "attribute10" ;
    rdfs:range <https://www.omg.org/spec/MOF2RDF/test/Profile/Stereotype6> .

<https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute9_attribute10.attribute10.Stereotype7>
    a owl:ObjectProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Profile/Stereotype6> ;
    rdfs:label "attribute10 link" ;
    rdfs:range <https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute9_attribute10> .

<https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute9_attribute10.attribute9>
    a owl:ObjectProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute9_attribute10> ;
    rdfs:label "attribute9" ;
    rdfs:range <https://www.omg.org/spec/MOF2RDF/test/Profile/Stereotype7> .

<https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute9_attribute10.attribute9.Stereotype6>
    a owl:ObjectProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Profile/Stereotype7> ;
    rdfs:label "attribute9 link" ;
    rdfs:range <https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute9_attribute10> .

<https://www.omg.org/spec/MOF2RDF/test/Profile/Color>
    a owl:Class ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Profile/"
    ] ;
    rdfs:label "Color" ;
    rdfs:subClassOf mof2rdf:Enumeration ;
    owl:equivalentClass [
        a owl:Class ;
        owl:oneOf (<https://www.omg.org/spec/MOF2RDF/test/Profile/Color.RED>
            <https://www.omg.org/spec/MOF2RDF/test/Profile/Color.GREEN>
        )
    ] .

<https://www.omg.org/spec/MOF2RDF/test/Profile/Color.GREEN>
    a owl:NamedIndividual, <https://www.omg.org/spec/MOF2RDF/test/Profile/Color> ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Profile/"
    ] ;
    rdfs:label "GREEN" .

<https://www.omg.org/spec/MOF2RDF/test/Profile/Color.RED>
    a owl:NamedIndividual, <https://www.omg.org/spec/MOF2RDF/test/Profile/Color> ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Profile/"
    ] ;
    rdfs:label "RED" .

<https://www.omg.org/spec/MOF2RDF/test/Profile/Stereotype1>
    a owl:Class ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Profile/"
    ] ;
    rdfs:label "Stereotype1" ;
    rdfs:subClassOf mof2rdf:Stereotype, [
        a owl:Restriction ;
        owl:onProperty rdf:type ;
        owl:someValuesFrom [
            a owl:Class ;
            owl:unionOf (<http://www.omg.org/spec/UML/20131001/UML/Operation>
                <http://www.omg.org/spec/UML/20131001/UML/Property>
            )
        ]
    ], [
        a owl:Restriction ;
        owl:allValuesFrom xsd:string ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Profile/Stereotype1.attribute1>
    ], [
        a owl:Restriction ;
        owl:onDataRange xsd:string ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Profile/Stereotype1.attribute1> ;
        owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger
    ], [
        a owl:Restriction ;
        owl:allValuesFrom xsd:integer ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Profile/Stereotype1.attribute2>
    ], [
        a owl:Restriction ;
        owl:onDataRange xsd:integer ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Profile/Stereotype1.attribute2> ;
```

```
        owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger
    ] .

<https://www.omg.org/spec/MOF2RDF/test/Profile/Stereotype1.attribute1>
    a owl:DatatypeProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Profile/Stereotype1> ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Profile/"
    ] ;
    rdfs:label "attribute1" ;
    rdfs:range xsd:string .

<https://www.omg.org/spec/MOF2RDF/test/Profile/Stereotype1.attribute2>
    a owl:DatatypeProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Profile/Stereotype1> ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Profile/"
    ] ;
    rdfs:label "attribute2" ;
    rdfs:range xsd:integer .

<https://www.omg.org/spec/MOF2RDF/test/Profile/Stereotype2>
    a owl:Class ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Profile/"
    ] ;
    rdfs:label "Stereotype2" ;
    rdfs:subClassOf mof2rdf:Stereotype, [
        a owl:Restriction ;
        owl:onProperty rdf:type ;
        owl:someValuesFrom <http://www.omg.org/spec/UML/20131001/UML/Property>
    ], [
        a owl:Restriction ;
        owl:allValuesFrom <https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute3_stereotype3> ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute3_stereotype3.attribute3.Stereotype2>
    ], [
        a owl:Restriction ;
        owl:maxQualifiedCardinality "1"^^xsd:nonNegativeInteger ;
        owl:onClass <https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute3_stereotype3> ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute3_stereotype3.attribute3.Stereotype2>
    ], [
        a owl:Restriction ;
        owl:allValuesFrom <https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute4_stereotype3> ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute4_stereotype3.attribute4.Stereotype2>
    ] .

<https://www.omg.org/spec/MOF2RDF/test/Profile/Stereotype2.attribute3>
    a owl:ObjectProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Profile/Stereotype2> ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Profile/"
    ] ;
    rdfs:label "attribute3" ;
    rdfs:range <http://www.omg.org/spec/UML/20131001/UML/Classifier> ;
    owl:propertyChainAxiom ([
            owl:inverseOf <https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute3_stereotype3.stereotype3>
        ]
        <https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute3_stereotype3.attribute3>
    ) .

<https://www.omg.org/spec/MOF2RDF/test/Profile/Stereotype2.attribute4>
    a owl:ObjectProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Profile/Stereotype2> ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Profile/"
    ] ;
    rdfs:label "attribute4" ;
    rdfs:range <http://www.omg.org/spec/UML/20131001/UML/Class> ;
    owl:propertyChainAxiom ([
            owl:inverseOf <https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute4_stereotype3.stereotype3>
        ]
        <https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute4_stereotype3.attribute4>
    ) .

<https://www.omg.org/spec/MOF2RDF/test/Profile/Stereotype3>
    mof2rdf:stereotypeIconPNG
    "89504E470D0A1A0A0000000D4948445200000018000000180803000000
    D7A9CDCA0000000173524742 00AECE1CE90000000467414D410000B18F0BFC6105000000206348524
    D00007A26000080840000FA00000080E800007530000 0EA6000003A98000017709CBA513C00000300
    504C5445000000080808101010181818202020303030383838404040484848585858606060686868 78
    78787F7F7F4CFF008787878F8F8F8F9797979F9F9FAFAFAFB7B7B7BFBFBFC7C7C7CFCFCFD7D7D7DFDFDFE
```

```
    7E7E7EFEFEFF7F7F7FFFFFF0000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000000000000000000000
    00000000000000000000000000000000000000000000000000000000000000000FB142D3A000000187
    4455874536F66747761726500506169 6E742E4E45542076332E3336A9E7E2250000009C494441542853
    CD91C912832010445B100D18708998B8FDFF6F861910E321D754FA02358F2EBA01FB17E1CFC1E2835E3
    9648AEB8D04ABD08FC818AC371AD541226C2AB611984B88BB4BEA24C4330185EA18D3AA2137764C90
    9F73E71406063DF415B468188C5057D0A065B049E4ABE9C4806289A9C602F5E9D12043ECE143FC52DB
    20A342A13E17DC579B8A03C2CC6773DA4D1D39AC3F1EEB175FFB068C1D327D1AB319A30000000049
    454E44AE426082"^^xsd:hexBinary ;
    a owl:Class ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Profile/"
    ] ;
    rdfs:label "Stereotype3" ;
    rdfs:subClassOf mof2rdf:Stereotype, [
        a owl:Restriction ;
        owl:onProperty rdf:type ;
        owl:someValuesFrom <http://www.omg.org/spec/UML/20131001/UML/Class>
    ], [
        a owl:Restriction ;
        owl:allValuesFrom <https://www.omg.org/spec/MOF2RDF/test/Profile/Color> ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Profile/Stereotype3.attribute5>
    ], [
        a owl:Restriction ;
        owl:onClass <https://www.omg.org/spec/MOF2RDF/test/Profile/Color> ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Profile/Stereotype3.attribute5> ;
        owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger
    ], [
        a owl:Restriction ;
        owl:allValuesFrom <https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute6_Stereotype3> ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute6_Stereotype3.attribute6.Stereotype3>
    ], [
        a owl:Restriction ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute6_Stereotype3.attribute6.Stereotype3> ;
        owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
        owl:someValuesFrom <https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute6_Stereotype3>
    ] .

<https://www.omg.org/spec/MOF2RDF/test/Profile/Stereotype3.attribute5>
    a owl:ObjectProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Profile/Stereotype3> ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Profile/"
    ] ;
    rdfs:label "attribute5" ;
    rdfs:range <https://www.omg.org/spec/MOF2RDF/test/Profile/Color> .

<https://www.omg.org/spec/MOF2RDF/test/Profile/Stereotype3.attribute6>
    a owl:ObjectProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Profile/Stereotype3> ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Profile/"
    ] ;
    rdfs:label "attribute6" ;
    rdfs:range <http://www.omg.org/spec/UML/20131001/UML/Property> ;
    owl:propertyChainAxiom ([
            owl:inverseOf <https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute6_Stereotype3.>
        ]
        <https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute6_Stereotype3.attribute6>
    ) .

<https://www.omg.org/spec/MOF2RDF/test/Profile/Stereotype4>
    a owl:Class ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Profile/"
    ] ;
    rdfs:label "Stereotype4" ;
```

```
    rdfs:subClassOf mof2rdf:Stereotype, [
        a owl:Restriction ;
        owl:onProperty rdf:type ;
        owl:someValuesFrom <http://www.omg.org/spec/UML/20131001/UML/Class>
    ], [
        a owl:Restriction ;
        owl:allValuesFrom xsd:string ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Profile/Stereotype4.attribute7>
    ], [
        a owl:Restriction ;
        owl:onDataRange xsd:string ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Profile/Stereotype4.attribute7> ;
        owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger
    ] .

<https://www.omg.org/spec/MOF2RDF/test/Profile/Stereotype4.attribute7>
    a owl:DatatypeProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Profile/Stereotype4> ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Profile/"
    ] ;
    rdfs:label "attribute7" ;
    rdfs:range xsd:string .

<https://www.omg.org/spec/MOF2RDF/test/Profile/Stereotype5>
    a owl:Class ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Profile/"
    ] ;
    rdfs:label "Stereotype5" ;
    rdfs:subClassOf mof2rdf:Stereotype, <https://www.omg.org/spec/MOF2RDF/test/Profile/Stereotype4>, [
        a owl:Restriction ;
        owl:allValuesFrom xsd:integer ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Profile/Stereotype5.attribute8>
    ], [
        a owl:Restriction ;
        owl:onDataRange xsd:integer ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Profile/Stereotype5.attribute8> ;
        owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger
    ] .

<https://www.omg.org/spec/MOF2RDF/test/Profile/Stereotype5.attribute8>
    a owl:DatatypeProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Profile/Stereotype5> ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Profile/"
    ] ;
    rdfs:label "attribute8" ;
    rdfs:range xsd:integer .

<https://www.omg.org/spec/MOF2RDF/test/Profile/Stereotype6>
    a owl:Class ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Profile/"
    ] ;
    rdfs:label "Stereotype6" ;
    rdfs:subClassOf mof2rdf:Stereotype, [
        a owl:Restriction ;
        owl:onProperty rdf:type ;
        owl:someValuesFrom <http://www.omg.org/spec/UML/20131001/UML/Interface>
    ], [
        a owl:Restriction ;
        owl:allValuesFrom <https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute9_attribute10> ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute9_attribute10.attribute9.Stereotype6>
    ], [
        a owl:Restriction ;
        owl:maxQualifiedCardinality "1"^^xsd:nonNegativeInteger ;
        owl:onClass <https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute9_attribute10> ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute9_attribute10.attribute9.Stereotype6>
    ] .

<https://www.omg.org/spec/MOF2RDF/test/Profile/Stereotype6.attribute9>
    a owl:ObjectProperty ;
    rdfs:domain <https://www.omg.org/spec/MOF2RDF/test/Profile/Stereotype6> ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Profile/"
    ] ;
    rdfs:label "attribute9" ;
    rdfs:range <https://www.omg.org/spec/MOF2RDF/test/Profile/Stereotype7> ;
    owl:propertyChainAxiom ([
            owl:inverseOf <https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute9_attribute10.attribute10>
```

```
        ]
        <https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute9_attribute10.attribute9>
    ) .

<https://www.omg.org/spec/MOF2RDF/test/Profile/Stereotype7>
    a owl:Class ;
    rdfs:isDefinedBy [
        rdfs:resource "https://www.omg.org/spec/MOF2RDF/test/Profile/"
    ] ;
    rdfs:label "Stereotype7" ;
    rdfs:subClassOf mof2rdf:Stereotype, [
        a owl:Restriction ;
        owl:onProperty rdf:type ;
        owl:someValuesFrom <http://www.omg.org/spec/UML/20131001/UML/Interface>
    ], [
        a owl:Restriction ;
        owl:allValuesFrom <https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute9_attribute10> ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute9_attribute10.attribute10.Stereotype7>
    ], [
        a owl:Restriction ;
        owl:maxQualifiedCardinality "1"^^xsd:nonNegativeInteger ;
        owl:onClass <https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute9_attribute10> ;
        owl:onProperty <https://www.omg.org/spec/MOF2RDF/test/Profile/A_attribute9_attribute10.attribute10.Stereotype7>
    ] .

[]
    a owl:AllDifferent ;
    owl:distinctMembers (<https://www.omg.org/spec/MOF2RDF/test/Profile/Color.RED>
        <https://www.omg.org/spec/MOF2RDF/test/Profile/Color.GREEN>
    ) .
```