Date: March 2005

Product Lifecycle Management Services

Convenience Document

dtc/05-03-08

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 organizations 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. The copyright holders listed above has agreed that no person shall be deemed to have infringed the copyright in the included material of any such copyright holder by reason of having used the specification set forth herein or having conformed any computer software to the specification.

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

PATENTS

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

GENERAL USE RESTRICTIONS

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

DISCLAIMER OF WARRANTY

WHILE THIS PUBLICATION IS BELIEVED TO BE ACCURATE, IT IS PROVIDED "AS IS" AND MAY CONTAIN ERRORS OR MISPRINTS. THE OBJECT MANAGEMENT GROUP AND THE COMPANIES LISTED ABOVE MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS PUBLICATION, INCLUDING BUT NOT LIMITED TO ANY WARRANTY OF TITLE OR OWNERSHIP, IMPLIED WARRANTY OF MERCHANTABILITY OR WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE OR USE. IN NO EVENT SHALL THE OBJECT MANAGEMENT GROUP OR ANY OF THE COMPANIES LISTED ABOVE BE LIABLE FOR ERRORS CONTAINED HEREIN OR FOR DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL, RELIANCE OR COVER DAMAGES, INCLUDING LOSS OF PROFITS, REVENUE, DATA OR USE, INCURRED BY ANY USER OR ANY THIRD PARTY IN CONNECTION WITH THE FURNISHING, PERFORMANCE, OR USE OF THIS MATERIAL, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

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

RESTRICTED RIGHTS LEGEND

Use, duplication or disclosure by the U.S. Government is subject to the restrictions set forth in subparagraph (c) (1) (ii) of The Rights in Technical Data and Computer Software Clause at DFARS 252.227-7013 or in subparagraph (c)(1) and (2) of the Commercial Computer Software - Restricted Rights clauses at 48 C.F.R. 52.227-19 or as specified in 48 C.F.R. 227-7202-2 of the DoD F.A.R. Supplement and its successors, or as specified in 48 C.F.R. 12.212 of the Federal Acquisition Regulations and its successors, as applicable. The specification copyright owners are as indicated above and may be contacted through the Object Management Group, 250 First Avenue, Needham, MA 02494, U.S.A.

TRADEMARKS

The OMG Object Management Group Logo®, CORBA®, CORBA Academy®, The Information Brokerage®, XMI® and IIOP® are registered trademarks of the Object Management Group. OMG™, Object Management Group™, CORBA logos™, OMG Interface Definition Language (IDL)™, The Architecture of Choice for a Changing World™, CORBAservices™, CORBAfacilities™, CORBAmed™, CORBAnet™, Integrate 2002™, Middleware That's Everywhere™, UML™, Unified Modeling Language™, The UML Cube logo™, MOF™, CWM™, The CWM Logo™, Model Driven Architecture™, Model Driven Architecture Logos™, MDA™, OMG Model Driven Architecture™, OMG MDA™ and the XMI Logo™ are trademarks of the Object Management Group. All other products or company names mentioned are used for identification purposes only, and may be trademarks of their respective owners.

COMPLIANCE

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

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

ISSUE REPORTING

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

Table of Contents

1 Scope

This specification defines a Platform Independent Model (PIM) for Product Lifecycle Management Services. Its informational model is derived from the ISO 10303-214 STEP model by an EXPRESS-X mapping specification and a EXPRESS-to-XMI mapping process. The functional model is derived from the OMG PDM Enablers V1.3 and to fulfill requirements of the PLM Services 1.0 RFP.

The specification defines a Platform Specific Model (PSM) applicable to the Web Services implementation defined by a WSDL specification, with a SOAP Binding, and a XML Schema specification.

2 Conformance

An implementation compliant to the XML Schema and Web Services PSM described in this specification shall be capable to deliver and to consume valid XML documents with respect to the XML Schema defined in Chapter 9.3

An implementation compliant to the XML Schema and Web Services PSM described in this specification shall support at least one of the Queries Conformance Points defined below.

A Queries Conformance Point consists of a set of specializations of the type `Query`. This specification defines four Queries Conformance Points:

- the Generic Queries Conformance Point (see Chapter 8.12),
- the XPath Queries Conformance Point (see Chapter 8.13),
- the Specific Queries Conformance Point (see Chapter 8.14), and
- the PDTnet Queries Conformance Point (see Chapter 8.15).

An implementation shall define the Queries Conformance Points it is realizing.

3 Normative References

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

- UML Specification
- XMI Specification
- MOF 2.0 Specification
- ISO 10303-11:1994 Description methods: The EXPRESS language reference manual
- ISO 10303-14:2001 Description methods: The EXPRESS-X language reference manual

- ISO CD 10303-25:2003 Implementation methods: EXPRESS to UML mapping
- ISO TS 10303-28:2002 XML representation for EXPRESS-driven data
- ISO 10303-203:2000 Configuration-controlled mechanical design
- ISO 10303-214:2000 Core data for automotive mechanical design process
- ISO 10303-232:2001 Technical Data Package
- ISO/IEC 10746: Reference Model for Object Distributed Computing (RM/ODP)

4 Terms and Definitions

None.

5 Symbols

None.

6 Additional Information

6.1 Changes to Adopted OMG Specifications

This specification completely replaces the PDM Enablers Version 1.3. It is recommended that “PDM Enablers Version 1.3” is retired as an adopted technology because of lack of vendor and user interest.

6.2 How to Read this Specification

The rest of this document contains the technical content of this specification.

Although the chapters are organized in a logical manner and can be read sequentially, this is a reference specification is intended to be read in a non-sequential manner. Consequently, extensive cross-references are provided to facilitate browsing and search.

6.3 Acknowledgements

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

- BMW AG

- Robert Bosch GmbH
- DaimlerChrysler AG
- Keiper GmbH & Co. KG
- PDTec GmbH
- PROSTEP AG
- Scania AB
- T-Systems International GmbH
- Volkswagen AG
- Zentrum für Graphische Datenverarbeitung e.V.
- 88solutions Corp.

7 Informational Viewpoint

7.1 Overview

The Information Model of the proposed PLM Service is based on the STEP PDM Schema [2] and extended by relevant subsets of STEP ISO 10303-214:2000 [8], especially the Configuration Management modelling parts according to CC8.

The selected scope of the Information Model is chosen based on the requirement analysis in the PDTnet project [1]. The use cases identified in this industrial project of European automotive companies are given in brief in Chapter 7.2. The chosen data model is derived from the STEP PDM Schema and the relevant subset of STEP ISO 10303-214:2000. The scope of both contributing sources is described shortly in Chapter 7.3. Both sources share a common AIM level representation for PLM related data models. The chosen data model itself is notated at ARM level according to the STEP nomenclature in EXPRESS language. It defines the scope of the so-called “Equivalence model” and denotes a Platform Independent Model (PIM). The PIM Equivalence model is described in Chapter 7.5. The transformation from the AIM representation into that PIM Equivalence model is described in Chapter 7.4. The mapping specification is formulated in EXPRESS-X (ISO 10303-14, [4]). The relationship of both the AIM and ARM EXPRESS models becomes tractable, executable, and normative.

The objectives for the PIM Equivalence model are twofold:

- To produce the desired reference model suitable for the mapping on to Web services.
- To produce the desired reference model suitable for the mapping on to CORBA PDM Enablers.

The PIM Equivalence model is mapped by STEP ISO 10303-25:2003 into an UML notation. This mapping is described in Chapter 7.6. The resulting UML model represents the PLM reference model in the informational viewpoint and is described in Chapter 7.7. The model is specified in UML 1.3.

7.2 Use cases

This section describes the uses cases that are subject to the PLM services specification. They are categorized according to the requirement analysis resulting from the PDTnet project [1]. They are documented in this section, and may be extended continuously.

The scope of the use cases is defined supporting an online PLM integration scenario which is characterized by a data access on remote systems using internet functionality and technology. This integration does not provide a real online integration, but due to the usage of data streaming techniques and due to the possibility of an immediate reply by a system it comes near to it. It is assumed, that a neutral PLM client provides access to different PLM data providers (these are usually different PLM systems in different companies).

7.2.1 Export of assembly data

Export of product data (meta data and geometry) of assemblies and parts from one partner to another partner via exchange of ENGDAT packages (STEP PDM files, CAD files).

7.2.1.1 Owner of the use case

This use case was defined by Work Group 1 of the PDTnet project.

7.2.1.2 Process purpose

Export of product data which consist of meta data and geometry information of assemblies and its components from one partner to another partner via exchange of ENGDAT formatted packages. The ENGDAT message contains the STEP PDM files and (optionally) the CAD files, in native or neutral format.

7.2.1.3 Partner role descriptions

Table 1 - Roles for export of assembly data

Role name	Role description	Role type
User	Party, that selects and processes product data to be exported.	Person
PLM System	Party, that provides the relevant product data and functionality for product data management. This is usually a company's PLM system, which also can be extended by a tool, that provides extended STEP processor functionality.	System
Data Exchange (DE) Tool	System, that provides communication with a network and functionality to automatically process and pack/unpack file packages (usually ENGDAT-based).	System

7.2.1.4 Process definition

The process steps are:

1. User selects parts/documents/CAD models (using the functionality of the PLM system):
   - Selection of root/top level assembly by assembly (version) number
   - Selection of affected sub-assemblies or parts (could be controlled by a context or specific algorithm)
   - Exclusion of elements from selected set is possible
2. PLM system generates STEP PDM file:
   - Passing assembly structure tree and collecting transformation matrices (if appropriate)
   - Generating STEP PDM file
3. User selects addressee of data (using the DE tool or PLM tool)
4. Download of digital files from PLM system
5. DE Tool generates ENGDAT package including message abstract, STEP PDM file(s) and digital files (CAD/CC2 files, etc.)
6. DE Tool initiates sending of ENGDAT message

The order of the process steps could differ depending on specific user requirements and system scenario. Examples for possible alternative process step orders are:

a): 1. ® 3. ® 4. ® 2. ® 5. ® 6.

b): 3 ® 1. ® 2. ® 4. ® 5. ® 6.

7.2.1.5 Process flow diagram

At the moment no flow diagram exists.

7.2.1.6 Process start and end states

Start state / precondition:

The user knows the assembly/part identifiers and digital file (CAD model) identifiers which are supposed to be exported. At least, the identifier of an assembly, which serves as an entry node, is provided. Additionally, a specific “context handle” is known (project, change status, work order, etc.) is known.

Alternative a): Depending on the user environment also a top-level document ID can be the entry node to a structure.

Alternative b): A top-level part and a specific configuration, which controls the way of the expansion of the tree (sub-parts, kind of documents,...), is known.

End state / post condition E1 (Success):

An ENGDAT package including the STEP PDM file and all selected digital files were successfully sent to the addressee.

End state / post condition E2 (Failure):

DE Tool delivers failure notification/report to user. The reasons can be:

- The STEP processor failed.
- The download of files from the PLM system failed.
- The DE Tool failed.

7.2.1.7 Constraints and assertions

Currently the number of STEP files included in one ENGDAT package is recommended to be restricted to one (VDA). Nevertheless, the intention is to allow more than one STEP file per ENGDAT message. See: Topics under discussion.

7.2.1.8 Relevant data

- Documents/digital files (CAD files)
- Document meta data
- Assembly/part master data
- Assembly structure data (including transformation data)

7.2.1.9 Topics under discussion / Remarks

Currently no engineering change information is included in the STEP PDM file.

Should more than one STEP file be allowed in an ENGDAT message?

7.2.2 Import of assembly data

Import of product data (meta data and geometry) of assemblies and parts from one partner to another via exchange of ENGDAT packages (STEP PDM files, CAD files).

7.2.2.1 Owner of the use case

This use case was defined by Work Group 1 of the PDTnet project.

7.2.2.2 Process purpose

Import of product data which consist of meta data and geometry information of assemblies and its components from one partner to another partner via exchange of ENGDAT formatted packages. The ENGDAT message contains the STEP PDM files and (optionally) the CAD files, in native or neutral format.

7.2.2.3 Partner role descriptions

Table 2 - Roles for import of assembly data

Role name	Role description	Role type
User	Party, that processes product data that has been imported.	Person
PLM System	Party, that provides the relevant product data and functionality for product data management. This is usually a company's PLM system, which also can be extended by a tool, that provides extended STEP processor functionality.	System
Data Exchange (DE) Tool	System, that provides communication with a network and functionality to automatically process and pack/unpack file packages (usually ENGDAT-based).	System

7.2.2.4 Process definition

The process steps are:

1. The DE tool receives an ENGDAT package.
2. The DE tool unpacks the ENGDAT package and stores STEP PDM and CAD files in defined directories (routing).
3. The PLM system evaluates the received STEP PDM file and displays the included data (assembly data, part data, CAD file meta data) and, optionally, generates an analysis report (comparison of existing data and data to be imported). This step can be initiated by the user or by the DE tool (if it is appropriately integrated). ® see Topics under discussion
4. The user manually processes the data and integrates it into the database of the PLM system or, alternatively, no manual interaction is done. ® see Topics under discussion

The DE tool can notify the user of the import process in different ways, e.g. via e-Mail, via PLM system message, a.s.o.

7.2.2.5 Process flow diagram

At the moment no flow diagram exists.

7.2.2.6 Process start and end states

Start state / precondition:

- An ENGDAT package including a STEP PDM file and one or more digital files (CAD files,...) has been received successfully. This means:
- The ENGDAT message contains the expected correct data.
- No inconsistencies between STEP file and references to digital files exist. ® see Topics under discussion
- User selected the mode for import (update, create, etc.)

End state / post condition E1 (Success):

The received PDM data has been successfully integrated in the PLM systems' database.

The received CAD files have been successfully stored in the defined storage areas.

Partial incorporation of data in the PLM system, if the user allowed it.

End state / post condition E2 (Failure):

The process results in a failure message. A failure can occur due to the following reasons:

- The ENGDAT message contains errors and can not be processed correctly.
- The STEP PDM file contains errors and can not be processed correctly (syntactically, semantically, e.g. STEP PDM Schema, etc.).
- The loading process into the PLM system caused errors.

7.2.2.7 Constraints and assertions

At the moment none are defined.

7.2.2.8 Relevant data

- Documents/digital files (CAD files)
- Document meta data
- Assembly/part master data
- Assembly structure data (including transformation data)

7.2.2.9 Topics under discussion

- Who or which system checks, whether the STEP file and the references to digital files included in an ENGDAT message are consistent? Definition of a separate use case?
- On supplier's side: How to handle product/document meta data, that is not managed by the own PLM system (or no PLM system exists) but that has to be re-exported to the OEM?
- Export of version/status information for re-exported assemblies/parts could be discussed. At the moment no version/status information is used.

- The CATIA model name must not be changed by the supplier.
- On supplier's side: How to associate product data identified by OEM identifiers to product data in the own PLM system?
- On supplier's side: How to manage different assembly structures?

7.2.3 Authentication/Start-Up of session

This process allows a user to be authenticated via a PLM client by one or more PLM server(s).

7.2.3.1 Owner of the use case

This use case was defined by the Work Group 2 of the PDTnet project.

7.2.3.2 Process purpose

This process allows a user to be authenticated via the PLM client by one or more PLM server(s).

7.2.3.3 Partner role descriptions

Table 3 - Roles for authentication and start-up of session

Role name	Role description	Role type
User	Party, that wishes to log in a remote PLM server. This could be a person, who interacts with the PLM client, or a system, that triggers the PLM client.	Person / System
PLM client	System, that provides the communication between user and PLM server	System
PLM server	System, that provides the relevant PLM data. This is usually a company's PLM system that acts as a server.	System

7.2.3.4 Process definition

This use case includes the initiation of the connection between PLM client and PLM server, the authentication and personalization of the user. This use case usually initiates all following communication and data transfer between a user, using the PLM client, and a PLM server (also called “site”).

Two alternative authentication processes are possible, which can also be combined:

1. The first attempt to access a remote PLM server will automatically start the authentication process.
2. The user explicitly starts a login procedure to authenticate in one or more PLM server(s) in the beginning of a session.

The following accesses to specific PDM data will be validated within the use case “Authorization.”

7.2.3.5 Process flow diagram

At the moment no flow diagram exists.

7.2.3.6 Process start and end states

Start state S1:

- The user owns a user name and a password valid for a certain PLM server (site).
- The client provides the necessary site information for the network connection.
- The user knows a valid development project to be authorized to access product data on the PLM server.
- The PLM server provides an authentication service based on user, password and session.

End state E1 (Success):

- The user is successfully logged in and, optionally, the PLM server returns a session id.

End state E2 (Failure):

- The process results in a failure message. A failure can occur due to the following reasons:
- The user is not allowed to access the PLM server (return message: “Permission denied”).
- The PLM server itself is not available.

7.2.3.7 Constraints and assertions

A development project defines a project in which persons work together on a certain set of product data. A development project can be a car/vehicle project, a module development project, etc.

7.2.3.8 Relevant data

User name, password, development project, site information (PLM server system), optional: session id.

7.2.4 Authorization

This process validates the access rights of a specific user (designer, group, department, company) to access specific product data on a PLM server.

7.2.4.1 Owner of the use case

This use case was defined by the Work Group 1 of the PDTnet project.

7.2.4.2 Process purpose

This process validates the access rights of a specific user (designer, group, department, company) to access specific product data on a PLM server.

7.2.4.3 Partner role descriptions

Table 4 -Roles for authorization

Role name	Role description	Role type
User	Party, that wishes to access PDM data on a remote PLM server. This could be a person, who interacts with the PLM client, or a system, that triggers the PLM client.	Person / System
PLM client	System, that provides the communication between user and PLM server	System
PLM server	System, that provides the relevant PDM data. This is usually a company's PLM system that acts as a server.	System

7.2.4.4 Process definition

This use case describes the authorization process of a user who attempts to request specific product data on a PLM server. It is used by all other use cases (e.g., when extracting product structure trees). The actual process description is dependent on the authorization mechanisms provided by the PLM server.

7.2.4.5 Process flow diagram

At the moment no flow diagram exists.

7.2.4.6 Process start and end states

Start state S1:

- A previous authentication process was successful (e.g. by given session id).
- The PLM server provides an authorization service based on user, password and session related to specific product data elements. Additionally, the association of product data elements to a development project has to be supported.
- Specific product data that is requested by a user.

End state E1 (Success):

- The user is identified to have the appropriate rights to access the requested product data. The calling process is enabled to provide the product data to the user.

End state E2 (Failure):

- The process results in a failure message. A failure can occur due to the following reason:
- The user is not allowed to access the requested product data. Since it could be in-tended to keep the existence of the requested data completely secret, the user should not get the information “Access denied.” Instead, he should get a failure message like “Data not found.”

7.2.4.7 Constraints and assertions

The PLM server provides an authorization service based on user, password and session related to specific product data elements. Additionally, the association of product data elements to a development project has to be supported. The detailed mechanisms of authorizing specific users to access specific product data elements depend on the PLM server's internal authorization features and company-specific customizing.

Specific assertions:

- The PLM server manages the association of user/development project to a specific server-internal role concept.
- The general role “owner” is provided having all rights for the owned data objects
- Defined access rights to all other (not owned) data objects are: View, Download, Write, Create

7.2.4.8 Relevant data

User name, password, development project, optional: session id

- Requested product data

7.2.4.9 Topics under discussion

The topic “Authorization and Network Security” is under discussion and will be documented in a separate specification.

7.2.5 Start node identification

Identify the start node of a product structure to enable browsing in the product structure.

7.2.5.1 Owner of the use case

This use case was defined by the Work Group 2 of the PDTnet project.

7.2.5.2 Process purpose

Identify the start node of a product structure to enable browsing in the product structure.

7.2.5.3 Partner role descriptions

Table 5 - Roles for start node identification

Role name	Role description	Role type
User	Party, that requests PDM data. This could be a per-son, who interacts with the PLM client, or a system, that triggers the PLM client.	Person / System
PLM client	System, that provides the communication between user and PLM server	System
PLM server	System, that provides the relevant PDM data. This is usually a company's PLM system that acts as a server.	System

7.2.5.4 Process definition

This use case defines the process of identifying the start node of a product structure in a PLM server. The end state / post condition of the use case is the precondition for the start of the use cases “Browsing down/up product structure data”.

The process steps are:

1. User enters ID (part number and optionally part version number) or Wild Card ("*" for "all").
2. PLM client submits search request ® Exception: The PLM server does not respond.
3. PLM server receives ID or Wildcard and triggers search in PLM system ® Exception: The connection between PLM client and PLM server is down.
4. PLM system executes query in its database ® Exceptions: Database is not available, no data found, user is not authorized to access the data, etc.
5. PLM server returns start node and list of views.
6. PLM client displays list of start nodes.

7.2.5.5 Process flow diagram

At the moment no flow diagram exists.

7.2.5.6 Process start and end states

Start state / precondition S1:

The user is correctly logged in, connected to the server, positively identified and authorized.

- The service is available.
- The user enters an ID ("Sachnummer" etc.) or wildcard for the structure start node.

End state / post condition E1 (Success):

- List of product structure nodes including their possible views / configurations

End state / post condition E2 (Failure):

- In case of missing authorization: Exception, message: "No items found or access denied."

7.2.5.7 Constraints and assertions

At the moment none are defined.

7.2.5.8 Relevant data

- Product structure data

7.2.5.9 Topics under discussion

The user should be able to enter either internal or external part master ids ("Alias-Query").

7.2.6 Browsing down product structure data

This process allows a user starting with the product structure to get a view on all product structure relevant data including document (structure) data that is relevant for this specific user or a specific project, independently of the provider of the data.

7.2.6.1 Owner of the use case

This use case was defined by the Work Group 2 of the PDTnet project.

7.2.6.2 Process purpose

This process allows a user starting with the product structure to get a view on all product structure relevant data including document (structure) data that is relevant for this specific user or a specific project, independently of the provider of the data.

7.2.6.3 Partner role descriptions

Table 6 -Roles for browsing down product structure data

Role name	Role description	Role type
User	Party, that requests PDM data. This could be a per-son, who interacts with the PLM client, or a system, that triggers the PLM client.	Person / System
PLM client	System, that provides the communication between user and PLM server	System
PLM server	System, that provides the relevant PDM data. This is usually a company's PLM system that acts as a server.	System

7.2.6.4 Process definition

This use case includes the browsing of product structure data down a product structure, ba-sic part classification data and associated document meta data. For browsing up a product structure ("where used" query), a separate use case is defined.

The following requirements are defined:

Multiple views on the product structure have to be supported, e.g. lead view, supplier's assembly structure, spare part structure, second tier supplier's view, etc.

- The relationship between different base classification data has to be handled (customer's and supplier's data).
- The assignment of structure and classification data to documents has to be consistent and browsing documents must result always in displaying identical information.
- The user defines a set of parameters (filter information), that specifies characteristics of the desired structure nodes in detail. Filtering the data will be defined as a separate use case “PLM filter”.
- Browsing in different PLM server systems has to be supported. This means, the change of a server site has to be possible (“Multi-site support”) when the user selects a structure node, which links to a supplied item provided by another PLM server. This enables the user to browse into a sub-structure of the development partner (e.g. OEM user browses into sub-structure of supplier or vice versa) and to see the information consistently in one single structure tree. The concept for this mechanism is the following:
  - Reference tables connecting the OEM part identifiers to the supplier part identifiers (“alias identifiers”) are managed by the PLM servers, containing for each exchange node:
    - Own part id (item_version to be supported)
    - Corresponding alias id on PLM server of partner
    - Unique identifier for partner PLM server site: harmonized organization ID (e.g., “bmw.de”).
- An additional reference table for the association of organization id and URL (server site connection) is provided on the PLM client site.

The process steps are:

1. PLM client sends a query for substructure specified by the user to the PLM server
   a. In case of the structure node being a “supplied item”, i.e., the selected structure node represents an alias identifier:
      •- Client retrieves alias site connection information (URL) from reference table.
      •- Client asks user for password for alias site (only in case of first request to this site).
      •- Client performs Login, Start node query on alias server site using current development project.

Steps repeated by PLM server for each product structure node in the scope of the query:

2. Check authorization regarding requested data ® Exception: Access denied (PLM server).
3. Collect requested data within PLM server.

End of repeated steps.

4. PLM server sends data to PLM client.
5. Display structure and items in PLM client.

7.2.6.5 Process flow diagram.

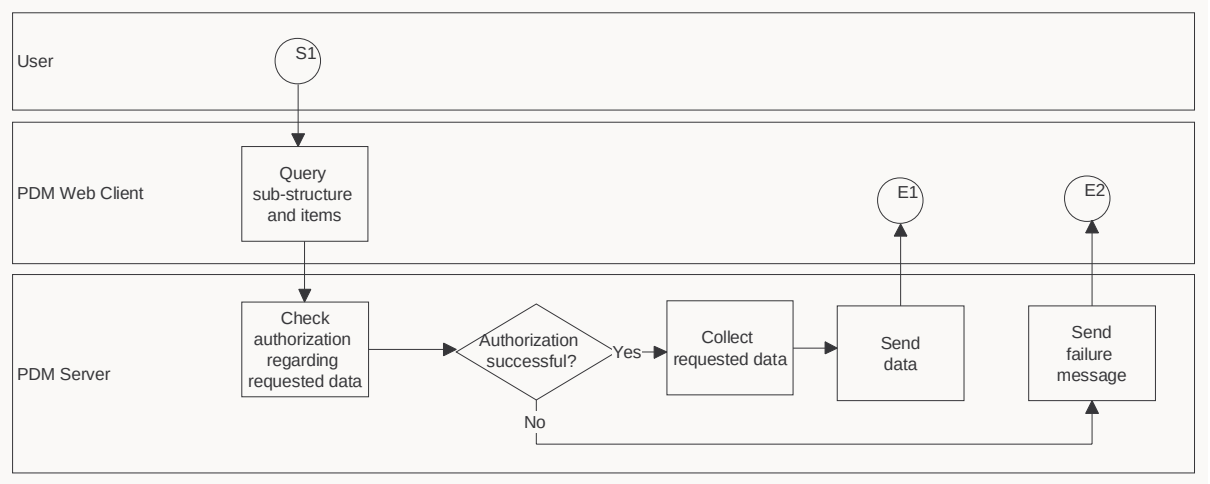


Figure 1 - Process flow diagram for browsing down product structure data

7.2.6.6 Process start and end states

Start state / precondition S1:

- A specific development project is defined, which itself defines certain items of product data (e.g. assemblies, parts, documents), that will be subject to change or creation during the project's life time. These items are identified by identifiers.
- The end state / post condition of use case “Start node identification” or one of the children of the start node.
- The user is correctly logged in and authorized to access the requested information.
- The level of depth down the start node / current node is defined (default: 1 level down the current node).
- The necessary filter information is defined, i.e., the result of the use case “PDM filter” is provided.

End state / post condition E1 (Success):

- The process results in a filtered list or a structure tree containing at least the identifiers of product data items, and additional information about the items (e.g. URLs to documents or additional item information to be downloaded).

End state / post condition E2 (Failure):

- The process results in a failure message. A failure can occur due to the following reasons:
- The user is not authorized to access the data.
- The requested data is not available on the PLM server.

7.2.6.7 Constraints and assertions

If process step 2 leads to an exception regarding a specific structure node, the whole process must continue. The structure node affected by the exception is not included in the collected data set.

7.2.6.8 Relevant data

Product structure data

- Basic part classification data
- Document meta data

7.2.7 Browsing up product structure data

This process allows a user starting with the product structure to get a view on all product structure relevant data including document (structure) data that is relevant for this specific user or a specific project, independently of the provider of the data.

7.2.7.1 Owner of the use case

This use case was defined by the Work Group 1 of the PDTnet project.

7.2.7.2 Process purpose

This process allows a user starting with a specific product structure node to get a view on all relevant product structure nodes in which this specific node is included (“Where used” query). For browsing down a product structure, a separate use case is defined.

7.2.7.3 Partner role descriptions

Table 7 -Roles for browsing up product structure data

Role name	Role description	Role type
User	Party, that requests PDM data. This could be a per-son, who interacts with the PLM client, or a system, that triggers the PLM client.	Person / System
PLM client	System, that provides the communication between user and PLM server.	System
PLM server	System, that provides the relevant PDM data. This is usually a company's PLM system that acts as a server.	System

7.2.7.4 Process definition

This use case includes the browsing of product structure data up a product structure (“where used” query).

The following requirements are defined

Multiple views on the product structure have to be supported, e.g. lead view, supplier's assembly structure, spare part structure, second tier supplier's view, etc.

- The user defines a set of parameters (filter information), that specifies characteristics of the desired structure nodes in detail.

The process steps are:

1. PLM client sends a query for “where used” nodes specified by the user to the PLM server

Steps repeated by PLM server for each product structure node in the scope of the query:

2. Check authorization regarding requested data ® Exception: Access denied (PLM server)
3. Collect requested data within PLM server

End of repeated steps.

4. PLM server sends data to PLM client
5. Display structure and items in PLM client. The way of presentation and needed interaction have to be defined by the application projects.

7.2.7.5 Process flow diagram

At the moment no flow diagram exists.

7.2.7.6 Process start and end states

Start state / precondition S1:

A specific engineering development project is defined, which itself defines certain items of product data (e.g. assemblies, parts, documents), that will be subject to change or creation during the project's life time. These items are identified by identifiers.

- The end state / post condition of use case “Start node identification” or one of the children of the start node, that means item or item_version. Item_version is optionally in order to enable the access of versioning information starting from the part number. Additionally, the single_instance can be identified (maybe by user interaction). This is “nice to have” in general, but required as precondition for a “Search in design space” functionality.
- The user is correctly logged in and authorized to access the requested information.
- The level of depth up the start node / current node is defined and restricted to direct parent or root node (default: direct parent node).

The necessary filter information is defined, i.e., the result of the use case “PLM filter” is provided.

End state / post condition E1 (Success):

- The process results in a filtered list or a structure tree containing only identifiers of product data items (root nodes or direct parent nodes). Only structure nodes which the user is authorized to see are included.

End state / post condition E2 (Failure):

- The process results in a failure message. A failure can occur due to the following reasons:
- The user is not authorized to access the data.
- The requested data is not available on the PLM server.

7.2.7.7 Constraints and assertions

Whenever the PLM System is providing a single_instance concept, the start node used may be the single_instance. If the single_instance is used, there is no necessity for repeating process steps 2 and 3. This statement needs to be evaluated!

- The level of depth up the start node / current node is defined and restricted to direct parent or root node (default: direct parent node).
- Exactly one root node exists for one development project.
- Need of unique filter, that displays the root node only once.
- Within the Client GUI the change to one of the resulting development project (in case of a result list containing root nodes) should be possible.

7.2.7.8 Relevant data

- Product structure data

7.2.8 Download of product data

7.2.8.1 Owner of the use case

This use case was defined by the Work Group 1 of the PDTnet project.

7.2.8.2 Process purpose

This use case has to be described under consideration of two main criteria:

What product data is to be downloaded?

- Download of a single digital file: either geometry (CATIA, STEP) or other binary formats (e.g. TIFF)
- Download of a set of digital files
- Download of structures including optionally digital files
- Download of product meta data of a (structure) node

How is the product data to be downloaded?

- Using online download: via HTTP, only for available documents - no conversion functionality provided
- Using offline download (e.g. via OFTP)

Due to this distinctions the use case “Download of product data” is divided into two use cases, which are described in the following sections 7.2.9 and 7.2.10.

7.2.9 Download meta data including structures

This use case allows the user to identify meta data including structures that he wants to store in a local file system, or that he wants to import into an own PLM system. The format of the transferred data differs:

Online download: The data is transmitted as an data stream (e.g. SOAP message response for web services based implementation). File representations are not supported in this case.

Offline download: The data is sent as an file within the download package. It can be a STEP AP214 Part21, which is specified in the server configuration and considers requirements at target side.

If the detail level covers digital documents the download of these files will be initiated. The download of existing Part 21 files is not covered by this use case either. For this, see use case “Download of a single digital file”. If the data is sent offline, the files may be added to the download package, which is specified in the server configuration and considers requirements at target side.

This functionality covers the access of multiple PLM server Interfaces. For this, two possibilities exist:

1. The user has access to the PDM data of his direct (!) partners. This is covered by the use cases.
2. All other alternate possibilities are managed by the PLM server interface (e.g. data in a 2nd-tier supplier's PLM system).

7.2.9.1 Partner role descriptions

Table 8 - Roles for downloading meta data including structures

Role name	Role description	Role type
User	Party, that requests PDM data. This could be a per-son, who interacts with the PLM client, or a system, that triggers the PLM client.	Person / System
PLM client	System, that provides the communication between user and PLM server.	System
PLM server	System, that provides the relevant PDM data. This is usually a company's PLM system that acts as a server.	System

7.2.9.2 Non-functional requirements

The following requirements with respect to the design of the PLM client GUI are defined:

- The level of detail (“configuration”) can be defined depending on the application project. The technology for defining this configuration is not defined yet.
- The approval status of the relevant data has to be managed by the PLM server interface via authentication and authorization use cases.
- The user is not able to exclude single objects that belong to the tree defined by the start node.
- An additional use case is needed: “PLM Filter”. This use case enables the user to define some special properties that restrict the following amount of managed data.

7.2.9.3 Process definition

The standard process consists of the following steps (the steps directly refer to elements of the user interface of the PLM client):

1. Using the context menu (“right mouse click”) for starting the use case. The user may use this menu only for items and documents in order to be STEP compliant in any cases.
2. By identifying the menu button “download of meta data” a submenu appears that provides all available levels of detail (called “configurations”): download of part master data, download of part and document master data, etc.
3. The user identifies the wished level of detail using the submenu.

4. If the user defined to download structure information the next submenu appears: “Level of structure depth”.
5. In the right frame a list of items appears that were defined for the download process. The user is able to use a scroll bar for browsing through the list.

Optionally: If the download information was not already received by the client the following steps will be performed:

5a. The client is calling the PLM server using a specified query.

5b.. The server generates the product data and sends the resulting data stream to the client interface.

Mandatory:

6. The User starts the download by choosing the Online or Offline Download entry in the right click menu.
7. Online Download:
8. The PLM client sends a query to the PLM server.
9. The PLM server sends the requested data as a data stream to the PLM client.
10. The client takes the data stream and:

10a..calls the “Upload Query” to the second PLM system or

10b. writes an data file.

Offline Download (see also “Initiation of an Offline Download”):

11. The PLM client sends a query to PLM server interface or to the involved EDI-Tool ® In-put to use case “Initiation of an Offline Download”.
12. A Client notification is created by the EDI-Tool.

7.2.9.4 Process flow diagram

At the moment no flow diagram exists.

7.2.9.5 Process start and end states

Start state S1:

- Successful results of Authorization and Browsing use cases.

End state E1 (Success):

- Offline Download: A notification of an additional exchange process is provided (e.g. “Off-line transfer is running”).
- Online Download: A notification for the User, if the download is finished (with success or not).
- The selected meta data including structures is stored in a data file on a local computer (file system), or generated as data stream as input for the Upload use case.

End state E2 (Failure):

- The process results in a failure message. A failure can occur due to the following reasons:

- The user is not authorized to access the PLM server.
- The PLM server interface detected a problem.
- The user is not authorized to download the requested data.
- The PLM server itself is not available.
- Offline Download: Triggering the EDI-Tool failed.
- Online Download: Not sufficient disc space for storing the file.

7.2.9.6 Relevant data

- All product data (part master, document master etc.)

2.2.9.7. Topics under discussion / Remarks

- This download use case ends by creating a data file or a data stream. This data can be re-used by Upload Use Cases.
- Definition of “configurations”: Should they be based on transformation rules?

7.2.10 Download a single digital file

This process allows a user to download a single specific digital file (geometry file, TIFF, etc.) from a remote PLM server to a local storage. The download also includes the viewing of digital files, as far as a viewing tool is automatically started on the user side after the download process has finished. This process is called “simple viewing”.

7.2.10.1 Partner role descriptions

Table 9 - Roles for downloading a single digital file

Role name	Role description	Role type
User	Party, that requests PDM data. This could be a per-son, who interacts with the PLM client, or a system, that triggers the PLM client.	Person / System
PLM client	System, that provides the communication between user and PLM server.	System
PLM server	System, that provides the relevant PDM data. This is usually a company's PLM system that acts as a server.	System

7.2.10.2 Process definition

This use case includes the identification of a single digital file to be downloaded, the start, the monitoring of the progress and the check of the success of the data transport from a PLM server to a local storage.

The process steps are as follows:

1. The user identifies the digital file to be downloaded from the PLM server.
2. The User starts the download by choosing the Online or Offline Download entry in the right click menu.

Online Download:

3. The PLM client sends a query to the PLM server.
4. The PLM server sends the requested digital file data to the PLM client.
5. The PLM client receives the digital file and displays it directly, opens an external application to display it or let the user store it in the local file system.
6. A notification is sent to the User (in case of success and in case of failure).

Offline Download (see also “Initiation of an Offline Download”):

7. The PLM client sends a query to PLM server interface or to the involved EDI-Tool ® In-put to use case “Initiation of an Offline Download”.
8. For the file export from the PDM Vault a copy of the document should be created, no file locking mechanism (for parallel use by other users) should be implemented. The export could be triggered by the PLM server or by the EDI-Tool.
9. A Client notification is created by the EDI-Tool.

7.2.10.3 Process flow diagram

At the moment no flow diagram exists.

7.2.10.4 Process start and end states

Start state S1:

- The user has been successfully authenticated.
- The user is authorized to know that the digital file exists.
- The user has got a list or a structure tree containing at least the identifier of the digital file and an appropriate URL.
- The kind of the access (viewing, changing) is specified. Currently only viewing functionality is considered.
- The final trigger is the selection in the context sensitive menu (“Download selected file online/offline”) that belongs to a selected single digital file.

End state E1 (Success):

- Offline Download: A notification of an additional exchange process is provided (e.g. “Offline transfer is running”).
- Online Download: A notification for the User, if the download is finished (with success or not).
- The digital file, that has been specified by the user for download, is opened and displayed or stored on the local storage.

End state E2 (Failure):

- The process results in a failure message. A failure can occur due to the following reasons:
- The user is not authorized to access the PLM server.
- The user is not authorized to download the digital file.
- The requested digital file is not available on the PLM server.
- The PLM server itself is not available.

- Offline Download: Triggering the EDI-Tool failed.
- Export (checkout) functionality failed (digital file doesn't exist, the file is already used by an other user).
- Online Download: Not sufficient disc space for storing the files.

7.2.10.5 Constraints and assertions

- The downloaded file is always not compressed if it is sent online. Then the file can be opened directly and maybe viewed using a client plug in or an external application. Compression is only allowed if an offline transfer process implies a package mechanism
- The file name is generated by server/system specific rules.

7.2.10.6 Relevant data

- Document meta data
- Document data (digital file)

7.2.11 Generic object query

This use case allows a user to generically access objects (e.g. items, documents) as result of a specified filter condition. Feasible filter parameters and the functionality for the collection and provision of these objects have to be provided by the PLM server. Therefore, this generic use case can be specialized to further detailed use cases. Examples for detailed use cases are:

- Find all parts contained in a design space by providing bounding box parameters.
- Find heat sensitive parts by providing temperature parameters.

7.2.11.1 Owner of the use case

This use case was defined by the Work Group 2 of the PDTnet project.

7.2.11.2 Process purpose

This use case allows a user to generically access objects (items, documents) as result of a specified filter condition. Feasible filter parameters and the functionality for the collection and provision of these objects have to be provided by the PLM server. Therefore, this generic use case can be specialized to further detailed use cases. Examples for detailed use cases are:

- Find all parts contained in a design space by providing bounding box parameters.
- Find heat sensitive parts by providing temperature parameters.

7.2.11.3 Partner role descriptions

Table 10 - Roles for generic object query

Role name	Role description	Role type
User	Party, that wishes to request information This could be a person, who interacts with the PLM client, or a system, that triggers the PLM client.	Person / System
PLM client	System, that provides the communication between user and PLM server.	System
PLM server	System, that provides the relevant PDM data. This is usually a company's PLM system that acts as a server.	System

7.2.11.4 Process definition

The process steps are:

1. User chooses the intended (and provided) functionality (specialized query).
2. User defines a development project or uses the existing one.
3. PLM client displays the parameter names, that have to be provided to filter out the correct data within the PLM server, according to the chosen functionality (see 1.).
4. User provides required parameter values (objects properties, bounding box information, etc.) and initiates query to PLM server interface (single PLM Interface).
5. PLM System is processing the query that results in an object list.
6. Object list is displayed within the PLM client.

7.2.11.5 Process flow diagram

The main mechanism for “Generic object query” is shown in the following diagram. For more details see the specialized use cases.

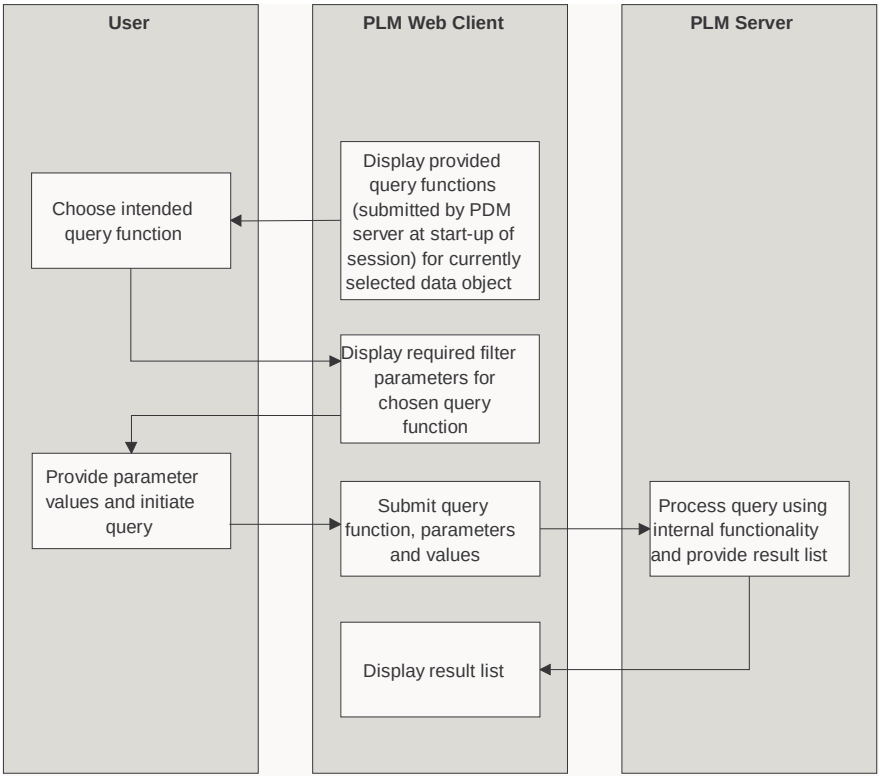


Figure 2 - Process flow diagram for generic object query

7.2.11.6 Process start and end states

Start state S1:

- The authentication and authorization of the user was successful.
- A valid development project is existing.
- The available specialized types of object queries related to specific objects have been previously submitted by the PLM server (see use case “Start-up of session”).

End state E1 (Success):

- List of objects that were requested according to the specialized query and filter parameters. Example for specialized query “Search in design space”: All parts contained in the defined design space as a list of items.

End state E2 (Failure):

- The process results in a failure message. A failure can occur due to the following reasons:
  - No development project defined.
  - The user is not authorized to access the data (see also use case “Authorization”).
  - The requested data is not available on the PLM server.
  - ·Functionality is not supported for this object type.

7.2.11.7 Constraints and assertions

Only one single PLM server is accessed. A generic object query that is sent simultaneously to more than one PLM server is not supported.

7.2.11.8 Relevant data

Product structure data

- Basic part classification data
- Document meta data
- Document data

7.2.11.9 Diagrams

UML diagrams are provided for the specialized use cases.

7.2.12 Search in design space

This use case is a specialization of the use case “Generic object query”.

7.2.12.1 Process purpose

Purpose of the “Search in design space” process is to query all parts which are located in the neighborhood of a given part. This use case allows a designer at the supplier site to search for parts which are positioned in a certain area around a specified part. The calculation of the neighborhood relation of parts will be done by using the “bounding boxes” of the parts. The user should be able to “blow up” the bounding box around a part in order to get all parts in a certain distance of the given part.

7.2.12.2 Partner/actor role descriptions

Table 11 - Roles for search in design space

Role name	Role description	Role type
User	Party, that requests PDM data. This could be a per-son, who interacts with the PLM client, or a system, that triggers the PLM client.	Person / System
PLM client	System, that provides the communication between user and PLM server.	System
PLM server	System, that provides the relevant PDM data. This is usually a company's PLM system that acts as a server.	System

7.2.12.3 Process definition

The process could be seen as a query in which the query parameters do not exist as discrete PDM data in the PLM system. Actually, the criteria for the evaluation of the result set is the geometrical relation between the given part and all other parts in a given assembly. For ex-ample, the designer has to modify the design of the oil pump of a car. He needs to know

which parts are located near to the pump to be able to check whether the modified pump fits into the space left for this device. With the search described here, he can find those parts easily. This use case would probably only be relevant for the OEM side of the PDTnet project.

The following requirements are defined:

- The parts found during the search are displayed in form of a “virtual container” which contains all parts meeting the design space criteria. The virtual container is an assembly which is only created temporarily and which does not represents any form of a real assembly. It is only meant as a set of objects and therefore can be displayed as an assembly with one and only one level.
- It should be possible to combine different search criteria (search in design space, search by defining PDM data filters). For example, all temperature sensitive parts in a certain distance of a hot part have to be found by the query.
- In order to ensure the clearness of visualization, the formerly displayed structures should be made available by means of a “Pull down list” or by “Tabs” which allow to go directly to the assigned structure display.
- The resulting set of items should allow to perform a download (online or offline) on certain items selectable by the user
- The user should optionally be able to define an assembly (“Start node”) in which the parts to find are contained. For example, all parts in an combustion engine should be found.
- Another option is to enter the depth of search, the levels of deepness in an assembly.

7.2.12.4 Process flow diagram

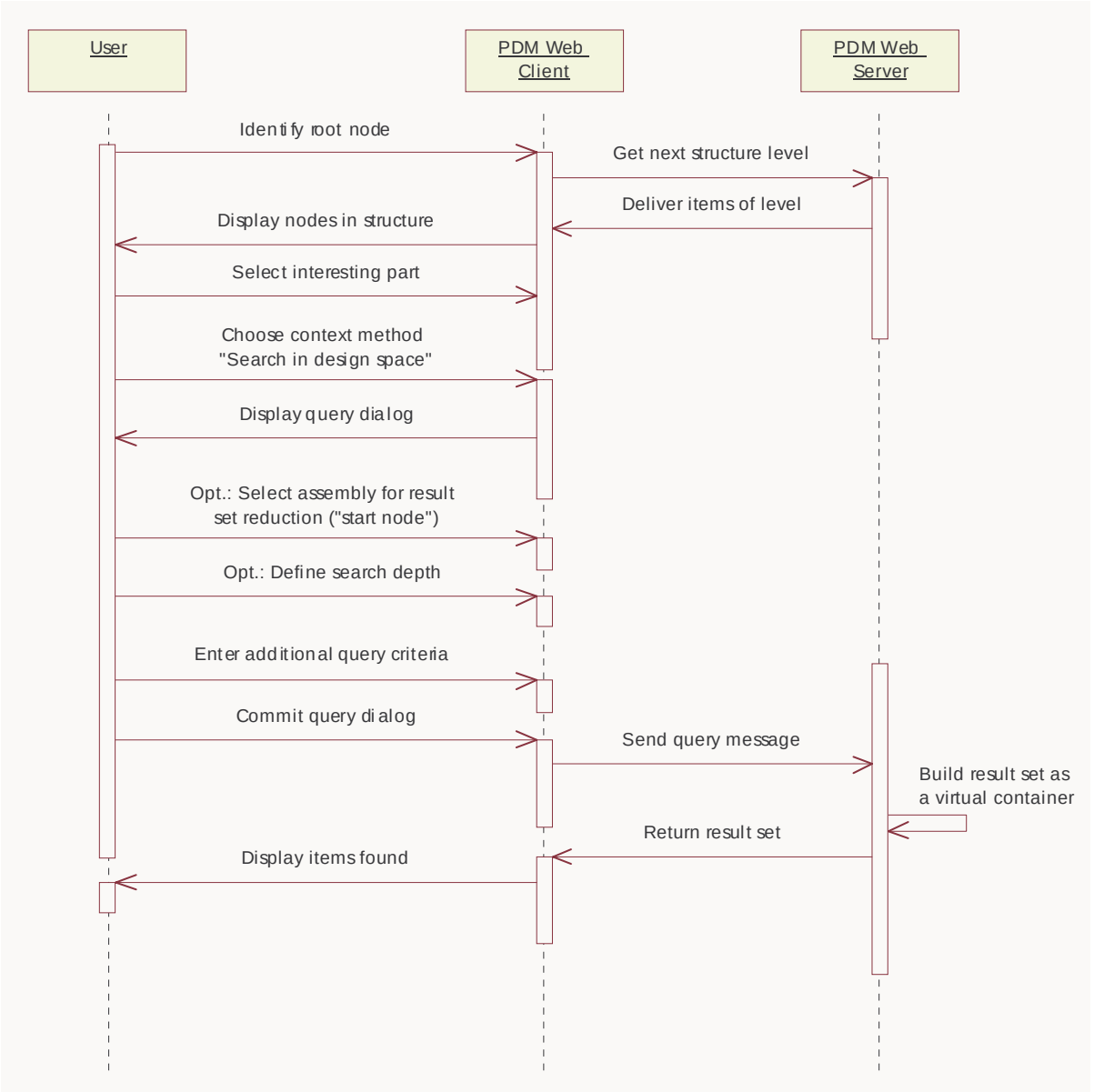


Figure 3 - Process flow diagram for search in design space

7.2.12.5 Process start and end states

Start state / precondition S1:

A specific engineering project is defined, which itself defines certain items of product data (e.g. assemblies, parts, documents), that will be subject to change or creation during the project's life time. These items are identified by identifiers.

- The end state / post condition of use case “Start node identification” or one of the children of the start node, that means an item
- The user is correctly logged in and authorized to access the requested information.

The necessary filter information is defined (see use case “Generic object query”).

End state / post condition E1 (Success):

- The process results in a virtual container (see 2.2.12) containing all the accessible parts found during the query. The number of parts found is displayed.
- The virtual container contains the transformation matrices of the parts in relation to the car origin
- If no parts or accessible parts were found, an empty virtual container is presented. The number of parts found is displayed, in this case it is 0.

End state / post condition E2 (Failure):

- The process results in a failure message. A failure can occur due to the following reasons:
- The selected part contains no geometry. Therefore, there is no possibility to find any parts in the neighborhood of the part. This should be reported by the message “Part contains no geometry.”

7.2.12.6 Constraints and assertions

- The selected part has to contain any geometry as a base for the query.

7.2.12.7 Relevant data

- Product structure data

7.2.13 Upload of product data

7.2.13.1 Process purpose

This use case allows a user to upload specific product data that was created or changed on a local storage to a remote PLM server.

This use case corresponds mainly to use case “Download of product data”. Additionally, it requires two functions:

- Identification of correct structure nodes for the integration of uploaded data.
- Creation/change of structures and/or structure nodes, if appropriate.

This functionality is closely related to the underlying access authorization concept. Due to the variety of PLM system-specific access authorization architectures this topic is closely de-pending on the PLM system functionality and/or company specific PLM system usage restrictions.

7.2.13.2 Owner of the use case

This use case was defined by the Work Group 2 of the PDTnet project.

7.2.14 Upload a single digital file (simple user interaction)

7.2.14.1 Process purpose

This process allows a user to upload a single file which were created or changed on a local storage to a remote PLM server.

7.2.14.2 Process definition

This use case corresponds mainly to use case “Download of a single digital file” (see section 2.2.10). Additionally, it requires two functions:

Identification of the correct structure node for the integration of uploaded data.

- Creation/change of structures and/or structure nodes, if appropriate. This functionality is closely related to the underlying access authorization concept. Due to the variety of PLM system-specific access authorization architectures this topic is closely depending on the PLM system functionality.

7.2.14.3 Process flow diagram

At the moment no flow diagram exists.

7.2.14.4 Partner role descriptions

Table 12 -Roles for uploading of a single digital file (simple user interaction)

Role name	Role description	Role type
User	Party, that wishes to store PDM data on a remote PLM server. This could be a person, who interacts with the PLM client, or a system, that triggers the PLM client.	Person / System
PLM client	System, that provides the communication between user and PLM server	System
PLM server	System, that provides the relevant PDM data. This is usually a company's PLM system that acts as a server. The PLM system can be extended by a Web Server to build the complete PLM server.	System

7.2.14.5 Process start and end states

Start state S1:

- The user has got a single file stored on his local file system to be uploaded.
- The user knows the correct structure node in the database of the PLM server for the integration of the data.

End state E1 (Success):

- Offline Upload: A notification of an additional exchange process is provided (e.g. “Offline transfer is running”).
- Online Upload: A notification for the User, if the upload is finished (with success or not). The displayed target structure is refreshed on the screen.

- The file, that had been specified by the user for upload, is stored on the remote PLM server and attached to the target structure. Maybe some new structure node were created to attach the file to.

End state E2 (Failure):

- The process results in a failure message. A failure can occur due to the following reasons:
- The user is not authorized to access the PLM server.
- The user is not authorized to upload the digital file.
- The user is not authorized to create needed structure nodes.
- The server can't create needed structure nodes with default values.
- The specified data could not be integrated in the database of the PLM server (e.g. the correct structure node for data integration could not be identified).
- The PLM server itself is not available.
- Offline Upload: Triggering the EDI-Tool failed.

7.2.14.6 Constraints and assertions

The uploaded file is always not compressed. Compression is only allowed if an offline transfer process implies a packaging mechanism.

The target element to assign an uploaded file to can be of type “Item_version” or “Document_version”. In case of a “Document_version” the file can be assigned directly. If an “Item_version” is selected, the server has to create a document with default values to assign the file to. If any creation is not possible, the action fails and the user is notified.

Any directives/parameters for the upload process are stored at server side.

7.2.14.7 Relevant data

Product structure data

- Document meta data
- Document data (digital file)

7.2.15 Upload meta data including structures

7.2.15.1 Process purpose

This process allows a user to upload meta data including structures to a remote PLM server. This data was created or changed on a local storage or is the result of a download process.

7.2.15.2 Process definition

This use case corresponds mainly to use case “Download of meta data including structures” (see section 2.2.9). Additionally, it requires two functions:

- Identification of correct structure nodes for the integration of uploaded data.

- Creation/change of structures and/or structure nodes, if appropriate. This functionality is closely related to the underlying access authorization concept. Due to the variety of PLM system-specific access authorization architectures this topic is closely depending on the PLM system functionality.

7.2.15.3 Process flow diagram

At the moment no flow diagram exists.

7.2.15.4 Partner role descriptions

Table 13 -Roles for uploading meta data including structures

Role name	Role description	Role type
User	Party, that wishes to store PDM data on a remote PLM server. This could be a person, who interacts with the PLM client, or a system, that triggers the PLM client.	Person / System
PLM client	System, that provides the communication between user and PLM server.	System
PLM server	System, that provides the relevant PDM data. This is usually a company's PLM system that acts as a server.	System

7.2.15.5 Process start and end states

Start state S1:

- The user has got data stored on his local file system or stored temporarily as a result of a download process.
- The user knows the correct structure nodes in the database of the PLM server for the integration of the data.

End state E1 (Success):

- Offline Upload: A notification of an additional exchange process is provided (e.g. “Offline transfer is running”).
- Online Upload: A notification for the User, if the upload is finished (with success or not). The displayed target structure is refreshed on the screen.
- The data, that had been specified by the user for upload, is stored on the remote PLM server and integrated into the target structure.

End state E2 (Failure):

- The process results in a failure message. A failure can occur due to the following reasons:
  - The user is not authorized to access the PLM server.
  - The user is not authorized to upload the data.
  - The specified data could not be integrated in the database of the PLM server (e.g. the correct structure nodes for data integration could not be identified).
  - The PLM server itself is not available.
  - Offline Upload: Triggering the EDI-Tool failed.

7.2.15.6 Constraints and assertions

The new structure is sent as message set to the server. The data can be assigned to one or more target elements. If the whole uploaded structure should be assigned to one single element, this will be selected within a message parameter. If there are more complex relations between the new and target elements, the message set also contains the target elements and the relationships to them. In case of an offline transfer, the message set can be replaced by a STEP Part 21 file, which is specified in the server configuration and considers requirements at target side. In case of an online transfer, STEP Part 21 is not supported.

Referenced files has to be uploaded separately using the use cases “Upload a single digital file” or “Upload a set of digital files.” If the data is sent offline, the files may be added to the upload package, which is specified in the server configuration and considers requirements at target side.

Any directives/parameters for the upload process are stored at server side.

7.2.15.7 Relevant data

Product structure data

- Basic part classification data
- Document meta data
- Document data

7.2.16 Change notification

7.2.16.1 Process purpose

The designer of a part needs notification when a change to a part happens which affects one of the parts he is responsible for. This could take place when a part in the neighborhood of a given part is changed in its dimensions or properties or when a part in an assembly is moved to another place than before. The user specifies the parts on which he wants to be notified by using the functionality of subscribing specified in use case “Change content of subscription list”.

7.2.16.2 Partner/actor role descriptions

Table 14 -Roles for change notification

Role name	Role description	Role type
User	Party, that requests PDM data. This could be a per-son, who interacts with the PLM client, or a system, that triggers the PLM client.	Person / System
E-Mail Client	System, that is able to maintain the user's e-mail.	System
PLM server	System, that provides the relevant PDM data. This is usually a company's PLM system that acts as a server.	System

7.2.16.3 Process definition

Target of the process is the evaluation of objects being changed since the last visit of the user to this object. When a modification of those object is being detected, an appropriate message has to be delivered to the user. Objects could be parts (and part versions), documents (and document versions) or models.

Changes to report could be:

- Creation of a new version of an object
- Change of the release status of an object
- Objects are deleted
- Geometry has changed
- Properties have changed

The following requirements are defined:

- Two possibilities of detecting changes on the server side are conceivable. Which of them is used is depending on the PLM server implementation:
- Whenever an object linked to anybody's subscription list is changed, an e-mail is sent to the user(s).
- In certain periods of time, the subscription lists of all users are checked against the objects they include. When a modification of a certain object is detected, an e-mail is sent to the user.
- The frequency and content of e-mail notifications (confidential data must not be included!) are defined server-specifically.

7.2.16.4 Process flow diagram

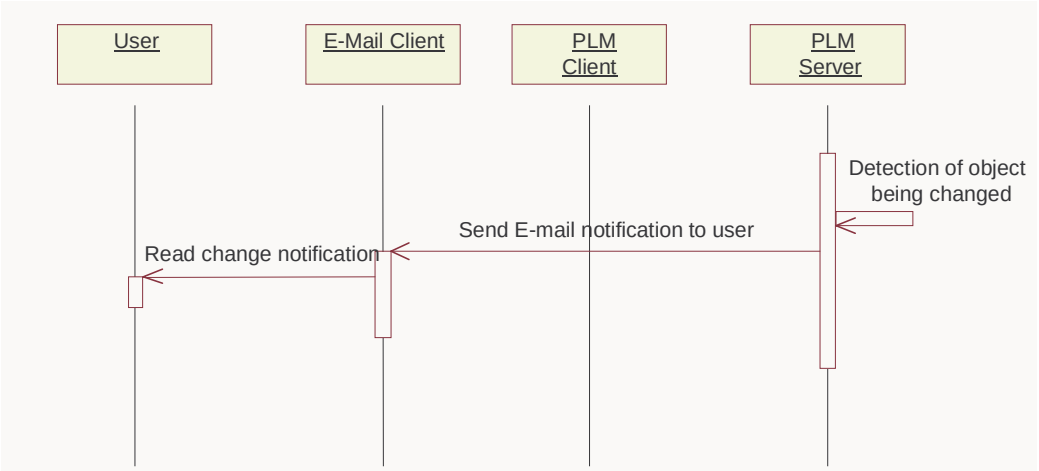


Figure 4 - Process flow diagram for change notification

7.2.16.5 Process start and end states

Start states / preconditions S1 and S2:

- User has access to his e-mail client.

End state / post condition E1 and E2 (Success):

- An e-mail notification about changes to one of his objects collected in the clipboard is sent to the user.

7.2.16.6 Constraints and assertions

Currently none are defined.

7.2.16.7 Relevant data

- Product meta data

7.2.17 Display content of subscription list and confirm changes

7.2.17.1 Process purpose

To get an overview about objects being changed on the PLM server, the user should be able to display the contents of his subscription list in which he collects all the objects to track. The changed objects should be displayed in an emphasized style to show the status of being changed.

The current content of the subscription list including notifications of changes can be re-quested by the PLM client:

- when logging in at the server
- when interactively initiated by the PLM client user.

7.2.17.2 Partner/actor role descriptions

Table 15 - Roles for displaying content of subscription list and confirm changes

Role name	Role description	Role type
User	Party, that requests PDM data. This could be a person, who interacts with the PLM client, or a system, that triggers the PLM client.	Person / System
PLM client	System, that provides the communication between user and PLM server.	System
PLM server	System, that provides the relevant PDM data. This is usually a company's PLM system that acts as a server.	System

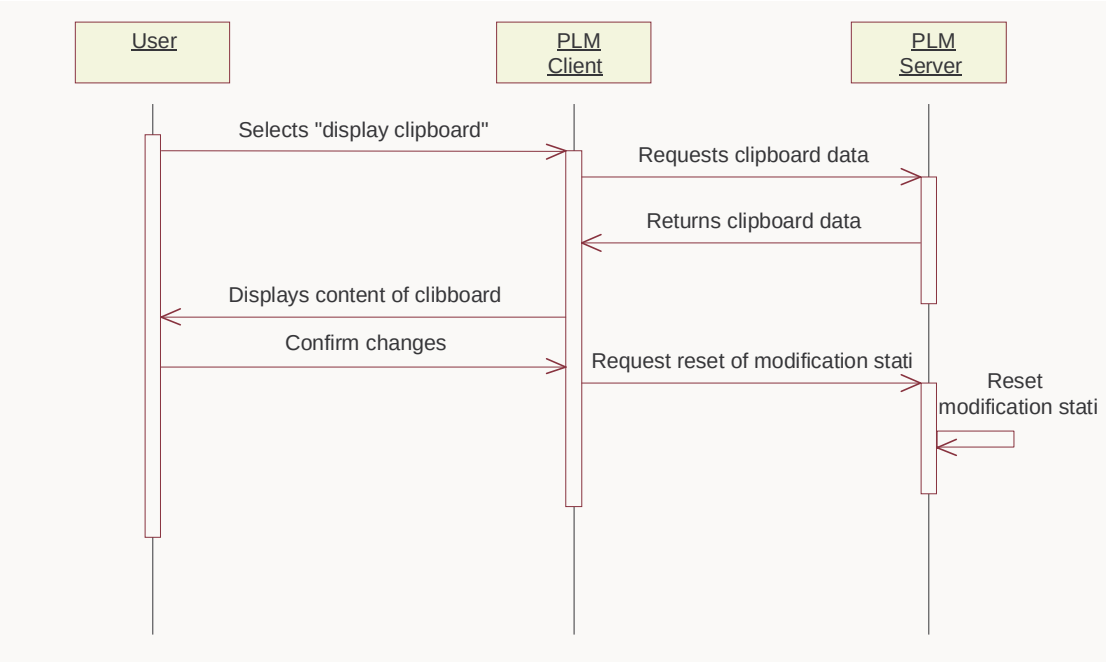
7.2.17.3 Process definition

Target of the process is the evaluation of objects being changed on the PLM server since the last visit of the user and the notification of the user by displaying the content of the subscription list. When a modification of those objects is being detected, the objects are marked as changed in the subscription list and the reasons of the changes are displayed.

The following requirements are defined:

- The user controls the start of the evaluation process via the client. The results of the evaluation process are displayed directly in the client.
- The change notification data is transferred by the PLM server using the data constructs provided by AP214 (work management information). An additional transfer of change management/notification documents (like PDF files) is currently not needed.
- The user must be able to define and to modify the content of his subscription list (see use case “Change content of subscription list”).
- The subscription list should be represented as a separate folder within the PLM client GUI.

7.2.17.4 Process flow diagram



7.2.17.5 Process start and end states

Start state / precondition S1:

A specific engineering project is defined, which itself defines certain items of product data (e.g. assemblies, parts, documents), that will be subject to change or creation during the project life time. These items are identified by identifiers.

- The user is correctly logged in and authorized to access the requested information.

End state / post condition E1 (Success):

- The process results in a virtual container (see use case “Search in design space”) containing all the objects in the subscription list.
- Objects modified since the last look on the subscription list are displayed emphasized. Deleted objects are displayed in a different style.
- After confirmation, the modification status of the objects is reset and in the case of deleted objects in the PLM system, they are also deleted from the subscription list.

7.2.17.6 Constraints and assertions

Currently none are defined.

7.2.17.7 Relevant data

- Product meta data
- Work management data

7.2.18 Change content of subscription list

7.2.18.1 Process purpose

The idea of the subscription list is, that the user needs a sort of folder in which he can collect objects. The purpose of the Subscription lists is to collect objects for which the change notification should be provided. The modification of the objects in this subscription lists is tracked and the user will be notified if such a modification takes place. The user should be able to change the content of his subscription list. The subscription list contains all objects the user wants to be notified when changes are applied to them.

7.2.18.2 Partner/actor role descriptions

Table 16 - Roles for changing content of subscription list

Role name	Role description	Role type
User	Party, that requests PDM data. This could be a per-son, who interacts with the PLM client, or a system, that triggers the PLM client.	Person / System
PLM client	System, that provides the communication between user and PLM server.	System
PLM server	System, that provides the relevant PDM data. This is usually a company's PLM system that acts as a server.	System

7.2.18.3 Process definition

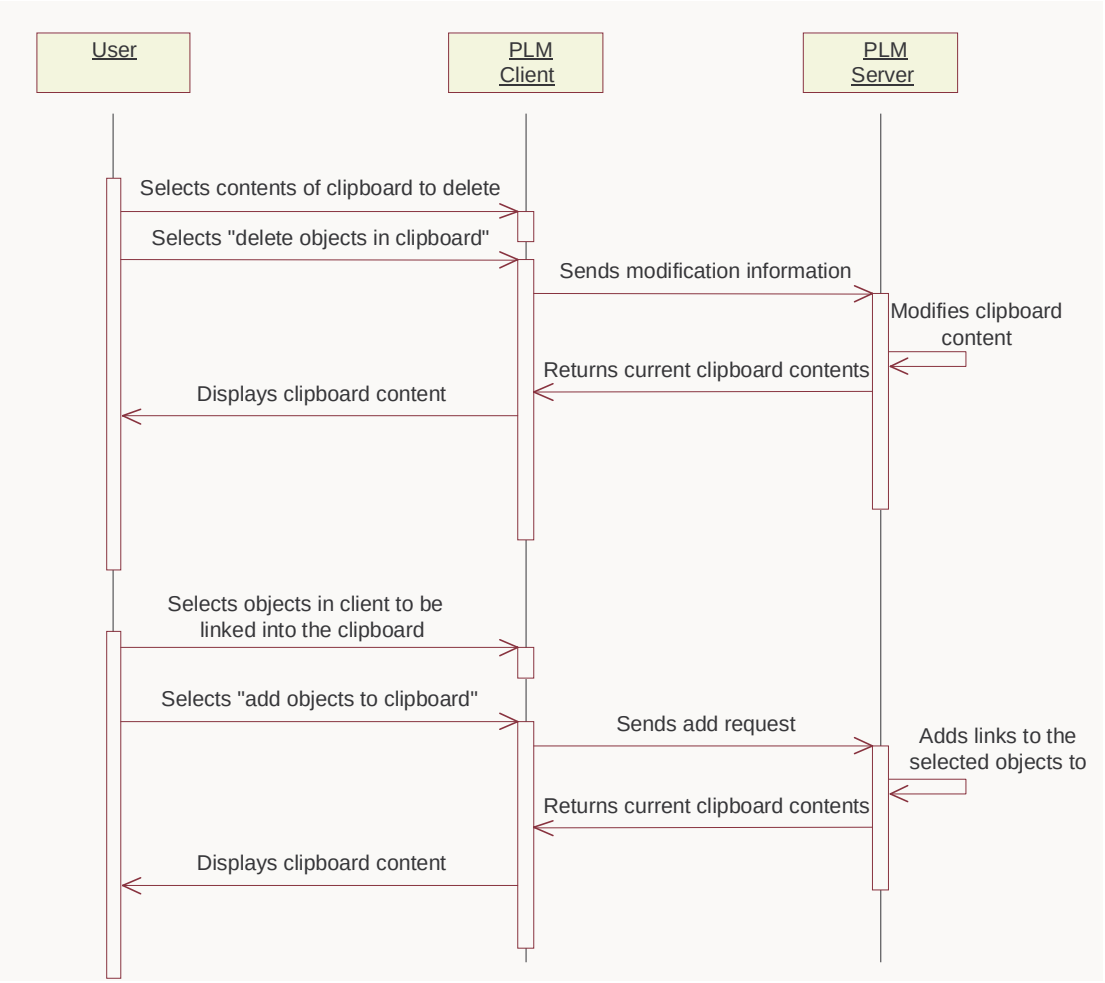
a.. The user selects objects in the subscription list and wants the PLM system to delete the objects from the subscription list.

b.. The user selects objects in the PLM client and wants the PLM system to link those objects into the subscription list.

The following requirements are defined:

- The user has got a subscription list in the PLM system
- For use case a), the content of the subscription list with the objects to delete have to be displayed.0
- For use case b), the objects to add have to be displayed in the client.

7.2.18.4 Process flow diagram



7.2.18.5 Process start and end states

Start state / precondition S1 (use case a):

A specific engineering project is defined, which itself defines certain items of product data (e.g. assemblies, parts, documents), that will be subject to change or creation during the project life time. These items are identified by identifiers.

- The user is correctly logged in and authorized to access the requested information.
- The content of the subscription list is being displayed in the client.

Start state / precondition E2 (use case b):

A specific engineering project is defined, which itself defines certain items of product data (e.g. assemblies, parts, documents), that will be subject to change or creation during the project life time. These items are identified by identifiers.

- The user is correctly logged in and authorized to access the requested information.
- Product data is displayed.

End state / post condition E1 and E2 (Success):

- The process results in an updated view to the subscription list.

7.2.18.6 Constraints and assertions

The user must own a subscription list.

7.2.18.7 Relevant data

Product meta data

7.2.19 Product Class Identification

7.2.19.1 Process purpose

Identification of a top level product_class to enable browsing of an abstract product structure.

7.2.19.2 Partner/actor role descriptions

Table 17 -Roles for product class identification

Role name	Role description	Role type
User	Party, that requests PDM data. This could be a per-son, who interacts with the PLM client, or a system, that triggers the PLM client.	Person / System
PLM client	System, that provides the communication between user and PLM server.	System
PLM server	System, that provides the relevant PDM data. This is usually a company's PLM system that acts as a server.	System

7.2.19.3 Process definition

This use case defines the process of identifying the start node of an abstract product structure in a PLM server. The end state / post condition of the use case is the precondition of the use case “Browsing of an abstract product structure.”

The process steps are:

- The user enters an ID or Wild Card.
- PLM server receives ID or Wild Card and triggers search in PLM System.
- Exception: The PLM server does not respond.

- PLM System executes query in its database

-> Exception: Database is not available, no data found, user is not authorized to access the data, etc.

- PLM server returns a list of product_class and product_component nodes.
- PLM client displays the resulting product_class nodes. If the list has only one member it shall be displayed as the root node of a tree. If the list contains more than one node than the result should be displayed as a list from which the user may select one node that is than displayed as the root node of a tree.

Remark: according to the AP214 CC8 Recommended Practices, each product_class is associated to one instance of product_component (with relation_type='realization') having the same attribute values. From this instance of product_component (not displayed within the client), the abstract product structure may be traversed (ProductStructureQuery).

7.2.19.4 Process flow diagram

At the moment no process flow diagram is provided.

7.2.19.5 Process start and end states

Start state / precondition S1:

- The user is correctly logged in and authorized to access the requested information.
- The service is available.
- The user enters an Id or Wild Card.

End state / post condition E1 (Success):

- The list of resulting nodes is displayed as described above.

End state / post condition E2 (Failure):

- The process results in a failure message.

7.2.19.6 Constraints and assertions

At the moment none are defined.

7.2.19.7 Relevant data

- Product_class information

7.2.20 Browsing of Abstract Product Structures

7.2.20.1 Process purpose

This process allows a user starting with an identified product_class, product_component or alternative_solution to get information on the subcomponents of an abstract product structure (product_component or item_instance).

7.2.20.2 Partner/actor role descriptions

Table 18 - Roles for browsing of abstract product structures

Role name	Role description	Role type
User	Party, that requests PDM data. This could be a per-son, who interacts with the PLM client, or a system, that triggers the PLM client.	Person / System
PLM client	System, that provides the communication between user and PLM server.	System
PLM server	System, that provides the relevant PDM data. This is usually a company's PLM system that acts as a server.	System

7.2.20.3 Process definition

The process steps are:

- The PLM client evaluates if the product structure information is already obtained then it is directly displayed in a table.
- The PLM client sends a query for a substructure of product_class, product_component or alternative_solution specified by the user to the PLM server.
- For each product structure node in the scope of the query the PLM server
- Checks the authorization regarding the requested data

-> Exception: Access denied

- Collects requested data within the PLM server
- PLM server sends data to the PLM client.
- PLM client displays the resulting nodes within the structure. The kind of relationship (e.g product_structure_relationship of kind “decomposition” or “realization”) and child node (product_component or item_instance) should be displayed within the PLM client.

Remark: only one level of the product structure is retrieved at a time.

Remark: only product_structure_relationships from product_component to product_component from alternative_solution to item_instance and from alternative_solution to product_component are supported.

Remark: all the subtypes of item_instance are supported (single, quantified and selected). selected_instance is used in the case of a quantity 'as needed': se-lected_instance.selection_quantity refers to an instance of value_limit with limit=0 and limit_qualifier='minimum'.

Remark: this functionality is also available on item_version nodes if they are handled both as part (for their usage) as well as product_component (having an own abstract product structure). In this case, the function handles the item_version just as if it was a product_component.

7.2.20.4 Process flow diagram

At the moment no flow diagram is provided.

7.2.20.5 Process start and end states

Start state / precondition S1:

- The user is correctly logged in and authorized to access the requested information.
- The service is available.
- The user enters an Id.

End state / post condition E1 (Success):

- The list of resulting of the resulting nodes is displayed as described above.

End state / post condition E2 (Failure):

- The process results in a failure message.

7.2.20.6 Constraints and assertions

At the moment none are defined.

7.2.20.7 Relevant data

- Product_structure_relationships, Product_components, Alternative_solutions, Item_instances

7.2.21 Browsing of Alternative Solutions within an Abstract Product Structure

7.2.21.1 Process purpose

This process allows a user starting with an identified product_component (or alternative_solution) to get information on the (sub-)alternative solutions of an abstract product structure.

7.2.21.2 Partner/actor role descriptions

Table 19 - Roles for browsing of alternate solutions within an abstract product structure

Role name	Role description	Role type
User	Party, that requests PDM data. This could be a per-son, who interacts with the PLM client, or a system, that triggers the PLM client.	Person / System
PLM client	System, that provides the communication between user and PLM server.	System
PLM server	System, that provides the relevant PDM data. This is usually a company's PLM system that acts as a server.	System

7.2.21.3 Process definition

The process steps are:

- The PLM client evaluates if the alternative_solutions are already obtained then it is directly displayed in a table.

- The PLM client sends a query for the alternative solutions of a product_component (or alternative_solution) specified by the user to the PLM server.
- For each alternative solution node in the scope of the query the PLM server
- Checks the authorization regarding the requested data

-> Exception: Access denied

- Collects requested data within the PLM server
- PLM server sends data to the PLM client.
- PLM client displays the resulting nodes within the structure. The kind of child node (alternative_solution, technical_solution, final_solution, supplier_solution) should be displayed within the PLM client.

7.2.21.4 Process flow diagram

At the moment no flow diagram is provided.

7.2.21.5 Process start and end states

Start state / precondition S1:

- The user is correctly logged in and authorized to access the requested information.
- The service is available.
- The user enters an Id.

End state / post condition E1 (Success):

- The list of resulting of the resulting nodes is displayed as described above.

End state / post condition E2 (Failure):

- The process results in a failure message.

7.2.21.6 Constraints and assertions

At the moment none are defined.

7.2.21.7 Relevant data

- Product_structure_relationships, Product_components

7.2.22 Retrieve Configuration Data within an Abstract Product Structure

7.2.22.1 Process purpose

This process allows a user starting with an identified alternative_solution or item_instance to get information on the configuration of an abstract product structure.

7.2.22.2 Partner/actor role descriptions

Table 20 - Roles for retrieving configuration data within an abstract product structure

Role name	Role description	Role type
User	Party, that requests PDM data. This could be a per-son, who interacts with the PLM client, or a system, that triggers the PLM client.	Person / System
PLM client	System, that provides the communication between user and PLM server.	System
PLM server	System, that provides the relevant PDM data. This is usually a company's PLM system that acts as a server.	System

7.2.22.3 Process definition

The process steps are:

- The PLM client evaluates if configuration information is already obtained then it is directly displayed in a table.
- The PLM client sends a query for the configuration[s] of an alternative_solution or item_instance specified by the user to the PLM server.
- For [each] configuration node in the scope of the query the PLM server
- Checks the authorization regarding the requested data

-> Exception: Access denied

- Collects requested data within the PLM server
- PLM server sends data to the PLM client.
- PLM client displays the resulting nodes within the structure. The associated Specification referenced through Configuration and Class_specification_association should be displayed within the PLM client as a property of the configuration.

Remark: currently, configuration may be only displayed on alternative_solution and item_instance, but not on product_component and product_function.

Remark: for complexity reason the specification_expression corresponding to the logical rule stored within the legacy system is mapped to a single string and mapped to a pseudo-Specification.id. This specification is directly referenced by the Class_specification_association. The category of this specification has id=/DUMMY.

Remark: the product_class referenced by the class_specification_association will not be displayed to the PLM client, since it is either derived from the root node of the abstract product structure, or is project independent (for example in the case on configured assembly structures) and would have to be instantiated with a product_class of kind 'enterprise'.

Remark: if the usage of a part or product_component is not configured (i.e. the associated logical rule is empty), this function will give no results.

7.2.22.4 Process flow diagram

At the moment no flow diagram is provided.

7.2.22.5 Process start and end states

Start state / precondition S1:

- The user is correctly logged in and authorized to access the requested information.
- The service is available.
- The user enters an Id.

End state / post condition E1 (Success):

- The list of resulting of the resulting nodes is displayed as described above.

End state / post condition E2 (Failure):

- The process results in a failure message.

7.2.22.6 Constraints and assertions

At the moment none are defined.

7.2.22.7 Relevant data

- Alternative_solution, Item_instance, Configuration, Product_class, Class_specification_association, Specification, Specification_category

7.2.23 Viewing of Change Management Information

7.2.23.1 Process purpose

Browsing through a product structure the user is able to see the assigned change management information.

7.2.23.2 Partner/actor role descriptions

Table 21 - Roles for viewing of change management information

Role name	Role description	Role type
User	Party, that requests PDM data. This could be a per-son, who interacts with the PLM client, or a system, that triggers the PLM client.	Person / System
PLM client	System, that provides the communication between user and PLM server.	System
PLM server	System, that provides the relevant PDM data. This is usually a company's PLM system that acts as a server.	System

7.2.23.3 Process definition

The process steps are:

- The user selects a node (product_class, product_component, item_version) within the PLM client.

- The PLM client evaluates if work management information is already obtained then it is directly displayed in a table.
- If work management information is not obtained the PLM client sends a query for this node to the PLM server.
- PLM System executes query in its database

-> Exception: Database is not available, no data found, user is not authorized to access the data, etc.

- PLM server sends obtained work management data to the PLM client.
- PLM client displays the resulting data in a table.

Remark: according to the CC8 Recommended Practices, the effectivity references an event_reference, which references again an activity. Effectivity_assignment.effective_element and Activity_Element.element both reference the product_class, product_component or item_version node.

Remark: other object nodes are not supported at this time.

7.2.23.4 Process flow diagram

At the moment no flow diagram is provided.

7.2.23.5 Process start and end states

Start state / precondition S1:

- The user is correctly logged in and authorized to access the requested information.
- The service is available.
- The user selects a node of kind product_class, product_component or item_version in the tree view.

End state / post condition E1 (Success):

- The resulting information is displayed as described above.

End state / post condition E2 (Failure):

- The process results in a failure message.

7.2.23.6 Constraints and assertions

At the moment none are defined.

7.2.23.7 Relevant data

- Activity, Activity_element, Effectivity, Effectivity_assignment, Event_reference

7.3 Relevant Subsets of STEP PDM Schema and STEP AP214

The relevant subsets of the STEP PDM Schema and the STEP AP214 are defined by the following functional modules:

- Part Identification,
- Part Structure,

- Document and File Management,
- Shape Definition and Transformation,
- Classification,
- Properties,
- Alias Identification,
- Authorization,
- Configuration Management,
- Change and Work Management,
- Process planning, and
- Multi-Language support.

7.3.1 Part Identification

This subset of the STEP PDM Schema includes the primary objects used for product data management. This subset provides the capability to represent product management information. It includes information about items that are either raw materials, parts, or tools, about versions and views of items. A part may represent one of a variety of physical entities used in discrete manufacturing; including raw material, semi-finished parts, assemblies, instruction manuals, kits, manufacturing by-products, and products. The manufacturing industry is de-fined by the design, production, and sales of parts, and almost every business activity in some way works with data that describes parts.

7.3.2 Part Structure

Base of this subset is the group of objects that define the bill of material relationships be-tween items for discrete manufacturing.

A part is not defined by a single object with a set of attributes, but a collection of objects and relationships, each describing different aspects of the part. For example, a part definition may consist of several engineering attributes, links to suppliers of the part, references to CAD drawings describing the parts geometry, and a list of components used to assemble the part. These different pieces of the part definition will be referred to as part data objects.This subset supports explicit hierarchical product structures representing assemblies and the constituents of those assemblies. This explicit part structure corresponds to the traditional engineering and manufacturing bill of material indentured parts list.

7.3.3 Document and File Management

The scope of this subset is the handling of electronic documents comprising one or more files and track documents that are not actively managed by the PLM system.

External files represent a simple external reference to a named file. An external file is not managed independently by the system - there is usually no revision control or any representation definitions of external files. Version identification may optionally be associated with an external file, but this is for information only and is not used for managed revision control.

If a file is under configuration control, it should be represented as a constituent of a document definition view/ representation. In this case it is actually the managed document that is under direct configuration control, the file is in this way indirectly under configuration control. A change to the file results in a change to the managed document (i.e. a new version) - the changed file would be mapped as a constituent of a view/representation definition of the new document version. A simple external reference alone is not configuration controlled; it is just an external file reference to product data. Documents may be associated with product data in a specified role, to represent some relationship between a document and other elements of product data. Constraints may also be specified on this association, in order to distinguish an applicable portion of an entire document or file in the association.

7.3.4 Shape Definition and Transformation

The scope of this subset provides the capability to associate items with shape or to identify aspects of the shape. It allows also to distinguish between geometric elements used as auxiliary elements and geometric elements that describe product data. Additionally, it contains the capability of an empty geometric model with only a geometric element for placement purposes and an unconstrained three-dimensional geometric model that may contain any geometric data elements.

This subset allows linking geometric structures that result from relating different shape representations with associated product structure when applicable, i.e., when the geometric structure directly corresponds to the assembly structure.

Two alternatives for the implementation of geometric structures related to assembly structures are recommended:

[1] The assembly is described with the components built in. With this approach the shape of the component is mapped into the shape of the assembly via mapped_item. The basic idea of the mapped_item is: an item will become part of another item. The assembly component geometry is used as a template in the assembly geometry.

[2] The components of an assembly are described together with the construction history. This approach uses the representation_relationship_with_transformation. The transformation describes the relation between different work-spaces.

The usage of both alternatives is considered reasonable, because both mechanisms make sense even in mixed combinations. With regard to the transformations in the context of assembly, a part is in principle incorporated in the assembly only by rigid motion (i.e., translation and/or rotation) excluding mirroring and scaling.

7.3.5 Classification

A simple basic type of classification of products in STEP works by assigning categories to product data items. These categories are identified by name labels that define the related classification. This type of classification is referred to as specific classification. A specific_item_classification_hierarchy is used to build up hierarchical structures of specific_item_classification.

7.3.6 Properties

The scope of this subset allows specifying properties associated with parts. A property is the definition of a special quality and may reflect physics or arbitrary, user defined measurements. A general pattern for instantiating property information is in this subset. A number of pre-defined property type names are also proposed for use when appropriate.

A special case of part properties is that of the part shape property - a representation of the geometrical shape model of the part, which are described in section 2.3.4.

7.3.7 Alias Identification

An alias identification is a mechanism to associate an object with an additional identifier that is used to identify the object of interest in a different context, either in another organization, or in some other context. The alias identification mechanism shall not be used to alias sup-plied parts.

The scope of the alias identification shall be specified either by the description of the associated identification_role or - if the scope is defined by an organization - with help of an applied_organization_assignment. The scope of an alias defines the context in which the id specified via applied_identification_assignment.assigned_id overrides the original id. A scenario might be that an object has an id in the context of the organization assigned in the role 'id owner' as a primary id and other ids defined via aliases that are valid in the context of some other organizations.

7.3.8 Authorization

The scope of this subset represents organizations and people in organizations as they per-form functions related to other product data and data relationships. A person in this scope must exist in the context of some organization. An organization or a person in an organization is then associated with the data or data relationship in some role indicating the function being performed. Both people and organizations may have addresses associated with them.

Approving in this scope is accomplished by establishing an approval entity and relating it to some construct through an applied_approval_assignment. The applied_approval_assignment entity may have a role associated with it through the entity role_association and its related object_role entity to indicate the reason/role of this approval related to the particular element of product data.

Approval may be represented as a simple basic approval, or it may represent a more complex approval cycle involving multiple provers, on different dates/times, and possibly with different status values.

7.3.9 Configuration Management

The purpose of this subset of the STEP PDM Schema [2] is meeting the requirements of enterprises that offer many possible configurations of their products for sale. In most cases, the different configurations of a product differ from each other in only minor ways. Configuration identification in the STEP PDM Schema [2] is the identification of product concepts and their associated configurations, the composition of which is to be managed. If a configuration of a product concept is implemented by a certain design, i.e. a particular part version, this version can be associated with the configuration and managed using configuration effectivity. Because this model is based on the configuration management model defined in STEP AP214, additional information and description of how to use the model can be found in the ARM model and other documentation on AP214.

7.3.10 Change and Work Management

This subset describes the process by which companies request, implement, and effect change to products, documents, components, assemblies, manufactured or purchased parts, processes, or even suppliers. This subset provides the capability to represent activity, project, and contract related information. Activities may be initiated by work requests and may be authorized by work orders. Activities may result in changes of models or of properties; such changes can also be represented.

7.3.11 Process planning

This subset provides the capability to represent process related data. This includes process plans, versions of process plans with version tracking, process operations and properties of processes. A process plan is decomposed into one or more occurrences of process operations. Process plans and process operations establish relationships among raw materials, in-process items, and final items, as well as the relationship between the items and the tools used to manufacture them. Additionally, the representation of the connection of parts in various kinds of mating is part of this subset.

7.3.12 Multi-Language support

This subset provides the capability to represent descriptive information about objects in different languages.

7.4 EXPRESS-X Mapping

Supposed that one has two EXPRESS Schemes which cover approximately the same context. Then the EXPRESS-X mapping gives rise to a method for mapping instances of one schema onto instances of the second schema. Information not contained in the second schema are neglected.

The EXPRESS-X mapping specification in this section specifies the mapping from the STEP PDM Schema [2] extended by the relevant subsets of ISO 10303-214:2000 [8] (Chapter 7.3), especially the Configuration Management modelling parts according to CC8, given as an AIM representation to the PIM Equivalence model (Chapter 7.5). In addition to the EXPRESS-X mapping specification, instance diagrams are supplied in order to illustrate the mapping specification. These diagrams follow the EXPRESS-G notation. They highlight the elements in scope of the illustrated mapping. Corresponding AIM and ARM constructs are shown in parallel.

Example: Entities needed to create an item.

```
ENTITY product;
   id : identifier;
   name : label;
   description : OPTIONAL text;
   frame_of_reference : SET[1:?] OF product_context;
END_ENTITY;

ENTITY product_related_product_category
   SUBTYPE OF (product_category);
   products : SET[1:?] OF product;
END_ENTITY;

ENTITY product_category;
   name : label;
   description : OPTIONAL text;
   DERIVE
      id : identifier := get_id_value(SELF);
   WHERE
      wr1 : SIZEOF(USEDIN(SELF,'AUTOMOTIVE_DESIGN.' + 'ID_ATTRIBUTE.' +
            'IDENTIFIED_ITEM')) <= 1;
END_ENTITY;
```

To reduce the complexity of the resulting reference model, the transformation in to the PIM represented in UML is based on a PIM Equivalence model which is similar to the STEP AP214 ARM representation. This PIM Equivalence model is described in Chapter 7.5.

Example: Entities needed to create an item.

```
ENTITY item;
   id : STRING;
   name : string_select;
   description : OPTIONAL string_select;
   INVERSE
      associated_version : SET[1:?] OF item_version FOR associated_item;
      item_classification : SET[1:?] OF specific_item_classification
                                 FOR associated_item;
END_ENTITY;
```

The relationship of the STEP AP214 ARM model representation described in EXPRESS to the underlying AIM model is normative and described by mapping tables as part of the STEP AP214 standard.

Nevertheless, an EXPRESS-X mapping exists between corresponding parts of the STEP PDM and AP214 Schema modelled with AIM elements and the STEP AP214 ARM model. This mapping is extended to reflect additional modelling requirements met by the PIM Equivalence model and to remove insufficiencies in the base models.

7.4.1 Part Identification

7.4.1.1 Item

A target instance of type Item is created out of a source Instance of type Product in the source schema which is referenced by an instance of type Product_related_product_category as products where the value of the name attribute is either 'part', 'raw material' or 'tool'.

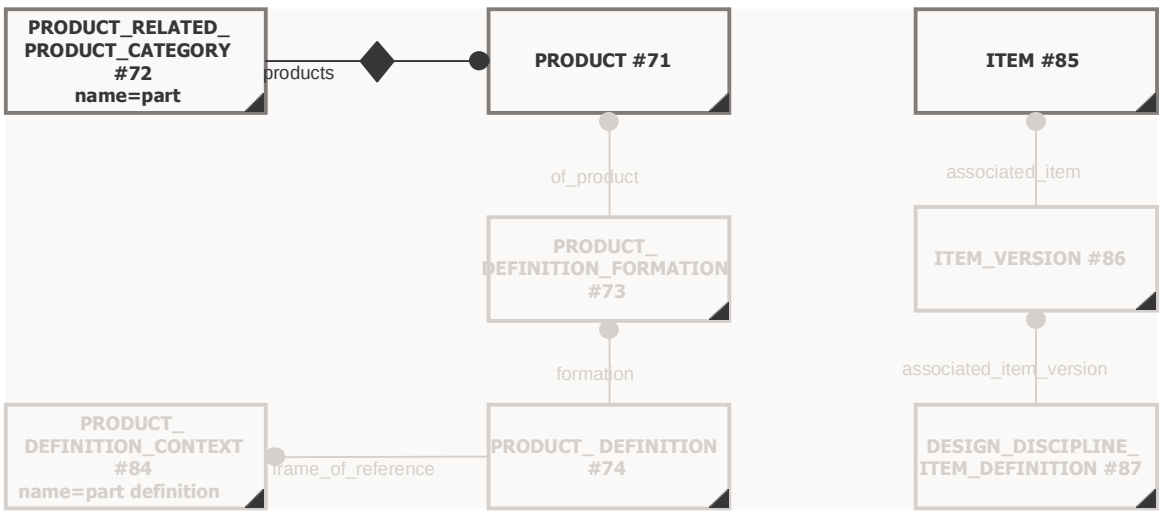


Figure 5 - Instance mapping for item

EXPRESS-X Mapping Specification:

```
MAP item_map AS
   it : item;
FROM
   p    : product;
   prpc : product_related_product_category;
WHERE
   wr1: p IN prpc.products;
   wr2: prpc.name IN ['part', 'raw material', 'tool'];
IDENTIFIED_BY p;
SELECT
   it.id          := p.id;
   it.name        := p.name;
   it.description := p.description;
END_MAP;
```

7.4.1.2 Item_version

A target instance of type Item_version in the target schema is created out of a source instance of type Product_definition_formation which references a Product instance which is mapped to an Item.

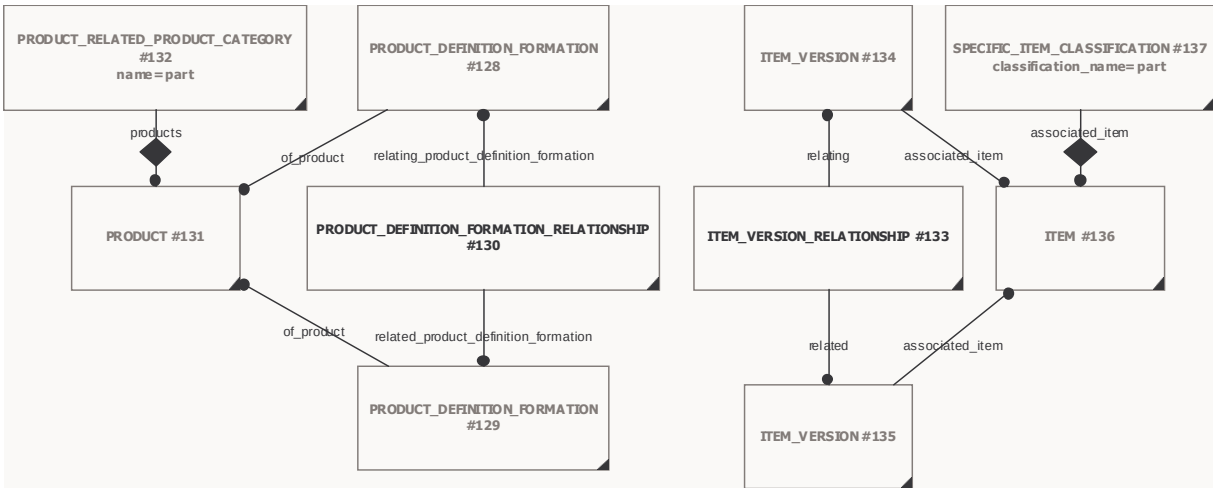


Figure 6 - Instance mapping for item version

EXPRESS-X Mapping Specification:

```
MAP item_version_map AS
   iv : item_version;
FROM
   pdf : product_definition_formation;
WHERE
   EXISTS(item_map(pdf.of_product));
SELECT
   iv.id := pdf.id;
   iv.associated_item := item_map(pdf.of_product);
   iv.description := pdf.description;
```

```
END_MAP;
```

7.4.1.3 Item_version_relationship

An target instance of type Item_version_relationship is created out of an instance of a source instance of type Product_definition_formation_relationship which references instances of type Product_definition_formation that are mapped to Item_versions as relating_product_definition_formation and as related_product_definition_formation.

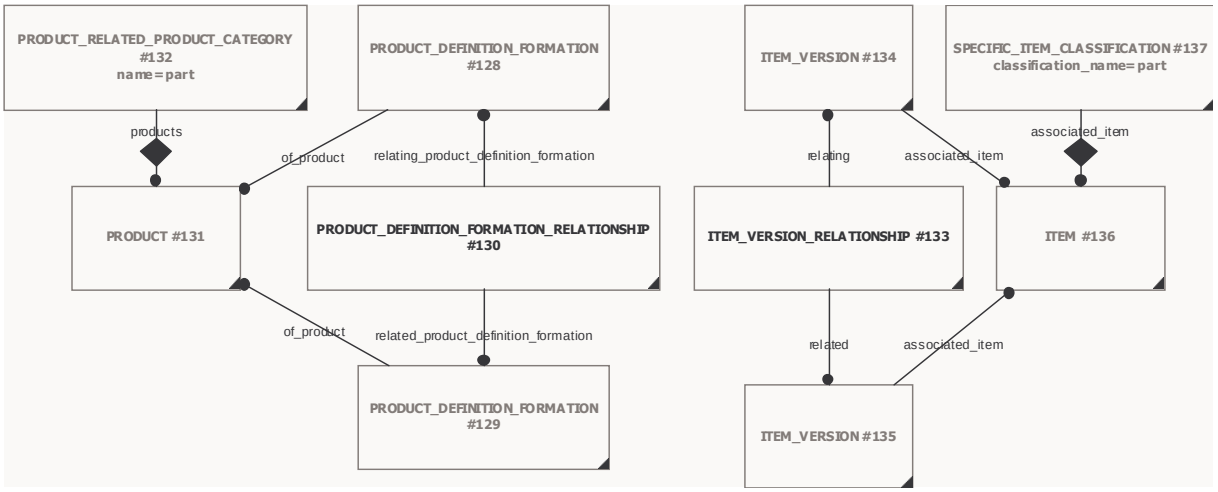


Figure 7 - Instance mapping for item version relationship

EXPRESS-X Mapping Specification:

```
MAP item_version_relationship_map AS
   ivr : item_version_relationship;
FROM
   pdfr : product_definition_formation_relationship;
WHERE
   EXISTS(item_version_map(pdfr.related_product_definition_formation)) AND
   EXISTS(item_version_map(pdfr.relating_product_definition_formation));
SELECT
   ivr.description   := pdfr.description;
   ivr.relation_type := pdfr.name;
   ivr.related       :=
          item_version_map(pdfr.related_product_definition_formation);
   ivr.relating      :=
          item_version_map(pdfr.relating_product_definition_formation);
END_MAP;
```

7.4.1.4 Application_context

A target instance of type Application_context is created out of a source instance of type Product_definition_context.

EXPRESS-X Mapping Specification:

```
MAP application_context_map AS
   actx : application_context;
FROM
   pdctx : product_definition_context;
SELECT
   actx.life_cycle_stage   := pdctx.life_cycle_stage;
   actx.application_domain := pdctx.frame_of_reference.application;
END_MAP;
```

7.4.2 Part Structure

7.4.2.1 Item definitions, and Process_state

A target instance of type Design_discipline_item_definition is created out of an source instance of type Product_definition in the source schema that references an instance of type Product_definition_context as frame_of_reference which has a name attribute with value 'part definition'.

If the source instance is referenced by an instance of type Product_definition_context_association with role name 'part definition type' as definition, which refers to an instance of type Product_definition_context as frame_of_reference then the type of the created instance is a subtype of Design_discipline_item_definition, depending on the value of the name attribute of the Product_definition_context. An Assembly_definition is created if the name is 'assembly definition', a Collection_definition is created if the name is 'collection definition' and a Process_state is created if the name is 'process state'

.

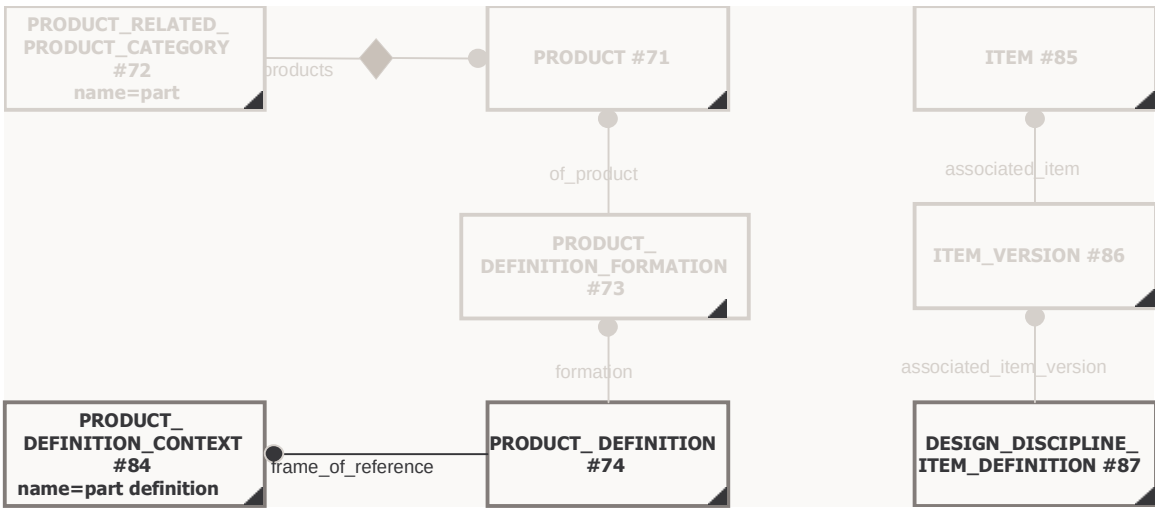


Figure 8 - Instance mapping for design discipline item definition

EXPRESS-X Mapping Specification for Design_discipline_item_definition

```
MAP ddid_map AS
   ddid : design_discipline_item_definition;
FROM
   pd : product_definition;
WHERE
```

```
   wr1: pd.frame_of_reference.name = 'part definition';
SELECT
   ddid.id                       := pd.id;
   ddid.name                     := pd.name;
   ddid.associated_item_version := item_version_map(pd.formation);
   ddid.initial_context := application_context_map(pd.frame_of_reference);
   ddid.additional_context :=
      FOR EACH pdca IN pd<-
         pd<-definition{product_definition_context_association |
                     role.name = 'additional context'};
         RETURN application_context_map(pdca.frame_of_reference);
END_MAP;
```

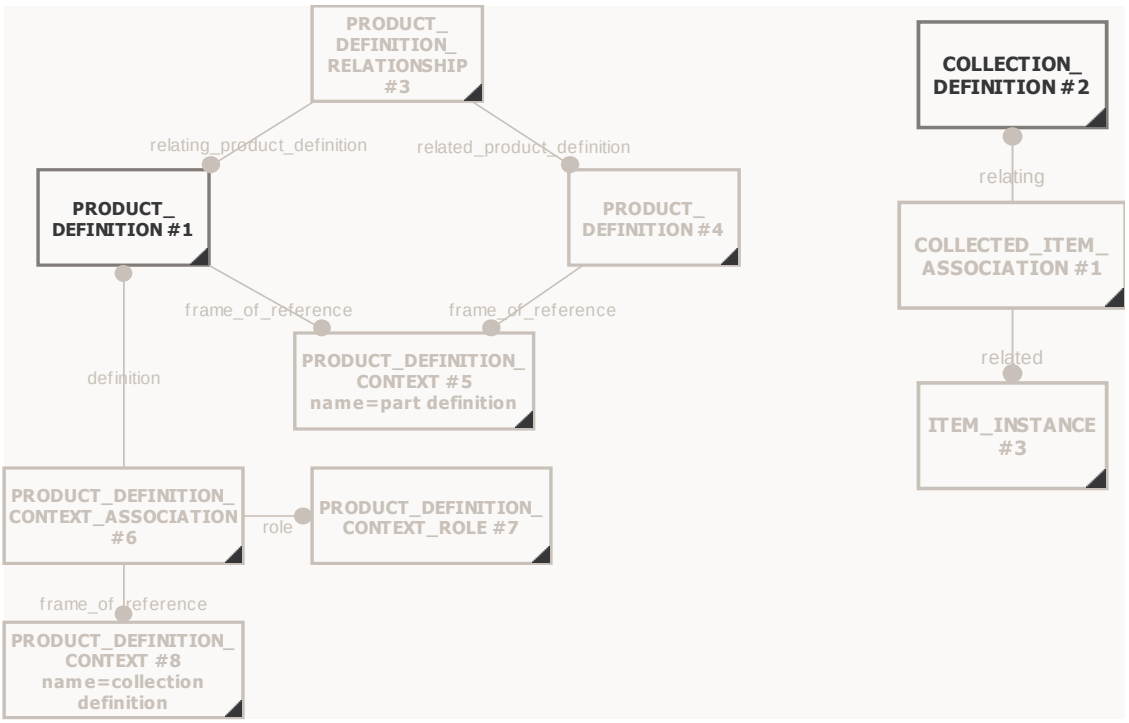


Figure 9 - Instance mapping for collection definition

EXPRESS-X Mapping Specification for Assembly_definition, Collection_definition, Process_state:

```
MAP assembly_definition_map AS
   ddid : assembly_definition;
SUBTYPE OF (ddid_map);
WHERE
   SIZEOF(pd<-definition{product_definition_context_association |
                        role.name = 'part definition type'}
            ::frame_of_reference{product_definition_context |
                                 name = 'assembly definition'}) > 0;
END_MAP;

MAP collection_definition_map AS
```

```
   ddid : collection_definition;
SUBTYPE OF (ddid_map);
WHERE
   (SIZEOF(pd<-definition{product_definition_context_association |
                         role.name = 'part definition type'}
           ::frame_of_reference{product_definition_context |
                               name = 'collection definition'}) > 0);
END_MAP;
MAP process_state_map AS
   ddid : process_state;
SUBTYPE OF (ddid_map);
WHERE
   SIZEOF(pd<-definition{product_definition_context_association |
                        role.name = 'part definition type'}
          ::frame_of_reference{product_definition_context |
                              name= 'process state'}) > 0;
SELECT
   ddid.related_item_definition :=
      ddid_map(pd<-related_product_definition
                 {product_definition_relationship |
                  name = 'process state to related item'}
                ::relating_product_definition[1]);
END_MAP;
```

7.4.2.2 Assembly relationships

A target instance of type Item_definition_instance_relationship is created out of a source instance of type Product_definition_relationship which refers to a Product_definition with a frame_of_reference name of 'part definition' as relating_product_definition. In addition the Product_definition_relationship source instance must either refer to an instance of type Product_definition with frame_of_reference name 'part occurrence' as related_product_definition or it must be of type Assembly_component_usage and refer to an instance of type Product_definition with frame_of_reference name 'part definition' as related_product_definition.

If the name of the source Product_definition_relationship is 'collection membership' then a target instance of subtype Collected_item_association is created.

If the source Product_definition_relationship is of type Next_assembly_usage_occurrence, then a target instance of subtype Next_higher_assembly is created.

If the source Product_definition_relationship is of type Assembly_component_usage then a target instance of subtype Assembly_component_relationship is created.

If all of the above conditions are false, a target instance of subtype General_item_definition_instance_relationship is created.

Figure 10 - Instance mapping for next higher assembly

EXPRESS-X Mapping Specification:

```
MAP item_definition_instance_relationship_map AS
   rel : item_definition_instance_relationship;
FROM
   pdr : product_definition_relationship;
WHERE
   wr1: pdr.relating_product_definition.frame_of_reference.name =
        'part definition';
   wr2: (pdr.related_product_definition.frame_of_reference.name =
        'part occurrence') OR
        ('AUTOMOTIVE_DESIGN.ASSEMBLY_COMPONENT_USAGE' IN TYPEOF(rel)) AND
        (pdr.related_product_definition.frame_of_reference.name =
        'part definition');
SELECT
   rel.relating := ddid_map(pdr.relating_product_definition);
   rel.related  := IF 'AUTOMOTIVE_DESIGN.ASSEMBLY_COMPONENT_RELATIONSHIP'
                      IN TYPEOF(rel)
                   THEN
                      item_instance_map(
                      pdr.related_product_definition
                      <-related_product_definition
                         {product_definition_relationship |
                         name = 'definition usage'}::
                         relating_product_definition{product_definition|
                            pdr IN product_definition<-occurence
                            {product_definition_occurrence_relationship}
                            ::occurrence_usage}[1]);
                   ELSE
                      item_instance_map(pdr.related_product_definition);
                   END_IF;
```

```
END_MAP;

MAP assembly_structure_map AS
   rel : assembly_component_relationship;
SUBTYPE OF (item_definition_instance_relationship_map)
WHERE
   'AUTOMOTIVE_DESIGN.ASSEMBLY_COMPONENT_RELATIONSHIP' IN TYPEOF(pdr);
SELECT
   acr.placement   :=
      model_relationship_map(pdr<-definition{product_definition_shape}
                             <-represented_product_relation
                               {context_dependent_shape_representation}
                             ::representation_relation[1]);
END_MAP;

MAP next_higher_assembly_map AS
   rel : next_higher_assembly;
SUBTYPE OF (assembly_structure_map);
WHERE
   'AUTOMOTIVE_DESIGN.NEXT_ASSEMBLY_USAGE_OCCURRENCE' IN TYPEOF(pdr);
END_MAP;

MAP collected_item_association_map AS
   rel : collected_item_association;
SUBTYPE OF (item_definition_instance_relationship);
WHERE
   pdr.name = 'collection membership';
END_MAP;

MAP general_item_definition_instance_relationship_map AS
   rel : general_item_definition_instance_relationship;
SUBTYPE OF (item_definition_instance_relationship);
WHERE
   OTHERWISE;
SELECT
   rel.description := pdr.description;
   rel.relation_type := pdr.name;
END_MAP;
```

7.4.2.3 Item relationships

A target instance of type Item_definition_relationship is created from a source instance of Product_definition_relationship, which is not of type Assembly_component_usage, where the related and relating Product_definition reference a Product_context as frame_of_reference with name attribute value 'part definition'. The exact type of the target instance depends on the subtype of the Product_definition_relationship.

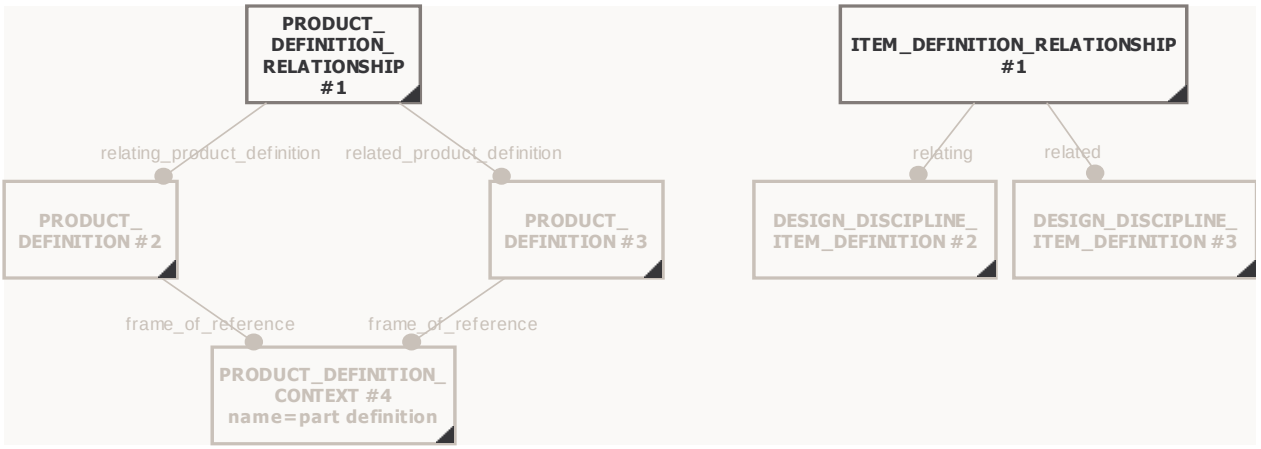


Figure 11 - Instance mapping for item relationships

EXPRESS-X Mapping Specification:

```
MAP item_definition_relationship_map AS
   idr : item_definition_relationship;
FROM
   pdrel : product_definition_relationship;
WHERE
   wr1: pdrel.relating_product_definition.frame_of_reference.name
        = 'part definition';
   wr2: pdrel.related_product_definition.frame_of_reference.name
        = 'part definition';
   wr3: NOT ('AUTOMOTIVE_DESIGN.ASSEMBLY_COMPONENT_USAGE' IN TYPEOF(rel));
SELECT
   idr.relating := ddid_map(pdrel.relating_product_definition);
   idr.related  := ddid_map(pdrel.related_product_definition);
END_MAP;

MAP make_from_relationship_map AS
   idr : make_from_relationship;
SUBTYPE_OF (item_definition_relationship_map);
WHERE
   wr3: 'AUTOMOTIVE_DESIGN.MAKE_FROM_USAGE_OPTION' IN TYPEOF(pdrel);
SELECT
   idr.description := pdrel.description;
END_MAP;

MAP replaced_definition_relationship_map AS
   idr : replaced_definition_relationship;
SUBTYPE OF (item_definition_relationship_map);
WHERE
   wr3: pdrel.name = 'definition replacement';
SELECT
   idr.description := pdrel.description;
```

```
END_MAP;

MAP geometrical_relationship_map AS
   idr : geometrical_relationship;
SUBTYPE OF (item_definition_relationship_map);
WHERE
   wr3: pdrel.name = 'geometrical relationship';
SELECT
   idr.description := pdrel.description;
   idr.definition_placement :=
      model_relationship_trafo_map(pdrel<-definition{product_definition_shape}
                                   <-represented_product_relation
                                     {context_dependent_shape_representation}
                                :  :representation_relation[1]);
END_MAP;

MAP tool_part_relationship_map AS
   idr : tool_part_relationship;
WHERE
   wr3: pdrel.name = 'tool part relationship';
SELECT
   idr.used_technology_description := pdrel.description;
   idr.placement :=
      model_relationship_trafo_map(pdrel<-definition{product_definition_shape}
                                   <-represented_product_relation
                                     {context_dependent_shape_representation}
                                :  :representation_relation[1]);
END_MAP;

MAP general_item_definition_relationship_map AS
   idr : general_item_definition_relationship;
SUBTYPE OF (item_definition_relationship_map);
WHERE
   OTHERWISE;
SELECT
   idr.description := pdrel.description;
   idr.relation_type := pdrel.name;
END_MAP;
```

7.4.2.4 Item instance entities

A target instance of type Item_instance is created out of a source instance of type Product_definition that references an instance of type Product_definition_context with name 'part occurrence' as frame_of_reference. Depending on the value of the name attribute of the source instance the target instance is of subtype Single_instance (for value 'single instance'), Quantified_instance (for value 'quantified instance'), Selected_instance (for value 'selected instance') or Specified_instance (for value 'specified instance').

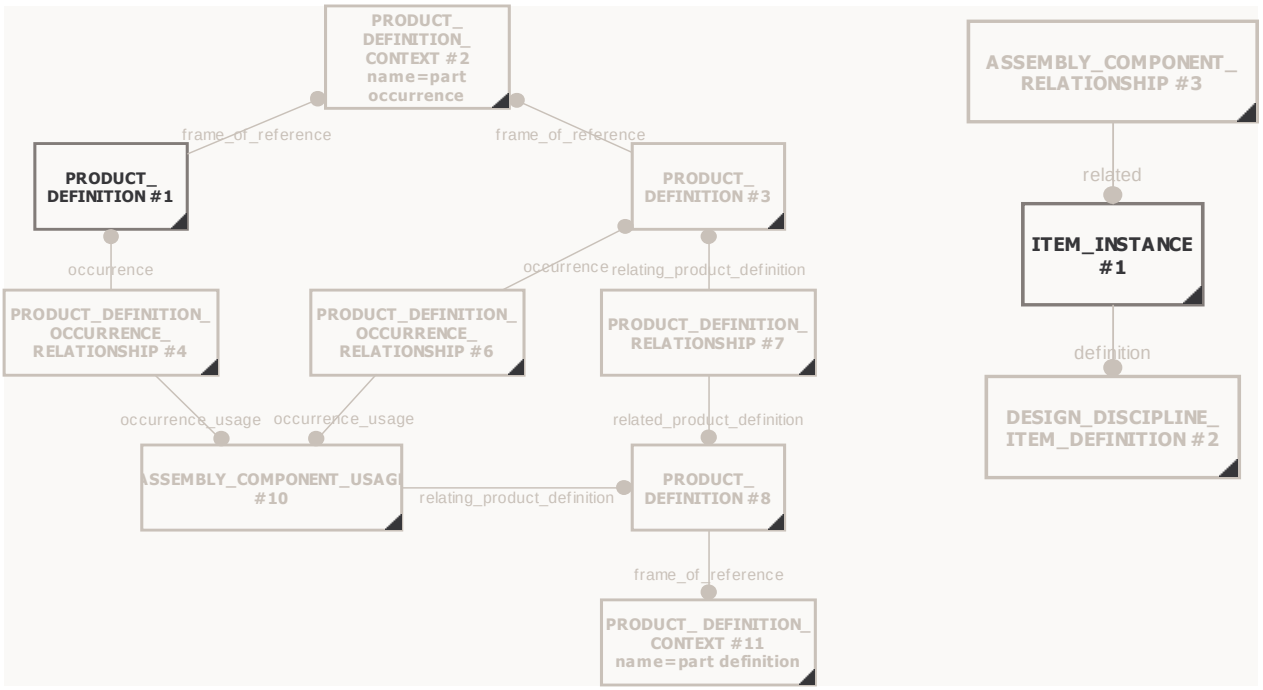


Figure 12 - Instance mapping for item instance entities

EXPRESS-X Mapping Specification:

```
MAP item_instance_map AS
   ii : item_instance;
FROM
   pd : product_definition;
WHERE
   pd.frame_of_reference.name = 'part occurrence';
SELECT
   ii.description := pd.description;
   ii.id := pd.id;
   ii.definition :=  IF SIZEOF(pd<-related_product_definition
                         {product_definition_relationship |
                         name = 'definition usage'}) > 0
                     THEN
                        ddid_map(pd<-related_product_definition
                           {product_definition_relationship |
                           name = 'definition usage'}
                              ::relating_product_definition[1]);
                     ELSE
                        product_identification_map
                           pd<-design{configuration_design |
                           name = 'occurrence usage definition'}
                              ::configuration{product_identification}[1]);
                     END_IF;
END_MAP;
```

```
MAP single_instance_map AS
   ii : single_instance;
SUBTYPE OF (item_instance);
WHERE
   pd.name = 'single instance';
END_MAP;

MAP quantified_instance_map AS
  ii : quantified_instance;
SUBTYPE OF (item_instance);
WHERE
   pd.name = 'quantified instance';
SELECT
   ii.quantity := numerical_value_map(
         pd<-definition{property_definition |
                  name = 'occurrence quantity'}
            <-definition{property_definition_representation}
            ::used_representation{representation | name='quantity'}
            ::items{measure_representation_item |
                  name = 'quantity measure'}[1]);
END_MAP;

MAP selected_instance_map AS
   ii : selected_instance;
SUBTYPE OF (item_instance);
WHERE
   pd.name = 'selected instance';
SELECT
   ii.selection_control :=
      pd<-definition{property_definition | name = 'occurrence selection'}
         <-definition{property_definition_representation}
         ::used_representation{representation | name = 'selection criteria'}
         ::items{descriptive_representation_item |
               name = 'selection control' }[1].description;
   ii.selected_quantity := value_with_unit_map(
      pd<-definition{property_definition | name = 'occurrence selection'}
         <-definition{property_definition_representation}
         ::used_representation{representation | name = 'selection criteria'}
         ::items{representation_item | name = 'selection quantity'}[1]);
END_MAP;

MAP specified_instance_map AS
   ii : specified_instance;
SUBTYPE OF (item_instance);
WHERE
   pd.name = 'specified instance';
SELECT
   ii.assembly_context := assembly_definition_map(
      pd<-occurrence{product_definition_occurrence_relationship}
         ::occurrence_usage{specified_higher_usage_occurrence}
```

```
      ::relating_product_definition[1]);
   ii.related_instance := item_instance_map(
      pd<-occurrence{product_definition_occurrence_relationship}
         ::occurrence_usage{specified_higher_usage_occurrence}
         ::next_usage{next_assembly_usage_occurrence}
               <-occurrence_usage{product_definition_occurrence_relationship}
         ::occurrence{product_definition}[1]);
   ii.upper_usage := item_instance_map(
      pd<-occurrence{product_definition_occurrence_relationship}
         ::upper_usage{next_assembly_usage_occurrence}
         ::occurrence_usage{specified_higher_usage_occurrence}
               <-occurrence_usage{product_definition_occurrence_relationship}
         ::occurrence{product_definition}[1]);
END_MAP;
```

7.4.2.5 Item instance relationships

A target instance of type Item_instance_relationship is created out of a source instance of type Product_definition_relationship that references instances of type Product_definition with a frame_of_reference name 'part occurrence' both as related_product_definition and as relating_product_definition. If the value of the name attribute of the source instance is 'usage re-placement' the target instance is of subtype Replaced_usage_relationship, otherwise the target instance is of subtype General_item_instance_relationship.

A target instance of type Replaced_usage_relationship is also created out of a source instance of type Product_definition_substitute that refers to a Product_definitions with a frame_of_reference name 'part occurrence' as substitute_definition and context_relationship.

EXPRESS-X Mapping Specification:

```
MAP item_instance_relationship_map AS
   iir : item_instance_relationship ;
PARTITION p_pdr ;
FROM
   pdr : product_definition_relationship ;
WHERE
   wr1 : pdr.relating_product_definition.frame_of_reference.name =
         'part occurrence' ;
   wr2 : pdr.related_product_definition.frame_of_reference.name =
         'part occurrence' ;
RETURN (item_instance_relationship_pdrel_map(pdr));
PARTITION p_psubst ;
FROM
   ps : product_definition_substitute;
WHERE
   ps.substitute_definition.frame_of_reference.name = 'part occurrence';
RETURN (replaced_usage_relationship_subst_map(ps));
END_MAP ;
DEPENDENT_MAP item_instance_relationship_pdrel_map AS
   iir : item_instance_relationship ;
FROM
   pdr : product_definition_relationship ;
```

```
SELECT
   iir.related := item_instance_map(pdr.related_product_definition) ;
END_DEPENDENT_MAP ;

DEPENDENT_MAP general_item_instance_relationship_pdrel_map AS
   iir : general_item_instance_relationship ;
SUBTYPE OF (item_instance_relationship_pdrel_map) ;
WHERE
   OTHERWISE ;
SELECT
   irr.relating := item_instance_map(pdr.related_product_definition);
   irr.relation_type := pdr.name ;
END_DEPENDENT_MAP ;

DEPENDENT_MAP replaced_usage_relationship_pdrel_map AS
   iir : replaced_usage_relationship ;
SUBTYPE OF (item_instance_relationship_pdrel_map) ;
WHERE
   (pdr.name = 'usage replacement') OR
   (pdr.name = 'process input or output replacement');
SELECT
   irr.description   := pdr.description ;
   irr.relating      := item_instance_map(pdr.relating_product_definition);
   irr.usage_context := process_operation_input_or_output_map(
      pdr.relating_product_definition
         <-defined_product{process_product_association}[1]);

END_DEPENDENT_MAP ;

DEPENDENT_MAP replaced_usage_relationship_subst_map AS
   rur : replaced_usage_relationship ;
FROM
   ps : product_definition_substitute ;
SELECT
   rur.related := item_instance_map(ps.substitute_definition) ;
   rur.relating :=
            item_instance_map(ps.context_relationship
            <-occurrence_usage{product_definition_occurrence_relationship}
            ::occurrence[1]) ;
   rur.usage_context :=
      IF 'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_USAGE' IN
         TYPEOF(ps.context_relationship)
      THEN
         product_structure_relationship(ps.context_relationship) ;
      ELSE
         item_definition_instance_relationship(ps.context_relationship) ;
      END_IF ;
END_DEPENDENT_MAP ;
```

7.4.2.6 Instance_placement

A target instance of type Instance_placement is created out of a source instance of type Representation_relation_with_transformation with name 'instance placement'.

EXPRESS-X Mapping Specification:

```
MAP instance_placement_map AS
   ip : instance_placement;
FROM
   rrel : representation_relation_with_transformation;
WHERE
   rrel.name = 'instance placement';
SELECT
   ip.placed_instance := single_instance_map(
      rrel::rep_1{shape_representation}
         <-used_representation{shape_definition_representation}
         :: represented_definition{product_definition_shape}
         ::definition{product_definition | (name = 'single instance')
               AND (frame_of_reference.name = 'part occurrence')}[1]);
   ip.placement := geometric_model_relationship(rrel);
   ip.reference_product_component :=
      IF 'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION' IN TYPEOF(rrel.rep_2) THEN
         product_component_map(rrel.rep_2
            <-used_representation{shape_definition_representation}
            :: represented_definition{product_definition_shape}
            ::definition{product_definition | frame_of_reference.name =
                                              'conceptual definition')}[1]);
      ELSE
         product_component_map(rrel::rep_2{representation |
                                     name = 'model property value'}
            <- used_representation{property_definition_representation}
            :: represented_definition{property_definition |
                                     name = 'positioning'}
            ::definition{product_definition | frame_of_reference.name =
                                       'conceptual definition')}[1]);
      END_IF;
   END_MAP;
```

7.4.2.7 Component_placement

A target instance of type Component_placement is created out of a source instance of type Representation_relation_with_transformation with name 'component placement'.

EXPRESS-X Mapping Specification:

```
MAP component_placement_map AS
   ip : compnent_placement;
FROM
   rrel : representation_relation_with_transformation;
WHERE
```

```
   rrel.name = 'component placement';
SELECT
   ip.placed_component :=
      IF 'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION' IN TYPEOF(rrel.rep_1) THEN
         product_component_map(rrel::rep_1{shape_representation}
            <-used_representation{shape_definition_representation}
            :: represented_definition{product_definition_shape}
            ::definition{product_definition | frame_of_reference.name =
                                              'conceptual definition}[1]);
      ELSE
         product_component_map(rrel::rep_1{representation |
                                           name = 'model property value'}
            <- used_representation{property_definition_representation}
            :: represented_definition{property_definition |
                                      name = 'positioning'}
            ::definition{product_definition | frame_of_reference.name =
                                              'conceptual definition')}[1]);
      END_IF;
   ip.placement := geometric_model_relationship(rrel);
   ip.reference_product_component :=
      IF 'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION' IN TYPEOF(rrel.rep_2) THEN
         product_component_map(rrel.rep_2
            <-used_representation{shape_definition_representation}
            :: represented_definition{product_definition_shape}
            ::definition{product_definition | frame_of_reference.name =
                                              'conceptual definition')}[1]);
      ELSE
         product_component_map(rrel::rep_2{representation |
                                           name = 'model property value'}
            <- used_representation{property_definition_representation}
            :: represented_definition{property_definition |
                                      name = 'positioning'}
            ::definition{product_definition | frame_of_reference.name =
                                              'conceptual definition')}[1]);
      END_IF;
   END_MAP;
```

7.4.3 Document and File Management

7.4.3.1 Document

An instance of type Document in the target schema is created out of an instance of type Product in the source schema.

Conditions: attribute 'products' of at least one instance of type Product_related_product_category where the value of the name attribute is 'document', refers to the Product instance.

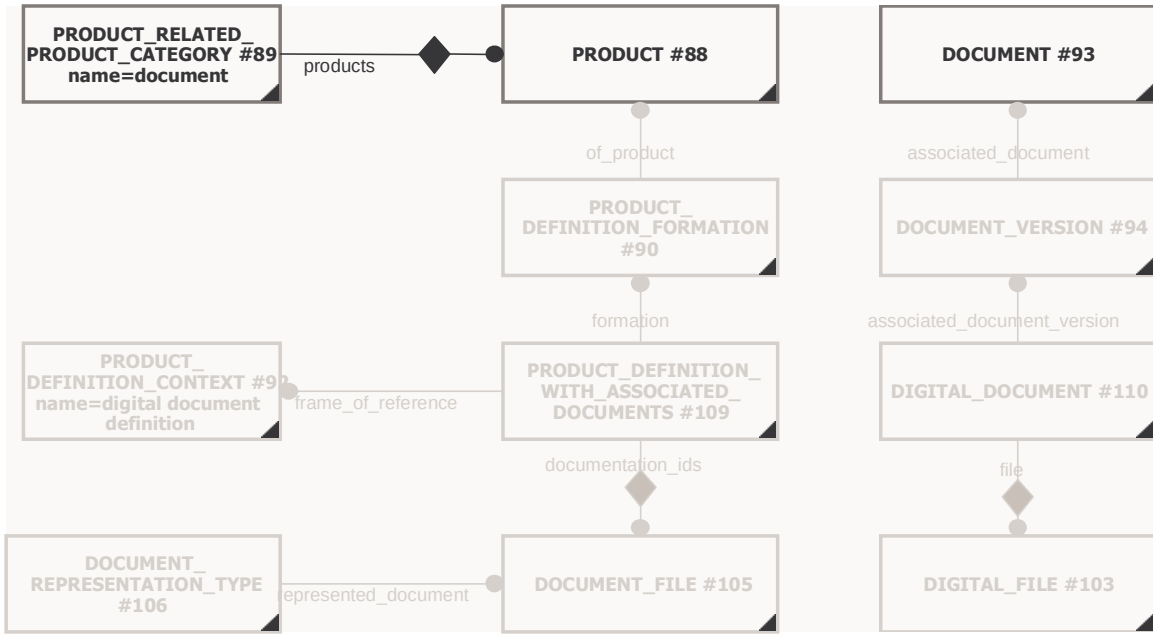


Figure 13 - Instance mapping for document

EXPRESS-X Mapping Specification:

```
MAP document_map AS
   doc : document;
FROM
   p    : product;
   prpc : product_related_product_category;
WHERE
   wr1: p IN prpc.products;
   wr2: prpc.name = 'document';
IDENTIFIED_BY p;
SELECT
   doc.document_id := p.id;
   doc.name        := p.name;
   doc.description := p.description;
END_MAP;
```

7.4.3.2 Document_version

An instance of type Document_version in the target schema is created out of an instance of type Product_definition_formation in the source schema.

Conditions: Attribute of_product refers to an instance of type Product that is mapped to a Document.

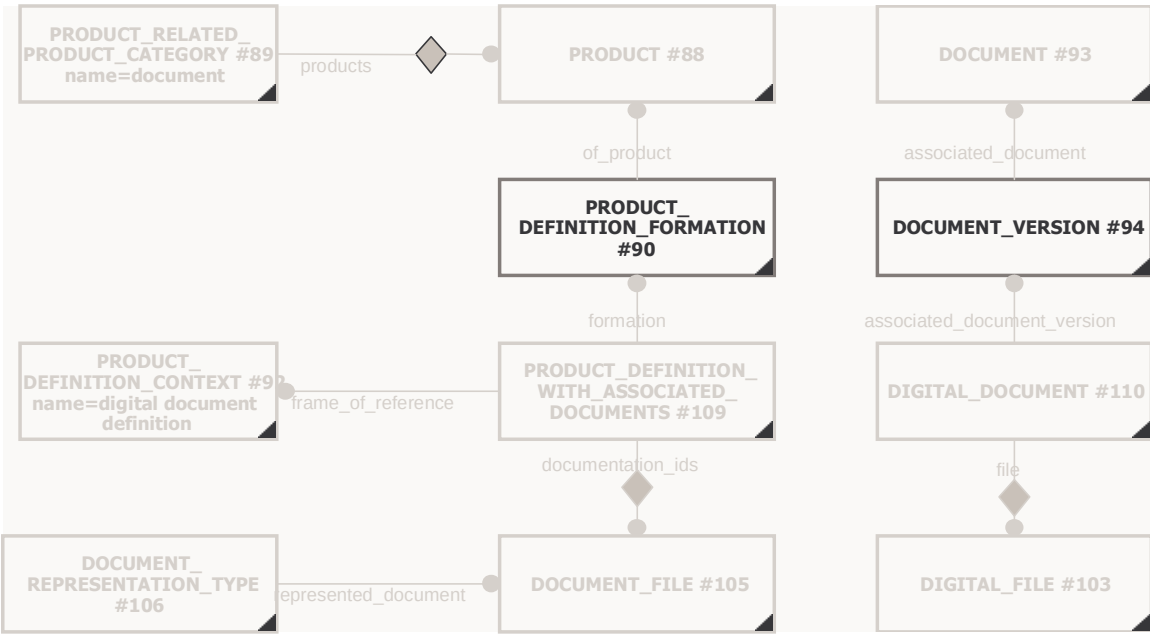


Figure 14 - Instance mapping for document version

EXPRESS-X Mapping Specification:

```
MAP document_version_map AS
   dv : document_version;
FROM
   pdf : product_definition_formation;
WHERE
   EXISTS(document_map(pdf.of_product));
SELECT
   dv.id := pdf.id;
   dv.associated_document := document_map(pdf.of_product);
   dv.description := pdf.description;
END_MAP;
```

7.4.3.3 Document_version_relationship

A target instance of Document_version_relationship is created out of a source instance of type Product_definition_formation_relationship which refers to instances of type Product_definition_formation that are mapped to Document_versions both as related_product_definition and as relating_product_definition.

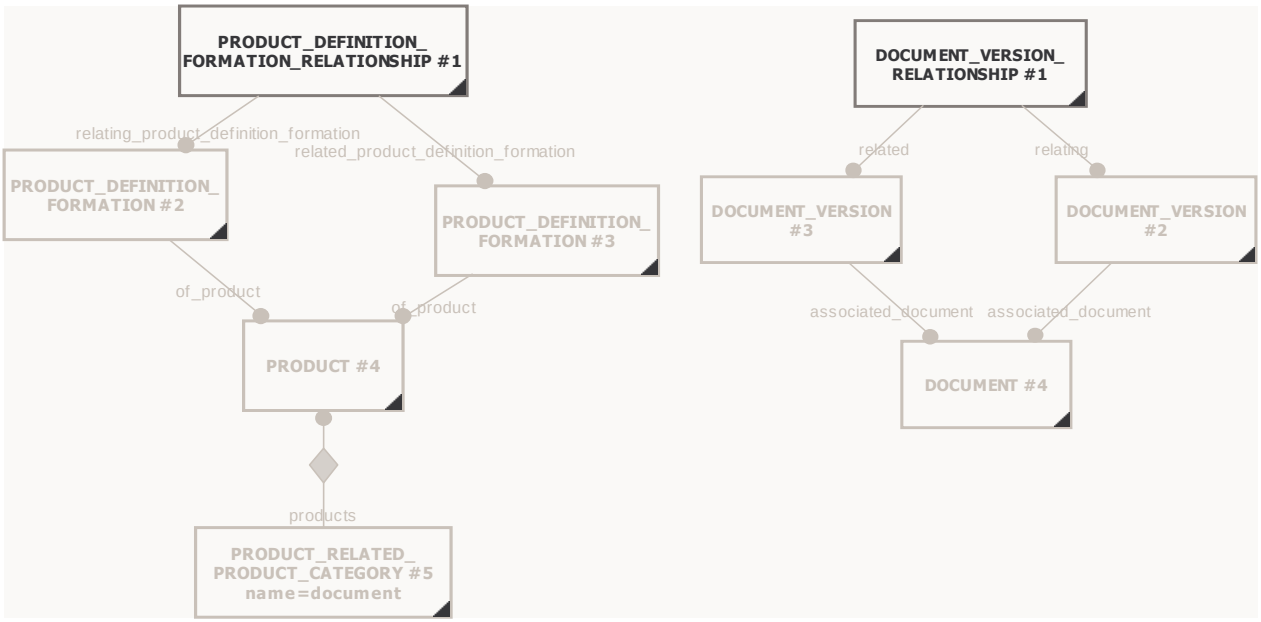


Figure 15 - Instance mapping for document version relationship

EXPRESS-X Mapping Specification:

```
MAP document_version_relationship_map AS
   ivr : document_version_relationship;
FROM
   pdfr : product_definition_formation_relationship;
WHERE
   wr1: EXISTS(document_version_map
               (pdfr.related_product_definition_formation));
   wr2: EXISTS(document_version_map
               (pdfr.relating_product_definition_formation));
SELECT
   ivr.description   := pdfr.description;
   ivr.relation_type := pdfr.name;
   ivr.related       :=
      document_version_map(pdfr.related_product_definition_formation);
   ivr.relating      :=
      document_version_map(pdfr.relating_product_definition_formation);
END_MAP;
```

7.4.3.4 Physical_document and Digital_document

A target instance of type Digital_document or Physical_document is created out of an instance of type Product_definition in the source schema.

[1] A Digital_document is created if the attribute frame_of_reference of the source instance refers to an instance of type Product_definition_context that has an attribute 'name' with value 'digital document definition'.

[2] A Physical_document is created if the attribute frame_of_reference of the source instance refers to an instance of type Product_definition_context that has an attribute 'name' with value 'physical document definition'.

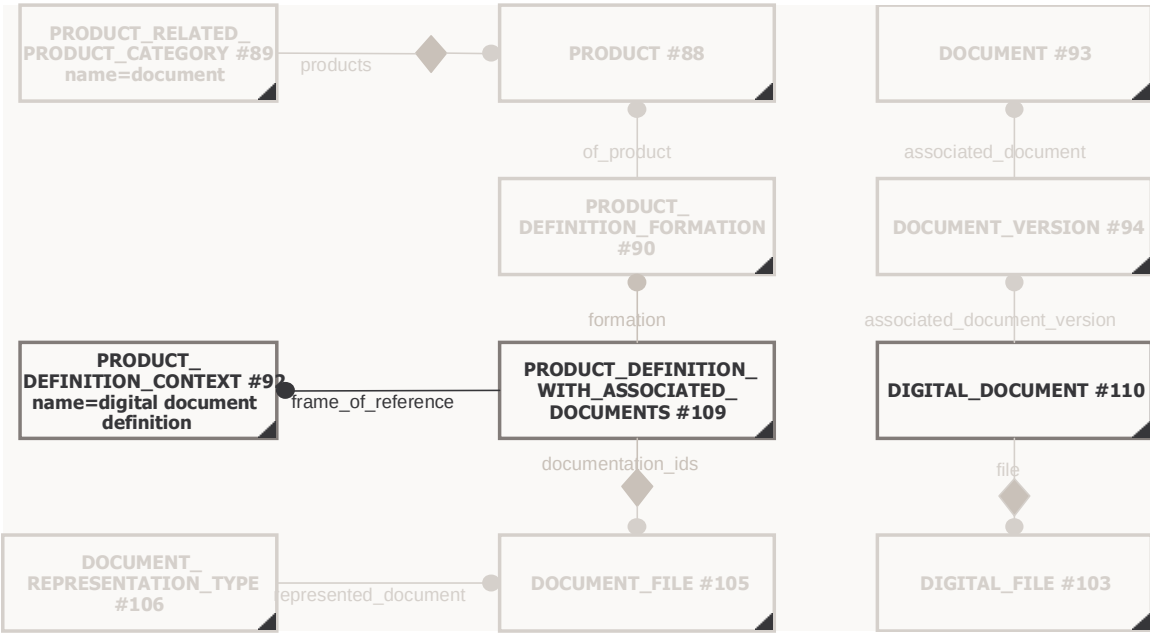


Figure 16 - Instance mapping for digital document

EXPRESS-X Mapping Specification:

```
MAP document_representation_map AS
   drep : document_representation;
FROM
   pd : product_definition;
WHERE
   pd.frame_of_reference.name = 'digital document definition';
SELECT
   drep.id                           := pd.id;
   drep.description                  := pd.description;
   drep.associated_document_version := document_version_map(pd.formation);
   drep.representation_format :=
      document_format_property_map(pd<-definition{property_definition |
                                        name = 'document property'}
                            <-definition{property_definition_representation}
                            ::used_representation{representation |
                                        name = 'document format'}[1]);
   drep.content :=
      document_content_property_map(pd<-definition{property_definition |
                                        name = 'document property'}
                           <-definition{property_definition_representation}
                           ::used_representation{representation |
                                        name = 'document content'}[1]);
   drep.creation :=
```

```
      document_creation_property_map(pd<-definition{property_definition |
                                          name = 'document property'}
                           <-definition{property_definition_representation}
                           ::used_representation{representation |
                                         name = 'document creation'}[1]);
   drep.common_location :=
      FOR EACH idx IN pd<-items{applied_external_identification_assignment |
                               role.name = 'common location'}
         RETURN document_location_property_map(idx.source);
END_MAP;

MAP digital_document_map AS
   drep : digital_document;
SUBTYPE OF (document_representation_map);
WHERE
   pd.frame_of_reference.name = 'digital document definition';
SELECT
   drep.file := IF 'AUTOMOTIVE_DESIGN.' +
                PRODUCT_DEFINITION_WITH_ASSOCIATED_DOCUMENTS' IN
                TYPEOF(pd)
                THEN
                 FOR EACH df IN pd.documentation_ids
                   RETURN digital_file_map(df);
                ELSE
                 [];
                END_IF;
END_MAP;

MAP physical_document_map AS
   drep : physical_document;
SUBTYPE OF (document_representation_map);
WHERE
   pd.frame_of_reference.name = 'physical document definition';
SELECT
   drep.component := IF 'AUTOMOTIVE_DESIGN.' +
                       PRODUCT_DEFINITION_WITH_ASSOCIATED_DOCUMENTS' IN
                       TYPEOF(pd)
                     THEN
                       FOR EACH df IN NVL(pd.documentation_ids,[]);
                         RETURN hardcopy_map(df);
                     ELSE
                       [];
                     END_IF;
END_MAP;
```

7.4.3.5 Digital_file and Hardcopy

A target instance of type Digital_file or Hardcopy is created out of a source instance of type Document_file.

Conditions:

[1] There is at least one instance of type Document_representation_type that references the Document_file instance with attribute 'represented_document'.

[2] If the value of attribute 'name' of the referencing Document_representation_type instance is 'digital' a Digital_file is created.

[3] If the value of attribute 'name' of the referencing Document_representation_type instance is 'physical' a Physical_file is created.

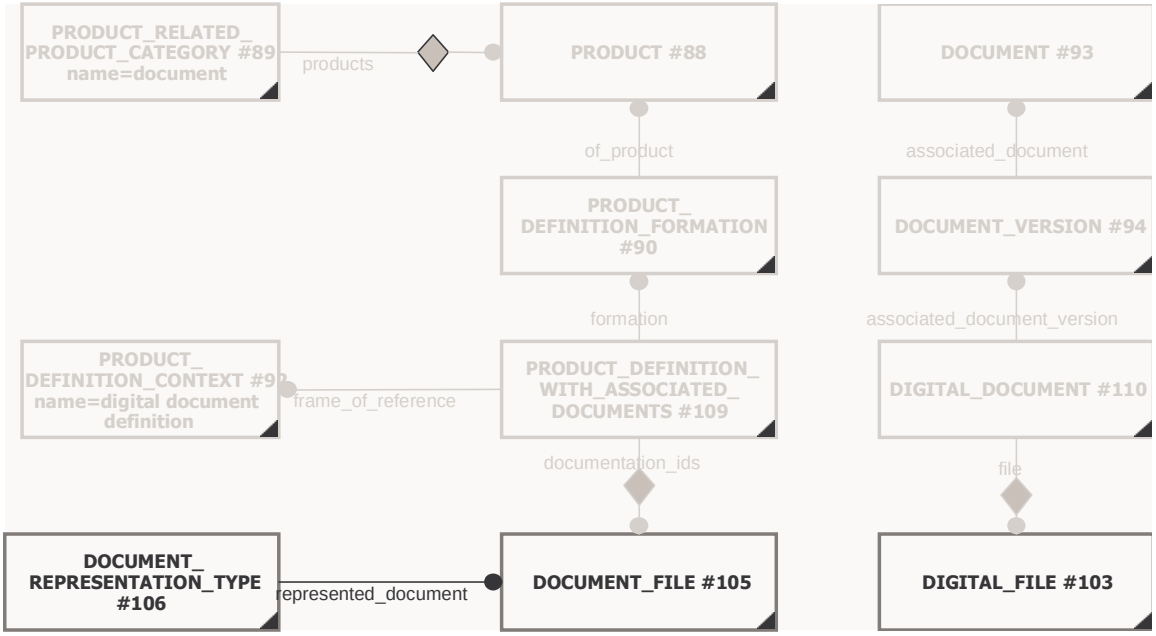


Figure 17 - Instance mapping for digital file

EXPRESS-X Mapping Specification:

```
MAP document_file_map AS
   docf : document_file;
FROM
   df  : document_file;
   drt : document_representation_type;
WHERE
   wr1: drt.represented_document :=: df;
   wr2: drt.name IN ['digital', 'physical'];
IDENTIFIED_BY df;
SELECT
   docf.file_id := df.id;
   docf.document_file_type := document_type_property_map(df.kind);
   docf.file_format :=
      document_format_property_map(df<-definition{property_definition |
                                    name = 'document property'}
                           <-definition{property_definition_representation}
                           ::used_representation{representation |
                                    name = 'document format'}[1]);
```

```
   docf.content :=
      document_content_property_map(df<-definition{property_definition |
                                          name = 'document property'}
                           <-definition{property_definition_representation}
                           ::used_representation{representation |
                                          name = 'document content'}[1]);
   docf.creation :=
      document_creation_property_map(df<-definition{property_definition |
                                          name = 'document property'}
                           <-definition{property_definition_representation}
                           ::used_representation{representation |
                                          name = 'document creation'}[1]);
   docf.external_id_and_location :=
      FOR EACH idx IN df<-items{applied_external_identification_assignment |
                            role.name = 'external document id and location'}
      RETURN external_file_id_and_location_map(idx);
END_MAP;

MAP digital_file_map AS
   docf : digital_file;
SUBTYPE OF (document_file_map);
WHERE
   drt.name = 'digital';
END_MAP;

MAP hardcopy_map AS
   docf : hardcopy;
SUBTYPE OF (document_file_map);
WHERE
   drt.name = 'physical';
END_MAP;
```

7.4.3.6 Document_structure

A target instance of type Document_structure is created out of a source instance of type Product_definition_relationship that references product_definition instances as relating_product_definition and related_product_definition which are both mapped to Document_representation instances.

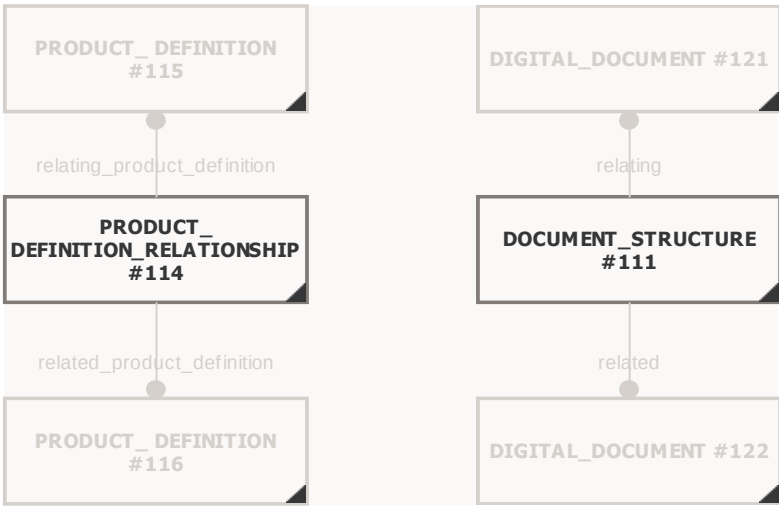


Figure 18 - Instance mapping for document structure

EXPRESS-X Mapping Specification:

```
MAP document_structure_map AS
   dstr : document_structure;
FROM
   pdr : product_definition_relationship;
WHERE
   wr1: EXISTS(document_representation _map(
               pdr.related_product_definition));
   wr2: EXISTS(document_representation_map(
               pdr.relating_product_definition));
SELECT
   dstr.related :=
     document_representation_map(pdr.related_product_definition);
   dstr.relating:=
     document_representation_map(pdr.relating_product_definition);
   dstr.relation_type := pdr.name;
   dstr.description := pdr.description;
END_MAP;
```

7.4.3.7 Document Assignments

A target instance of type Document_assignment is created out of a source instance of type Applied_document_reference.

:

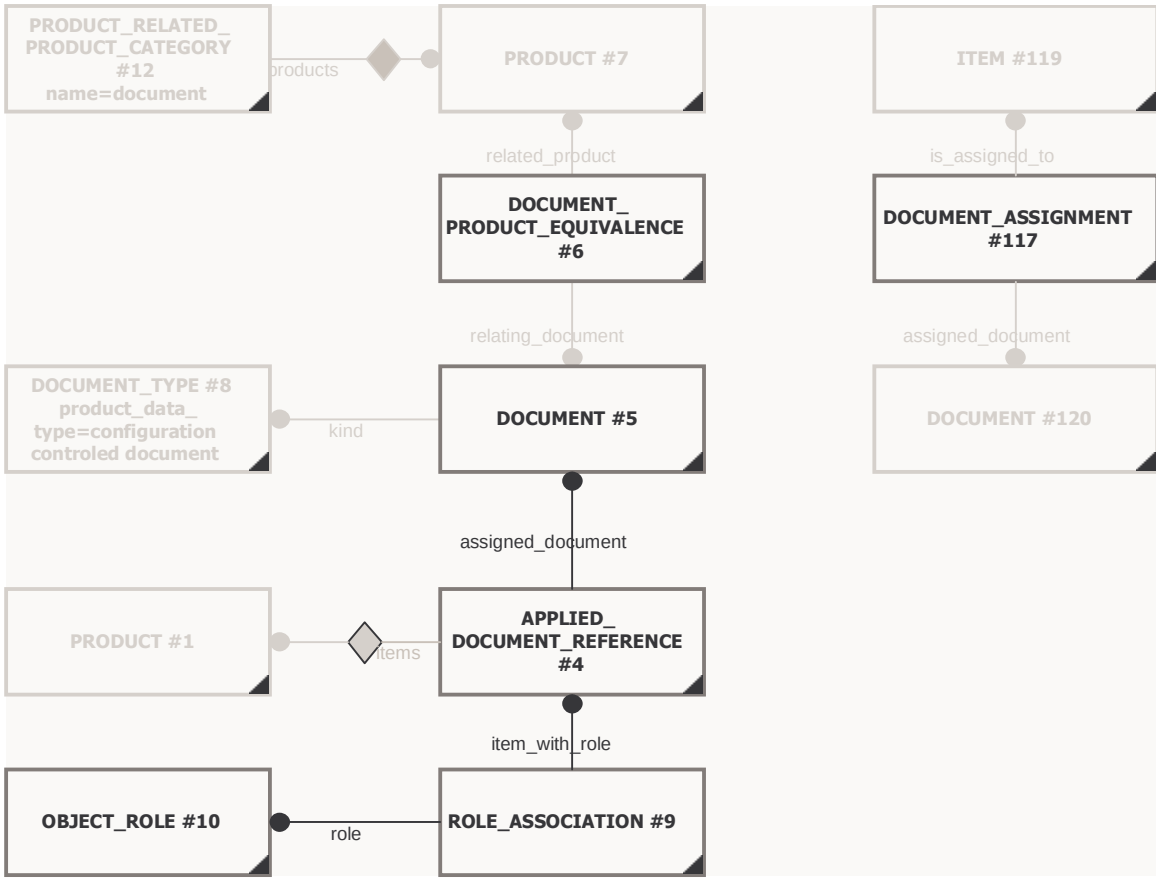


Figure 19 - Instance mapping for document assignment

EXPRESS-X Mapping Specification:

```
MAP item_document_assignment_map AS
   da : document_assignment;
FROM
   adr  : applied_document_reference;
   p    : product;
   dpa  : document_product_equivalence;
   dp   : product;
WHERE
   wr1: p IN adr.items;
   wr2: dpa.relating_document :=: adr.assigned_document;
   wr3: dpa.related_product   :=: dp;
   wr4: adr.assigned_document.kind.product_data_type =
'configuration controlled document';
SELECT
   da.assigned_document := document_map(dp);
   da.is_assigned_to    := item_map(p);
   da.role              := adr.role.name;
END_MAP;
```

```
MAP item_document_version_assignment_map AS
   da : document_assignment;
FROM
   adr : applied_document_reference;
   p  : product;
   dpa : document_product_equivalence;
   dpdf : product_definition_formation;
WHERE
   wr1: p IN adr.items;
   wr2: dpa.relating_document :=: adr.assigned_document;
   wr3: dpa.related_product   :=: dpdf;
   wr4: adr.assigned_document.kind.product_data_type =
'configuration controlled document version';
SELECT
   da.assigned_document := document_version_map(dpdf);
   da.is_assigned_to    := item_map(p);
   da.role              := adr.role.name;
END_MAP;


MAP item_document_representation_assignment_map AS
   da : document_assignment;
FROM
   adr : applied_document_reference;
   p   : product;
   dpa : document_product_equivalence;
   dpd : product_definition;
WHERE
   wr1: p IN adr.items;
   wr2: dpa.relating_document :=: adr.assigned_document;
   wr3: dpa.related_product   :=: dpd;
   wr4: adr.assigned_document.kind.product_data_type =
           'configuration controlled document representation';
SELECT
   da.assigned_document := document_representation_map(dpd);
   da.is_assigned_to    := item_map(p);
   da.role              := adr.role.name;
END_MAP;

MAP item_digital_file_assignment_map AS
   da : document_assignment;
FROM
   p   : product;
   adr : applied_document_reference;
   df  : document_file;
WHERE
   wr1: p IN adr.items;
   wr2: adr.assigned_document :=: df;
SELECT
   da.assigned_document := digital_file_map(df);
```

```
   da.is_assigned_to    := item_map(p);
   da.role              := adr.role.name;
END_MAP;

----- item_version ---

MAP item_version_document_assignment_map AS
   da : document_assignment;
FROM
   adr : applied_document_reference;
   pdf : product_definition_formation;
   dpa : document_product_equivalence;
   dp  : product;
WHERE
   wr1: pdf IN adr.items;
   wr2: dpa.relating_document :=: adr.assigned_document;
   wr3: dpa.related_product   :=: dp;
   wr4: adr.assigned_document.kind.product_data_type =
'configuration controlled document';
SELECT
   da.assigned_document := document_map(dp);
   da.is_assigned_to    := item_version_map(pdf);
   da.role              := adr.role.name;
END_MAP;

MAP item_version_document_version_assignment_map AS
   da : document_assignment;
FROM
   adr  : applied_document_reference;
   pdf  : product_definition_formation;
   dpa  : document_product_equivalence;
   dp   : product_definition_formation;
WHERE
   wr1: pdf IN adr.items;
   wr2: dpa.relating_document :=: adr.assigned_document;
   wr3: dpa.related_product   :=: dp;
   wr4: adr.assigned_document.kind.product_data_type =
'configuration controlled document version';
SELECT
   da.assigned_document := document_version_map(dp);
   da.is_assigned_to    := item_version_map(pdf);
   da.role              := adr.role.name;
END_MAP;

MAP item_version_document_representation_assignment_map AS
   da : document_assignment;
FROM
   adr  : applied_document_reference;
   pdf  : product_definition_formation;
   dpa  : document_product_equivalence;
   dp   : product_definition;
```

```
WHERE
   wr1: pdf IN adr.items;
   wr2: dpa.relating_document :=: adr.assigned_document;
   wr3: dpa.related_product   :=: dp;
   wr4: adr.assigned_document.kind.product_data_type =
         'configuration controlled document representation';
SELECT
   da.assigned_document := document_representation_map(dp);
   da.is_assigned_to    := item_version_map(pdf);
   da.role              := adr.role.name;
END_MAP;

MAP item_version_digital_file_assignment_map AS
   da : document_assignment;
FROM
   pdf : product_definition_formation;
   adr : applied_document_reference;
   df  : document_file;
WHERE
   wr1: pdf IN adr.items;
   wr2: adr.assigned_document :=: df;
SELECT
   da.assigned_document := digital_file_map(df);
   da.is_assigned_to    := item_version_map(pdf);
   da.role              := adr.role.name;
END_MAP;

MAP ddid_document_assignment_map AS
   da : document_assignment;
FROM
   adr : applied_document_reference;
   pd  : product_definition;
   dpa : document_product_equivalence;
   dp  : product;
WHERE
   wr1: pd IN adr.items;
   wr2: dpa.relating_document :=: adr.assigned_document;
   wr3: dpa.related_product   :=: dp;
   wr4: adr.assigned_document.kind.product_data_type =
'configuration controlled document';
SELECT
   da.assigned_document := document_map(dp);
   da.is_assigned_to    := ddid_map(pd);
   da.role              := adr.role.name;
END_MAP;

MAP ddid_document_version_assignment_map AS
   da : document_assignment;
FROM
   adr  : applied_document_reference;
   pd: product_definition;
```

```
   dpa  : document_product_equivalence;
   dp  : product_definition_formation;
WHERE
   wr1: pd IN adr.items;
   wr2: dpa.relating_document :=: adr.assigned_document;
   wr3: dpa.related_product   :=: dp;
   wr4: adr.assigned_document.kind.product_data_type =
'configuration controlled document version';
SELECT
   da.assigned_document := document_version_map(dp);
   da.is_assigned_to    := ddid_map(pd);
   da.role              := adr.role.name;
END_MAP;

MAP ddid_document_representation_assignment_map AS
   da : document_assignment;
FROM
   adr  : applied_document_reference;
   pd   : product_definition;
   dpa  : document_product_equivalence;
   dp   : product_definition;
WHERE
   wr1: pd IN adr.items;
   wr2: dpa.relating_document :=: adr.assigned_document;
   wr3: dpa.related_product   :=: dp;
   wr4: adr.assigned_document.kind.product_data_type =
         'configuration controlled document representation';
SELECT
   da.assigned_document := document_representation_map(dp);
   da.is_assigned_to    := ddid_map(pd);
   da.role              := adr.role.name;
END_MAP;

MAP ddid_digital_file_assignment_map AS
   da : document_assignment;
FROM
   pd  : product_definition;
   adr : applied_document_reference;
   df  : document_file;
WHERE
   wr1: pd IN adr.items;
   wr2: adr.assigned_document :=: df;
SELECT
   da.assigned_document := digital_file_map(df);
   da.is_assigned_to    := ddid_map(pd);
   da.role              := adr.role.name;
END_MAP;
```

7.4.3.8 Document_content, Document_format, Document_creation

Document property instances are referenced by instances of type Document_representation or Document_file. The reference between the the Document_property instances and the referring instances are build by map calls in digital_file_map and document_representation_map (see Chapter 7.4.3.4 and Chapter 7.4.3.6).

Document property instances are created out of instances of type Representation, depending on the name of the Representation different types of Document_property instances are created:

value of AIM representation name	Type of PIM equivalence model property
document content	document_content_property
document format	document_format_property
document creation	document_creation_property
document size	document_size_property

Additional conditions:

The value of the attribute context_type of the Representation_context instance referenced by the Representation must be 'document parameters'.

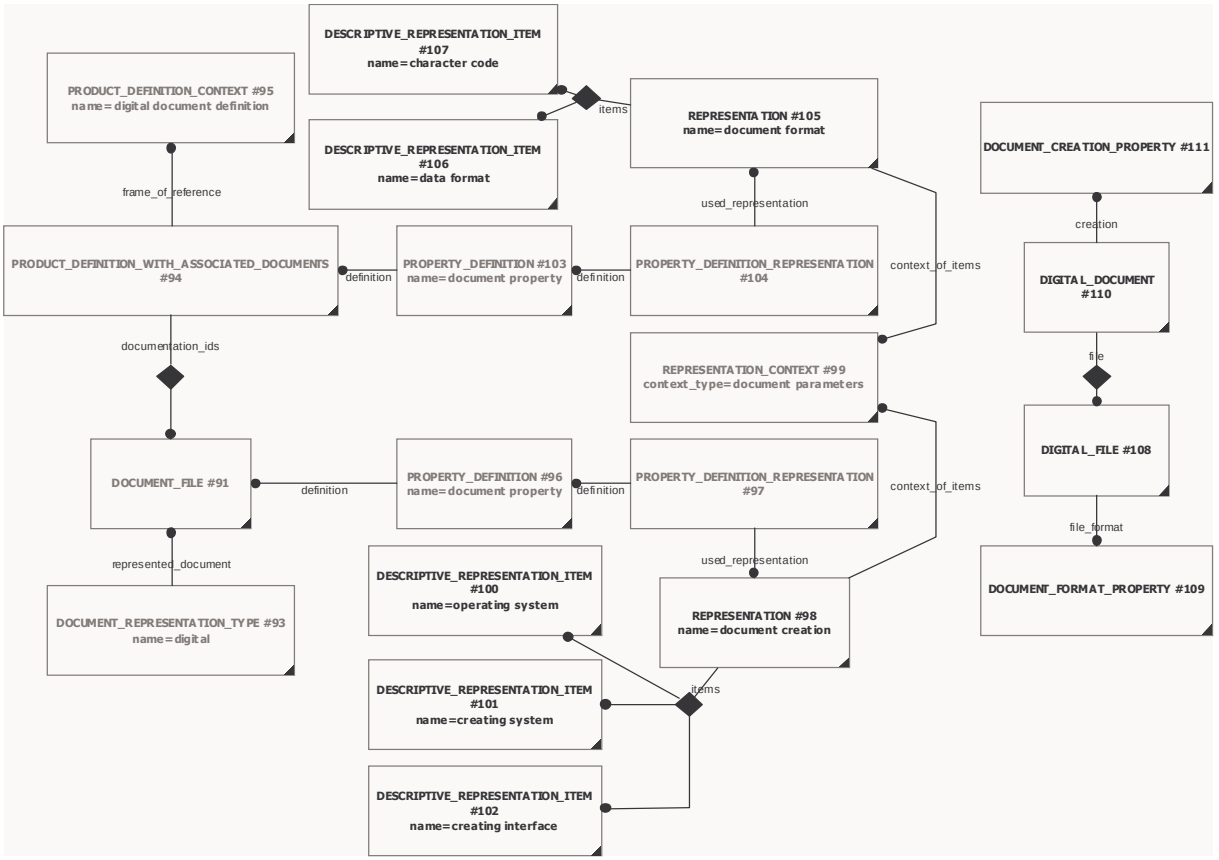


Figure 20 - Instance mapping for document content properties

EXPRESS-X Mapping Specification:

```
MAP document_content_property_map AS
   dcp : document_content_property;
FROM
   rep : representation;
WHERE
   wr1: rep.context_of_items.context_type = 'document parameters';
   wr2: rep.name = 'doument content';
SELECT
   dcp.detail_level  := rep::items{descriptive_representation_item |
                                   name = 'detail level'}[1].description;
   dcp.geometry_type := rep::items{descriptive_representation_item |
                                   name = 'geometry type'}[1].description;
   dcp.real_world_scale := numerical_value_map(rep::items
      {measure_representation_item |
                             name = 'real world scale'}[1]);
   dcp.languages := FOR EACH lang IN rep<-items{language_assignment |
                                                role.name = 'language'}
                                          ::assigned_class{language}
                    RETURN language_map(lang);
END_ MAP;

MAP document_format_property_map AS
   dfp : document_format_property;
FROM
   rep : representation;
WHERE
   wr1: rep.context_of_items.context_type = 'document parameters';
   wr2: rep.name = 'document format';
SELECT
   dfp.character_code := rep::items{descriptive_representation_item |
                                    name = 'character code'}[1].description;
   dfp.data_format := rep::items{descriptive_representation_item |
                                 name = 'data format'}[1].description;
END_MAP;

MAP document_creation_property_map AS
   dcp : document_creation_property;
FROM
   rep : representation;
WHERE
   wr1: rep.context_of_items.context_type = 'document parameters';
   wr2: rep.name = 'document creation';
SELECT
   dcp.creating_interface := rep::items{descriptive_representation_item |
                                        name = 'creating interface'}
                                        [1].description;
   dcp.creating_system    := rep::items{descriptive_representation_item |
                                        name = 'creating system'}
                                        [1].description;
```

```
   dcp.operating_system   := rep::items{descriptive_representation_item |
                                        name = 'operating system'}
                                        [1].description;
END_MAP;

MAP document_size_property_map AS
   dcp : document_size_property;
FROM
   rep : representation;
WHERE
   wr1: rep.context_of_items.context_type = 'document parameters';
   wr2: rep.name = 'document size';
SELECT
   dcp.file_size := value_with_unit_map(rep::items{representation_item |
name = 'file size'}[1]);
   dcp.page_count := value_with_unit_map(rep::items{representation_item |
name = 'page count'}[1]);
END_MAP;
```

7.4.3.9 Document properties

An instance of type Document_location_property is created out of an instance of type External_source which is attached to a Document_representation or Document_file instance by an instance of type Applied_external_identification_assignment.

An instance of type Document_type_property is created out of an instance of type Document_type which is referenced by a Document_file.

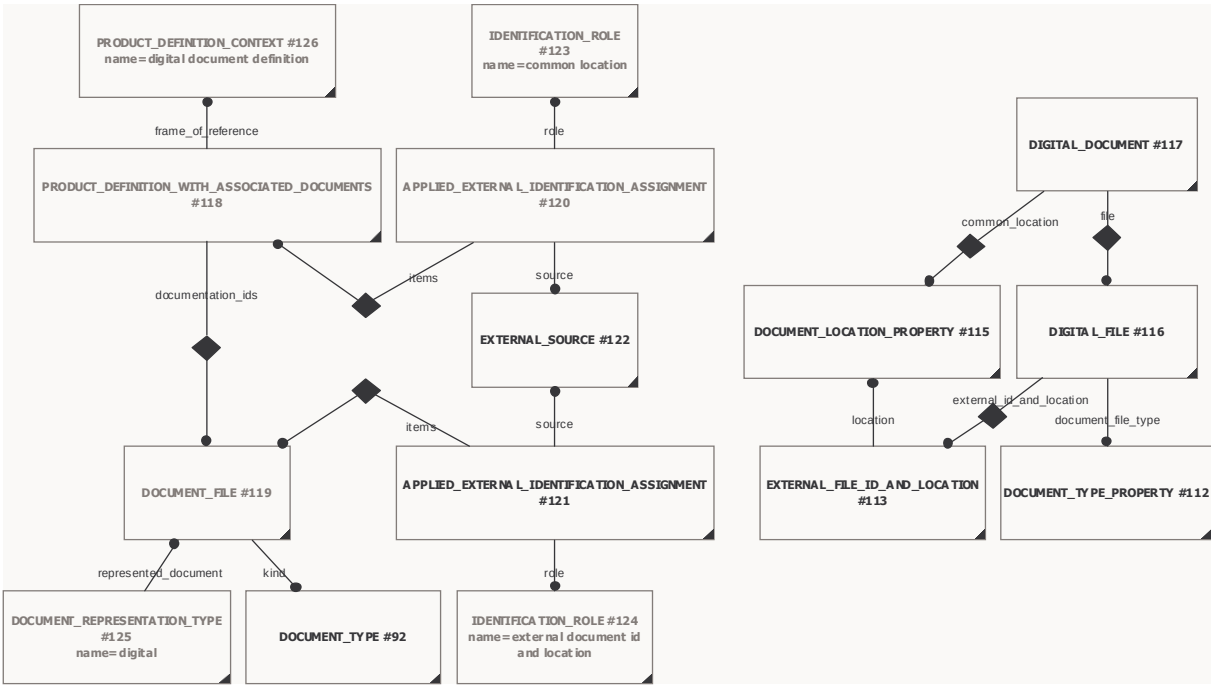


Figure 21 - Instance mapping for document properties

EXPRESS-X Mapping Specification:

```
DEPENDENT_MAP document_location_property_map AS
   dlp : document_location_property;
FROM
   src : external_source;
SELECT
   dlp.location_name := src.source_id;
END_DEPENDENT_MAP;

DEPENDENT_MAP external_file_id_and_location_map AS
   efl : external_file_id_and_location;
FROM
   idx : applied_external_identification_assignment;
SELECT
   efl.external_id := idx.assigned_id;
   efl.location := document_location_property_map(idx.source);
END_DEPENDENT_MAP;

DEPENDENT_MAP document_type_property_map AS
   dt : document_type_property;
FROM
   dtp : document_type;
SELECT
   dt.document_type_name := dtp.product_data_type;
DEPENDENT_END_MAP;
```

7.4.3.10 Rectangular_size, Named_size

A target instance of Rectangular_size is created out of a source instance of complex type Planar_extent with name 'size format'.

If the source instance is of complex type Planar_extent and Descriptive_representation_item, a the target instance of subtype Named_size is created.

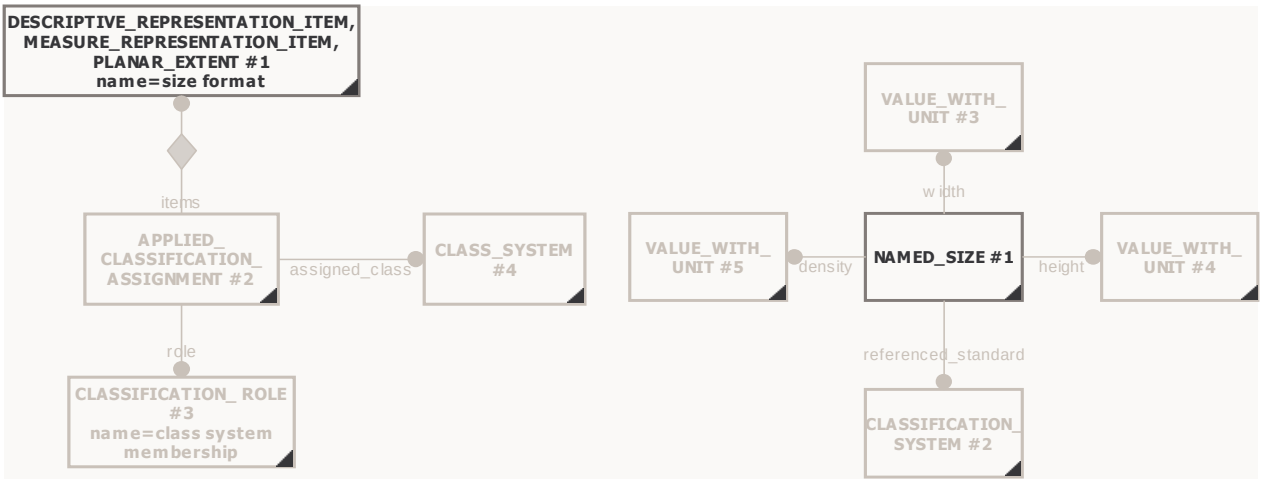


Figure 22 - Instance mapping for named size

EXPRESS-X Mapping Specification:

```
MAP rectangular_size_map AS
   rs : rectangular_size;
FROM
   px : planar_extent;
WHERE
   wr1 : px.name = 'size format';
SELECT
   rs.density := IF 'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM'
                    IN TYPEOF(px)
                 THEN
                  value_with_unit_map(px);
                 END_IF;
   rs.height := value_with_unit_map(px.size_in_y);
   rs.width  := value_with_unit_map(px.size_in_x);
END_MAP;

MAP named_size_map AS
   rs : named_size;
SUBTYPE OF (rectangular_size_map);
   wr2: 'AUTOMOTIVE_DESIGN.DESCRITPITVE_REPRESENTATION_ITEM' IN TYPEOF(px);
SELECT
   nas.size := px.description;
   nas.referenced_standard := classification_system_map(
      px<-items{applied_classification_assignment |
              role.name = 'class system membership'}
         ::assigned_class{class_system}[1]);
END_MAP;
```

7.4.4 Shape Definition and Transformation

7.4.4.1 Item_shape

An instance of type Item_shape is created out of an instance of type Product_definition_shape, that references a Product_definition as definition, which is mapped to an instance of type Design_discipline_item_definition.

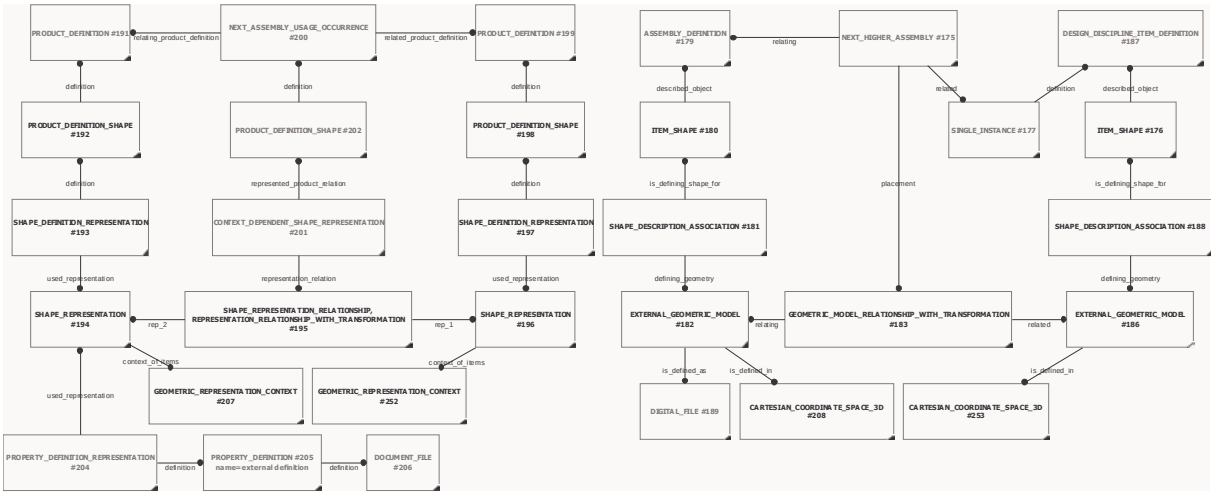


Figure 23 - Instance mapping for item shape

EXPRESS-X Mapping Specification:

```
MAP item_shape_map AS
   is : item_shape;
FROM
   pds : product_definition_shape;
WHERE
   pds.definition.frame_of_reference.name = 'part definition';
SELECT
   is.description      := pds.description;
   is.described_object := ddid_map(pds.definition);
END_MAP;
```

7.4.4.2 Shape_description_association

An instance of type Shape_description_association is created out of an instance of type Shape_definition_representation, that references a Product_definition_shape as definition, which is mapped to Item_shape.

EXPRESS-X Mapping Specification:

```
MAP shape_description_association_map AS
   sda : shape_description_association;
FROM
   sdr : shape_definition_representation;
WHERE
```

```
   sdr.definition.definition.frame_of_reference.name = 'part definition';
SELECT
   sda.is_defining_shape_for :=
               IF 'AUTOMOTIVE_DESIGN.SHAPE_ASPECT'
                  IN TYPEOF(sdr.definition.definition)
               THEN
                 shape_element_map(sdr.definition.definition);
               ELSE
                 item_shape_map(sdr.definition);
               END_IF;
   sda.defining_geometry := geometric_model_map(sdr.used_representation);
   sda.role              := sdr.name;
END_MAP;
```

7.4.4.3 Geometric_model

A target instance of type Geometric_model is created out of source instance of type Shape_representation. that does not refer to an instance of type Representation_context with context_type value 'external'.

EXPRESS-X Mapping Specification:

```
MAP geometric_model_map AS
   gm : geometric_model;
FROM
   sr : shape_representation;
WHERE
   sr.context_of_items.context_type <> 'external';
SELECT
   gm.description := sr.description ;
   gm.model_extent := sr<-rep_1{representation_relationship |
                               name = 'model extent association'}
                       ::rep_2{representation |
                               name = 'model extent representation}
                       ::items{representation_item |
                               name = 'model extent value'}
                       ::value_component;
   gm.model_id     := sr.id;

   gm.is_defined_in :=
      cartesian_coordinate_space_map(sr.context_of_items);
END_MAP;
```

7.4.4.4 External models

An instance of type External_model is created out of an instance of type Representation that references a Representation_context with context_type 'external' and that refers to an instance of type Axis2_placement_3d or Axis2_placement_2d as items.

If the source instance is of type Shape_representation then a target instance of type External_geometric_model is created.

If the source instance refers to a Geometric_representation_context with a coordinate_space_dimension value of 2 as context_of_items, a target instance of type External_picture is created.

EXPRESS-X Mapping Specification:

```
MAP external_model_map AS
   emod : external_model;
FROM
   rep : representation;
WHERE
   wr1 : rep.context_of_items.context_type = 'external';
   wr2 : SIZEOF(rep::items{axis2_placement_3d} +
                rep::items{axis2_placement_2d}) > 0 ;
SELECT
   emod.model_id      := rep.name;
   emod.description   := rep.description ;
   emod.is_defined_as :=
       digital_file_map(sr<-used_representation
                        {property_definition_representation}
                        ::definition{property_definition |
                                     name = 'external definition'}
                        ::definition{document_file}[1]);
   emod.is_defined_in :=
      cartesian_coordinate_space_map(sr.context_of_items);
END_MAP;

MAP external_geometric_model_map AS
   emod : external_geometric_model ;
SUBTYPE OF (external_model_map) ;
WHERE
   'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION' IN TYPEOF(rep);
SELECT
   emod.model_extent := sr<-rep_1{representation_relationship |
                                  name = 'model extent association'}
                        ::rep_2{representation |
                                  name = 'model extent representation}
                        ::items{representation_item |
                                  name = 'model extent value'}
                        ::value_component;
END_MAP;

MAP external_picture_map AS
   emod : external_picture ;
SUBTYPE OF (external_model_map) ;
WHERE
   wr1 : 'AUTOMOTIVE_DESIGN.GEOMETRIC_REPRESENTATION_CONTEXT'
          IN TYPEOF(rep.context_of_items);
   wr2 : rep.context_of_items.coordinate_space_dimension = 2
END_MAP;
```

7.4.4.5 Cartesian_coordinate_space and subtypes

An target Instance of type Cartesian_coordinate_space is created out of a source instance of type Geometric_representation_context.

If the value of the coordinate_space_dimension attribute of the source instance is 2 then a target instance of subtype Cartesian_coordinate_space_2d is created. If the attribute has the value 3, a target instance of type Cartesian_coordinate_space_3d is created.

EXPRESS-X Mapping Specification:

```
MAP cartesian_coordinate_space_map AS
   ccs : cartesian_coordinate_space;
FROM
   grc : geometric_representation_context;
SELECT
   ccs.unit_of_values := FOR EACH un IN grc.units;
                         RETURN unit_map(un);
END_MAP;

MAP cartesian_coordinate_space_2d_map AS
   ccs : cartesian_coordinate_space_2d;
SUBTYPE OF (cartesian_coordinate_space_map);
WHERE
   grc.coordinate_space_dimension = 2 ;
END_MAP;

MAP cartesian_coordinate_space_3d_map AS
   ccs : cartesian_coordinate_space_3d;
SUBTYPE OF (cartesian_coordinate_space_map);
WHERE
   grc.coordinate_space_dimension = 3 ;
END_MAP;
```

7.4.4.6 Accuracy

An target instance of type Accuracy is created out of a source instance of type Global_uncertainty_assigned_context or Uncertainty_assigned_representation or Qualified_representation_item which refers to an instance of type Standard_uncertainty as qualifiers.

EXPRESS-X Mapping Specification:

```
MAP accuracy_map AS
   ac : accurracy;
PARTITION p_gl;
FROM
   gl : global_uncertainty_assigned_context;
SELECT
   ac.accuracy_type := gl.name;
   ac.accuracy_value :=
      gl.uncertainty{ uncertainty_measure_with_unit}[1].value_component;
```

```
   ac.description :=
      gl.uncertainty{ uncertainty_measure_with_unit}[1].description;
   ac.is_defined_for :=
      geometric_model_map(gl<-context_of_items{shape_representation}[1]);
PARTITION p_uar;
FROM
   uar : uncertainty_assigned_context;
SELECT
   ac.accuracy_type := uar.name;
   ac.accuracy_value :=;
      uar.uncertainty{ uncertainty_measure_with_unit}[1].value_component;
   ac.description :=
      uar.uncertainty{ uncertainty_measure_with_unit}[1].description;
   ac.is_defined_for := geometric_model_map(uar);
PARTITION p_qual;
FROM
   qrep : qualified_representation_item;
   stu: standard_uncertainty;
WHERE
   stu IN qrep.qualifiers;
IDENTIFIED_BY qrep;
SELECT
   ac.accuracy_type := stu.measure_name;
   ac.accuracy_value := stu.uncertainty_value;
   ac.description := stu.description;
END_MAP;
```

7.4.4.7 Shape_element

An instance of type Shape_element is created out of an instance of type Shape_aspect.

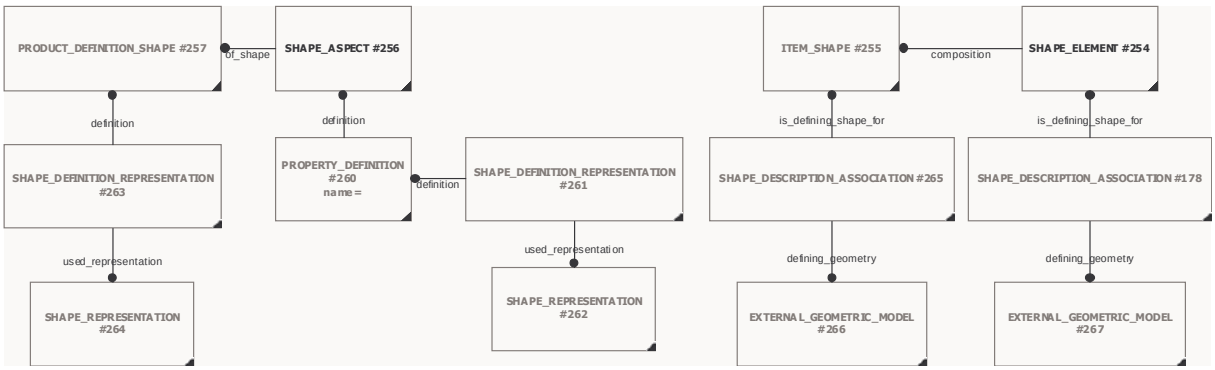


Figure 24 - Instance mapping for shape element

EXPRESS-X Mapping Specification:

```
MAP shape_element_map AS
   se : shape_element;
FROM
```

```
   sa : shape_aspect;
SELECT
   se.description  := sa.description;
   se.element_name := sa.name;
   se.composition  := item_shape_map(sa.of_shape);
END_MAP;
2.4.4.8.Shape_element_relationship
A target instance of shape_element_relationship is created out of a source instance
of type shape_aspect_relationship.
Instance Diagrams:
AIM (PDM SCHEMA, AP214)PIM equivalence model

EXPRESS-X Mapping Specification:
MAP shape_element_relationship_map AS
   ser : shape_element_relationship;
FROM
   sar : shape_aspect_relationship;
SELECT
   ser.description := sar.description;
   ser.relation_type := sar.name;
   ser.related := shape_element_map(sar.related_shape_aspect);
   ser.relating := shape_element_map(sar.relating_shape_aspect);
END_MAP;
```

7.4.4.8 Geometric_model_relationship

A target instance of type Geometric_model_relationship is created out of a source instance of type Shape_representation_relationship. If the source instance is of type Representation_relationship_with_transformation, the target instance will be of type Geometric_model_relationship_with_transformation.

EXPRESS-X Mapping Specification:

```
MAP model_relationship_map AS
   gmr : geometric_model_relationship;
FROM
   srr : shape_representation_relationship;
SELECT
   gmr.relating      := geometric_model_map(srr.rep_2);
   gmr.related       := geometric_model_map(srr.rep_1);
   gmr.relation_type := srr.name;
END_MAP;

MAP model_relationship_trafo_map AS
   gmr : geometric_model_relationship_with_transformation
SUBTYPE OF (model_relationship_map);
WHERE
   'AUTOMOTIVE_DESIGN.REPRESENTATION_RELATIONSHIP_WITH_TRANSFORMATION'
   IN TYPEOF(srr);
SELECT
   gmr.model_placement :=
   transformation_map(srr.transformation_operator);
```

```
END_MAP;
```

7.4.4.9 Transformation, Transformation_3d, Axis2_placement_3d

A target instance of type Transformation_3d is created out of a source instance of type Mapped_item or Item_defined_transformation or Cartesian_transformation_operator_3d. If the source instance is of type Item_defined_transformation or if the source instance is a Mapped_item referencing an Axis2_placement_3d as mapping_target then a target instance of subtype Implicit_transformation_3d is created, otherwise a target instance of subtype Explicit_transformation_3d is created.

EXPRESS-X Mapping Specification:

```
MAP transformation_map AS
   tr : transformation_3d;
PARTITION p_mapped;
FROM
   mit : mapped_item ;
RETURN (mapped_item_map(idt));
PARITITION p_idt ;
FROM
   idt : item_defined_transformation ;
RETURN (implicit_transformation_3d_map(idt));
PARTITION p_fdt ;
FROM
   pdt: cartesian_transformation_operator_3d;
RETURN (explicit_transformation_3d_map(idt));
END_MAP;
```

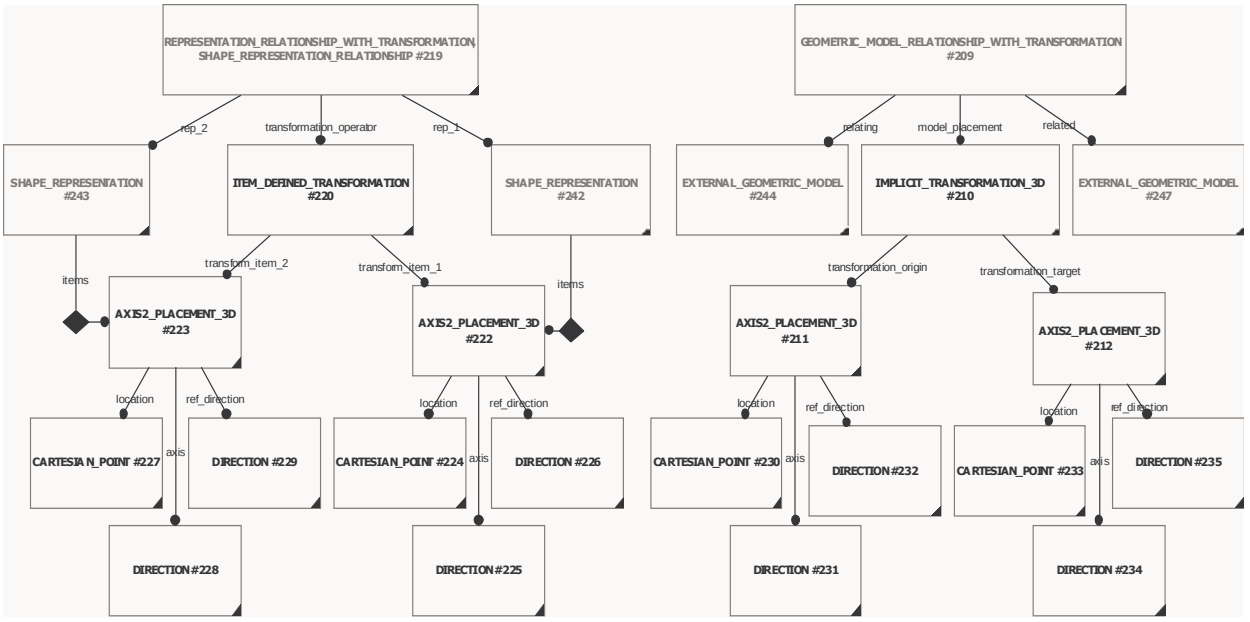


Figure 25 - Instance mapping for implicit transformation

EXPRESS-X Mapping Specification (implicit_transformation):

```
DEPENDENT_MAP implicit_transformation_3d_map AS
   tr : implicit_transformation_3d;
FROM
   idt : item_defined_transformation ;
SELECT
   tr.transformation_target := axis_placement_map(trop.transform_item_2);
   tr.transformation_origin := axis_placement_map(trop.transform_item_1);
END_DEPENDENT_MAP;
```

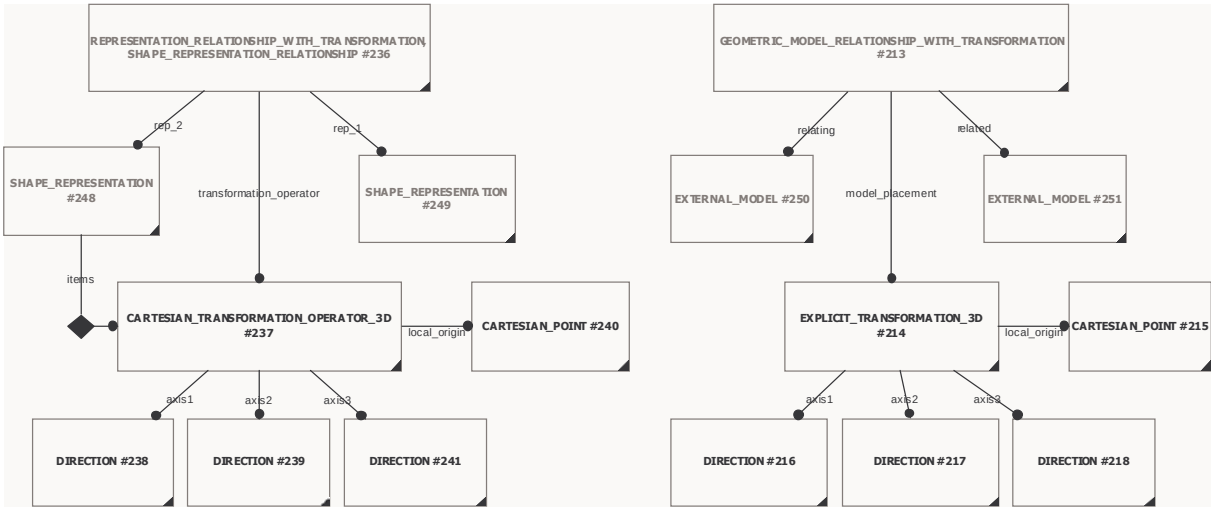


Figure 26 - Instance mapping for explicit transformation

EXPRESS-X Mapping Specification (explicit_transformation):

```
DEPENDENT_MAP explicit_transformation_3d_map AS
   tr : explicit_transformation_3d;
FROM
   cto : cartesian_transformation_operator_3d ;
SELECT
   tr.axis1 := direction_map(trop.axis1);
   tr.axis2 := direction_map(trop.axis2);
   tr.axis3 := direction_map(trop.axis3);
   tr.local_origin := cartesian_point_map(trop.local_origin);
END_DEPENDENT_MAP;

DEPENDENT_MAP mapped_item_map AS
   tr : transformation_3d ;
FROM
   mi : mapped_item ;
END_DEPENDENT_MAP ;

DEPENDENT_MAP mapped_item_explicit_trafo_map AS
   tr : explicit_transformation_3d;
```

```
SUBTYPE OF (mapped_item_map) ;
WHERE
   'AUTOMOTIVE_DESIGN.CARTESIAN_TRANSFORMATION_3D' IN
   TYPEOF(mi.mapping_target) ;
SELECT
   tr.axis1 := direction_map(mi.mapping_target.axis1);
   tr.axis2 := direction_map(mi.mapping_target.axis2);
   tr.axis3 := direction_map(mi.mapping_target.axis3);
   tr.local_origin := cartesian_point_map(mi.mapping_target.local_origin);
END_DEPENDENT_MAP ;

DEPENDENT_MAP mapped_item_implicit_trafo_map AS
   tr : implicit_tranformation_3d ;
SUBTYPE OF (mapped_item_map) ;
WHERE
   'AUTOMOTIVE_DESIGN.AXIS2_PLACEMENT_3D' IN TYPEOF(mi.mapping_target) ;
SELECT
   tr.transformation_target := axis_placement_map(mi.mapping_target);
   tr.transformation_origin := axis_placement_map(mi.mapping_origin);
END_DEPENDENT_MAP ;
```

7.4.4.10 Axis2_placement_3d, Cartesian_point and Direction

Target instances of type Cartesian_point and Direction are created out of the correspondingly named source instances.

EXPRESS-X Mapping Specification:

```
MAP axis_placement_map AS
   t_axpl : axis2_placement_3d;
FROM
   s_axpl : axis2_placement_3d;
SELECT
   t_axpl.location      := cartesian_point_map(s_axpl.location);
   t_axpl.ref_direction := direction_map(s_axpl.ref_direction);
   t_axpl.axis          := direction_map(s_axpl.axis);
END_MAP;

MAP cartesian_point_map AS
   t_cp : cartesian_point;
FROM
   s_cp : cartesian_point;
SELECT
   t_cp.coordinates := s_cp.coordinates;
END_MAP;

MAP direction_map AS
   t_dr : direction;
FROM
   s_dr : direction;
SELECT
   t_dr.direction_ratios := s_dr.direction_ratios;
```

```
END_MAP;
```

7.4.5 Classification

7.4.5.1 Specific item and document classification

An instance of type Specific_item_classification is created out of an instance of type Product_related_product_category if the referenced Product is mapped to an Item. If the referenced Product is mapped to a Document, the instance is mapped to an instance of type Specific_document_classification.

:

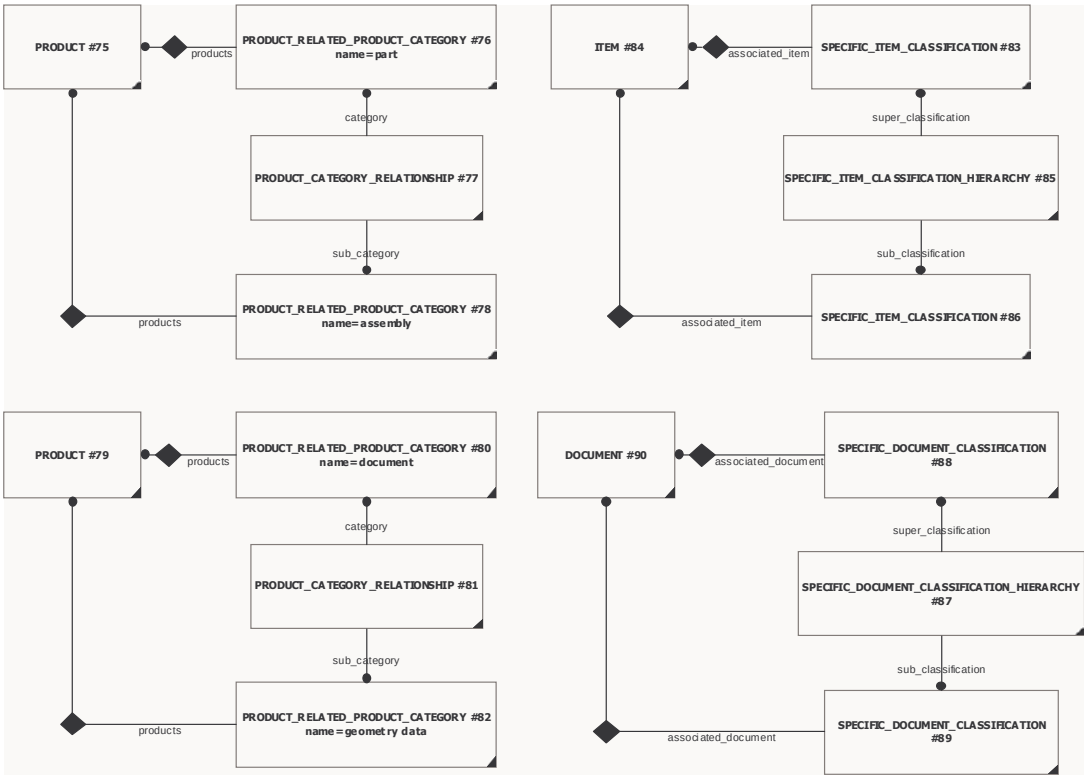


Figure 27 - Instance mapping for specific item and document classification

EXPRESS-X Mapping Specification:

```
MAP item_classification_map AS
   sic : specific_item_classification;
FROM
   prpc : product_related_product_category;
WHERE
   EXISTS(item_map(prpc.products[1]));
SELECT
   sic.associated_item     := FOR EACH p IN prpc.products
                              RETURN item_map(p);
   sic.classification_name := prpc.name;
   sic.description         := prpc.description;
```

```
END_MAP;

MAP item_classification_hierarchy_map AS
   ich : specific_item_classification_hierarchy;
FROM
   prpc : product_category_relationship;
WHERE
   EXISTS(item_classification_map(prpc.category));
SELECT
   ich.sub_classification   := item_classification_map(prpc.sub_category);
   ich.super_classification := item_classification_map(prpc.category);
END_MAP;

MAP document_classification_map AS
   sic : specific_document_classification;
FROM
   prpc : product_related_product_category;
WHERE
   EXISTS(document_map(prpc.products[1]));
SELECT
   sic.associated_document := FOR EACH p IN prpc.products
                              RETURN document_map(p);
   sic.classification_name := prpc.name;
   sic.description         := prpc.description;
END_MAP;

MAP document_classification_hierarchy_map AS
   dch : specific_document_classification_hierarchy;
FROM
   prpc : product_category_relationship;
WHERE
   EXISTS(document_classification_map(prpc.category));
SELECT
   dch.sub_classification := document_classification_map(prpc.sub_category);
   dch.super_classification := document_classification_map(prpc.category);
END_MAP;
```

7.4.5.2 General_classification, General_classification_hierarchy, Classification_system

A target instance of type General_classification is created out of a source instance of type Class. A target instance of type General_classification_hierarchy is created out of a source instance of type Group_relationship with role name 'class system membership'. A target instance of type Classification_sytem is created out of a source instance of type Class_system.

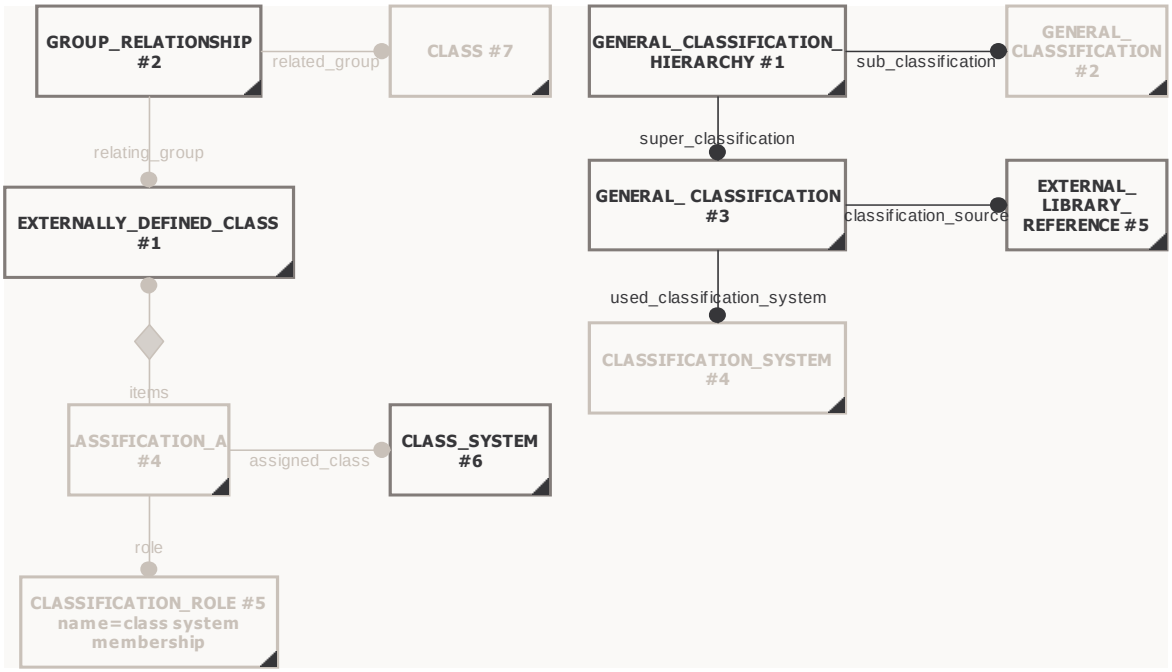


Figure 28 - Instance mapping for general classification

EXPRESS-X Mapping Specification:

```
MAP general_classification_map AS
   gc : general_classification;
FROM
   cl : class;
SELECT
   gc.id := cl\group.name;
   gc.description := cl\group.description;
   gc.version_id := aia<-items{applied_identification_assignment |
                              role.name = 'version'}[1].assigned_id
   gc.classification_source :=
      IF 'AUTOMOTIVE_DESIGN.EXTERNALLY_DEFINED_CLASS' IN TYPEOF(cl) THEN
         external_library_reference_map(cl.source);
      END_IF;
   gc.used_classification_system := classification_system_map(
      cl<-items{applied_classification_assignment |
                role.name = 'class system membership'}
        ::assigned_class{class_system}[1]);
END_MAP;

MAP classification_system_map AS
   cls : classification_system;
FROM
   cs : class_system;
SELECT
```

```
   cls.id := cs.name;
   cls.description := cs.description;
END_MAP;

MAP general_classification_hierarchy_map AS
   gch : general_classification_hierarchy;
FROM
   grel : group_relationship;
WHERE
   grel.name = 'class hierarchy';
SELECT
   gch.sub_classification := general_classification_map(grel.related_group);
   gch.super_classification :=
      general_classification_map(grel.relating_group);
END_MAP;
```

7.4.5.3 External_library_reference

A target instance of type External_library_reference is created out of an instance of type Externally_defined_class or Externally_defined_property which references an instance of exact type External_source.

Figure 29 - Instance mapping for external library reference

EXPRESS-X Mapping Specification:

```
MAP external_library_reference_map AS
   elr : external_library_reference;
FROM
   edi : externally_defined_item;
WHERE
   wr1: SIZEOF(['AUTOMOTIVE_DESIGN.EXTERNALLY_DEFINED_CLASS',
                'AUTOMOTIVE_DESIGN.EXTERNALLY_DEFINED_PROPERTY'] *
               TYPEOF(edi)) > 0;
```

```
   wr2: NOT('AUTOMOTIVE_DESIGN.KNOWN_SOURCE' IN TYPEOF(cls.source));
SELECT
   elr.description  := edi.source.description;
   elr.external_id  := edi.item_id;
   elr.library_type := edi.source.source_id;
END_MAP;
```

7.4.5.4 Classification_association

A target instance of type Classification_association is created out of a source instance of type Applied_classification_assignment.

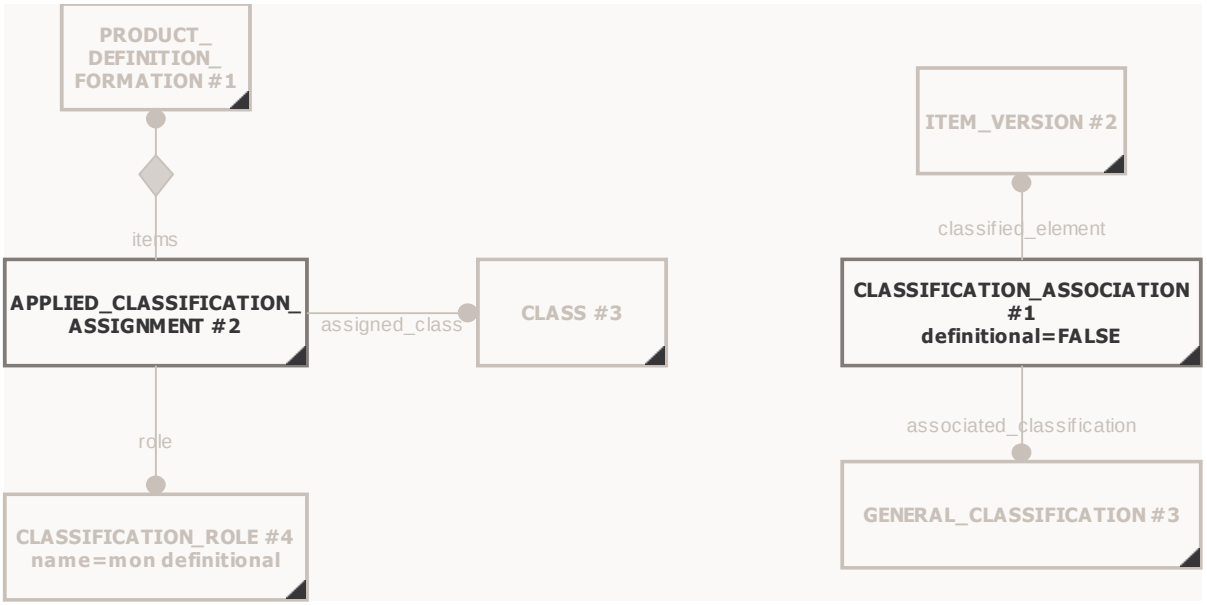


Figure 30 - Instance mapping for classification association

EXPRESS-X Mapping Specification:

```
MAP classification_association_map AS
   ca : classification_association;
FROM
   aca : applied_classification_assignment;
SELECT
   ca.definitional := IF aca.role.name = 'non-definitional' THEN
                        FALSE;
                      ELSE
                        IF aca.role.name = 'definitional' THEN
                          TRUE;
                        END_IF;
                      END_IF;
   ca.role := aca.role.description;
   ca.associated_classification :=
      general_classification_map(ca.assigned_class);
```

```
   ca.classified_element := classified_element_select_map(aca.items[1]);
END_MAP;
```

7.4.5.5 Classification_attribute

A target instance of type Classification_attribute is created out of a source instance of type Property_definition which references an instance of type Characterized_class as definition.

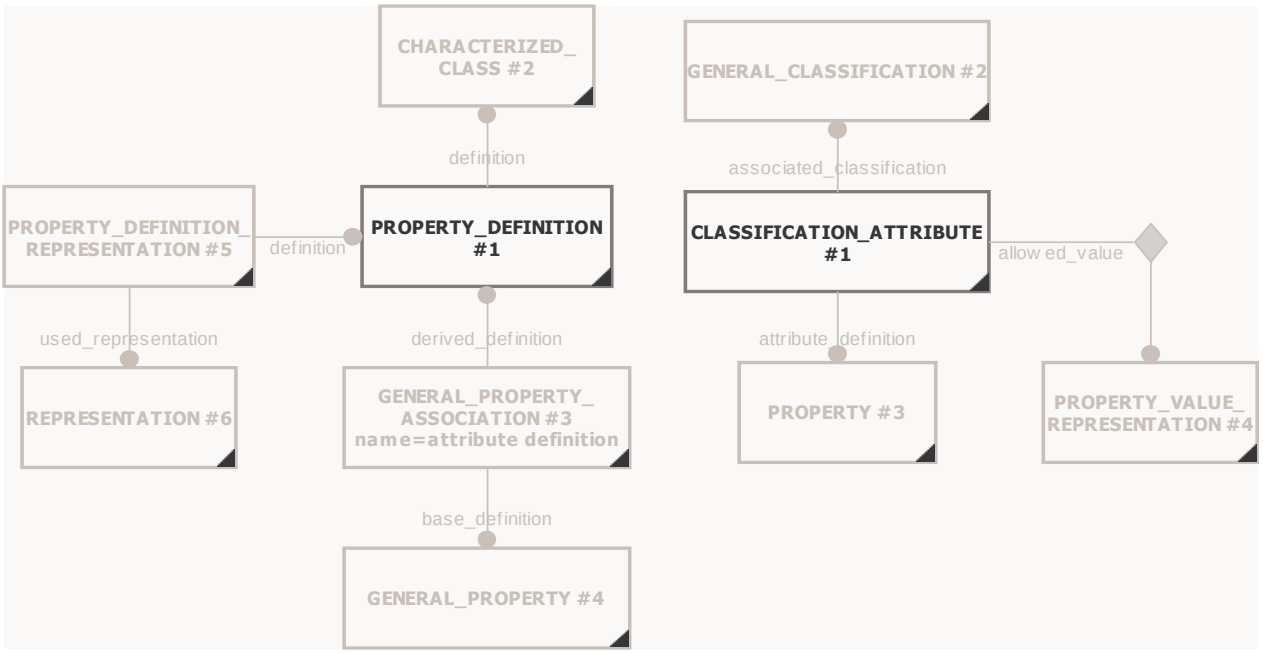


Figure 31 - Instance mapping for classification attribute

EXPRESS-X Mapping Specification:

```
MAP classification_attribute_map AS
   ca : classification_attribute;
FROM
   pd : property_definition;
   cd : characterized_class;
WHERE
   pd.definition :=: cd;
IDENTIFIED_BY pd;
SELECT
   ca.description := pd.description;
   ca.id := pd.id;
   ca.name := pd.name;
   ca.associated_classification := general_classification_map(cd);
   ca.attribute_definition :=
      property_map(pd<-derived_definition{general_property_association |
                                          name = 'attribute definition'}
                   ::base_definition{general_property}[1]);
```

```
   ca.allowed_value := property_value_representation_map(
      pd<-represented_definition{property_definition_representation}
         ::used_representation{representation}[1]);
END_MAP;
```

7.4.6 Properties

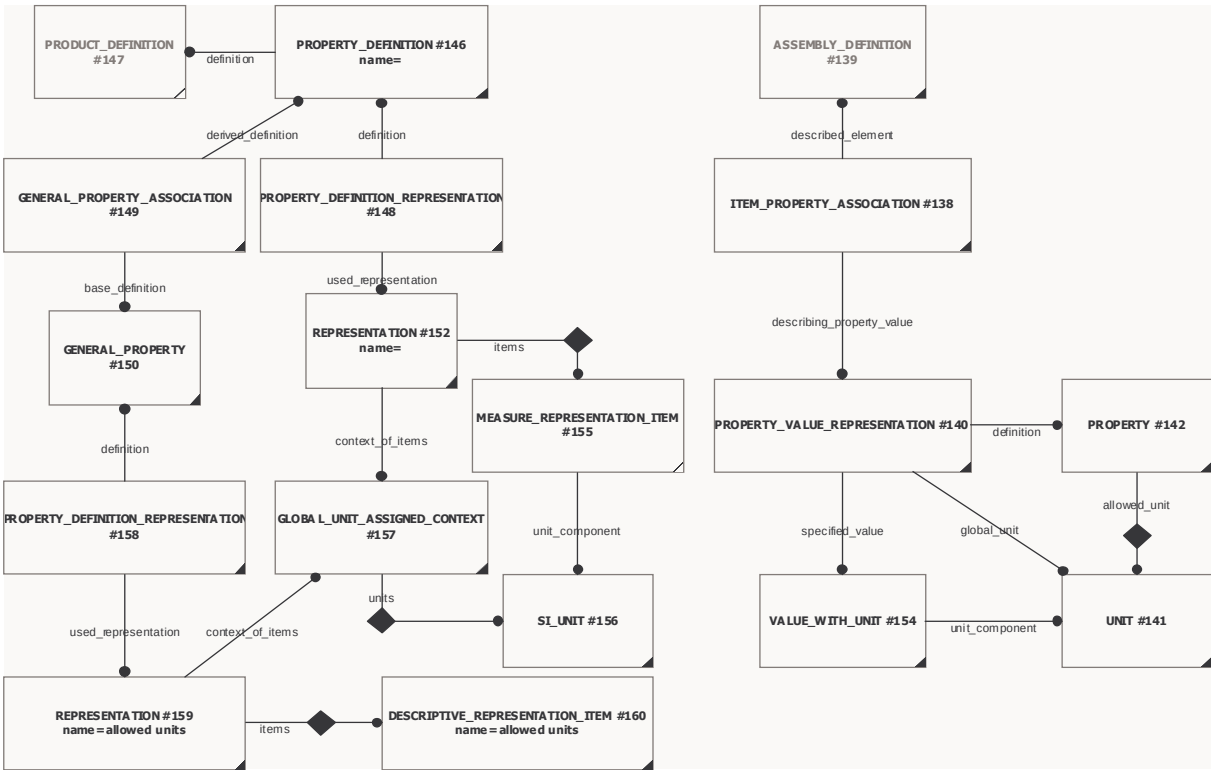


Figure 32 - Instance mapping for properties

7.4.6.1 Property_value_association, Item_property_association

An instance of type Item_property_association is created out of an instance of type Property_definition which is referenced by an instance of type General_property_association as derived_definition.

EXPRESS-X Mapping Specification:

```
MAP property_value_association_map AS
   pv : property_value_association ;
PARTITION p_pdef ;
FROM
   pdef : property_definition ;
RETURN (item_property_association_map(pdef)) ;
PARTITION p_act ;
FROM
   apr : action_property ;
```

```
RETURN (process_property_association_map(apr));
PARTITION p_res ;
FROM
   res : resource_property ;
RETURN (process_property_association_map(res));
END_MAP ;

MAP item_property_association_map AS
   ipa : item_property_association;
FROM
   pdef : property_definition;
   gpa  : general_property_association;
WHERE
   gpa.derived_definition :=: pdef;
IDENTIFIED_BY pdef;
SELECT
   ipa.described_element := item_property_select_map(pdef.definition);
   ipa.describing_property_value :=
         property_value_rep_map(pdef<-definition
                                {property_definition_representation}[1]);
   ipa.definitional := CASE gpa.name OF
                        'definitional'     : TRUE;
                        'non-definitional' : FALSE;
                     END_CASE;
   ipa.description := pdef.description;
   ipa.validity_context :=
      IF SIZEOF(ipa<-items{applied_organization_assignment |
                           name = 'validity context'}) > 0 THEN
      organization_map(ipa<-items{applied_organization_assignment |
                                  name = 'validity context'}
                           ::assigned_organization[1]) ;
      ELSE
         IF SIZEOF(ipa<-related_property_definition
                   {property_definition_relationship |
                    name = 'validity context'}
                  ::relating_property_definition{property_definition |
                    name = 'context definition'}
                  ::definition{product_class}) > 0
         THEN
            product_class_map(ipa<-related_property_definition
                    {property_definition_relationship |
                     name = 'validity context'}
                   ::relating_property_definition{property_definition |
                     name = 'context definition'}
                   ::definition{product_class}[1]);
         ELSE
            product_identification_map(ipa<-related_property_definition
                    {property_definition_relationship |
                     name = 'validity context'}
                   ::relating_property_definition{property_definition |
                     name = 'context definition'}
```

```
                ::definition{product_identification}[1]);
      END_IF;
   END_IF;
END_MAP;

DEPENDENT_MAP item_property_select_map AS
   ips : item_property_select;
PARTITION p_ddid;
FROM
   pd : product_definition;
WHERE
   pd.frame_of_reference.name = 'part definition';
RETURN ddid_map(pd);
PARTITION p_docrep;
FROM
   pd : product_definition;
WHERE
   pd.frame_of_reference.name = 'digital document definition';
RETURN digital_document_map(pd);
PARTITION p_docfile;
FROM
   df : document_file;
WHERE
   EXISTS(digital_file_map(df));
RETURN digital_file_map(df);
END_DEPENDENT_MAP;
```

7.4.6.2 Property

An instance of type Property is created out of an instance of type General_property. If the name of the General_propery is 'mass', an instance of subtype Mass_property is created instead.

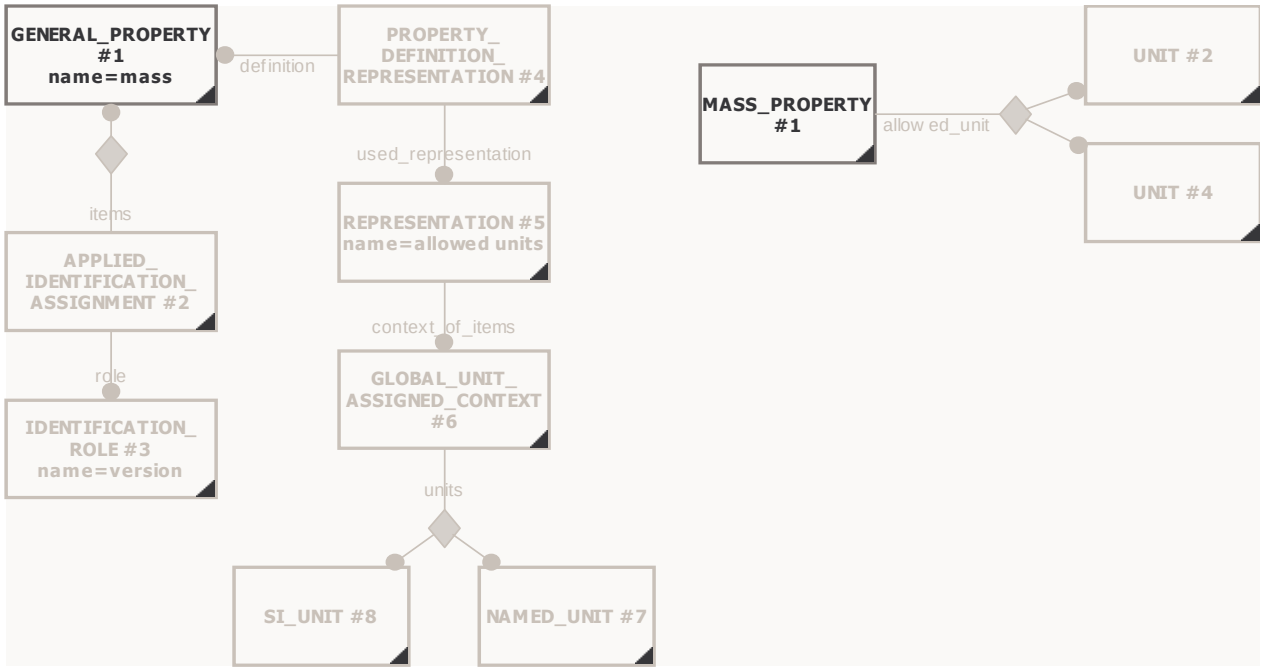


Figure 33 - Instance mapping for mass properties

EXPRESS-X Mapping Specification:

```
MAP property_map AS
   prop : property;
FROM
   gp : general_property;
SELECT
   prop.id              := gp.id;
   prop.description     := gp.description;
   prop.version_id      := gp<-items{applied_identification_assignment |
                                role.name = 'version'}[1].assigned_id;
   prop.allowed_unit := FOR EACH un IN
                         gp<-definition{property_definition_representation}
                           ::used_representation{representation |
                                                  name = 'allowed units'}
                           ::context_of_items{global_unit_assigned_context}
                           ::units
                         RETURN unit_map(un);
END_MAP;

MAP mass_property_map AS
   prop : mass_property;
SUBTYPE OF (property_map);
WHERE
   gp.name = 'mass';
END_MAP;
```

```
MAP cost_property_map AS
   prop : cost_property;
SUBTYPE OF (property_map);
WHERE
   gp.name = 'cost property';
END_MAP;

MAP duration_property_map AS
   prop : duration_property;
SUBTYPE OF (property_map);
WHERE
   gp.name = 'duration property';
END_MAP;

MAP recyclability_property_map AS
   prop : recyclability_property ;
SUBTYPE OF (propery_map) ;
WHERE
   gp.name = 'recyclability property' ;
END_MAP ;

MAP quality_property AS
   prop : quality_property ;
SUBTYPE OF (propery_map) ;
WHERE
   gp.name = 'quality property' ;
END_MAP ;

MAP material_property_map AS
   prop : material_property ;
SUBTYPE OF (property_map);
WHERE
   'AUTOMOTIVE_DESIGN.GENERAL_MATERIAL_PROPERTY' IN TYPEOF(gp) ;
SELECT
   prop.property_name := gp.name ;
END_MAP ;

MAP general_property_map AS
   prop : general_property;
SUBTYPE OF (property_map);
WHERE
   OTHERWISE ;
SELECT
   prop.property_type := gp.name ;
END_MAP;
```

7.4.6.3 Material, Material_property_association

A target instance of Material is created out of a source instance of type Material_designation.

A target instance of type Material_property_association is created out of a source instance of type Material_designation_characterization.

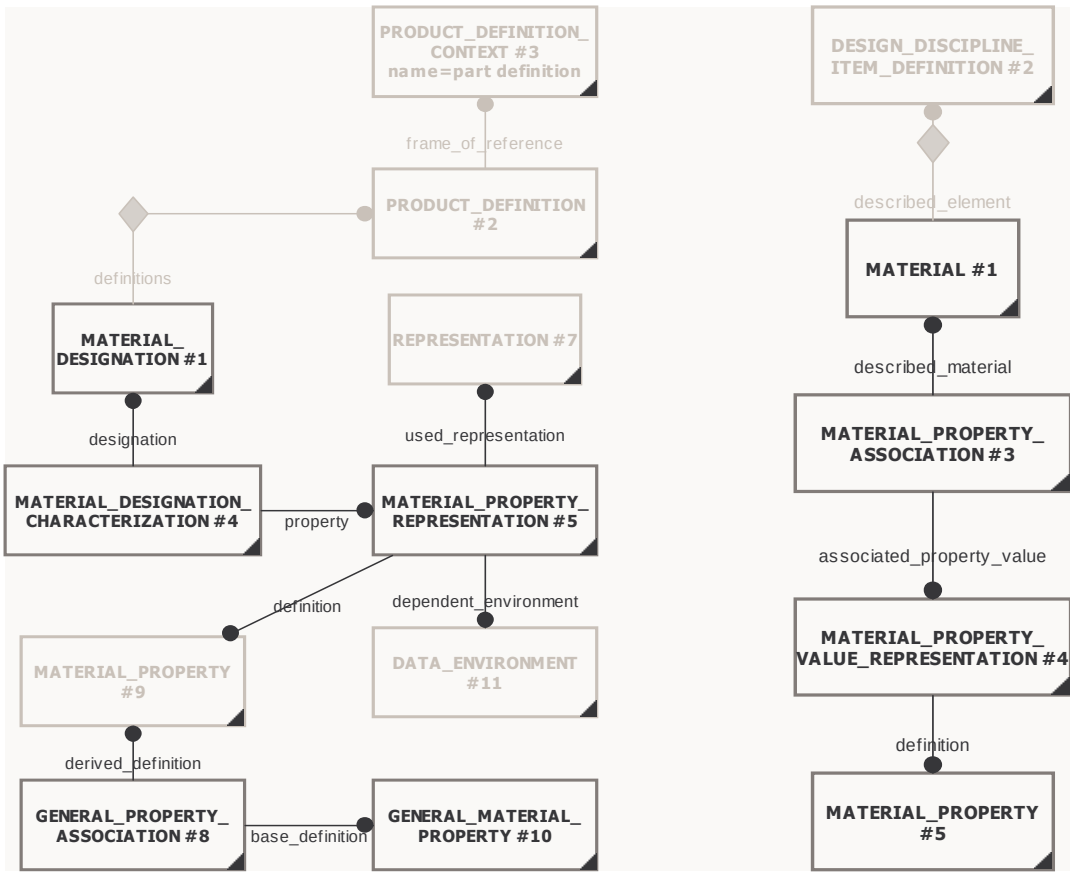


Figure 34 - Instance mapping for material

EXPRESS-X Mapping Specification:

```
MAP material_map AS
   ma : material;
FROM
   mdes : material_designation;
SELECT
   ma.material_name := mdes.name;
   ma.described_element := FOR EACH def IN mdes.definitions;
                           RETURN (item_property_select_map(def));
END_MAP;

MAP material_property_association_map AS
   mpa : material_property_association;
FORM
   mdc : material_designation_characterization;
SELECT
```

```
   mpa.definitional := IF mdc.name = 'definitional' THEN
                         TRUE;
                       ELSE
                         IF mdc.name = 'non-definitional' THEN
                           FALSE;
                         END_IF;
                       END_IF;
   mpa.associated_property_value :=
      material_property_value_representation_map(
         mdc.property.used_representation);
   mpa.described_material := material_map(mdc.designation);
END_MAP;
```

7.4.6.4 Property value representations

An instance of type Property_value_representation is created out of an instance of type Representation which is referenced by an instance of type Property_definition_representation that referenced an instance of type Property_definition as definition, which gets mapped to an instance of type Item_property_association or Material_property_association.

If the Property_definition_representation source instance is of subtype Material_property_representation then a target instance of subtype Material_property_value_representation is created.

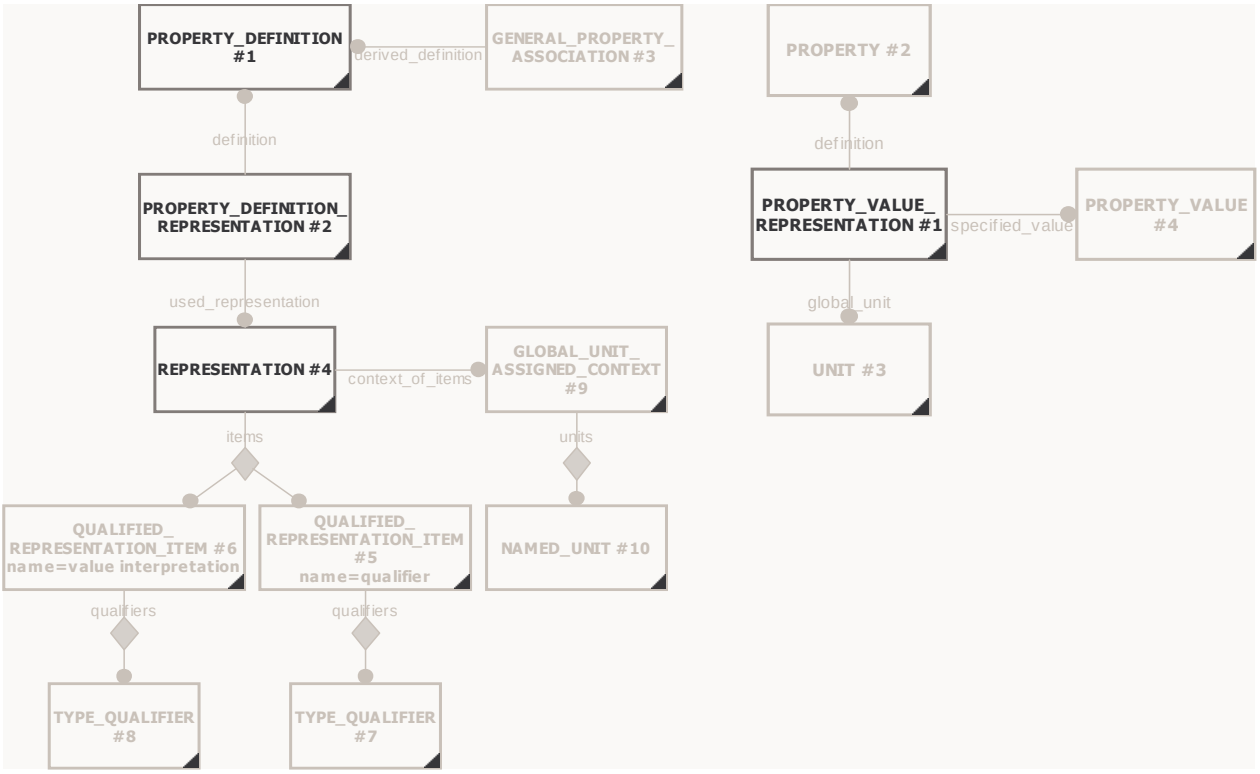


Figure 35 - Instance mapping for property value representation

EXPRESS-X Mapping Specification:

```
MAP property_value_rep_map AS
   pval : property_value_representation;
FROM
   pdef : property_definition;
   pdr  : property_definition_representation;
   gpa  : general_property_association;
   rep  : representation;
WHERE
   wr1: pdr.definition          :=: pdef;
   wr2: gpa.derived_definition  :=: pdef;
   wr3: pdr.used_representation :=: rep;
IDENTIFIED_BY rep;
SELECT
   pval.definition  := property_map(gpa.base_definition);
   pval.global_unit := unit_map(rep
                       ::context_of_items{global_unit_assigned_context}
                       ::units[1]);
   pval.qualifier   := rep::items{qualified_representation_item |
                                  name = 'qualifier'}
                          ::qualifiers{type_qualifier}[1].name;
   pval.value_determination := rep::items{qualified_representation_item |
                                          name = 'value interpretation'}
                                  ::qualifiers{type_qualifier}[1].name;
   pval.specified_value := property_value_map(rep::items{representation_item
                             | (name <> 'qualifier') AND
                               (name <> 'value interpretation')}[1]);
END_MAP;

MAP material_property_value_representation_map AS
   pval : material_property_value_representation;
SUBTYPE OF (property_value_rep_map);
WHERE
   wr4: 'AUTOMOTIVE_DESIGN.MATERIAL_PROPERTY_REPRESENTATION'
        IN TYPEOF(pdr);
SELECT
   pval.environment_condition :=
      data_environment_map(pdr.dependent_environment);
END_MAP;

MAP data_environment_map AS
   t_de : data_environment;
FROM
   s_de : data_environment;
SELECT
   t_de.description := s_de.description;
   t_de.environment_name := s_de.name;
END_MAP;
```

7.4.6.5 Property values

An instance of type Property_value_representation is created out of an instance of type Representation_item which is referenced by a Representation mapped to a Property_value_represenation by an instance as Definition which is mapped to an instance of type Item_property_association.

Depending on the type of the representation_item instance, the target instance will be either a string_value (for Descriptive_representation_item), a Numerical_value or a Value_limit (for Measure_representation_item), a Value_range.(for Value_range) or a Value_list (for Compound_representation_item).

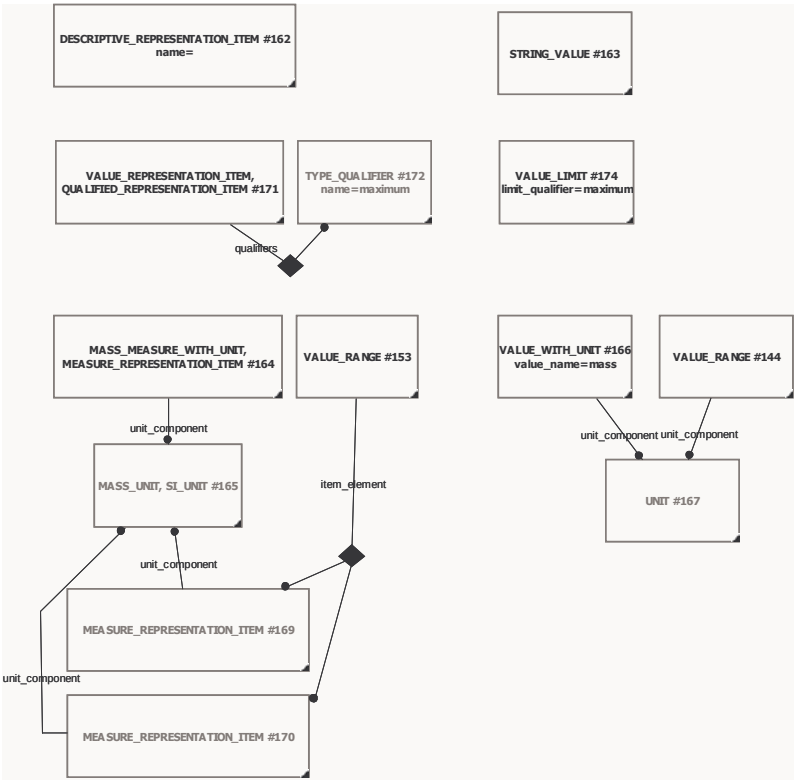


Figure 36 - Instance mapping for property values

EXPRESS-X Mapping Specification:

```
DEPENDENT_MAP property_value_map AS
   pval : property_value;
FROM
   ri : representation_item;
SELECT
   pval.value_name := ri.name;
END_DEPENDENT_MAP;

DEPENDENT_MAP string_value_map AS
   pval : string_value;
SUBTYPE OF (property_value_map);
```

```
WHERE
   'AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(ri);
SELECT
   pval.value_specification := ri.description;
END_DEPENDENT_MAP;

DEPENDENT_MAP value_with_unit_map AS
   pval : value_with_unit;
SUBTYPE OF (property_value_map);
WHERE
   OTHERWISE;
SELECT
   pval.significant_digits := ri::qualifiers{precision_qualifier}
                              [1].precision_value;
END_DEPENDENT_MAP;

DEPENDENT_MAP value_limit_map AS
   pval : value_limit;
SUBTYPE OF (value_with_unit_map);
WHERE
   SIZEOF(ri::qualifiers{type_qualifier |
                         (name = 'maximum') OR
                         (name = 'minimum')}) > 0;
SELECT
   pval.limit_qualifier := ri::qualifiers{type_qualifier |
                           (name = 'maximum') OR
                           (name = 'minimum')}[1].name;
   pval.unit_component  := unit_map(ri::unit_component[1]);
   pval.limit           := ri.value_component;
END_DEPENDENT_MAP;

DEPENDENT_MAP value_range_map AS
   pval : value_range;
SUBTYPE OF (value_with_unit_map);
WHERE
   'AUTOMOTIVE_DESIGN.VALUE_RANGE' IN TYPEOF(ri);
SELECT
   pval.lower_limit    := ri::item_element{representation_item |
                                           name = 'lower limit'}
                           ::value_component[1];
   pval.upper_limit    := ri::item_element{representation_item |
                                           name = 'upper limit'}
                           ::value_component[1];
   pval.unit_component := unit_map(ri::item_element{measure_with_unit}
                                   ::unit_component[1]);
END_DEPENDENT_MAP;

DEPENDENT_MAP numerical_value_map AS
   pval : numerical_value;
SUBTYPE OF (value_with_unit_map);
WHERE
```

```
   OTHERWISE;
SELECT
   pval.value_component := ri.value_component;
   pval.unit_component  := unit_map(ri::unit_component[1]);
END_DEPENDENT_MAP;

DEPENDENT_MAP unit_map AS
   t_un : unit;
FROM
   s_un : unit;
SELECT
   t_un.unit_name := get_unit_name(s_un);
END_DEPENDENT_MAP;
```

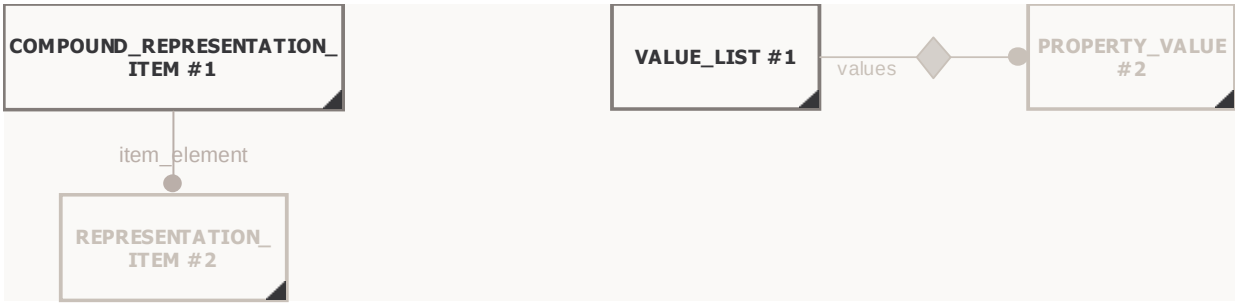


Figure 37 - Instance mapping for value list

EXPRESS-X Mapping Specification for Value_list:

```
DEPENDENT_MAP value_list_map AS
   pval : value_list;
SUBTYPE OF (property_value_map);
WHERE
   'AUTOMOTIVE_DESIGN.COMPOUND_REPRESENTATION_ITEM' IN TYPEOF(ri);
SELECT
   pval.values := FOR EACH elem IN ri.item_element;
                  RETURN (property_value_map(elem));
END_DEPENDENT_MAP;
```

7.4.6.6 Design_contraint, Design_constraint_version, Design_contrained_relationship

A target instance of type Design_constraint is created out of a source instance of type Product_definition_formation which references a Product which is references by a Product_related_product_category with name 'requirement'. In addition the Product_definition_formation source instance must be referenced by an instance of type Product_definition which refers to a Product_definition_context with name 'design constrained definition' as frame_of_reference.

If the id attribute of the Product_definition_formation source instances has a valid value, the target instance is of subtype Design_contrained_version.

A target instance of type Design_contrained_relationship is created out of a source instance of type Product_definition_relationship which refers to Product_definition instances with frame_of_reference name 'design constraint definition' as related_product_definition and as relating_product_definition.

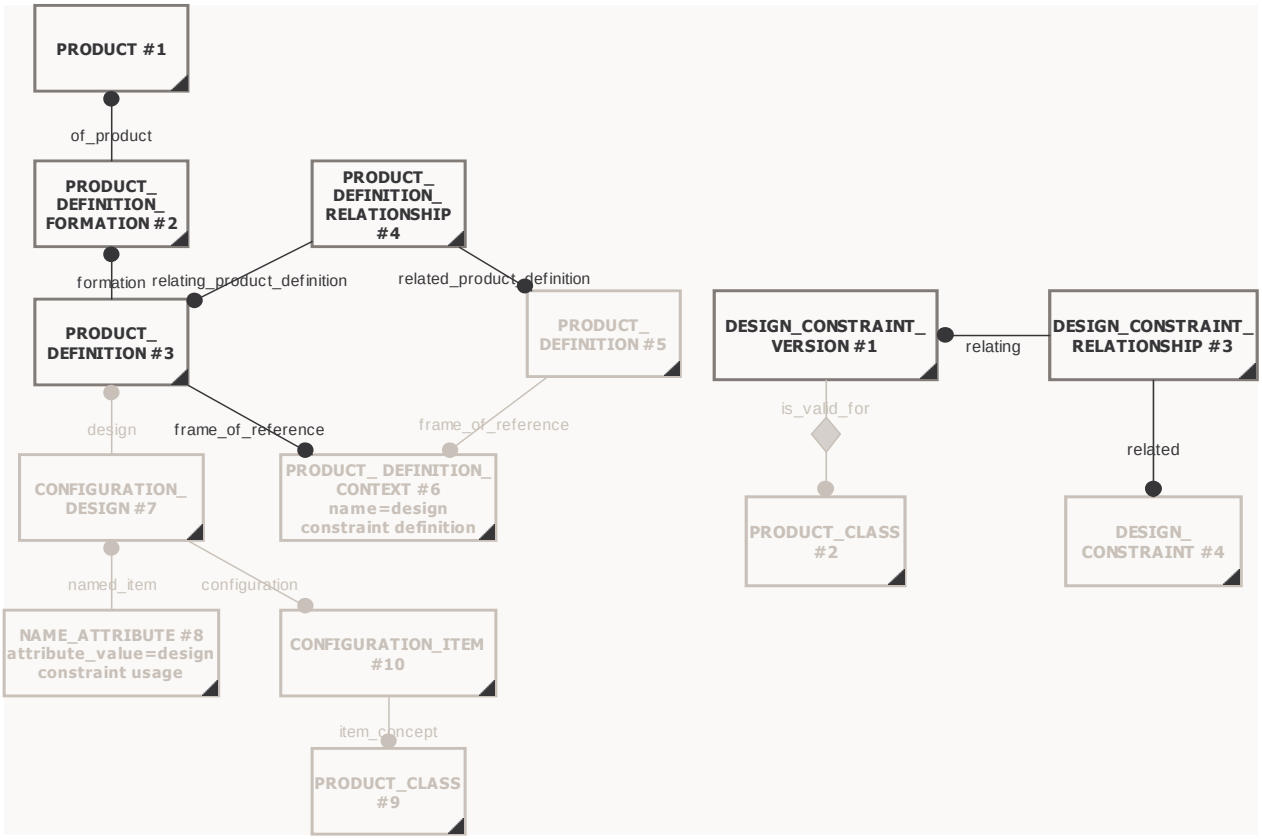


Figure 38 - Instance mapping for design constraint version

EXPRESS-X Mapping Specification:

```
MAP design_constraint_map AS
   dc : design_contraint;
FROM
   pdf : product_definition_formation;
WHERE
   wr1: SIZEOF(pdf<-formation{product_definition |
              frame_of_reference.name='design contrained definition'}) > 0;
   wr2: SIZEOF(pdf.of_product<-products{product_related_product_category |
                                        name = 'requirement'}) > 0;
SELECT
   dc.constraint_id := pdf.of_product.id;
   dc.description := pdf.description;
   dc.name := pdf.of_product.name;
   dc.is_valid_for := product_class_map(
      pdf<-formation{product_definition |
```

```
                  frame_of_reference.name='design constraint definition'}
       <-design{configuration_design | name = 'design constrained usage'}
       ::configuration{configuration_item}
       ::item_concept{product_class}[1]);
END_MAP;

MAP design_contraint_version_map AS
   dc : design_contraint_version;
SUBTYPE OF (design_constraint_map);
WHERE
   wr3: (pdf.id <> '') AND (pdf.id <> '/ANY') AND (pdf.id <> '/NULL');
SELECT
   dc.version_id := pdf.id;
END_MAP;

MAP design_contraint_relationship_map AS
   dcr : design_contraint_relationship;
FROM
   pdr : product_definition_relationship;
WHERE
   wr1: pdr.related_product_definition.frame_of_reference.name =
      'design contraint definition';
   wr2: pdr.relating_product_definition.frame_of_reference.name =
       'design contraint definition';
SELECT
   dcr.related := design_contraint_map(pdr.related_product_definition);
   dcr.relating := design_contraint_map(pdr.relating_product_definition);
   dcr.relation_type := pdr.name;
   drc.description := pdr.description;
END_MAP;
```

7.4.6.7 Design_constraint_association

A target instance of Design_constraint_association is created out of a source instance of type Product_definition_relationship with name 'design constraint association' which refers to a Product_definition with frame_of_reference name 'design constraint definition' as relating_product_definition and which refers to a Product_definition with frame_of_reference name 'alternative definition' or 'conceptual definition' or 'functional definition' as related_product_definition.

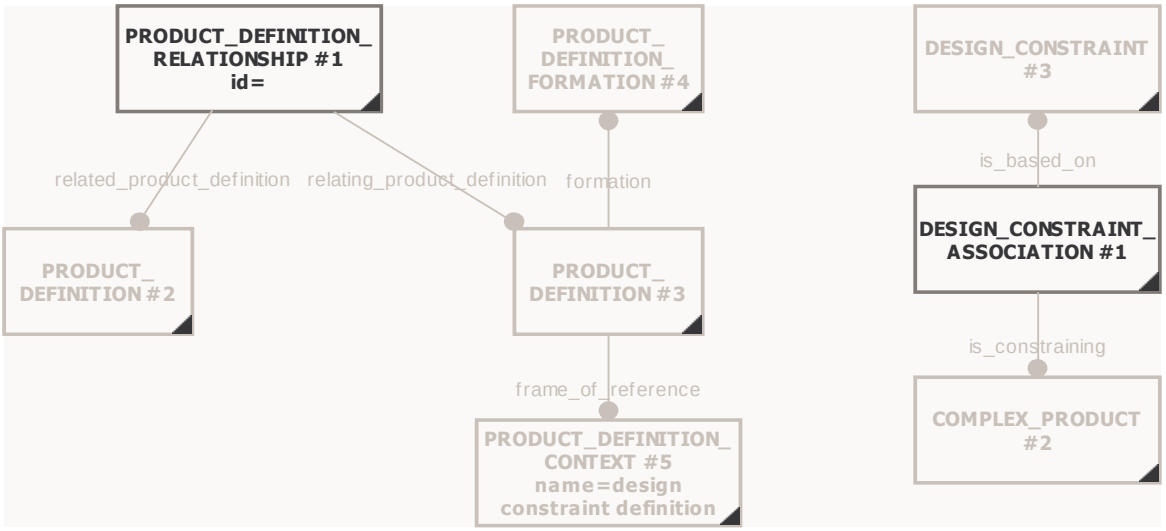


Figure 39 - Instance mapping for design constraint association

EXPRESS-X Mapping Specification:

```
MAP design_constraint_association_map AS
   dca : design_constraint_association;
FROM
   pdr : product_definition_relationship;
WHERE
   wr1: pdr.name = 'design contraint association';
   wr2: pdr.relating_product_definition.frame_of_reference.name =
        'design contraint definition';
   wr3: pdr.related_product_definition.frame_of_reference.name IN
        ['alternative definition', 'conceptual definition',
         'functional definition'];
SELECT
   dca.name := pdr.description;
   dca.is_based_on := design_containt_map(pdr.relating_product_definition);
   dca.is_contraining :=complex_product_map(pdr.related_product_definition);
END_MAP;
```

7.4.6.8 Change

A target instance of Change is created out of a source instance of type Product_definition_formation_relationship or Product_definition_relationship or Action_relationship or Shape_aspect_relationship with name 'change'.

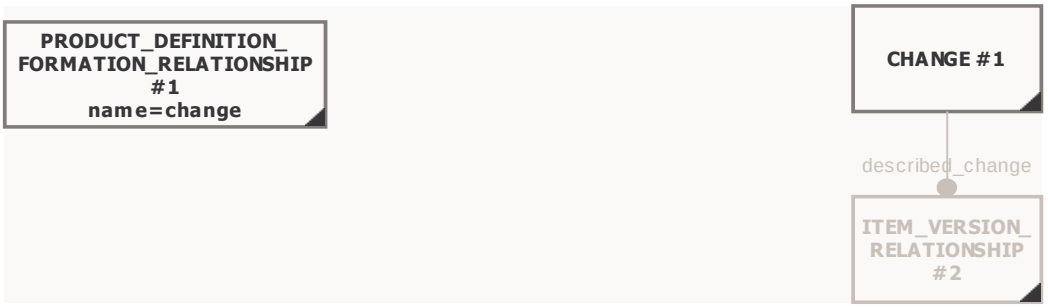


Figure 40 - Instance mapping for change

EXPRESS-X Mapping Specification:

```
MAP change_map AS
   ch : change;
PARTITION p_pdr;
FROM
   pdr : product_definition_relationship;
WHERE
   pdr.name = 'change';
SELECT
   ch.description := pdr.description;
   ch.described_change := change_relationship_select_map(pdr);
PARTITION p_pdfr;
FROM
   pdfr : product_definition_formation_relationship;
WHERE
   pdfr.name = 'change';
SELECT
   ch.description := pdfr.description;
   ch.described_change := change_relationship_select_map(pdfr);
PARTITION p_arel;
FROM
   arel : action_relationship;
WHERE
   arel.name = 'change';
SELECT
   ch.description := arel.description;
   ch.described_change := change_relationship_select_map(arel);
PARTITION p_sarel;
FROM
   sarel : shape_aspect_relationship;
WHERE
   sarel.name = 'change';
SELECT
   ch.description := sarel.description;
   ch.described_change := change_relationship_select_map(sarel);
END_MAP;
```

7.4.7 Alias Identification

7.4.7.1 Alias Identification

An instance of type Alias_identification is created out of an instance of type Applied_identification_assignment, which references an instance of type Identification_role that contains the value 'alias' in its name attribute.

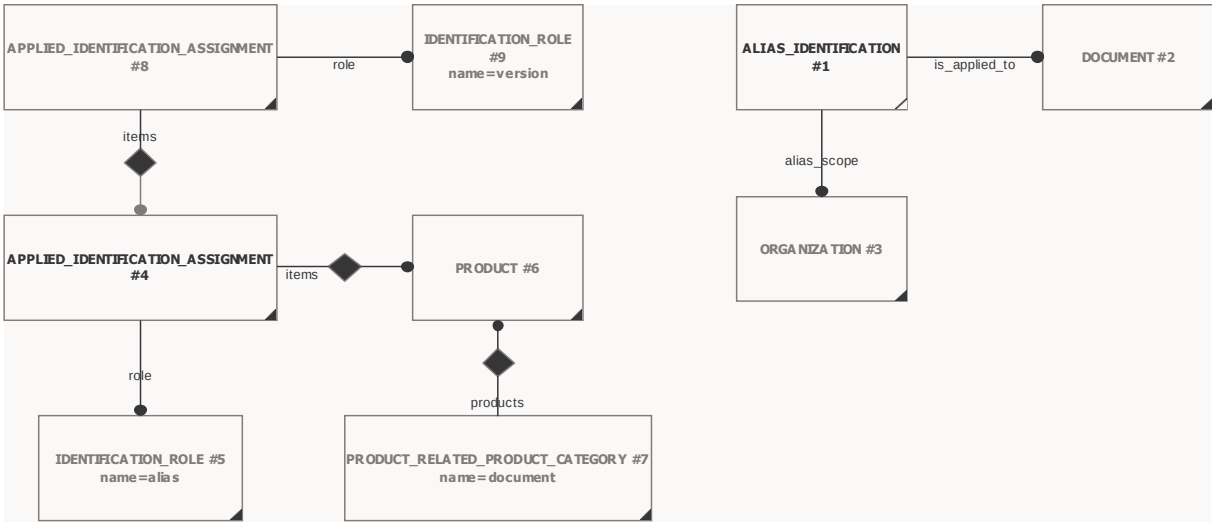


Figure 41 - Instance mapping for alias identification

EXPRESS-X Mapping Specification:

```
MAP alias_identification_map AS
   aid : alias_identification;
FROM
   aia : applied_identification_assignment;
WHERE
   aia.role.name = 'alias';
SELECT
   aid.alias_id         := aia.assigned_id;
   aid.alias_scope      := organization_map(aia
                           <-items{applied_organization_assignment
                           | role.name = 'alias_scope'}[1]
                           .assigned_organization);
   aid.alias_version_id := aia<-items{applied_identification_assignment |
                                   role.name = 'version'}[1].assigned_id;
   aid.description      := aia<-items{applied_identification_assignment |
                           role.name = 'version'}[1].role.description;
   aid.is_applied_to    := IF SIZEOF(aia.items[1]
                              <-products{product_related_product_category
                                        | name = 'document'}) > 0
                           THEN
                             document_map(aia.items[1]);
                           ELSE
```

```
                    item_map(aia.items[1]);
                END_IF;
END_MAP;
```

7.4.8 Authorization

7.4.8.1 Organization, Person and Address

Instances of type Person, Organization and Address are created out of the corresponding instances in AIM (PDM SCHEMA, AP214). An instance of type Person_in_organization in the PIM equivalence model is created out of an instance of type Person_and_organization in AIM (PDM SCHEMA, AP214).

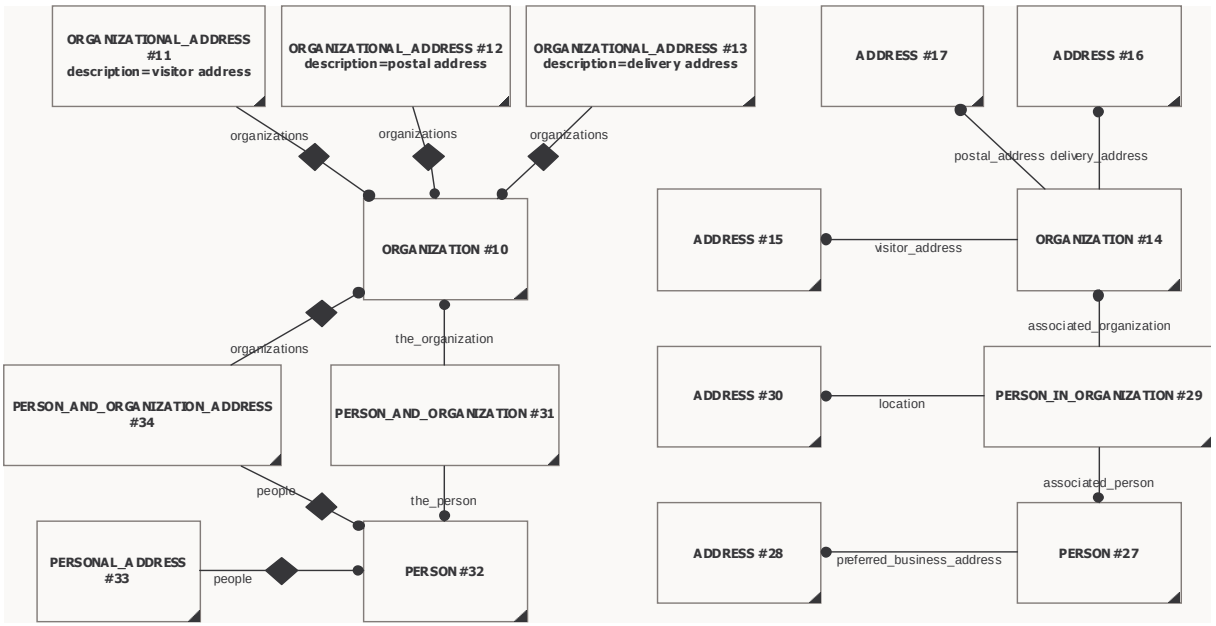


Figure 42 - Instance mapping for organization, person, and address

EXPRESS-X Mapping Specification:

```
MAP address_map AS
   pdtnet_adr : address;
FROM
   ap214_adr : address;
SELECT
   pdtnet_adr.internal_location       := ap214_adr.internal_location;
   pdtnet_adr.street_number           := ap214_adr.street_number;
   pdtnet_adr.street                  := ap214_adr.street;
   pdtnet_adr.postal_box              := ap214_adr.postal_box;
   pdtnet_adr.postal_code             := ap214_adr.postal_code;
   pdtnet_adr.town                    := ap214_adr.town;
   pdtnet_adr.region                  := ap214_adr.region;
   pdtnet_adr.country                 := ap214_adr.country;
   pdtnet_adr.facsimile_number        := ap214_adr.facsimile_number;
```

```
   pdtnet_adr.telephone_number        := ap214_adr.telephone_number;
   pdtnet_adr.telex_number            := ap214_adr.telex_number;
   pdtnet_adr.electronic_mail_address := ap214_adr.electronic_mail_address;
END_MAP;

MAP organization_map AS
   pdtnet_org : organization;
FROM
   ap214_org : organization;
SELECT
   pdtnet_org.organization_name := ap214_org.name;
   pdtnet_org.id                := ap214_org.id;
   pdtnet_org.organization_type := ap214_org.description;
   pdtnet_org.visitor_address   := address_map(ap214_org<- organizations
                                   {organizational_address | description =
                                   'visitor address'}[1]);
   pdtnet_org.delivery_address  := address_map(ap214_org<-organizations
                                   {organizational_address | description =
                                   'delivery address'}[1]);
   pdtnet_org.postal_address    := address_map(ap214_org<-organizations
                                   {organizational_address | description =
                                   'postal address'}[1]);
END_MAP;

MAP person_map AS
   pdtnet_pers : person;
FROM
   ap214_pers : person;
SELECT
   pdtnet_pers.person_name := ap214_pers.first_name + ' ' +
                              ap214_pers.last_name;
   pdtnet_pers.preferred_business_address :=
                              address_map(ap214_pers<-people
                                        {personal_address}[1]);
END_MAP;

MAP person_in_organization_map AS
   pio : person_in_organization;
FROM
   pao : person_and_organization;
SELECT
   pio.id                      := pao<-items
                                  {applied_identification_assignment |
                                  role.name = 'id'}[1].assigned_id;
   pio.associated_person       := person_map(pao.the_person);
   pio.associated_organization := organization_map(pao.the_organization);
   pio.role                    := pao.name;
   pio.location                := address_map(pao.the_person<-people
                                  {person_and_organization_address |
                                  pao.the_organization IN
                                 organizations}[1]);
```

```
END_MAP;
```

7.4.8.2 Date and Time

An instance of type Date_time is created out of an instance of type Date_time or of and instance of type Calendar_date, which is not referenced as Date_component by an instance of type Date_time.

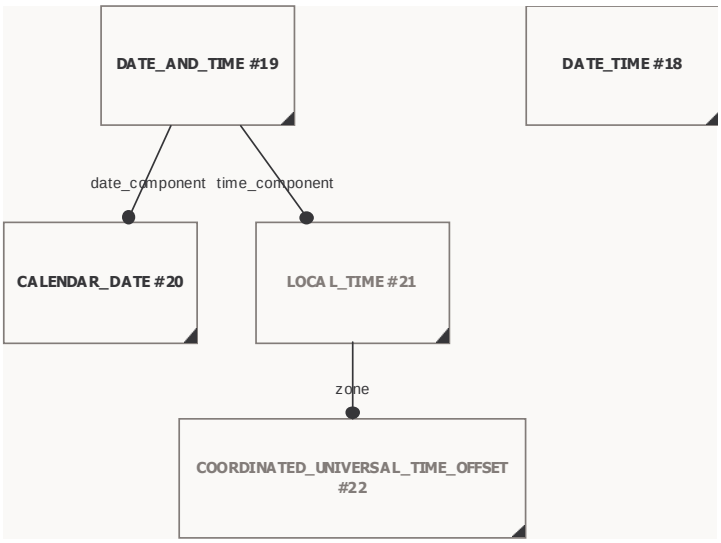


Figure 43 - Instance mapping for date and time

EXPRESS-X Mapping Specification:

```
MAP date_time_map AS
   dat : date_time;
PARTITION p_date_and_time;
FROM
   dt : date_and_time;
SELECT
   dat.date := FORMAT(dt.date_component.year_component,'####') + '/' +
               FORMAT(dt.date_component.month_component,'02I') + '/' +
               FORMAT(dt.date_component.day_component,'02I');
   dat.time := FORMAT(dt.time_component.hour_component, '02I') + ':' +
               FORMAT(dt.time_component.minute_component, '02I') + ':' +
               FORMAT(dt.time_component.second_component, '02I');
PARTITION p_calendar_date;
FROM
   cd : calendar_date;
WHERE
   SIZEOF(cd<-date_component{date_and_time}) = 0;
SELECT
   dat.date := FORMAT(cd.year_component, '####') + '/' +
               FORMAT(cd.month_component, '02I') + '/' +
               FORMAT(cd.day_component, '02I');
END_MAP;
```

7.4.8.3 Date, person and organization

An instance of type Date_and_person_organization in the PIM equivalence model is created out of an instance of type Person_and_organization or Organization which is referenced by an instance of type Applied_date_assignment or Applied_date_and_time_assignment as items. The role name of the Applied_date_assignment or Applied_date_and_time_assignment must have either the value 'actual' or if the date_item is an Approval, the value 'sign off'.

Figure 44 - Instance mapping for date, person and organization

EXPRESS-X Mapping Specification:

```
MAP date_person_organization_map AS
   dpo : date_and_person_organization;
PARTITION p_date_time_pers_org;
FROM
   pao : person_and_organization;
   dta : applied_date_and_time_assignment;
WHERE
```

```
   wr1: pao IN dta.items;
   wr2: dta.role.name = 'actual';
SELECT
   dpo.actual_date            := date_time_map(dta.assigned_date_and_time);
   dpo.person_or_organization := person_in_organization_map(pao);
PARTITION p_date_pers_org;
FROM
   apa : applied_person_and_organization_assignment;
   da  : applied_date_assignment;
WHERE
   wr1: apa IN da.items;
   wr2: da.role.name = 'actual';
SELECT
   dpo.actual_date            := date_time_map(da.assigned_date);
   dpo.person_or_organization := per-
son_in_organization_map(apa.assigned_person_and_organization);
PARTITION p_date_time_org;
FROM
   aoa : applied_organization_assignment;
   dta : applied_date_and_time_assignment;
WHERE
   wr1: aoa IN dta.items;
   wr2: dta.role.name = 'actual';
SELECT
   dpo.actual_date            := date_time_map(dta.assigned_date_and_time);
   dpo.person_or_organization := organization_map(aoa.assigned_organization);
PARTITION p_date_org;
FROM
   aoa : applied_organization_assignment;
   da  : applied_date_assignment;
WHERE
   wr1: aoa IN da.items;
   wr2: da.role.name = 'actual';
SELECT
   dpo.actual_date           := date_time_map(da.assigned_date);
   dpo.person_or_organization:= organization_map(aoa.assigned_organization);
PARTITION p_approval_date_time;
FROM
   apo : approval_person_organization;
   dta : applied_date_and_time_assignment;
WHERE
   wr1: apo IN dta.items;
   wr2: dta.role.name = 'sign off';
SELECT
   dpo.actual_date            := date_time_map(dta.assigned_date_and_time);
   dpo.person_or_organization :=
                      person_in_organization_map(apo.person_organization);
PARTITION p_approval_date;
FROM
   apo : approval_person_organization;
   da : applied_date_assignment;
```

```
WHERE
   wr1: apo IN da.items;
   wr2: da.role.name = 'sign off';
SELECT
   dpo.actual_date            := date_time_map(da.assigned_date);
   dpo.person_or_organization :=
            person_in_organization_map(apo.person_organization);
END_MAP;
```

7.4.8.4 Person organization assignment

An instance of type Person_organization_assignment is created out of an instance of type Applied_person_and_organization_assignment or Applied_organization_assignment.

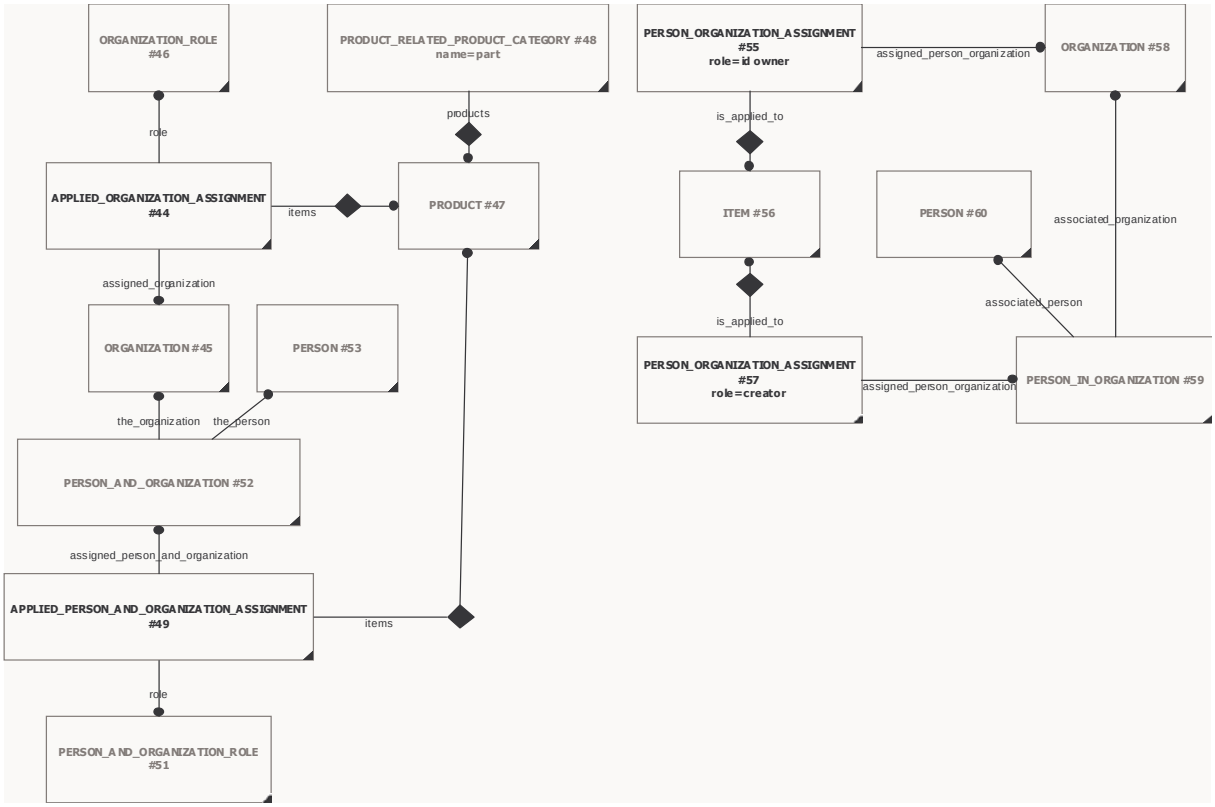


Figure 45 - Instance mapping for person organization assignment

EXPRESS-X Mapping Specification:

```
MAP person_organization_assignment_map AS
  poa : person_organization_assignment;
PARTITION p_org;
FROM
  aoa : applied_organization_assignment;
SELECT
```

```
  poa.assigned_person_organization :=
                      organization_map(aoa.assigned_organization);
  poa.role                          := aoa.role.name;
  poa.description                   := aoa.role.description;
  poa.is_applied_to                 := FOR EACH it IN aoa.items
                                         RETURN org_select_map(it);
PARTITION p_pers_org;
FROM
  apa : applied_person_and_organization_assignment;
SELECT
  poa.assigned_person_organization :=
         person_in_organization_map(apa.assigned_person_and_organization);
  poa.role                          := apa.role.name;
  poa.description                   := apa.role.description;
  poa.is_applied_to                 := FOR EACH it IN apa.items
                                         RETURN org_select_map(it);
END_MAP;

DEPENDENT_MAP org _select_map AS
  god : general_organizational_data_select;
PARTITION p_item;
FROM
  p : product;
WHERE
  EXISTS(item_map(p));
RETURN item_map(p);
PARTITION p_item_version;
FROM
  pdf : product_definition_formation;
WHERE
  EXISTS(item_version_map(pdf));
RETURN item_version_map(pdf);
PARTITION p_document;
FROM
  p : product;
WHERE
  EXISTS(document_map(p));
RETURN document_map(p);
PARTITION p_document_version;
FROM
  pdf : product_definition_formation;
WHERE
  EXISTS(document_version_map(pdf));
RETURN document_version_map(pdf);
END_DEPENDENT_MAP;
```

7.4.8.5 Date and person assignment

An instance of type Date_and_person_assignment is created out of an instance of type Applied_person_and_organization_assignment or Applied_organization_assignment.

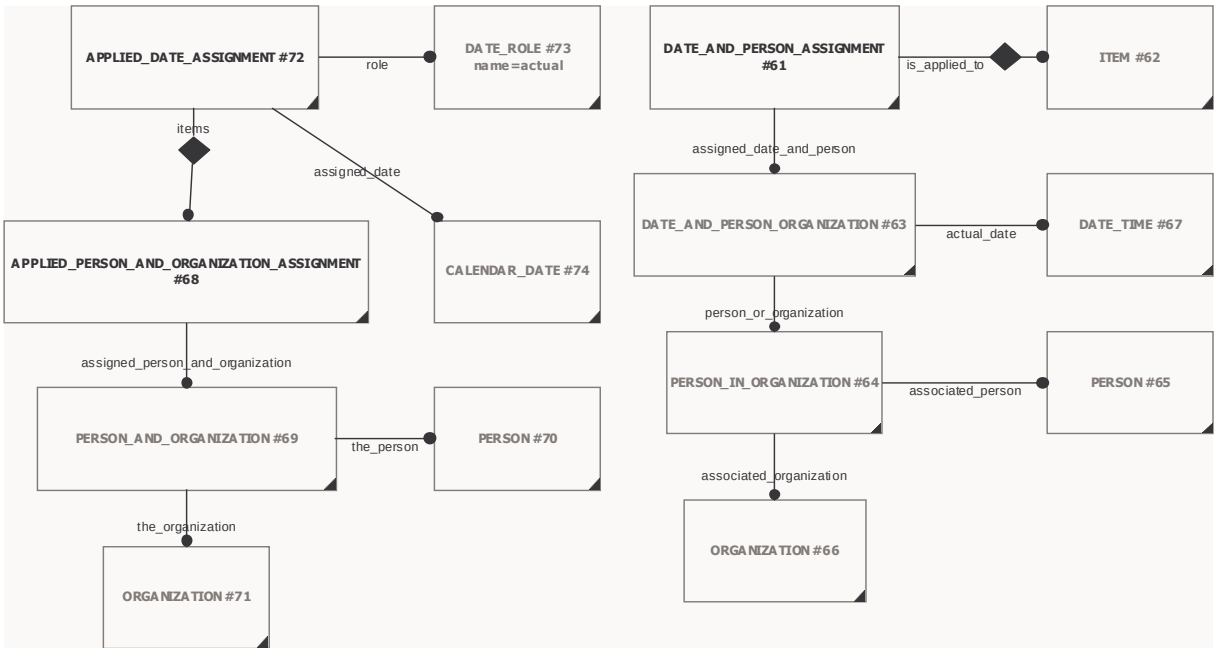


Figure 46 Instance mapping for date and person assignment

EXPRESS-X Mapping Specification:

```
MAP date_and_person_assignment_map AS
   dpa : date_and_person_assignment;
PARTITION p_date_time_pers_org;
FROM
   aoa : applied_organization_assignment;
   dta : applied_date_and_time_assignment;
WHERE
   wr1: aoa IN dta.items;
   wr2: dta.role.name = 'actual';
SELECT
   dpa.assigned_date_and_person := date_person_organization_map(aoa, dta);
   dpa.is_applied_to            := FOR EACH it IN aoa.items
                                       RETURN org_select_map(it);
   dpa.role                     := aoa.role.name;
   dpa.description             := aoa.role.description;
PARTITION p_date_pers_org;
FROM
   apa : applied_person_and_organization_assignment;
   da  : applied_date_assignment;
WHERE
   wr1: apa IN da.items;
   wr2: da.role.name = 'actual';
SELECT
   dpa.assigned_date_and_person := date_person_organization_map(apa, da);
   dpa.is_applied_to            := FOR EACH it IN apa.items
```

```
                                    RETURN org_select_map(it);
   dpa.role                      := apa.role.name;
   dpa.description               := apa.role.description;
END_MAP;
```

7.4.8.6 Date_time_assignment

A target Instance of Date_time_assignment is created out of a source instance of type Applied_date_and_time_assignment or Applied_date_assignment.

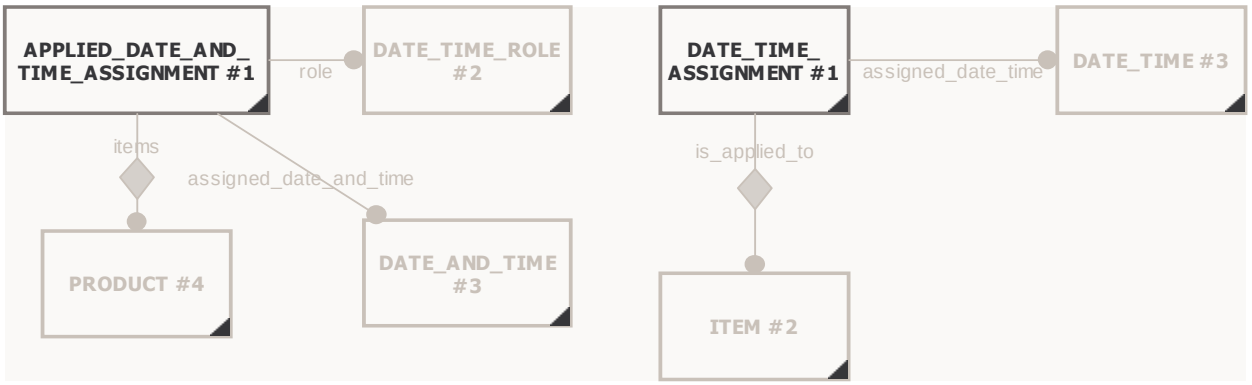


Figure 47 - Instance mapping for date time assignment

EXPRESS-X Mapping Specification:

```
MAP date_time_assignment_map AS
   dta : date_time_assignment;
PARTITION p_date_time;
FROM
   adta : applied_date_and_time_assignment;
SELECT
   dta.description := adta.role.description;
   dta.role        := adta.role.name;
   dta.assigned_date_time := data_time_map(adta.assigned_date_and_time);
   dta.is_applied_to :=
      FOR EACH it IN adta.items;
         RETURN (data_time_person_organization_element_select_map(it));
PARTITION p_date;
FROM
   ada : applied_date_assignment;
SELECT
   dta.description := ada.role.description;
   dta.role        := ada.role.name;
   dta.assigned_date_time := data_time_map(ada.assigned_date);
   dta.is_applied_to :=
      FOR EACH it IN ada.items;
         RETURN (data_time_person_organization_element_select_map(it));
END_MAP;
```

7.4.8.7 Approval, Approval_status and Approval_relationship

A target instance of type approval is created out of a source instance of type Approval, a target instance of type Approval_status is created out of a source instance of type Approval_status. Target instances of type Approval_relationship are created out of source instances of type Approval_relationship.

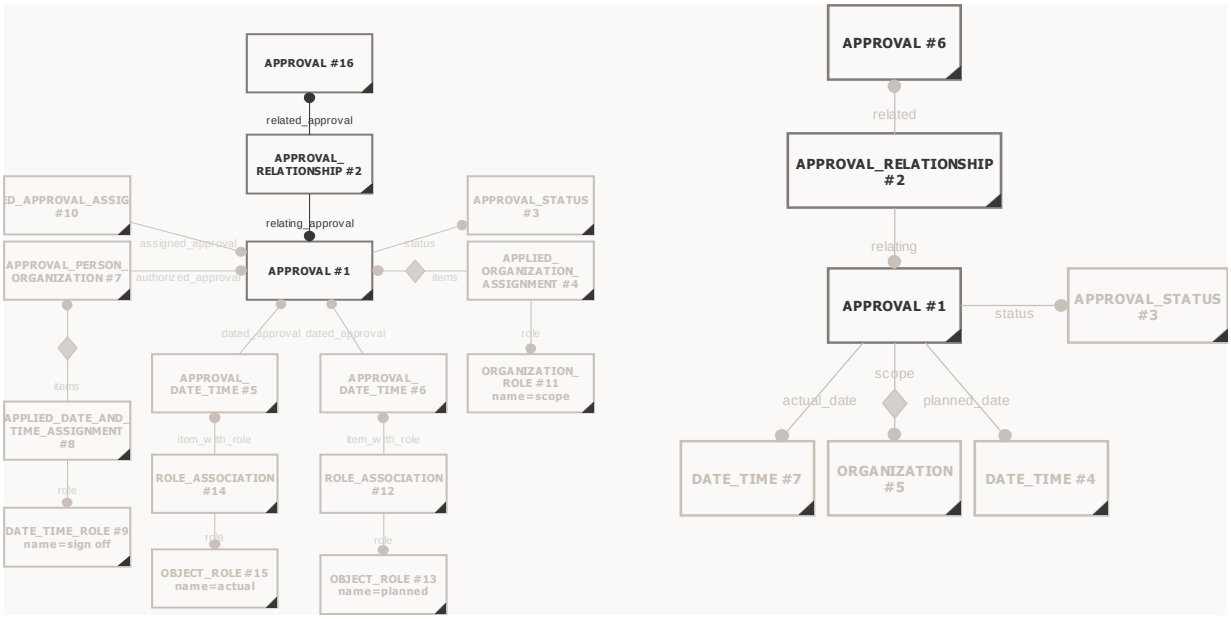


Figure 48 - Instance mapping for approval

EXPRESS-X Mapping Specification:

```
MAP approval_map AS
   tgt : approval;
FROM
   src : approval;
SELECT
   tgt.level          := src.level;
   tgt.status         := approval_status_map(src.status);
   tgt.is_applied_to  :=
      FOR EACH it IN
         src_app<-assigned_approval{applied_approval_assignment}::items;
         RETURN (approval_element_select_map(it));
   tgt.scope          :=
      organization_map(src<-items{applied_organization_assignment |
                                  role.name = 'scope'}
                        ::assigned_organization{organization}[1]);
   tgt.planned_date:=
      date_time_map(src<-dated_approval{approval_date_time |
                                        role.name = 'planned'}
                     ::date_time[1]);
   tgt.actual_date    :=
```

```
      date_time_map(src<-dated_approval{approval_date_time |
                                        role.name = 'actual'}
                    ::date_time[1]);
   tgt.is_approved_by := FOR EACH it IN
      src<-authorized_approval{approval_person_organization |
      approval_person_organization<-items{applied_date_and_time_assignment |
                                         role.name = 'sign off'};
      RETURN date_person_organization_map(it.person_organization,
             it<-items{applied_date_and_time_assignment}[1]);
END_MAP;

MAP approval_status_map AS
   tgt : approval_status;
FROM
   src : approval_status;
SELECT
   tgt_stat.status_name := src_stat.name;
   tgt_stat.used_classification_system :=
      classification_system_map(
         src<-items{applied_classification_assignment |
                 role.name = 'class system membership'}
         :: assigned_class{class_system}[1]);
END_MAP;

MAP approval_relationship_map AS
   tgt_apr : approval_relationship;
FROM
   src_apr : approval_relationship;
SELECT
   tgt_apr.description   := src_apr.description;
   tgt_apr.relation_type := src_apr.name;
   tgt_apr.related       := approval_map(src_apr.related_approval);
   tgt_apr.relating      := approval_map(src_apr.relating_approval);
END_MAP;
```

7.4.9 Configuration Management

7.4.9.1 Product_class and relationships

A target instance of type Product_class is created out of a source instance of type Product_class. An instance of type Product_class_relationship is created of an instance of type Product_concept_relationship, which references instances of type Product_class as related_product_concept and as relating_product_concept. An instance of type Class_structure_relationship is created of of an instance of type Configuration_design that references an instance of type Product_class as Configuration.item_concept and that references Product_definitions with frame_of_reference.name values 'conceptual definition' or 'function definition'.

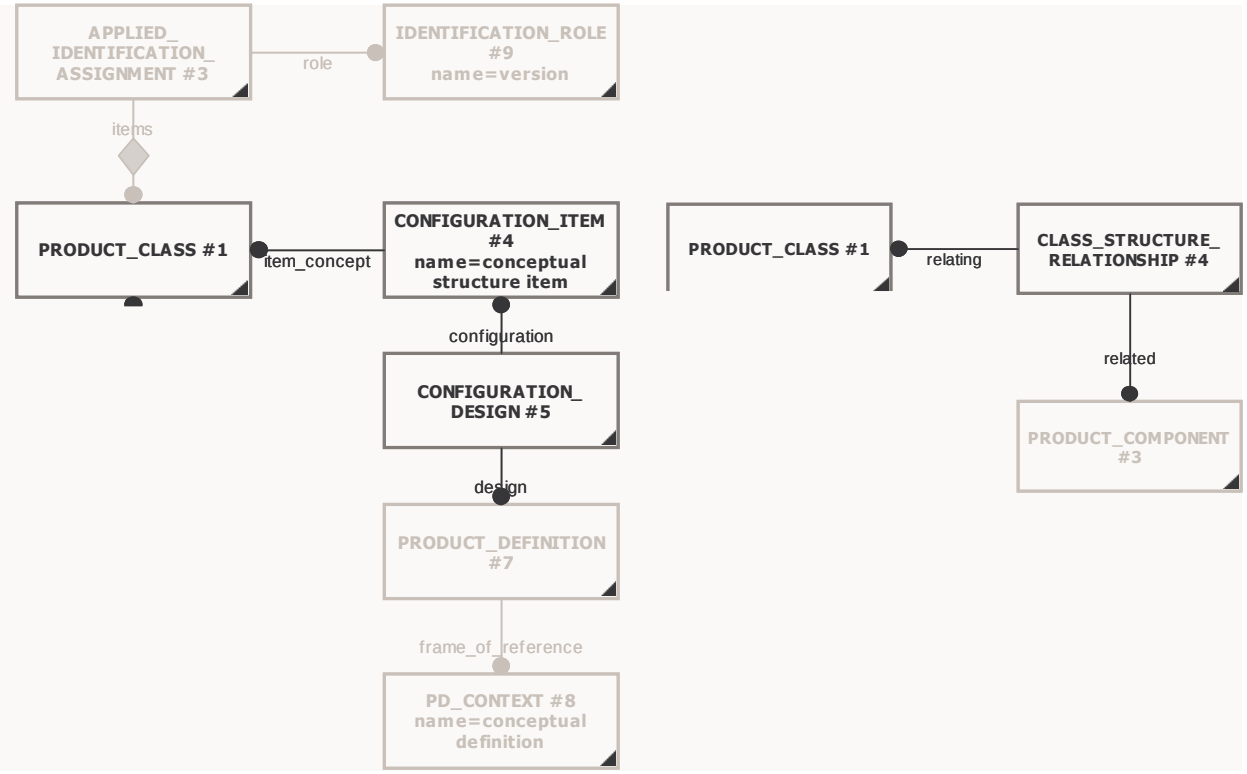


Figure 49 - Instance mapping for product class

EXPRESS-X Mapping Specification:

```
MAP product_class_map AS
   arm_pcl : product_class;
FROM
   aim_pcl : product_class;
SELECT
   arm_pcl.id          := aim_pcl.id;
   arm_pcl.name        := aim_pcl\product_concept.name;
   arm_pcl.description := aim_pcl\product_concept.description;
   arm_pcl.level_type  := aim_pcl\characterized_object.name;
   arm_pcl.version_id  := aim_pcl<-items{applied_identification_assignment |
                                  role.name = 'version'} ::assigned_id[1]
END_MAP;

MAP class_structure_rel_map AS
   csr : class_structure_relationship;
FROM
   cd : configuration_design;
WHERE
   wr1: AUTOMOTIVE_DESIGN.PRODUCT_CLASS' IN
        TYPEOF(cd.configuration.item_concept);
   wr2: cd.design.frame_of_reference IN ['conceptual definition',
```

```
                                        'functional definition'];
SELECT
   csr.relation_type := cd.name ;
   csr.description   := cd.description ;
   csr.related       := IF cd.design.frame_of_reference.name =
                           'conceptual definition'
                        THEN
                          product_component_map(cd.design.formation) ;
                        ELSE
                          product_function_map(cd.design.formation) ;
                        END_IF;
   csr.relating      := product_class_map(cd.configuration.item_concept);
END_MAP;
```

7.4.9.2 Complex_product, Product_component, Product_function and solution types

A target instances of type Complex_product is created out of a source instance of Product_definition_formation that is referenced by an instance of type Product_definition as formation. Depending on the value of the attribute frame_of_reference.name of the Product_definition, the target instance is either of subtype Product_component (for value 'conceptual definition'), Product_function (for value 'functional definition') or Alternative_solution (for value 'alternative definition'). If the target instance is of type Alternative_solution its exact type depends on the value of the name attribute of the Product_definition source instance: it is either Technical_solution (for value 'technical'), Final_solution (for value 'final') or Supplier_solution (for value 'supplier').

EXPRESS-X Mapping Specification:

```
MAP complex_product_map AS
   cp : complex_product;
FROM
   pdf : product_definition_formation;
   pd  : product_definition;
IDENTIFIED_BY pdf;
WHERE
   wr1: pd.formation :=: pdf;
   wr2: pd.frame_of_reference.name IN
        ['conceptual definition','functional definition',
         'alternative definition'];
SELECT
   cp.id          := pdf.of_product.id;
   cp.version_id  := pdf.id;
END_MAP;
```

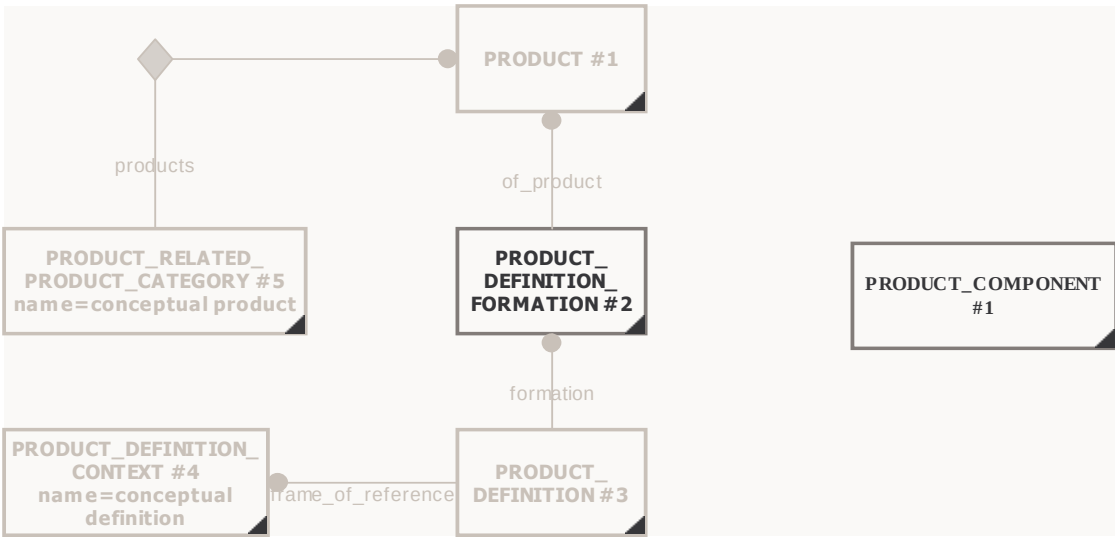


Figure 50 - Instance mapping for product component

EXPRESS-X Mapping Specification:

```
MAP product_component_map AS
   cp : product_component;
SUBTYPE OF (complex_product_map);
WHERE
   pd.frame_of_reference.name = 'conceptual definition';
SELECT
   cp.instance_required := IF pd.name = 'instance required' THEN
                             TRUE;
                           ELSE
                             IF pd.name = 'no instance required' THEN
                               FALSE;
                             END_IF;
                           END_IF;
   cp.name := pdf.of_product.name;
   cp.description := pdf.description;
   cp.is_relevant_for :=
      FOR EACH pdc IN pd<-definition{product_definition_context_association |
                                     role.name = 'application context'}
                      ::frame_of_reference{product_definition_context};
         RETURN (app_context_map(pdc));
END_MAP;
```

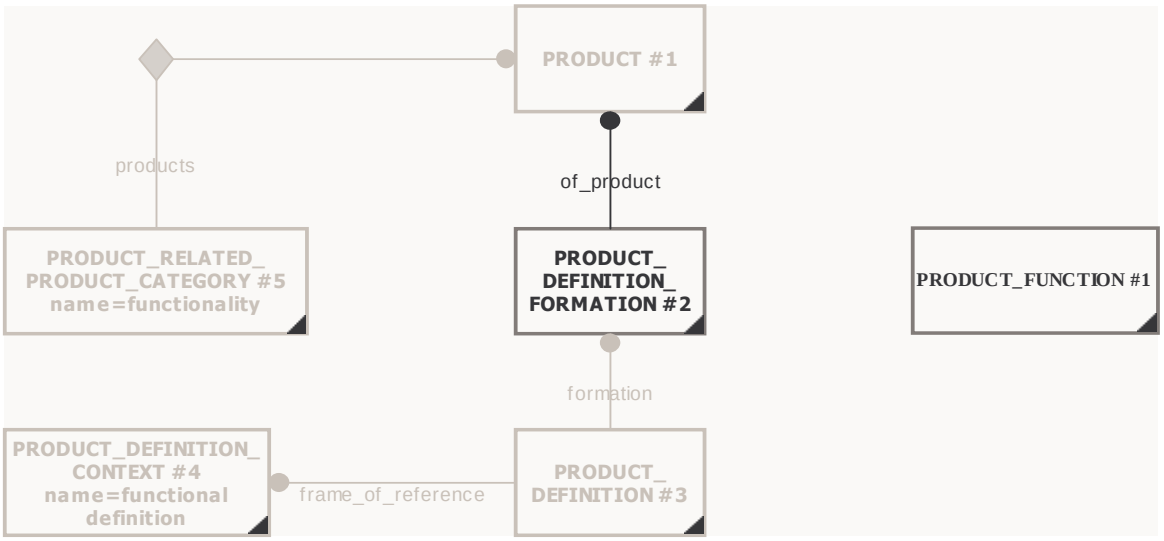


Figure 51 - Instance mapping for product function

EXPRESS-X Mapping Specification:

```
MAP product_function_map AS
   cp : product_function;
SUBTYPE OF (complex_product_map);
WHERE
   pd.frame_of_reference.name = 'functional definition';
SELECT
   cp.name := pdf.of_product.name;
   cp.description := pdf.description;
   cp.is_relevant_for :=
      FOR EACH pdc IN pd<-definition{product_definition_context_association |
                                    role.name = 'application context'}
                     ::frame_of_reference{product_definition_context};
         RETURN (app_context_map(pdc));
END_MAP;

MAP alternative_solution_map AS
   cp : alternative_solution;
SUBTYPE OF (complex_product_map);
WHERE
   pd.frame_of_reference.name = 'alternative definition';
SELECT
   cp.base_element :=
      complex_product_map(
         pd<-related_product_definition{product_definition_relationship |
                              name = 'solution alternative definition'}
            ::relating_product_definition::formation[1]);
END_MAP;
```

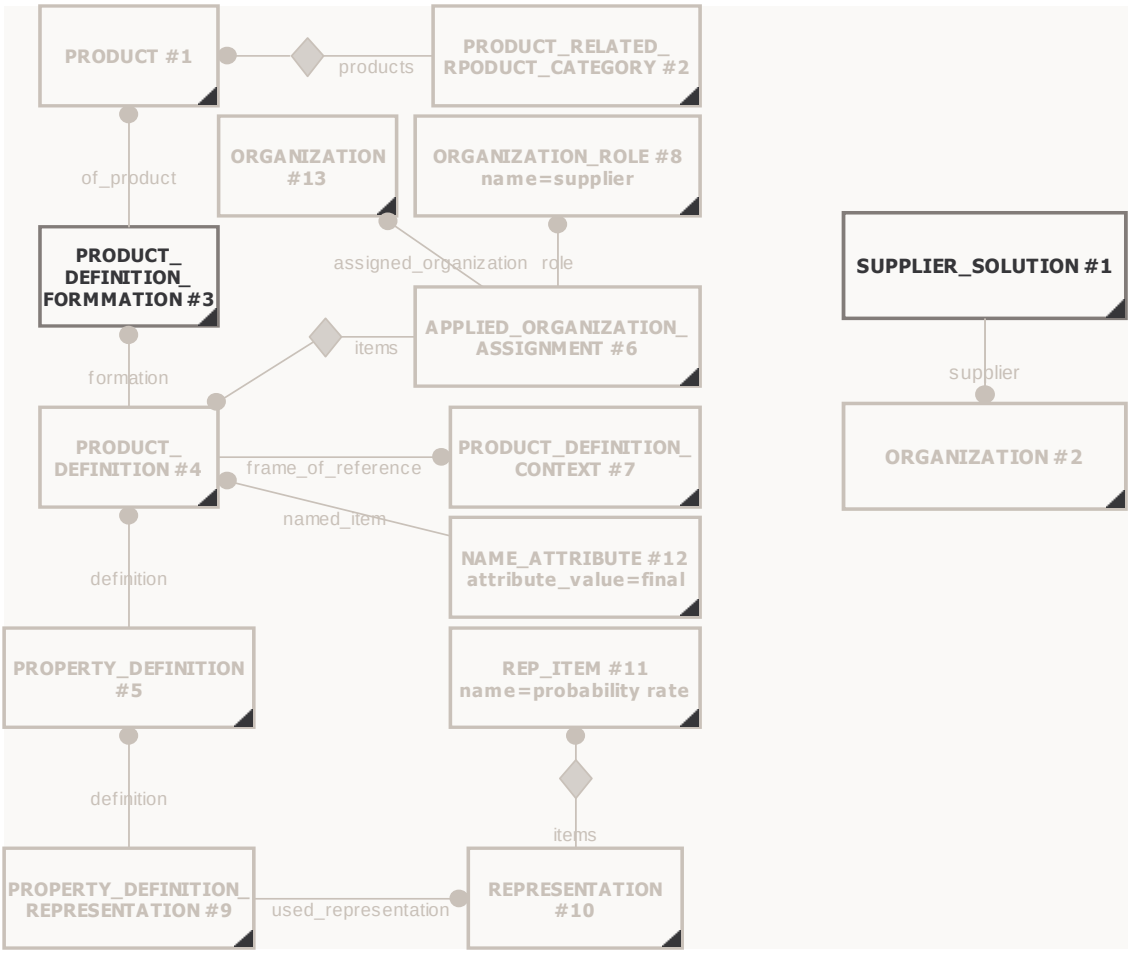


Figure 52 - Instance mapping for supplier solution

EXPRESS-X Mapping Specification:

```
MAP supplier_solution_map AS
   cp : supplier_solution;
SUBTYPE OF (alternative_solution_map);
WHERE
   pd.name = 'supplier';
SELECT
   cp.supplier :=
      organization_map(pd<-items{applied_organization_assignment |
                                 role.name = 'supplier'}[1]);
   cp.probability_rate :=
      pd<-definition{property_definition}
         <-definition{property_definition_representation}
         ::used_representation{representation | name = 'supplier probability'}
         ::items{measure_representation_item | name = 'probability rate'}
         [1].value_component;
END_MAP;
```

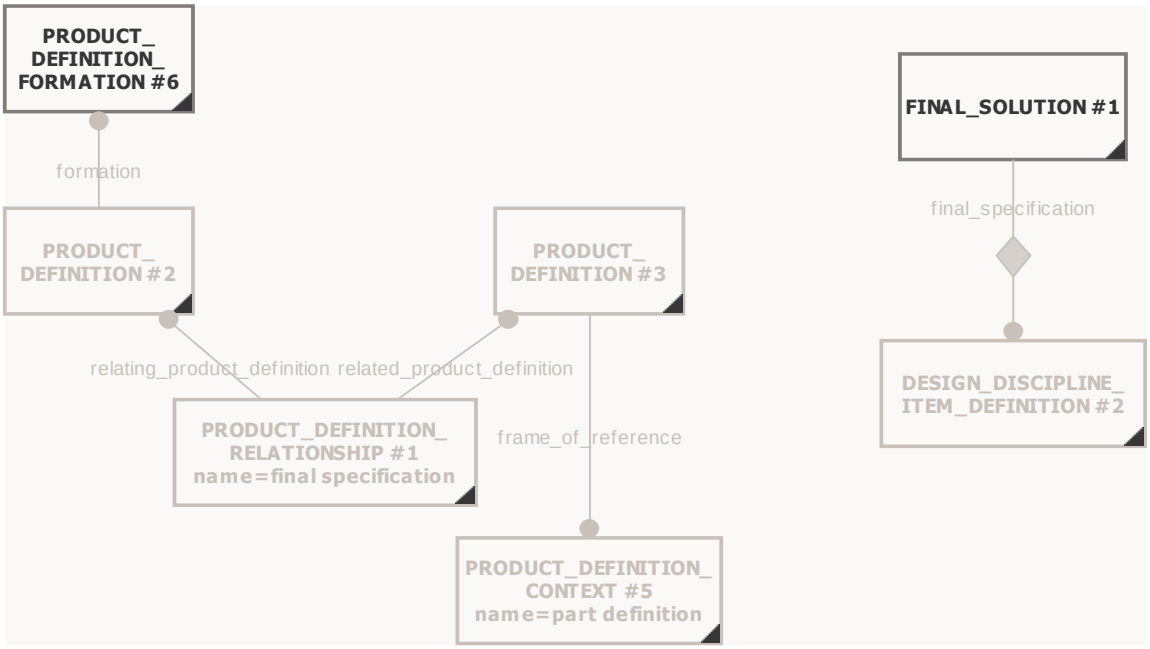


Figure 53 - Instance mapping for final solution

EXPRESS-X Mapping Specification:

```
MAP final_solution_map AS
   cp : final_solution;
SUBTYPE OF (alternative_solution_map);
WHERE
   pd.name = 'final';
SELECT
   cp.final_status :=
      pd<-definition{property_definition }
         <-definition{property_definition_representation}
         ::used_representation{representation |
                              name = 'final item characteristics'}
         ::items{descriptive_representation_item | name='final item status'}
         [1].description;
   cp.final_specification := FOR EACH pd IN
      pd<-relating_product_definition{product_definition_relationship |
                                      name = 'final specification'}
         ::related_product_definition{product_definition |
                                      frame_of_reference.name IN
                                      ['part definition',
                                      'physical occurrence']};
         RETURN (IF pd_fs.frame_of_reference.name = 'part definition' THEN
                  ddid_map(pd_fs);
               ELSE
                 it := physical_instance_map(pd);
               END_IF);
```

```
END_MAP;
```

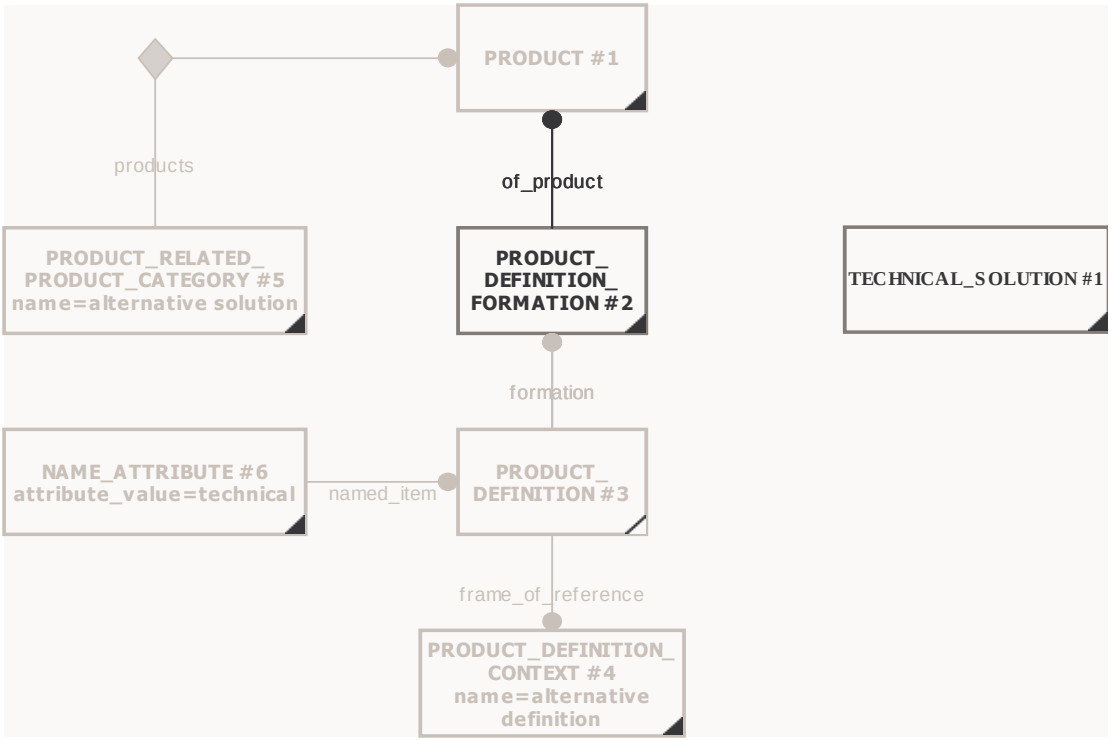


Figure 54 - Instance mapping for technical solution

EXPRESS-X Mapping Specification:

```
MAP technical_solution_map AS
   cp : technical_solution;
SUBTYPE OF (alternative_solution_map);
WHERE
   pd.name = 'technical';
SELECT
   cp.description := pdf.description;
END_MAP;
```

7.4.9.3 Product relationships

A target instance of type Complex_product_relationship is created out of a source instance of type Product_definition_formation_relationship where the relating_product_definition_formation and the related_product_definition_formation both refer to Product_definition_formation instances that are mapped to Complex_product target instances.

A target instance of type Product_structure_relationship is created out of a source instance of type Product_definition_usage where the Product_definition_formation of the relating_product_definition is mapped to a Complex_product and the related_product_definition is mapped to a Product_constituent.

A target instance of type Item_function_association is created out of a source instance of type Product_definition_relationship which refers to a relating product_definition with frame_of_reference.name value 'functionality' and to a related product_definition with frame_of_reference.name value 'part_definition'.

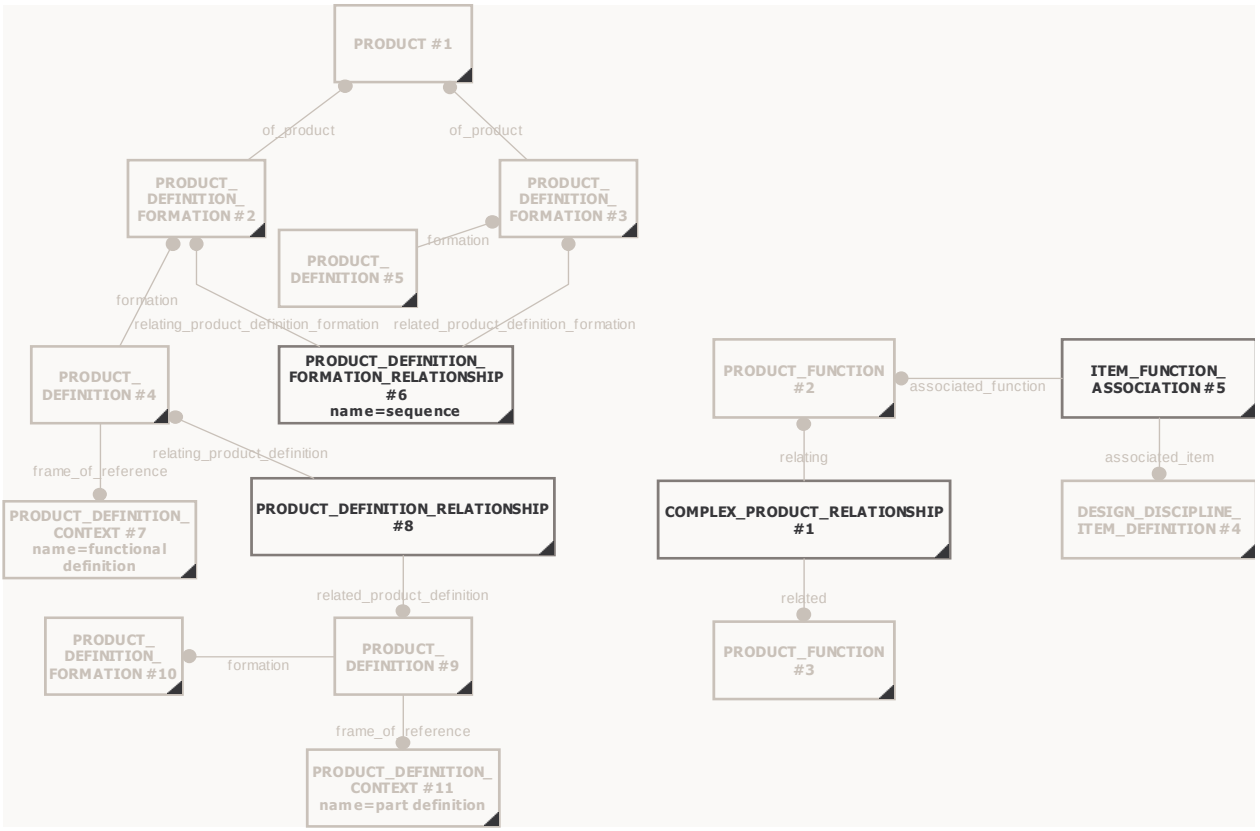


Figure 55 - Instance mapping for item function association

EXPRESS-X Mapping Specification:

```
MAP complex_product_relationship_map AS
   cpr : complex_product_relationship;
FROM
   pdfr : product_definition_formation_relationship;
WHERE
   wr1: EXISTS(complex_product_map(
                pdrf.relating_product_definition_formation));
   wr2: EXISTS(complex_product_map(
                pdrf.related_product_definition_formation));
SELECT
   cpr.relation_type := pdfr.name;
   cpr.description   := pdfr.description;
   cpr.relating :=
      complex_product_map(pdrf.relating_product_definition_formation);
   cpr.related  :=
      complex_product_map(pdrf.related_product_definition_formation);
```

```
END_MAP;
```

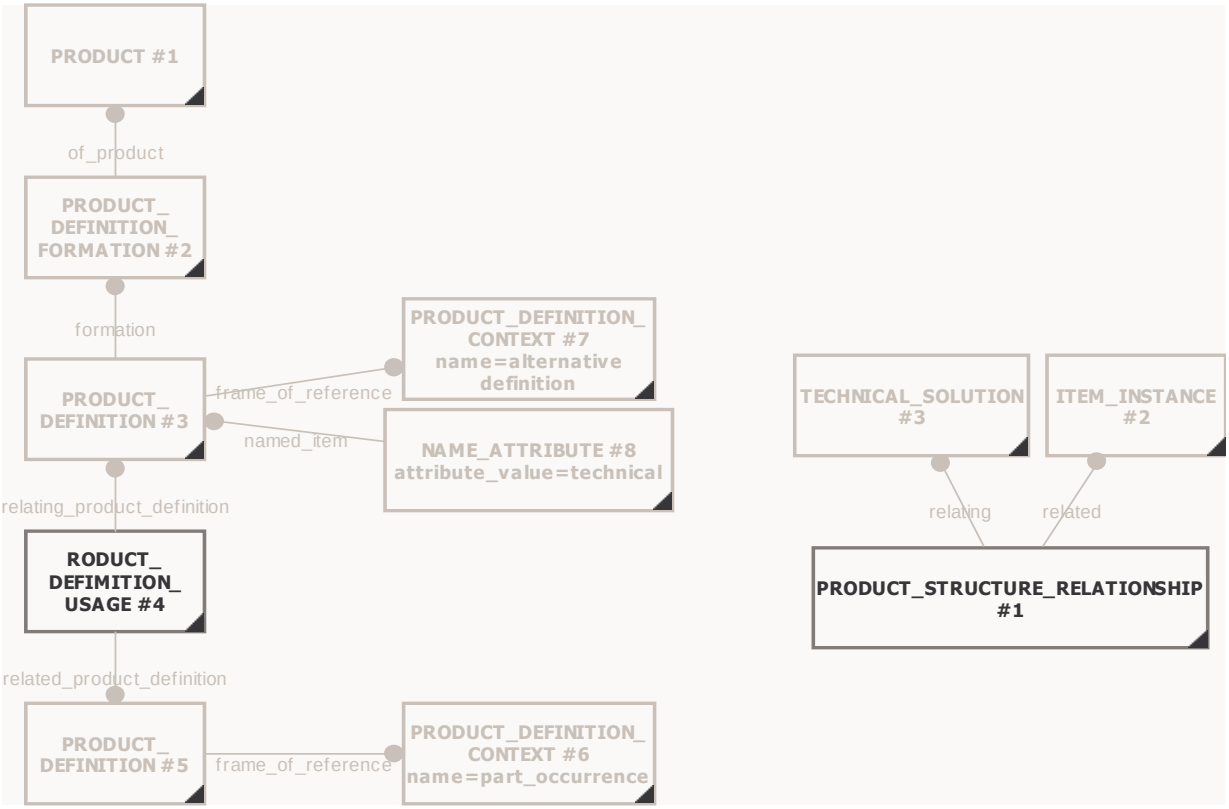


Figure 56 - Instance mapping for product structure relationship

EXPRESS-X Mapping Specification:

```
MAP product_structure_relationship_map AS
   cpr : product_structure_relationship;
FROM
   pdu : product_definition_usage;
WHERE
   wr1: EXISTS(complex_product_map(
                pdrf.relating_product_definition.formation));
   wr2: EXISTS(complex_product_map(
                pdrf.related_product_definition.formation)) OR
        EXISTS(item_instance_map(pdrf.related_product_definition));
SELECT
   cpr.relation_type := pdu.name;
   cpr.description   := pdu.description;
   cpr.relating :=
      complex_product_map(pdu.relating_product_definition.formation);
   cpr.related  :=
      IF EXISTS(item_instance_map(pdu.related_product_definition))
      THEN
```

```
         item_instance_map(pdu.related_product_definition);
      ELSE
         complex_product_map(pdu.related_product_definition.formation);
      END_IF;
END_MAP;

MAP item_function_association_map AS
   ifa : item_function_association;
FROM
   pdr : product_definition_relationship;
WHERE
   wr1: pdr.relating_product_definition.frame_of_reference.name =
       'functional definition';
   wr2: pdr.related_product_definition.frame_of_reference.name =
       'part definition';
SELECT
   ifa.associated_function :=
      product_function_map(pdr.relating_product_definition.formation);
   ifa.assoicated_item  := ddid_map(pdr.related_product_definition);
   ifa.association_type := pdr.name;
   ifa.description      := pdr.description;
END_MAP;
```

7.4.9.4 Class associations

A target instance of type Class_inclusion_association is created out of a source instance of type Product_concept_feature_association that references an instance of type Inclusion_product_concept_feature as feature. A target instance of type Class_condition_association is created out of a source instance of type Product_concept_feature_association that references an instance of type Conditional_product_concept_feature as feature. A target instance of type Class_specification_association is created out of a source instance of type Product_concept_feature_association that references an instance of type Product_concept_feature as feature, but not an instance of type Inclusion_product_concept_feature or Conditional_concept_feature.

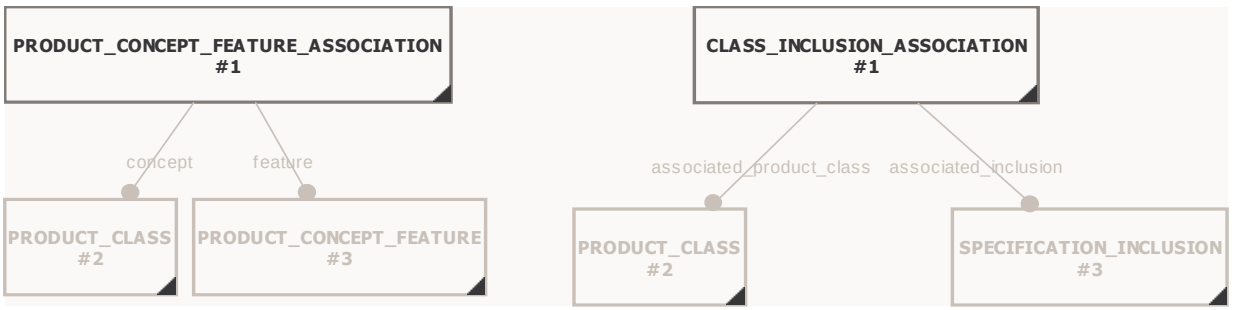


Figure 57 - Instance mapping for class inclusion association

EXPRESS-X Mapping Specification:

```
MAP class_inclusion_association_map AS
   cia : class_inclusion_association;
FROM
```

```
   pcfa : product_concept_feature_association;
   icf  : inclusion_product_concept_feature;
WHERE
   pcfa.feature :=: icf;
IDENTIFIED_BY pcfa;
SELECT
   cia.description := pcfa.description;
   cia.associated_product_class := product_class_map(pcfa.concept);
   cia.associated_inclusion := specification_inclusion_map(icf);
END_MAP;
```

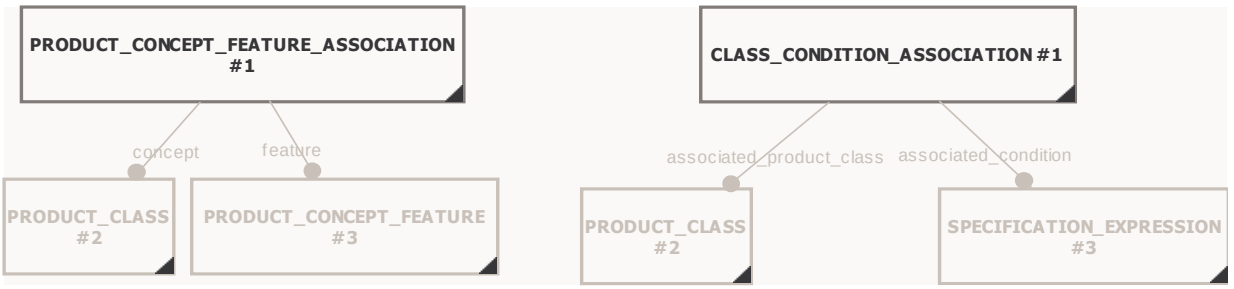


Figure 58 - Instance mapping for class condition association

EXPRESS-X Mapping Specification:

```
MAP class_condition_association_map AS
   cia : class_condition_association;
FROM
   pcfa : product_concept_feature_association;
   ccf : conditional_concept_feature;
WHERE
   pcfa.feature :=: ccf;
IDENTIFIED_BY pcfa;
SELECT
   cia.description := pcfa.description;
   cia.condition_type := pcfa.name;
   cia.associated_product_class := product_class_map(pcfa.concept);
   cia.associated_condition := specification_expression_map(ccf);
END_MAP;
```

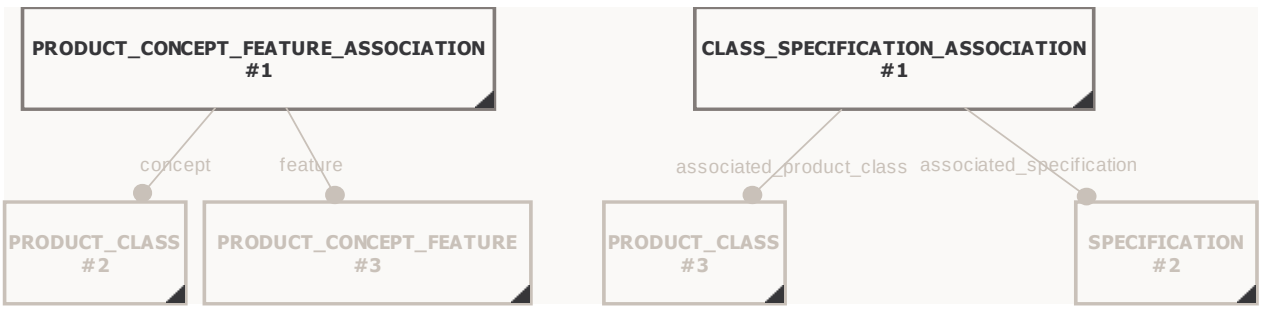


Figure 59 - Instance mapping for class specification association

EXPRESS-X Mapping Specification:

```
MAP class_specification_association_map AS
   csa : class_specification_association;
FROM
   pcfa : product_concept_feature_association;
   pcf  : product_concept_feature;
WHERE
   wr1: pcfa.feature :=: pcf;
   wr2: NOT('AUTOMOTIVE_DESIGN.CONDITIONAL_CONCEPT_FEATURE' IN
           TYPEOF(pcf));
   wr3: NOT('AUTOMOTIVE_DESIGN.INCLUSION_CONCEPT_FEATURE' IN
           TYPEOF(pcf));
SELECT
   csa.association_type := pcfa.name;
   csa.associated_product_class := product_class_map(pcfa.concept);
   csa.associated_specification := specification_map(pcf);
END_MAP;
```

7.4.9.5 Class category types

A target instance of type Class_category_association is created out of a source instance of type Product_concept_feature_category_usage. A target instance of type Specification_category is created out of a source instance of type Product_concept_feature_category.

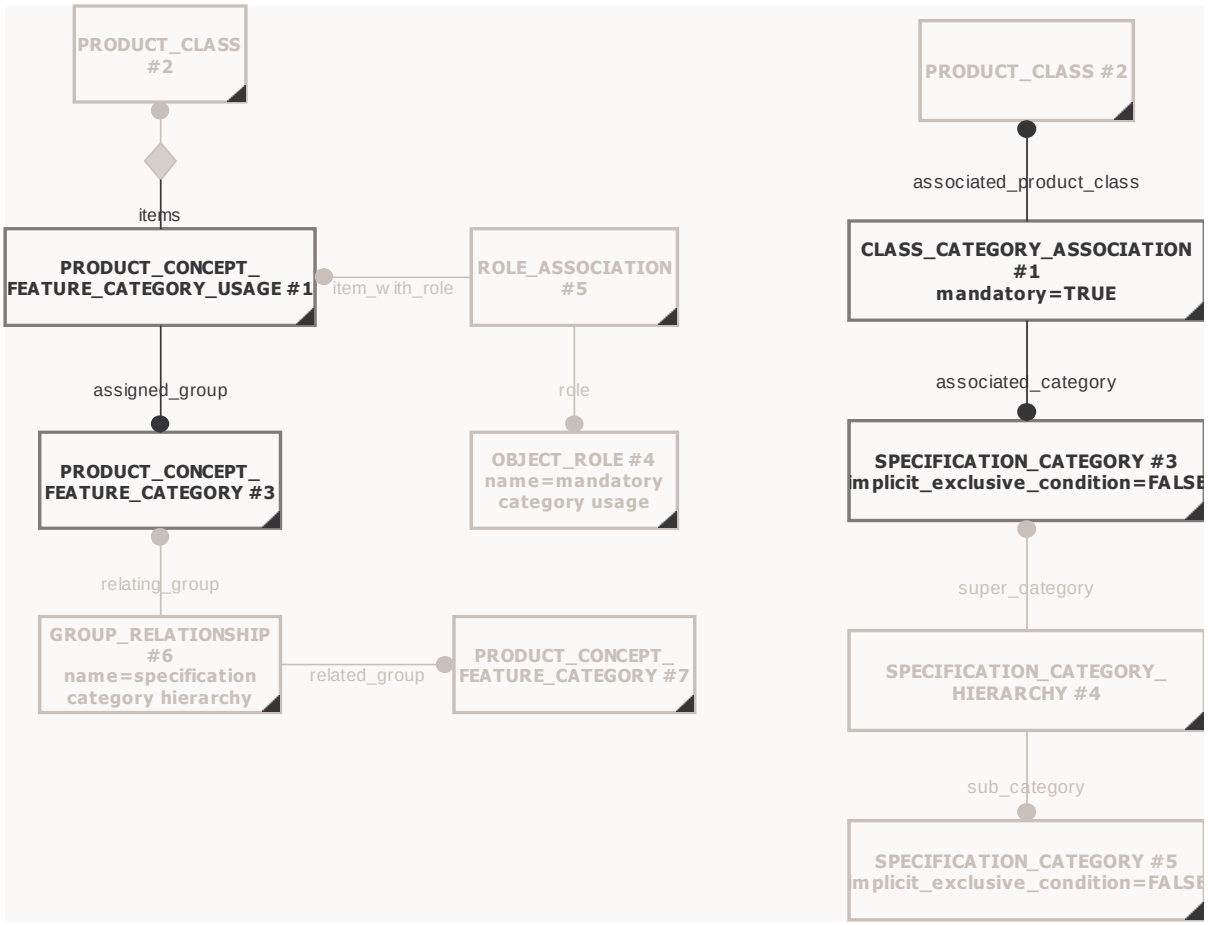
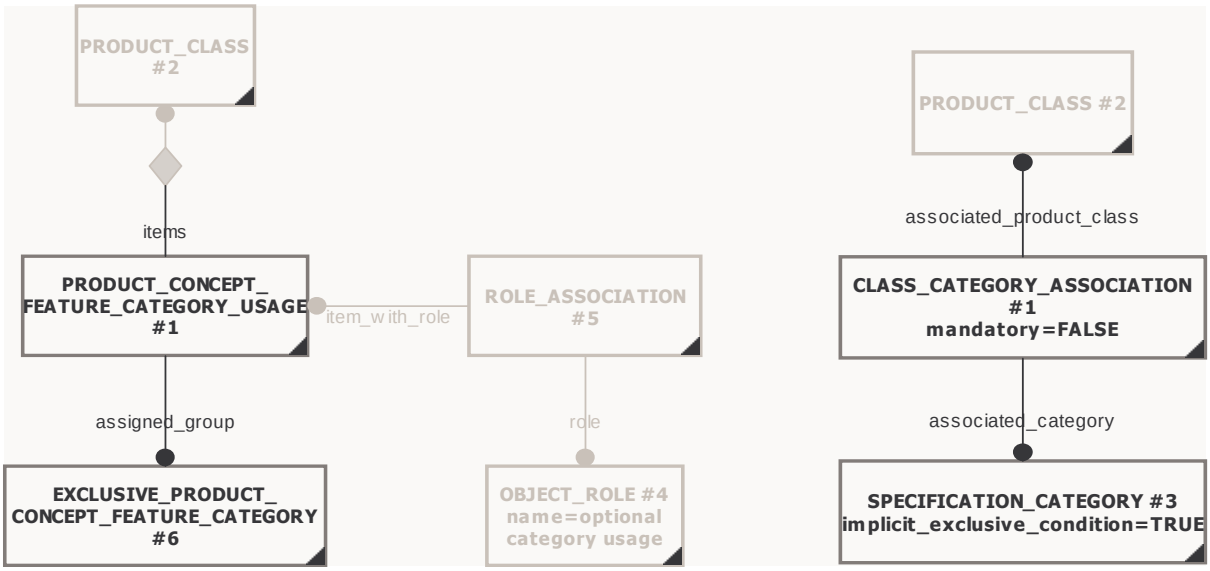


Figure 60 - Instance mapping for class category association

EXPRESS-X Mapping Specification:

```
MAP class_category_assocation_map AS
   cca : class_category_assocation;
FROM
   pcfc : product_concept_feature_category_usage;
SELECT
   cca.mandatory := IF pcfc.role.name = 'mandatory category usage' THEN
                     TRUE;
                    ELSE
                      IF pcfc.role.name = 'optional category usage' THEN
                        FALSE;
                      END_IF;
                    END_IF;
   cca.associated_product_class := product_class_map(pcfc.items[1]);
   cca.associated_category :=
      specification_category_map(pcfc.assigned_group);
END_MAP;
```

Instance Diagrams:



EXPRESS-X Mapping Specification:

```
MAP specification_category_map AS
   sc : specification_category;
FROM
   pcfc : product_concept_feature_category;
SELECT
   sc.description := pcfc.description;
   sc.id          := pcfc.id;
   sc.implicit_exclusive_condition :=
      IF 'AUTOMOTIVE_DESIGN.EXCLUSIVE_CONCEPT_FEATURE_CATEGORY'
         IN TYPEOF(pcfc)
      THEN
         TRUE;
      ELSE
         FALSE;
      END_IF;
END_MAP;

MAP specification_category_hierarchy_map AS
   sch : specification_category_hierarchy;
FROM
   grel : group_relationship;
WHERE
   grel.name = 'specification category hierarchy';
SELECT
   sch.super_category := specification_category_map(grel.relating_group);
   sch.sub_category   := specification_category_map(grel.related_group);
END_MAP;
```

7.4.9.6 Specification types

A target instance of type Specification is created out of a source instance of exact type Product_concept_feature. A target instance of type Specification_expression is created out of a source instance of type Conditional_concept_feature. A target instance of type Specification_inclusion is created out of a source instance of type Inclusion_concept_feature.

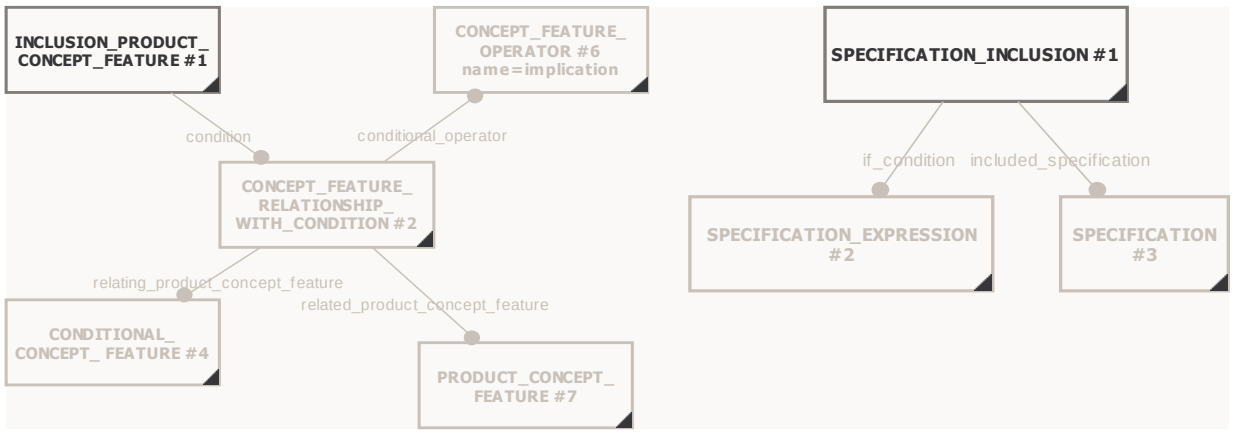


Figure 61 - Instance mapping for specification inclusion

EXPRESS-X Mapping Specification:

```
MAP specification_inclusion_map AS
   si : specification_inclusion;
FROM
   ipcf : inclusion_product_concept_feature;
SELECT
   si.description := ipcf.description;
   si.id := ipcf.id;
   si.if_condition := specification_operand_map(
                      ipcf.condition.relating_product_concept_feature);
   si.included_specification := specification_operand_map(
                      ipcf.condition.related_product_concept_feature);
END_MAP;

MAP specification_operand_map AS
   sp : specification_operand_select;
FROM
   pcf : product_concept_feature;
END_MAP;
```

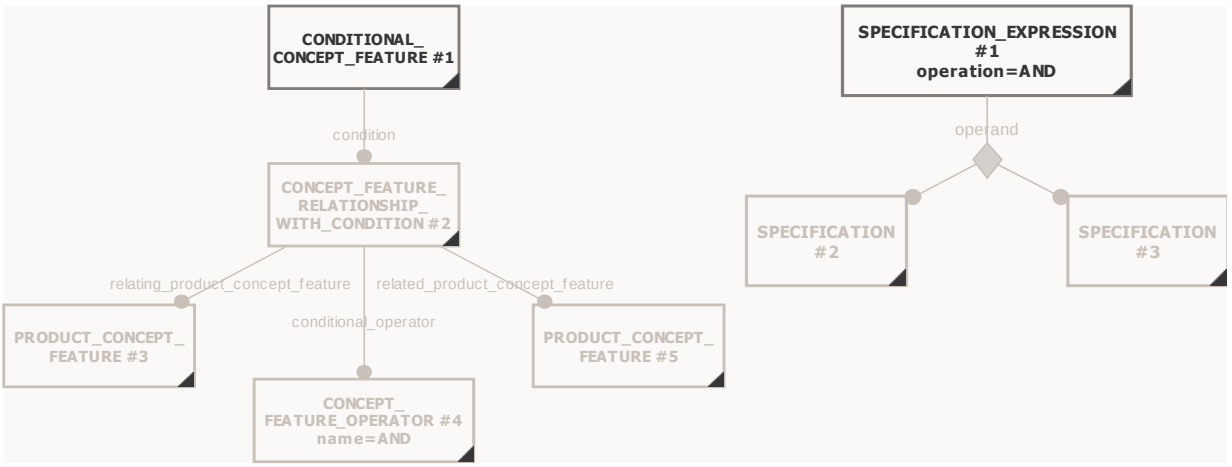


Figure 62 - Instance mapping for specification expression

EXPRESS-X Mapping Specification:

```
MAP specification_expression_map AS
   sp : specification_expression;
SUBTYPE OF (specification_operand_map);
WHERE
   wr1: 'AUTOMOTIVE_DESIGN.CONDITIONAL_CONCEPT_FEATURE' IN TYPEOF(pcf);
SELECT
   sp.id          := pcf.id;
   sp.description := pcf.description;
   sp.operation   := pcf.condition.conditional_operator.name;
   sp.operand     := FOR EACH op IN
                     [pcf.condition.related_product_concept_feature,
                     pcf.condition.relating_product_concept_feature];
                     RETURN (specification_operand_map(op));
END_MAP;
```

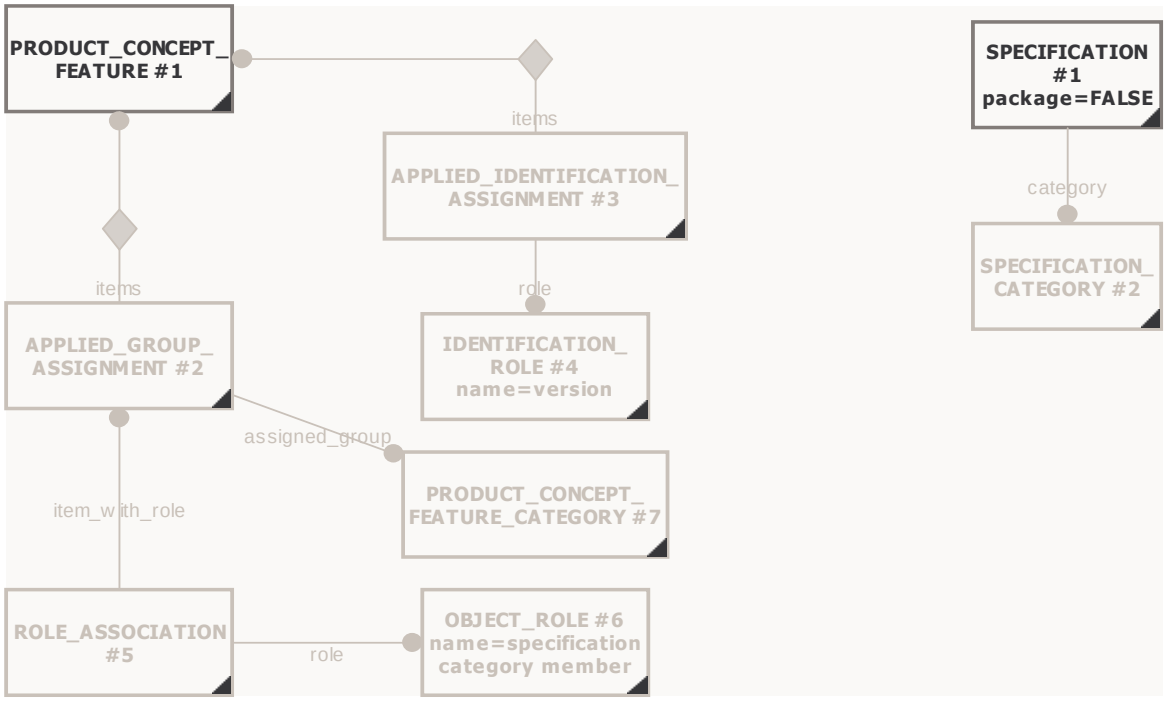


Figure 63 - Instance mapping for specification

EXPRESS-X Mapping Specification:

```
MAP specification_map AS
   sp : specification;
SUBTYPE OF (specification_operand_map);
WHERE
   OTHERWISE;
SELECT
   sp.id          := pcf.id;
   sp.description := pcf.description;
   sp.package     := 'AUTOMOTIVE_DESIGN.PACKAGE_CONCEPT_FEATURE'
                      IN TYPEOF(pcf);
   sp.name        := pcf.name;
   sp.version_id  := aia<-items{applied_identification_assignment |
                               role.name = 'version'}[1].assigned_id;
   sp.category    :=
      specification_category_map(pcf<-items{applied_group_assignment |
                    role.name = 'specification category member'}
                   ::assigned_group{product_concept_feature_category}[1]);
END_MAP;
```

7.4.9.7 Configuration

A target instance of type Configuration is created out of a source instance of type Configured_effectivity_assignment, which references an instance of type Effectivity with id value 'configuration validity'.

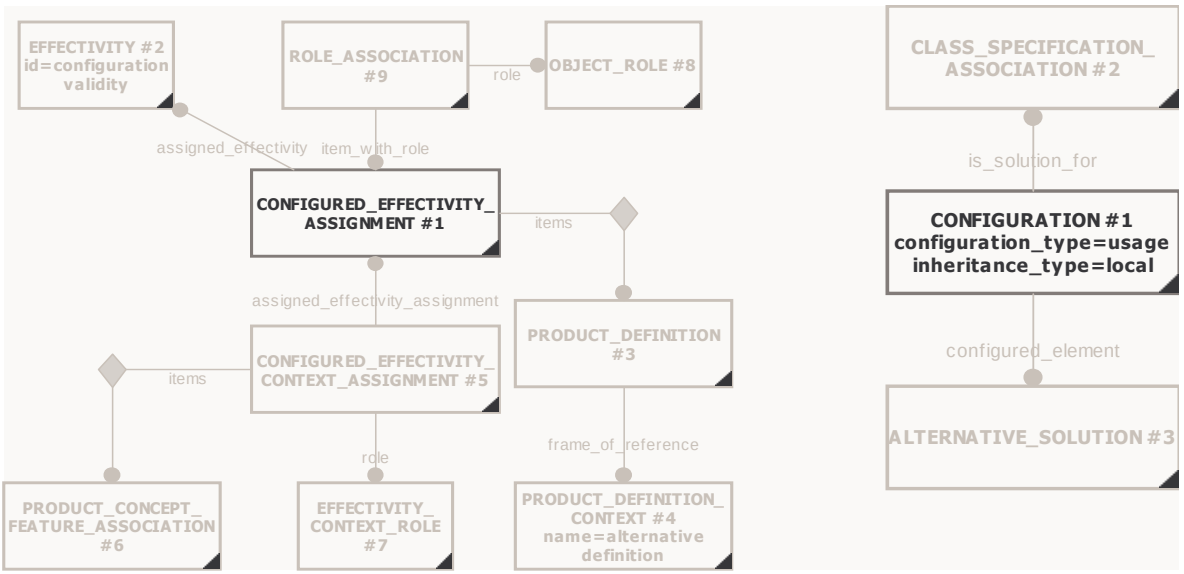


Figure 64 - Instance mapping for configuration

EXPRESS-X Mapping Specification:

```
MAP configuration_map AS
   cfg : configuration;
FROM
   cea : configured_effectivity_assignment;
LOCAL
   pcfa : product_context_feature_association;
END_LOCAL;
WHERE
   cea.assigned_effectivity.id = 'configuration validity';
SELECT
   cfg.configuration_type := cea.role.name;
   cfg.inheritance_type   := cea.role.description;

   cfg.configured_element :=
      IF 'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION' IN TYPEOF(cea.items[1])
      THEN
         CASE cea.items[1].frame_of_reference.name OF
         'alternative definition',
         'conceptual definition',
         'functional definition' : complex_product_map(cea.items[1]);
         'part occurrence'       : item_instance_map(cea.items[1]);
         END_CASE;
      END_IF;
   pcfa := cea<-assigned_effectivity_assignment
           {configured_effectivity_context_assignment |
           role.name = 'specification based condition'}
           ::items{product_context_feature_association}[1];
```

```
   cfg.is_solution_for :=
      IF 'AUTOMOTIVE_DESIGN.CONDITIONAL_CONCEPT_FEATURE' IN
         TYPEOF(pcfa.feature)
      THEN
         class_condition_association_map(pcfa);
      ELSE
         class_specification_association_map(pcfa);
      END_IF;
END_MAP;
```

7.4.9.8 Product_design

A target instance of Product_design is created out of a source instance of type Configuration_design.

Figure 65 - Instance mapping for product design

EXPRESS-X Mapping Specification:

```
MAP product_design_map AS
   pdes : product_design;
FROM
   cd : configuration_design;
WHERE
   cd.name = 'product design';
SELECT
   pdes.design  := item_version_map(cd.design);
   pdes.product := product_identification_map(cd.configuration);
END_MAP
```

7.4.9.9 Product_identification and Product_specification

A target instance of Product_identification is created out of a source instance of type Product_identification.

A target instance of Product_specification is created out of a source instance of type Product_specification.

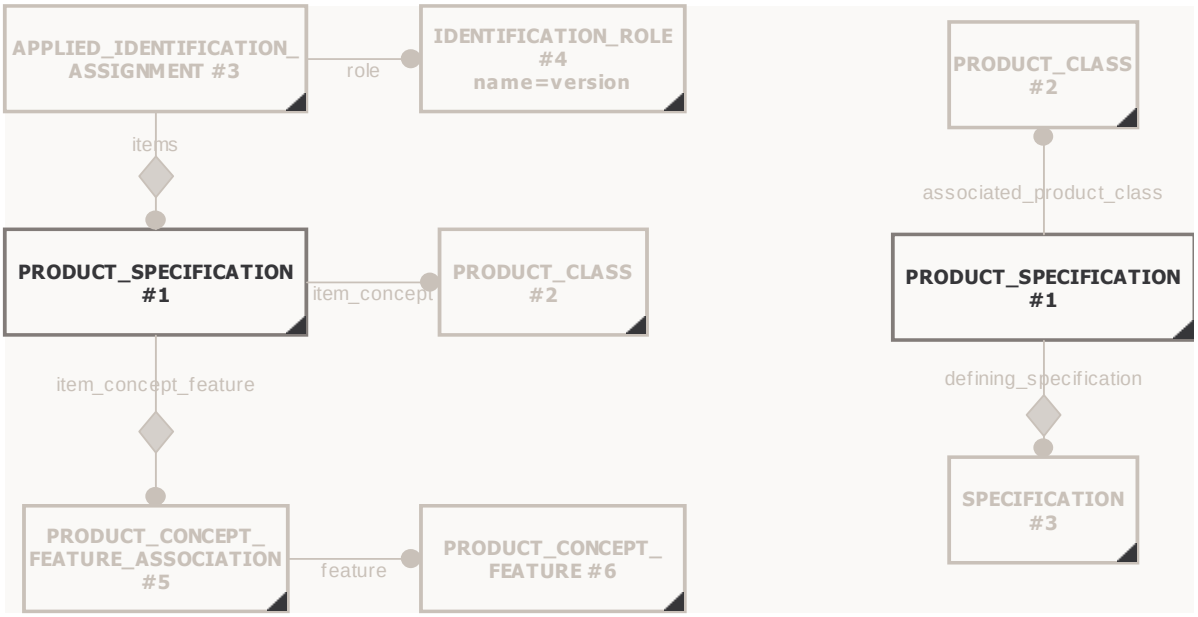


Figure 66 - Instance mapping for product specification

EXPRESS-X Mapping Specification:

```
MAP product_identification_map AS
   t_pid : product_identification;
FROM
   s_pid : product_identification;
SELECT
   t_pid.id   := s_pid.id;
   t_pid.name := s_pid.name;
   t_pid.description := s_pid.description;
   t_pid.version_id := s_pid<-items{applied_identification_assignment |
                                    role.name = 'version'}[1].assigned_id;
   t_pid.associated_product_class := product_class_map(s_pid.item_concept);
   t_pid.
END_MAP;

MAP product_specification_map AS
   t_pid : product_specification;
SUBTYPE OF (product_identification_map);
WHERE
   'AUTOMOTIVE_DESIGN.PRODUCT_SPECIFICATION' IN TYPEOF(s_pid);
SELECT
   t_pid.defining_specification :=
      FOR EACH pcf IN s_pid.item_concept_feature::feature;
         RETURN (specification_map());
END_MAP;
```

7.4.9.10 Physical_instance

A target instance of Physical_instance is created out of a source instance of type Product_definition which refers to a Product_definition_context as frame_of_reference with name 'physical occurrence'.

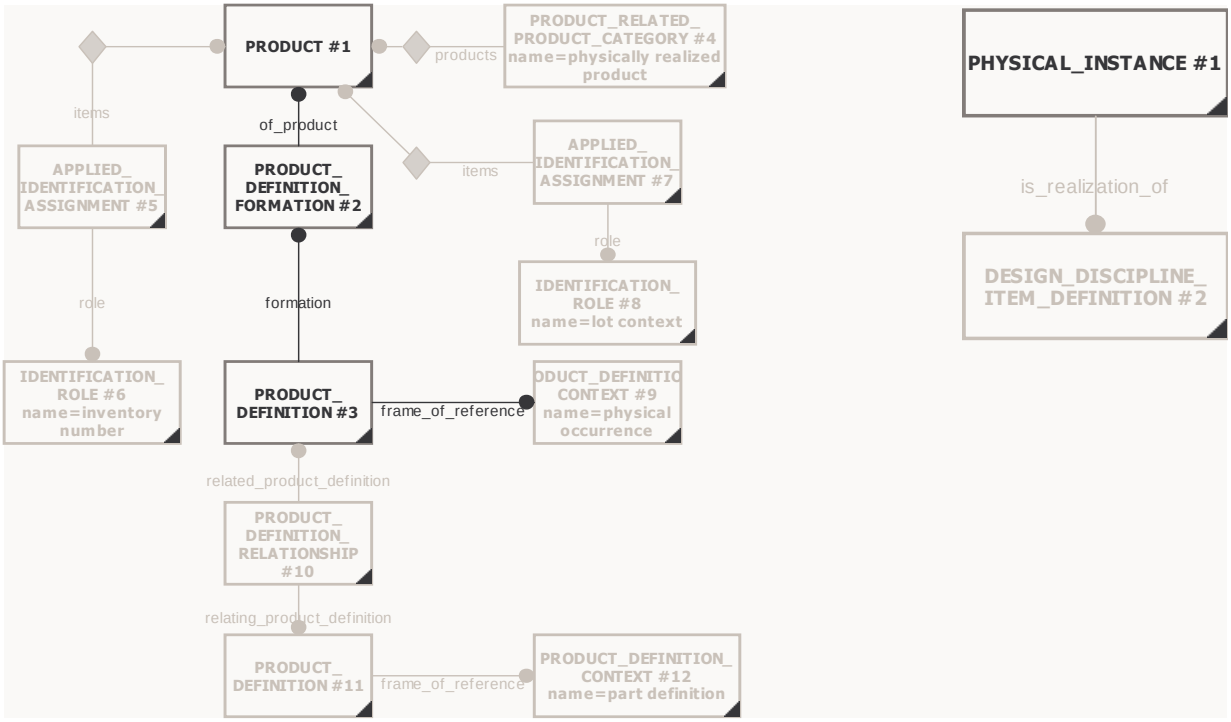


Figure 67 - Instance mapping for physical instance

EXPRESS-X Mapping Specification:

```
MAP physical_instance_map AS
   phi : physical_instance;
FROM
   pd : product_definition;
WHERE
   pd.frame_of_reference.name = 'physical occurrence';
SELECT
   phi.description := pd.formation.of_product.description;
   phi.inventory_number :=
      pd.formation.of_product<-items{applied_identification_assignment |
                     role.name = 'inventroy number'}[1].assigned_id;
   phi.lot_id :=
      pd.formation.of_product<-items{applied_identification_assignment |
                     role.name = 'lot context'}[1].assigned_id;
   phi.serial_number := pd.formation.of_product.id;
   phi.is_realization_of :=
      IF SIZEOF(pd<-related_product_definition
              {product_definition_relationship |
```

```
            name = 'physical realization'}) > 0
      THEN
         ddid_map(pd<-related_product_definition
              {product_definition_relationship |
               name = 'physical realization'}
              ::relating_product_definition[1]);
      ELSE
         product_identification_map(pd<-design{configuration_design |
                                 name ='physical instance basis'}
                                 ::configuration[1]);
      END_IF;
END_MAP;
```

7.4.9.11 Physical_instance_test_result

A target instance of Physical_instance_test_result is created out of a source instance of type Property_definition with name 'test result'.

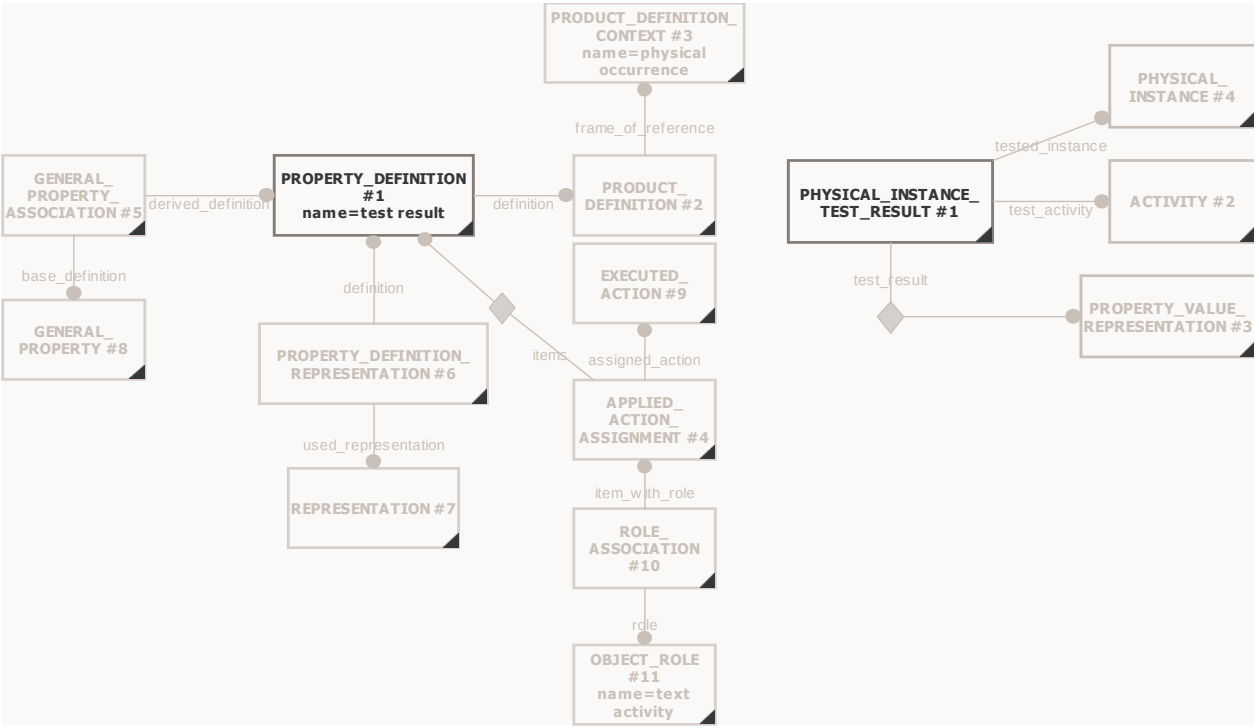


Figure 68 - Instance mapping for physical instance test result

EXPRESS-X Mapping Specification:

```
MAP physical_instance_test_result_map AS
   pitr : physical_instance_test_reslult;
FROM
   prd : property_definition;
WHERE
```

```
   prd.name = 'test result';
SELECT
   pitr.description := prd.description;
   pitr.id := prd.id;
   pitr.tested_instance := physical_instance_map(prd.definition);
   pitr.test_result     :=
      property_value_representation_map(prd<-definition
                                        {property_definition_representation}
                                       ::used_representation[1]);
   pitr.test_activity :=
      IF SIZEOF(pd<-items{applied_action_assignment |
                          role.name = 'test activity'}
                ::assigned_action{action}
                <-related_action{action_relationship |
                                 name = 'process operation occurrence'}
      THEN
         process_operation_occurrence_map(
            pd<-items{applied_action_assignment |
                role.name = 'test activity'}
               ::assigned_action{action
               <-related_action{action_relationship |
                           name = 'process operation occurrence'}[1]);
      ELSE
         activity_map(pd<-items{applied_action_assignment |
               role.name = 'test activity'}
               ::assigned_action{executed_action}[1]);
    END_IF;
END_MAP;
```

7.4.9.12 Physical_assembly_relationship

A target instance of Physical_assembly_relationship is created out of a source instance of type Assembly_component_usage with name 'physical occurrence usage'.

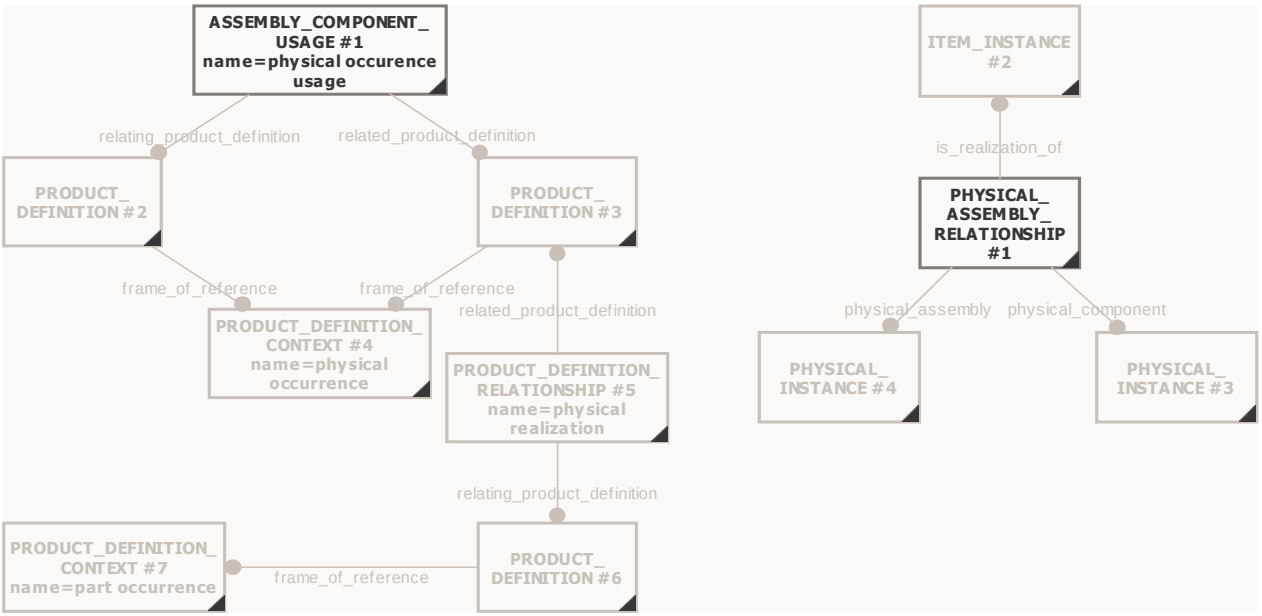


Figure 69 - Instance mapping for physical assembly relationship

EXPRESS-X Mapping Specification:

```
MAP physical_assembly_relationship_map AS
   par : physical_assembly_relationship;
FROM
   acu : assembly_component_usage;
WHERE
   acu.name = 'physical occurrence usage';
SELECT
   par.physical_assembly :=
      physical_instance_map(acu.relating_product_definition);
   par.physical_component :=
      physical_instance_map(acu.related_product_definition);
   par.is_realization_of :=
      item_instance_map(acu.related_product_definition
         <-related_product_definition{product_definition_relationship
                                      | name = 'physical realization'}
         ::relating_product_definition[1]});
END_MAP;
```

7.4.9.13 Effectivity

A target instance of type Effectivity is created out of a source instance of type Effectivity that is referenced by an instance of type Effectivity_relationship with name 'inheritance' as related_effectivity or which is of subtype Dated_effectivity or Time_interval_based_effectivity.

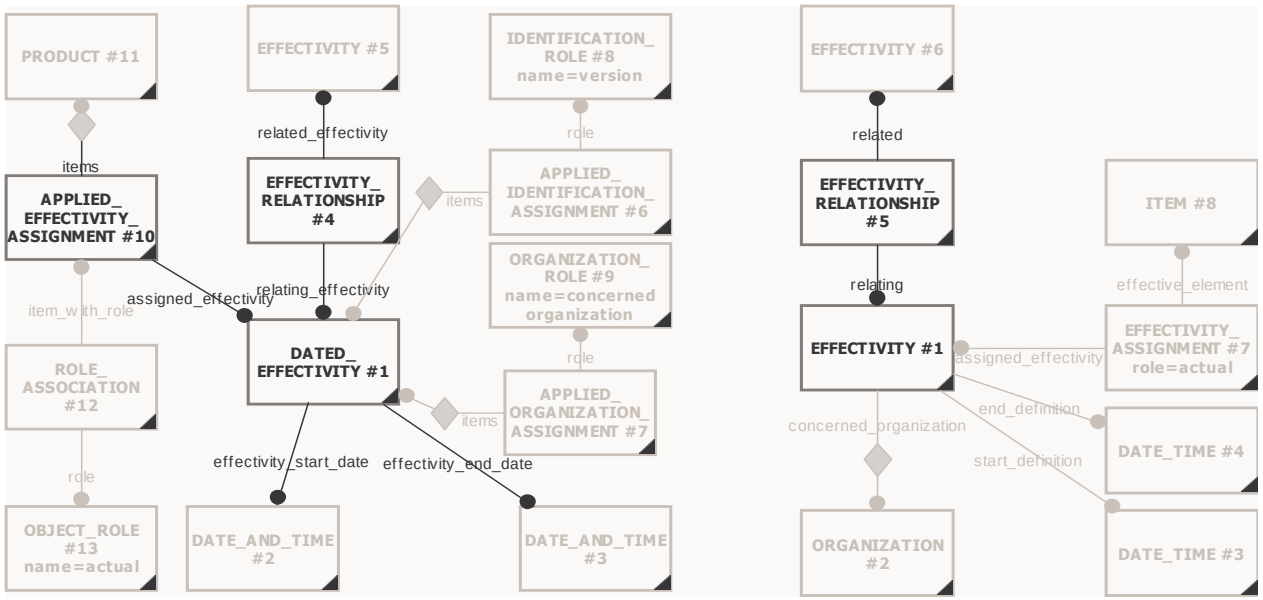


Figure 70 - Instance mapping for effectivity

EXPRESS-X Mapping Specification:

```
MAP effectivity_map AS
   t_eff : effectivity;
FROM
   s_eff : effectivity;
WHERE
   wr1: (SIZEOF(['AUTOMOTIVE_DESIGN.DATED_EFFECTIVITY',
                 'AUTOMOTIVE_DESIGN.TIME_INTERVAL_BASED_EFFECTIVITY']
                * TYPEOF(s_eff)) > 0) OR
        (SIZEOF(s_eff<-related_effectivity{effectivity_relationship |
                                           name = 'inheritance'}) > 0;
SELECT
   t_eff.description := s_eff.description;
   t_eff.effectivity_context := s_eff.name;
   t_eff.id := s_eff.id;
   t_eff.version_id := s_eff<-items{applied_identification_assignment |
                                    role.name = 'version'}[1].assigned_id;
   t_eff.concerned_organization :=
      FOR EACH org IN s_eff<-items{applied_organization_assignment |
                                   role.name = 'concerned organization'}
                           ::assigned_organization);
         RETURN organization_map(org);
END_MAP;

MAP dated_effectivity_map AS
   t_eff : effectivity;
SUBTYPE OF (effectivity_map);
WHERE
```

```
   wr2: 'AUTOMOTIVE_DESIGN.DATED_EFFECTIVITY' IN TYPEOF(s_eff);
SELECT
   t_eff.start_definition :=
      IF 'AUTOMOTIVE_DESIGN.EVENT_OCCURRENCE'
            IN TYPEOF(s_eff.effectivity_end_date) THEN
         event_reference_map(s_eff.effectivity_end_date);
      ELSE
         date_time_map(s_eff.effectivity_end_date);
      END_IF;
   t_eff.end_definition :=
      IF 'AUTOMOTIVE_DESIGN.EVENT_OCCURRENCE'
            IN TYPEOF(s_eff.effectivity_end_date) THEN
         event_reference_map(s_eff.effectivity_end_date);
      ELSE
         date_time_map(s_eff.effectivity_end_date);
      END_IF;
END_MAP;

MAP time_interval_based_effectivity_map AS
   t_eff : effectivity;
SUBTYPE OF (effectivity_map);
WHERE
   wr2: 'AUTOMOTIVE_DESIGN.TIME_INTERVAL_BASED_EFFECTIVITY'
            IN TYPEOF(s_eff);
SELECT
   t_eff.period := duration_map(s_eff.effectivity_period.duration);
   t_eff.start_definition :=
      IF 'AUTOMOTIVE_DESIGN.EVENT_OCCURRENCE'
            IN TYPEOF(s_eff.effectivity_period.primary_bound) THEN
         event_reference_map(s_eff.effectivity_period.primary_bound);
      ELSE
         date_time_map(s_eff.effectivity_period.primary_bound);
      END_IF;
   t_eff.end_definition :=
      IF 'AUTOMOTIVE_DESIGN.EVENT_OCCURRENCE'
            IN TYPEOF(s_eff.effectivity_period.primary_bound) THEN
         event_reference_map(s_eff.effectivity_period.secondary_bound);
      ELSE
         date_time_map(s_eff.effectivity_period.secondary_bound);
      END_IF;
END_MAP;
```

7.4.9.14 Specific configurations

A target instance of type Manufacturing_configuration is created out of a source instance of type Configuration_effectivity.

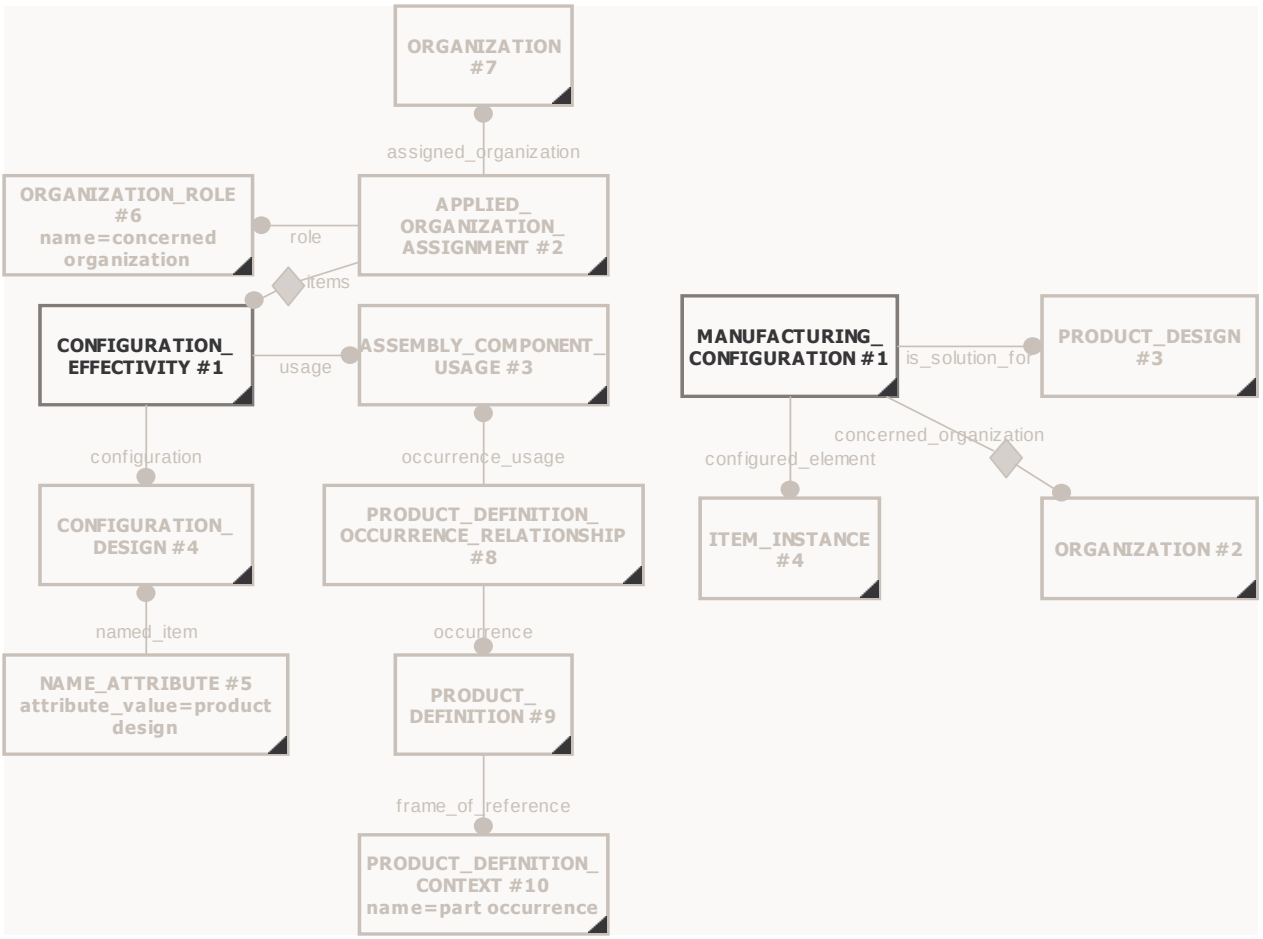


Figure 71 - Instance mapping for manufacturing configuration

EXPRESS-X Mapping Specification:

```
MAP manufacturing_configuration_map AS
   mc : manufacturing_configuration;
FROM
   ce : configuration_effectivity;
SELECT
   mc.concerned_organization :=
      organization_map(ce<-items{applied_organization_assignment |
                                 role.name = 'concerned organization'}
                       ::assigned_organization[1]);
   mc.configured_element := item_instance_map(
         ce::usage{assembly_component_usage}
         <-occurrence_usage{product_definition_occurrence_relationship}
         ::occurrence{product_definition}[1]);
   mc.is_solution_for := product_design_map(ce.configuration);
END_MAP;
```

```
MAP lot_configuration_map AS
   mc : lot_configuration;
SUBTYPE OF (manufacturing_configuration_map);
WHERE
   'AUTOMOTIVE_DESIGN.LOT_EFFECTIVITY' IN TYPEOF(ce);
SELECT
   mc.lot_id := ec.effectivity_lot_id;
   mc.lot_size := ec.lot_size.value_component;
END_MAP;

MAP serial_configuration_map AS
   mc : serial_configuration;
SUBTYPE OF (manufacturing_configuration_map);
WHERE
   'AUTOMOTIVE_DESIGN.SERIAL_NUMBERED_EFFECTIVITY' IN TYPEOF(ce);
SELECT
   mc.serial_end_number := ce.effectivity_end_id;
   mc.serial_start_number := ce.effectivity_start_id;
END_MAP;

MAP dated_configuration_map AS
   mc : dated_configuration;
SUBTYPE OF (manufacturing_configuration_map);
WHERE
   'AUTOMOTIVE_DESIGN.DATED_EFFECTIVITY' IN TYPEOF(ce);
SELECT
   mc.end_date := date_time_map(ce.effectivity_end_date);
   mc.start_date := date_time_map(ce.effectivity_start_date);
END_MAP;
```

7.4.9.15 Event_reference

A target instance of type Event_reference is created out of a source instance of type Event_occurrence.

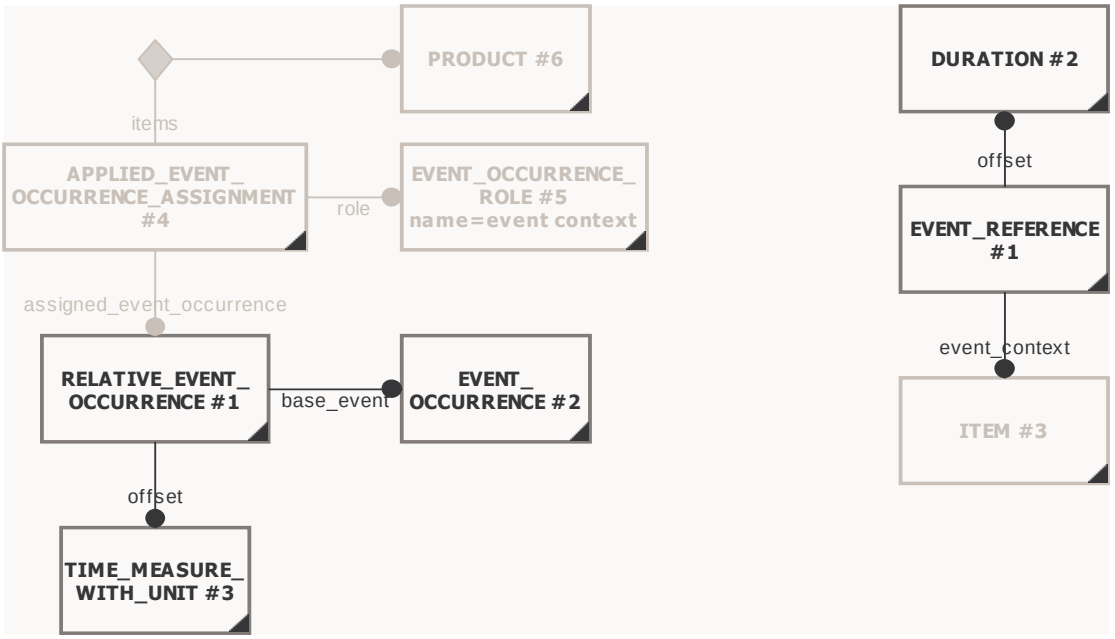


Figure 72 - Instance mapping for duration and event

EXPRESS-X Mapping Specification:

```
MAP event_reference_map AS
   ref : event_reference;
FROM
   occ : event_occurrence;
SELECT
   ref.description := occ.description;
   ref.event_type := occ.name;
   ref.event_context :=
      general_organizational_data_select_map(
         occ<-assigned_event_occurrence{applied_event_occurrence_assignment |
                                        role.name = 'event context')
            ::items[1])
   ref.offset := IF 'AUTOMOTIVE_DESIGN.RELATIVE_EVENT_OCCURRENCE'
                    IN TYPEOF(occ)
                 THEN
                   duration_map(occ.offset);
                 END_IF;
END_MAP;
```

7.4.9.16 Duration

A target instance of type Duration is created out of a source instance of type Time_measure_with_unit.

EXPRESS-X Mapping Specification:

```
DEPENDENT_MAP duration_map AS
```

```
   d : duration;
FROM
   tmu : time_measure_with_unit;
SELECT
   d.time := tmu.value_component;
   d.time_unit := get_unit_name(tmu.unit_component);
END_DEPENDENT_MAP;
```

7.4.10 Change and Work Management

7.4.10.1 Activity and related types

A target instance of type Activity is created out of a source instance of type Executed_action.

A target instance of type Activity_relationship is created out of a source instance of type Action_relationship. A target instance of type Activity_method is created out of a source instance of type Action_method. A target instance of type Activity_element is created out of a source instance of type Applied_action_assignment, which references an instance of type Object_role with description value 'activity element'.

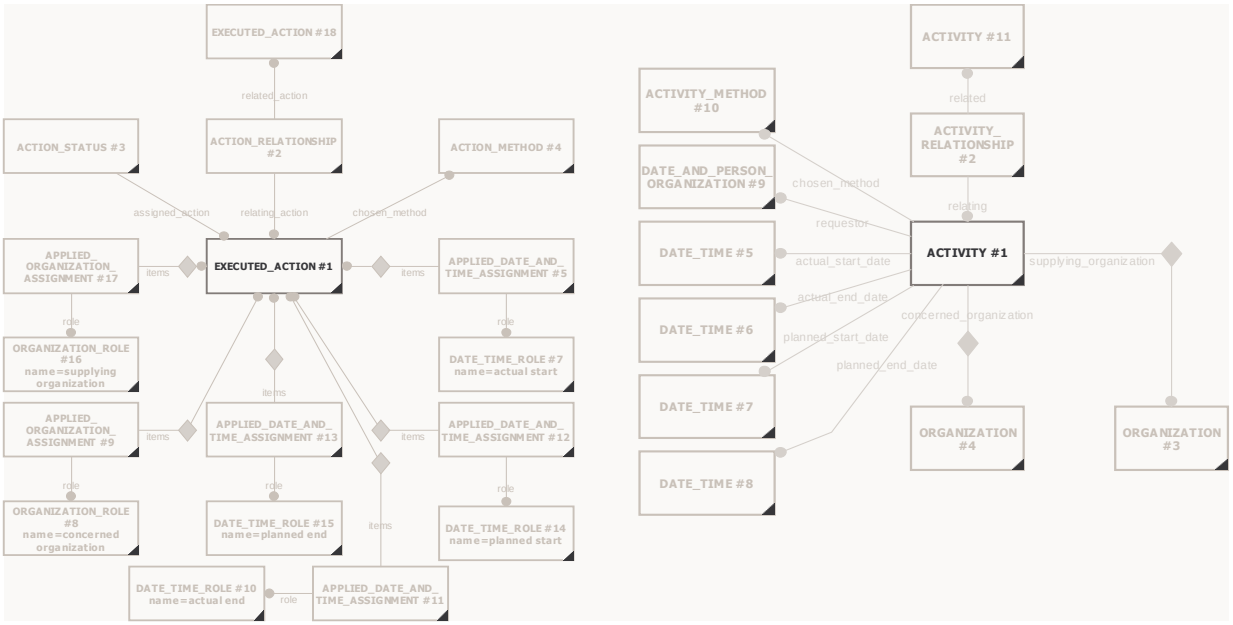


Figure 73 - Instance mapping for activity

EXPRESS-X Specification:

```
MAP action_map AS
   acv : activity;
FROM
   act : executed_action;
LOCAL
   dpos : SET OF date_and_person_organization;
END_LOCAL;
```

```
SELECT
   acv.activity_type := act.name;
   acv.description   := act.description
   acv.id            := act.id
   acv.internal      := CASE act.chosen_method.purpose OF
                          'internal' : TRUE;
                          'external' : FALSE;
                        OTHERWISE : ?;
                        END_CASE;
   acv.status        := act<-assigned_action{action_status}[1].status
   acv.chosen_method := activity_method_map(act.chosen_method);
   acv.actual_end_date := date_time_map(
                           (act<-items{applied_date_and_time_assignment |
                                       role.name = 'actual end'}
                              ::assigned_date_and_time +
                            act<-items{applied_date_assignment |
                                       role.name = 'actual end'}
                              ::assigned_date)[1]);
   acv.actual_start_date := date_time_map(
                           (act<-items{applied_date_and_time_assignment |
                                       role.name = 'actual start'}
                              ::assigned_date_and_time +
                            act<-items{applied_date_assignment |
                                       role.name = 'actual start'}
                              ::assigned_date)[1]);
   acv.planned_end_date := date_time_map(
                           (act<-items{applied_date_and_time_assignment |
                                       role.name = 'planned end'}
                              ::assigned_date_and_time +
                            act<-items{applied_date_assignment |
                                       role.name = 'planned end'}
                              ::assigned_date)[1]);
   acv.planned_start_date     := date_time_map(
                           (act<-items{applied_date_and_time_assignment |
                                       role.name = 'planned start'}
                              ::assigned_date_and_time +
                            act<-items{applied_date_assignment |
                                       role.name = 'planned start'}
                              ::assigned_date)[1]);
   acv.concerned_organization :=
      FOR EACH org IN act<-items{applied_organization_assignment |
                                 role.name = 'concerned organization'}
                      ::assigned_organization;
      RETURN (organization_map(orig));
   acv.supplying_organization :=
      FOR EACH org IN act<-items{applied_organization_assignment |
                                 role.name = 'supplying organization'}
                      ::assigned_organization;
         RETURN (organization_map(orig));
   acv.requestor := person_organization_map(
                    act<-items{applied_person_and_organization_assignment |
```

```
                                role.name = 'requestor'}
                           ::assigned_person_and_organization[1]);
END_MAP;

MAP activity_relationship_map AS
   actirel : activity_relationship ;
FROM
   actrel : action_relationship ;
SELECT
   actirel.related := activity_map(actrel.related_action) ;
   actirel.relating := activity_map(actrel.relating_action) ;
   actirel.relation_type := actrel.name ;
   actirel.description := actrel.description ;
END_MAP ;

MAP activity_method_map AS
   am : activity_method ;
FROM
   actm : action_method ;
SELECT
   am.consequence := actm.consequence ;
   am.description := actm.description ;
   am.name        := actm.name ;
END_MAP ;

MAP activity_element_map AS
   ae : activity_element ;
FROM
   aaa : applied_action_assignment ;
WHERE
   aaa.role.description = 'activity element' ;
SELECT
   ae.role := aaa.role.name ;
   ae.associated_activity := activity_map(aaa.assigned_action) ;
   ae.element := activity_element_select_map(aaa.items[1]) ;
END_MAP ;
```

7.4.10.2 Work_request, Activity_method_assignment

A target instance of type Work_request is created out of a source instance of type Versioned_action_request which is referenced by an instance of type Action_request_status. A target instance of type Activity_method_assignment is created out of a source instance of type Action_request_solution.

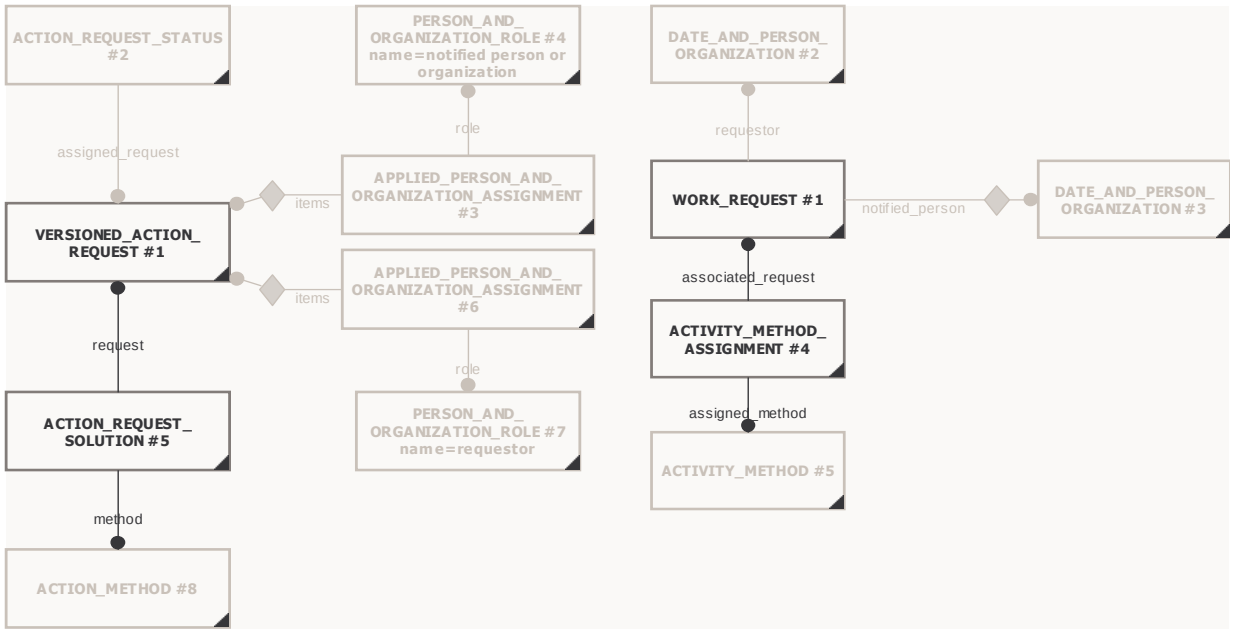


Figure 74 - Instance mapping for work request and activity method assignment

EXPRESS-X Specification:

```
MAP work_request_map AS
   wr : work_request ;
FROM
   va : versioned_action_request ;
   ar : action_request_status ;
WHERE
   ar.assigned_request := : va ;
SELECT
   wr.description := va.description ;
   wr.id := va.id ;
   wr.request_type := va.purpose ;
   wr.status := ar.status ;
   wr.version_id := va.version_id ;
   wr.scope :=
      FOR EACH it IN
         vaq<-assigned_action_request{applied_action_request_assignment |
                                    role.name = 'scope'}::items ;
         RETURN (activity_element_select_map(it));
   wr.notified_person :=
      FOR EACH po IN act<-items{applied_person_and_organization_assignment |
                             role.name='notified person or organization'}
                       ::assigned_person_and_organization;
         RETURN (person_organization_map(po)) ;
   wr.requestor := person_organization_map(
                    act<-items{applied_person_and_organization_assignment |
                              role.name = 'requestor'}
```

```
                        ::assigned_person_and_organization[1]);
END_MAP ;

MAP activity_method_assignment_map AS
   ara : activity_method_assignment ;
FROM
   ars : action_request_solution ;
SELECT
   ara.relation_type := ars.name ;
   ara.assigned_method := activity_method_map(ars.method) ;
   ara.associated_request := work_request_map(ars.request) ;
END_MAP ;
```

7.4.10.3 Work_order

A target instance of type Work_order is created out of a source instance of type Action_directive.

Figure 75 - Instance mapping for work order

EXPRESS-X Mapping Specification:

```
MAP work_order_map AS
   wo : work_order ;
FROM
   ad : action_directive ;
SELECT
   wo.description     := ad.comment
   wo.id              := ad.name;
   wo.work_order_type := ad.description;
   wo.version_id      := ad<-items{applied_identification_assignment |
                                role.name = 'version'} ::assigned_id[1] ;
   wo.is_controlling := FOR EACH da IN adr<-directive{directed_action};
                        RETURN (activity_map(da));
END_MAP ;
```

7.4.10.4 Project

A target instance of type Project is created out of a source instance of type Organizational_project. A target instance of type Project_relationship is created out of a source instance of type Organizational_project_relationship.

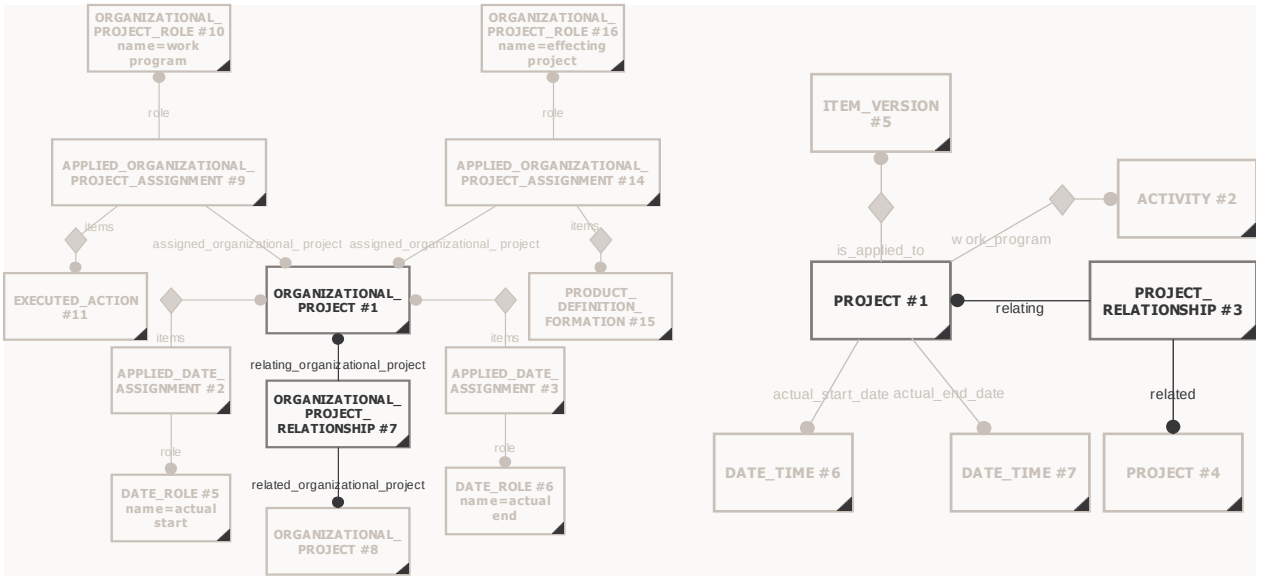


Figure 76 - Instance mapping for project

EXPRESS-X Mapping Specification:

```
MAP project_map AS
   pro : project;
FROM
   opr : organizational_project;
SELECT
   pro.description := opr.description;
   pro.id := opr.id;
   pro.name := opr.name;
   pro.actual_end_date :=
   pro.actual_start_date :=
   pro.planned_end_date :=
   pro.planned_start_date :=
   pro.work_program := activity_map(opr<-assigned_organizational_project
                                       {organizational_product_assignment|
                                        role.name = 'work program'}
                                     ::items{executed_action}[1]);
   pro.is_applied_to := FOR EACH it IN
                          opr<-assigned_organizational_project
                               {organizational_product_assignment |
                                role.name = 'affected item'}::items;
                        RETURN (project_information_select_map(it));
END_MAP;

MAP project_relationship_map AS
   prel : project_relationship;
FROM
   orel : organizational_project_relationship;
```

```
SELECT
   prel.related := project_map(orel.related_organizational_project);
   prel.relating := project_map(orel.relating_organizational_project);
   prel.relation_type := orel.name;
   prel.description := orel.description;
END_MAP;
```

7.4.10.5 Element_delivery

A target instance of Element_delivery is created out of a source instance of type Element_delivery.

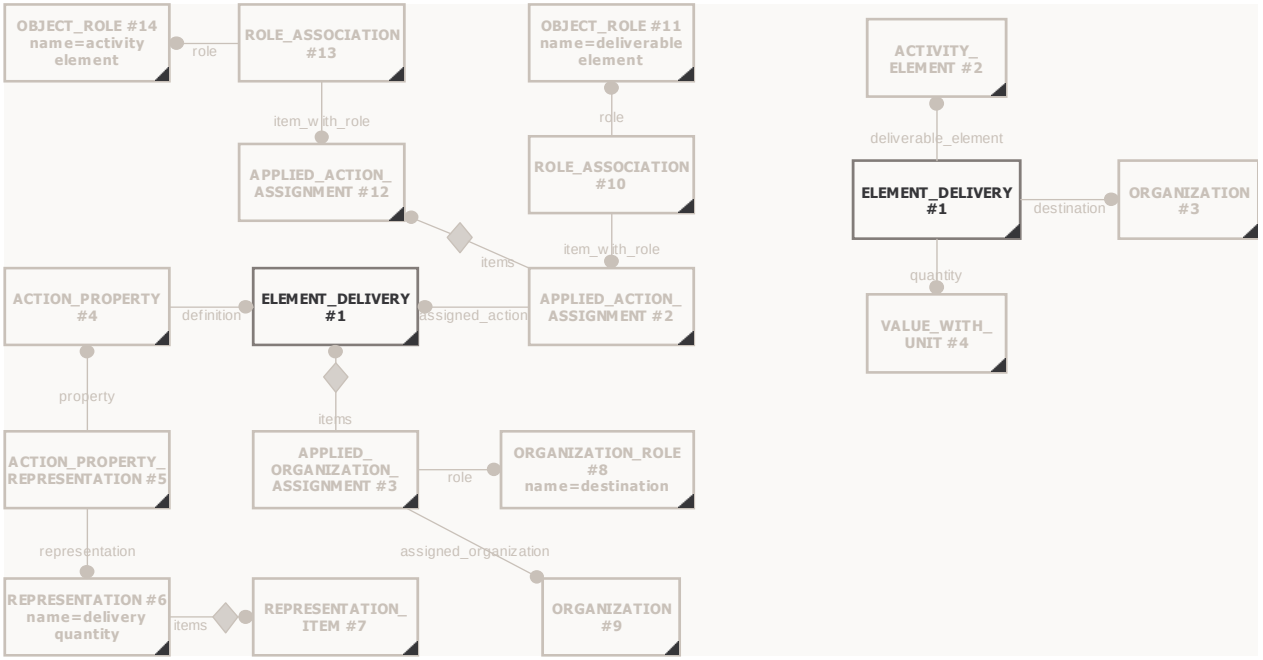


Figure 77 - Instance mapping for element delivery

EXPRESS-X Mapping Specification:

```
MAP element_delivery_map AS
   t_ed : element_delivery;
FROM
   s_ed : element_delivery;
SELECT
   t_ed.deliverable_element :=
      activity_element_map(s_ed<-assigned_action{applied_action_assignment |
                                        role.name = 'delivarable element'}
                           ::items{applied_action_assignment |
                             role.description = 'activity element'}[1]);;
   t_ed.destination :=
      organization_map(s_ed<-items{applied_organization_assignment |
                     role.name = 'destination'}::assigned_organization[1]);
   t_ed.quantity :=
```

```
      value_with_unit_map(t_ed<-definition{action_property}
                              <-property{action_property_representation}
                              ::respresentation{representation |
                                 name = 'derlivery quantity'}::items[1])
END_MAP;
```

7.4.11 Process planning

7.4.11.1 Process_plan, Process_plan_relationship

A target instance of type Process_plan is created out of a source instance of type Process_plan. If an Applied_identification_assignment with role name 'version' references the Product_process_plan as items, the created target instance is of subtype Product_plan_version. A target instance of type Process_plan_relationship is created out of a source instance of type Action_relationship that referenced instances of type Process_plan as relating_action and as related_action.

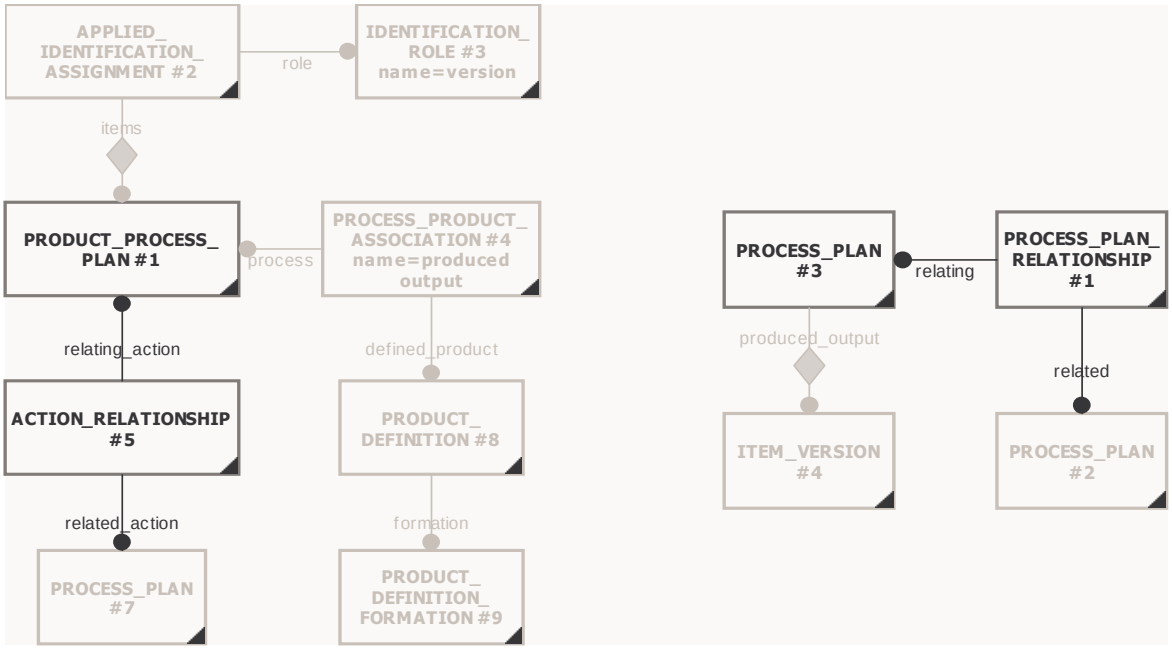


Figure 78 - Instance mapping for process plan

EXPRESS-X Mapping Specification:

```
MAP process_plan_map AS
   pp : process_plan;
FROM
   ppp : process_plan;
SELECT
   pp.description := ppp.description;
   pp.name := ppp.name;
   pp.plan_id := ppp.id;
   pp.produced_output :=
```

```
      item_version_map(ppp<-process{process_product_association |
                                    name = 'produced output'}
                          ::defined_product{product_definition}
                          ::formation[1]);
END_MAP;

MAP process_plan_version_map AS
   pp : process_plan_version;
SUBTYPE OF (process_plan_map);
WHERE
   SIZEOF(ppp<-items{applied_identification_assignment |
                     role.name = 'version'}) > 0;
SELECT
   pp.version_id := ppp<-items{applied_identification_assignment |
                               role.name = 'version'}[1].assigned_id;
END_MAP;

MAP process_plan_relationship_map AS
   ppr : process_plan_relationship;
FROM
   arel : action_relationship;
WHERE
   wr1: 'AUTOMOTIVE_DESIGN.PROCESS_PLAN' IN TYPEOF(arel.related_action);
   wr2: 'AUTOMOTIVE_DESIGN.PROCESS_PLAN' IN TYPEOF(arel.relating_action);
SELECT
   ppr.related := process_plan_map(arel.related_action);
   ppr.relating := process_plan_map(arel.relating_action);
   ppr.relation_type := arel.name;
   ppr.description := arel.description;
END_MAP;
```

7.4.11.2 Process_operation_definition

A target instance of Process_operation_definition is created out of a source instance of type Process_operation.

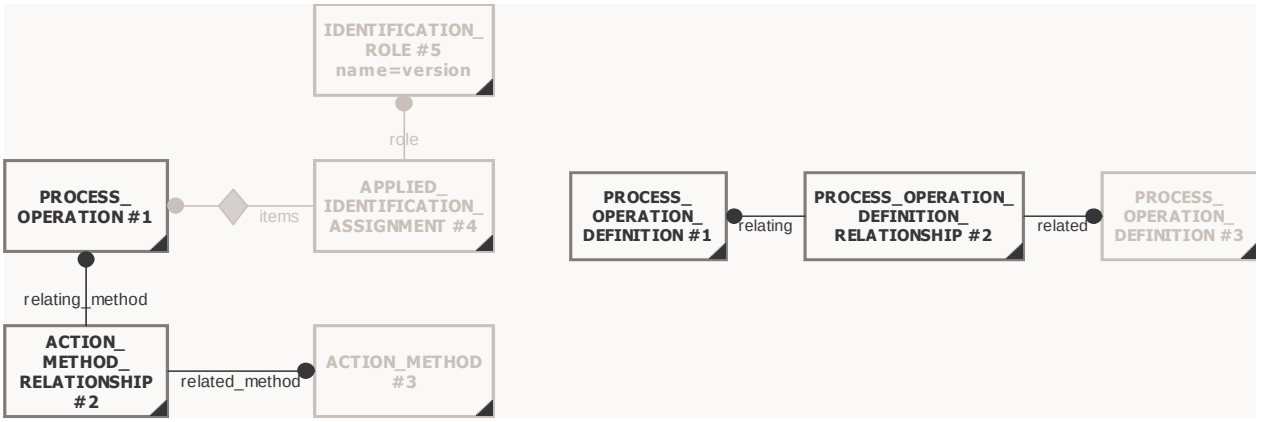


Figure 79 - Instance mapping for process operation definition

EXPRESS-X Mapping Specification:

```
MAP process_operation_definition_map AS
   pod : process_operation_definition;
FROM
   po : process_operation;
SELECT
   pod.description := po.description;
   pod.id := po.name;
   pod.name := po.purpose;
   pod.process_type := po.consequence;
   pod.version_id := po<-items{applied_identification_assignment |
                               role.name = 'version'}[1].assigned_id;
END_MAP;
```

7.4.11.3 Process_operation_occurrence

A target instance of type Process_operation_occurrence is created out of a source instance of type Action_relationship with name 'process operation occurrence'.

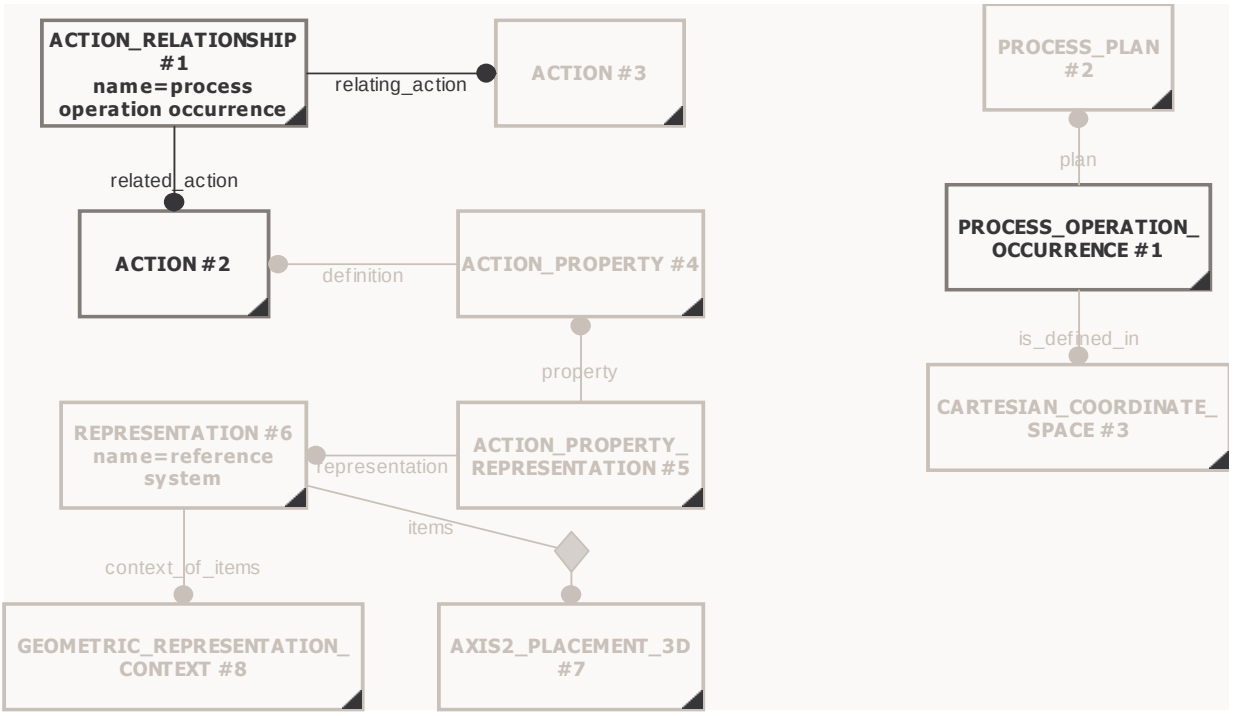


Figure 80 - Instance mapping for process operation occurence

EXPRESS-X Mapping Specification:

```
MAP process_operation_occurrence_map AS
   poo : process_operation_occurrence;
FROM
```

```
   arel : action_relationship;
SELECT
   poo.id := arel.related_action.id;
   poo.operation_definition :=
      process_operation_definition_map(arel.related_action.choosen_method[1]);
   poo.plan := process_plan_map(arel.relating_action);
   poo.is_defined_in :=
      cartesian_coordinate_space_map(arel.related_action
         <-definition{action_property}
         ::representation{representation |
                     name = 'reference system'}::context_of_items[1]);
END_MAP;
```

7.4.11.4 Process_operation_occurrence_relationship

A target instance of type Process_operation_occurrence_relationship is created out of a source instance of type Action_relationship which refers to action instances as related_action and relating_action which are both referenced by instances of type Action_relationship with name 'process operation occurrence' as related_action.

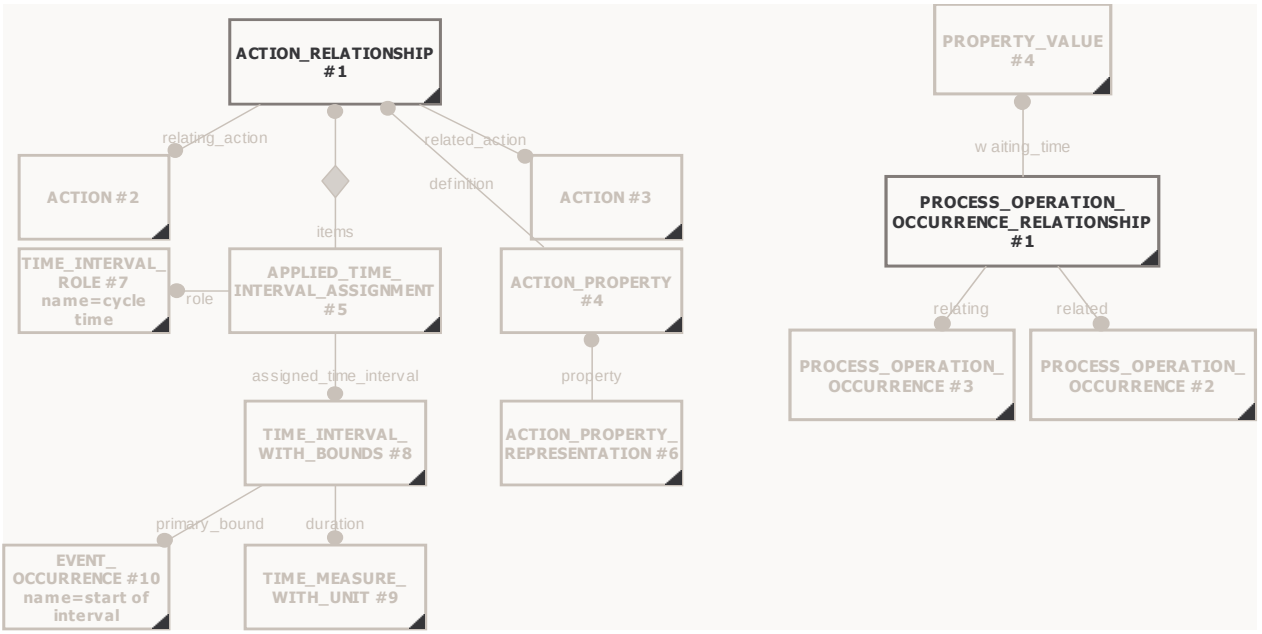


Figure 81 - Instance mapping for process operation occurence relationship

EXPRESS-X Mapping Specification:

```
MAP process_operation_occurrence_relationship_map AS
   por : process_operation_occurrence_relationship;
FROM
   arel : action_relationship;
WHERE
   wr1: SIZEOF(arel.relating_action<-related_action{action_relationship |
              name = 'process operation occurrence'}) > 0;
```

```
   wr2: SIZEOF(arel.related_action<-related_action{action_relationship |
              name = 'process operation occurrence'}) > 0;
SELECT
   por.cycle_time :=
   por.description := arel.description;
   por.relation_type := arel.name;
   por.related := process_operation_occurrence_map(arel.related_action);
   por.relating := process_operation_occurrence_map(arel.relating_action);
   por.cycle_time :=
      arel<-items{applied_time_interval_assignment | role.name='cycle time'}
        ::assigned_time_interval{time_interval_with_bounds |
                                 primary_bound.name = 'start of interval'}
        ::duration.value_component;
   por.waiting_time :=
      property_value_map(arel<-definition{action_property}
                          <-property{action_property_representation}
                         ::representation{representation}
                         ::items[1]);
END_MAP;
```

7.4.11.5 Process_property_association

A target instance of Process_property_association is created out of a source instance of type Action_property or Resource_property.

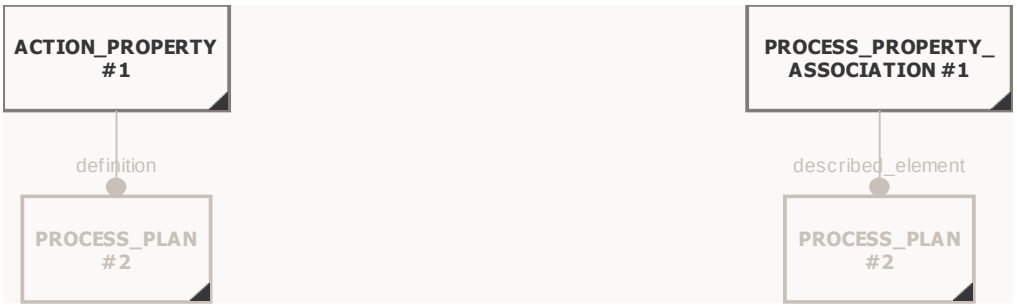


Figure 82 - Instance mapping for process property association

EXPRESS-X Mapping Specification:

```
MAP process_property_association_map AS
   ppa : process_property_association;
PARTITION p_ap;
FROM
   ap : action_property;
SELECT
   ppa.describing_property_value :=
      property_value_representation_map(
         ap<-property{action_property_representation}
            ::representation[1]);
   ppa.described_element := process_property_select_map(ppa.definition);
```

```
   ppa.validity_context :=
      IF SIZEOF(ap<-items{applied_organization_assignment |
                          name = 'validity context'}) > 0 THEN
         organization_map(ap<-items{applied_organization_assignment |
                                    name = 'validity context'}
                           ::assigned_organization[1]) ;
      ELSE
         IF SIZEOF(ap<-related_property_definition
                     {property_definition_relationship |
                      name = 'validity context'}
                    ::relating_property_definition{property_definition |
                      name = 'context definition'}
                    ::definition{product_class}) > 0
      THEN
         product_class_map(ap<-related_property_definition
                     {property_definition_relationship |
                      name = 'validity context'}
                    ::relating_property_definition{property_definition |
                      name = 'context definition'}
                    ::definition{product_class}[1]);
      ELSE
         product_identification_map(ap<-related_property_definition
                     {property_definition_relationship |
                      name = 'validity context'}
                    ::relating_property_definition{property_definition |
                      name = 'context definition'}
                    ::definition{product_identification}[1]);
         END_IF;
      END_IF;
PARTITION p_res;
FROM
   rp : resource_property;
SELECT
   ppa.describing_property_value :=
      property_value_representation_map(
         rp<-property{resource_property_representation}
            ::representation[1]);
   ppa.described_element := process_property_select_map(ppa.definition);
   ppa.validity_context :=
      IF SIZEOF(rp<-items{applied_organization_assignment |
                          name = 'validity context'}) > 0 THEN
         organization_map(rp<-items{applied_organization_assignment |
                                    name = 'validity context'}
                           ::assigned_organization[1]) ;
      ELSE
         IF SIZEOF(rp<-related_property_definition
                     {property_definition_relationship |
                      name = 'validity context'}
                    ::relating_property_definition{property_definition |
                      name = 'context definition'}
                    ::definition{product_class}) > 0
```

```
        THEN
           product_class_map(rp<-related_property_definition
                    {property_definition_relationship |
                     name = 'validity context'}
                   ::relating_property_definition{property_definition |
                     name = 'context definition'}
                   ::definition{product_class}[1]);
        ELSE
           product_identification_map(rp<-related_property_definition
                    {property_definition_relationship |
                     name = 'validity context'}
                   ::relating_property_definition{property_definition |
                     name = 'context definition'}
                   ::definition{product_identification}[1]);
        END_IF;
      END_IF;
END_MAP;
```

7.4.11.6 Process_operation_resource_assignment

A target instance of Process_operation_resource_assignment is created out of a source instance of type Requirement_for_action_resource.

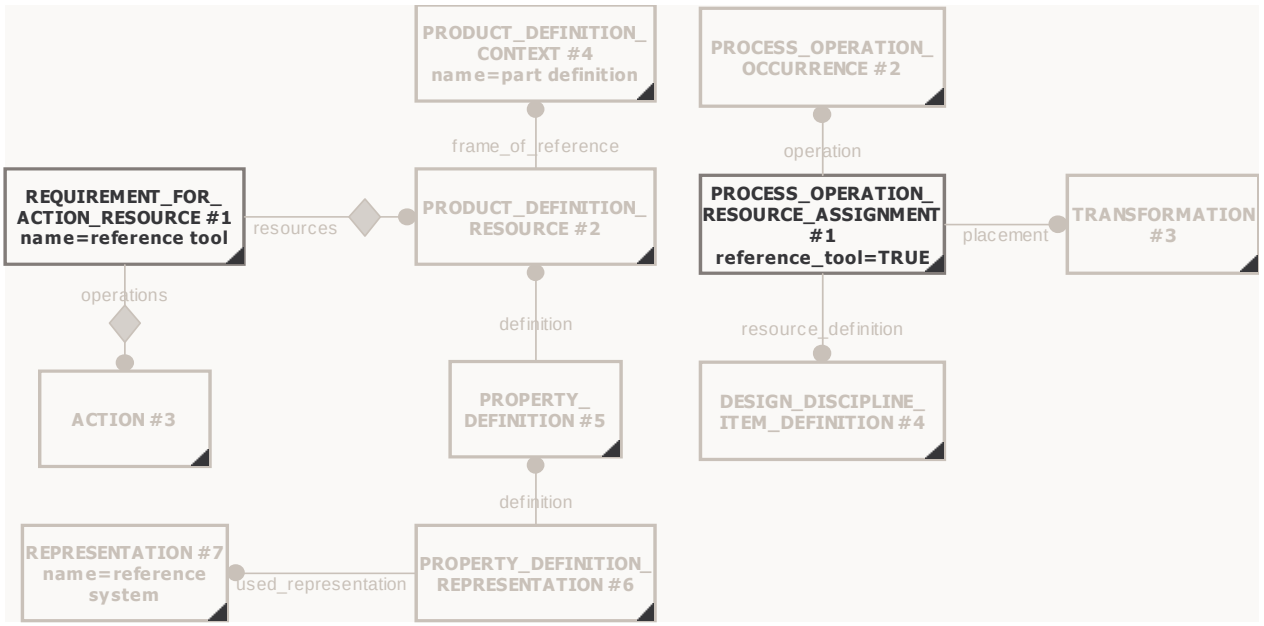


Figure 83 - Instance mapping for process operation resource assignment

EXPRESS-X Mapping Specification:

```
MAP process_operation_resouce_assignment_map AS
   pora : process_operation_resource_assignment;
FROM
```

```
   rfar : requirement_for_action_resource;
SELECT
   pora.reason := rfar.description;
   pora.reference_tool := rfar.name = 'reference tool';
   pora.operation :=
      product_operation_occurrence_map(rfar.operations{action}
         <-related_action{action_relationship |
                        name = 'process operation occurrence'}[1]);
END_MAP;
```

7.4.11.7 Process_operation_input_or_output

A target instance of Process_operation_input_or_output is created out of a source instance of type Process_product_association.

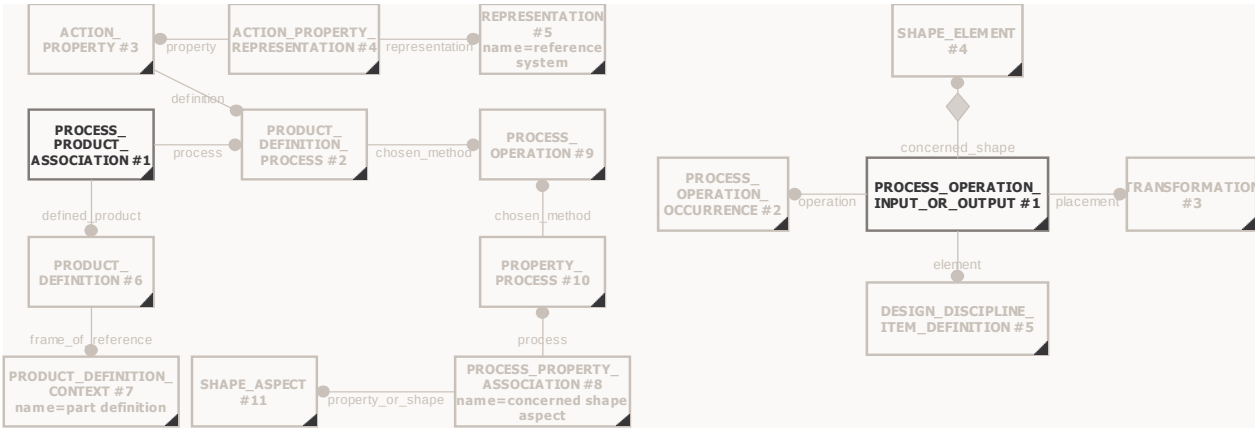


Figure 84 - Instance mapping for process operation input or output

EXPRESS-X Mapping Specification:

```
MAP process_operation_input_or_output_map AS
   poio : process_operation_input_or_output;
FROM
   ppa : process_product_association;
SELECT
   poio.description := ppa.description;
   poio.role := ppa.name;
   poio.concerned_shape := FOR EACH sasp IN ppa.process
                           <-process{process_property_association |
                                     name = 'concerned shape aspect'}
                           ::property_or_shape{shape_aspect};
                           RETURN (shape_element_map(saps));
   poio.element :=
      process_operation_input_or_output_select_map(ppa.defined_product);
END_MAP;
```

7.4.11.8 Descriptive_specification

A target instance of Descriptive_specification is created out of a source instance of type Descriptive_representation_item.

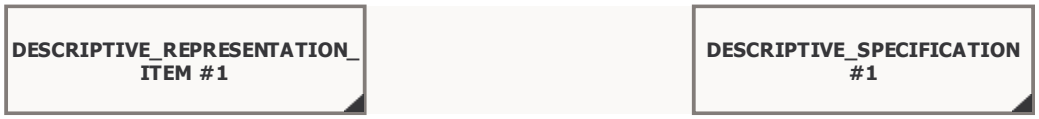


Figure 85 - Instance mapping for descriptive specification

EXPRESS-X Mapping Specification:

```
DEPENDENT_MAP descriptive_specification_map AS
   desp : descriptive_specification;
FROM
   deri : descriptive_representation_item;
SELECT
   desp.description := deri.description;
   desp.id := deri.name;
END_DEPENDENT_MAP;
```

7.4.12 Multi-Language support

7.4.12.1 Language

A target instance of type Language is created out of a source instance of type Language.

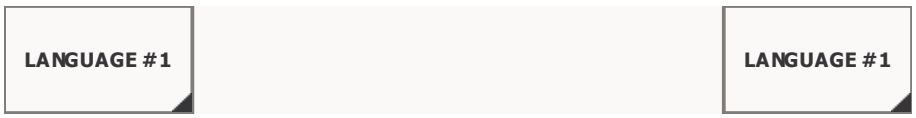


Figure 86 - Instance mapping for language

EXPRESS-X Mapping Specification:

```
MAP language_map AS
   t_lan : language;
FROM
   s_lan : language;
SELECT
   t_lan.country_code := s_lan.description;
   t_lan.language_code := s_lan.name;
END_MAP;
```

7.4.12.2 String_with_language

A target instance of type String_with_language is created out of a source instance of type Attribute_language_assignment.

EXPRESS-X Mapping Specification:

```
MAP string_with_language_map AS
  sl : string_with_language;
FROM
  ala : attribute_language_assignment;
SELECT
  sl.contents := ala.attribute_value;
  sl.language_specification := language_map(ala.assigned_class);
END_MAP;
```

7.4.12.3 Multi_language_string

A target instance of type Multi_language_string is created out of a source instance of type Multi_language_attribute_assignment.

EXPRESS-X Mapping Specification:

```
MAP string_with_language_map AS
   mls : multi_language_string;
FROM
   mla : multi_language_attribute_assignment;
SELECT
   mla.primary_language_dependent_string := string_with_language_map(mla);
   mla.additional_language_dependent_string :=
      FOR EACH it IN mla<-items{multi_language_attribute_assignment |
                               role.name = 'alternate language'};
         RETURN (string_with_language_map(it));
END_MAP;
```

7.5 PIM Equivalence Model

The PIM Equivalence Model is given in Appendix A.

7.6 EXPRESS to XMI Mapping

The mapping of EXPRESS to XMI is a two step process to ensure that the semantic information is transformed from EXPRESS into UML and then partly rearranged into a more compact model.

7.6.1 Standard mapping

This mapping is based on the ISO 10303-25 [5] Technical Specification, which defines a mapping between EXPRESS Schema and XMI. The standardization of the Technical Specification is still in progress, therefore the Committee Draft of February 24th 2003 was considered. Some rules were adapted or added to fulfill all needed requirements.

The mapping is applied to the PIM equivalence model described in Chapter 7.5. The result of the mapping is the PLM reference model represented in UML and is serving as the informational PIM as described in Chapter 7.7.

To reduce the complexity of the model obtained by original ISO TS 10303-25, some of the rules were adapted or added. These rules are explained here.

[1] Throughout the whole Part 25, navigation in UML is not explicitly discussed. Therefore associations between classes are unidirectional if the corresponding construct in EXPRESS does not explicitly define an inverse attribute, otherwise bi-directional with role names given by the attribute names.

[2] EXPRESS SELECT types are mapped to empty interfaces. Corresponding choices realize this interface. Each interface is named as the corresponding SELECT type. Nested SELECT type hierarchies are flattened before mapping them to interfaces. Therefore all sub-SELECT types of a SELECT type are replaced by their underlying types. This is done recursively till a SELECT type only contains non-SELECT data types.
If one of the sub-SELECT types is not used anymore (e.g. by an attribute of an ENTITY) it is not mapped into the UML model.

Example:

```
TYPE shape_information_select = SELECT (
   shape_element_relationship,
   shaped_element_select
   );
END_TYPE;

TYPE shaped_element_select = SELECT (
   shape_element,
   item_shape
   );
END_TYPE;
```

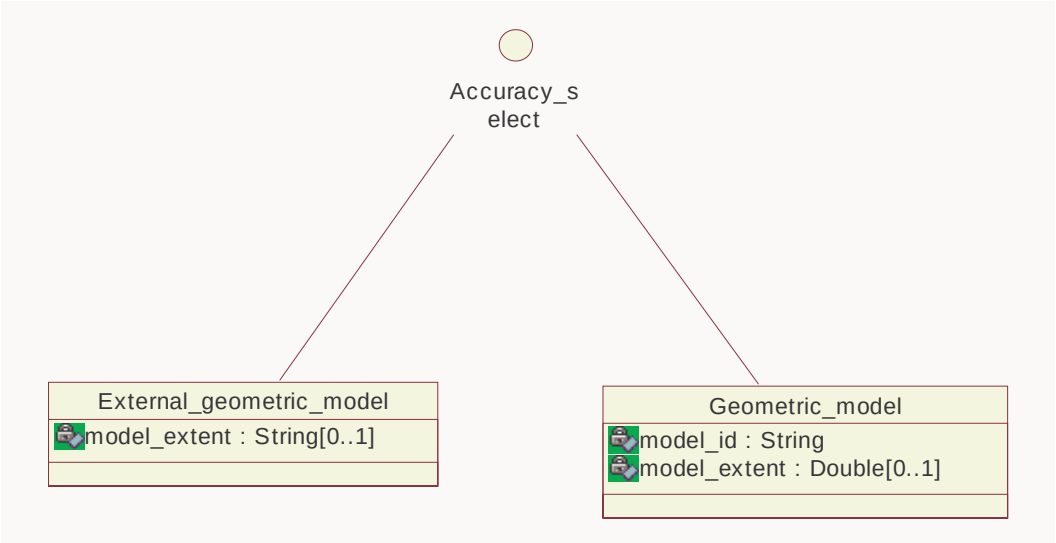


Figure 87 - UML interface modelled from EXPRESS SELECT

The following SELECT types were flattened:

- date_time_person_organization_element_select
- general_organizational_data_select

- configured_item_select
- documented_element_select
- simple_property_select
- shape_information_select

These SELECT types became unnecessary and were not mapped:

- general_organizational_data_sub_select
- documented_element_sub_select
- shaped_element_select

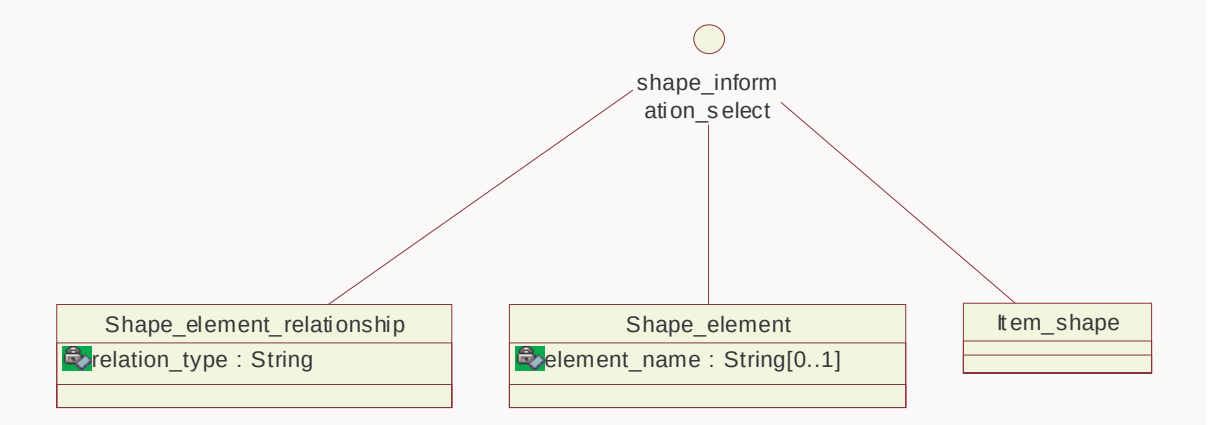


Figure 88 - UML interface modelled from nested EXPRESS SELECTS

[1] Optional attributes are mapped using the lower multiplicity "0" of attributes and relations in the UML model.

[2] Additionally, the following general mapping restrictions were defined:

- AND/OR inheritance is not supported (all classes inherit in a simple way).
- SELF statements are not mapped to a UML model construct. The restrictions are mentioned in the descriptions of the UML attributes.
- INVERSE attributes are mapped but later removed in the 2nd step (see Chapter 7.6.2: only influences on existing cardinalities are taken over but no inverse attribute roles).
- OPTIONAL SET [2:?] is treated as SET [0:?]. The prohibited multiplicity of exactly 1 is mentioned in the description of the UML attribute.
- Comments are not mapped.

The mapping of simple data types is defined as follows:

EXPRESS fragment	Resulting UML Interchange Model fragment
STRING	UML Datatype with name String
BOOLEAN	UML Datatype with name Boolean
NUMBER	UML Datatype with name Double
REAL	UML Datatype with name Double
INTEGER	UML Datatype with name Integer

Because in UML all relations between classes are considered as mathematical sets, special attention has to be paid to all EXPRESS aggregation types. This is reflected in the following table with a relevant subset of mapping rules taken from [5], Annex E. Some rules were adapted or added. The table summarizes all used mappings for complex modeling constructs.

EXPRESS fragment	Resulting UML Interchange Model fragment
SCHEMA s1	UML Model with name s1
ENTITY e2; ENTITY e1 SUBTYPE OF (e2);	UML Class with name E2· UML Class with name E1· UML Generalization with UML Class E1 as child and UML Class E2 as parent
ENTITY e2; ENTITY e1 ABSTRACT SUBTYPE OF (e2);	UML Class with name E2· Abstract UML Class with name E1· UML Generalization with UML Class E1 as child, and UML Class E2 as parent
TYPE t1 = SELECT (e1,e2);	UML Interface with name T1 · UML Class with name E1 which implements Interface T1· UML Class with name E2 which implements Interface T1
TYPE t1 = String;	UML Datatype with name T1 · UML Generalization with Datatype T1 as child, and the UML Datatype of the underlying simple type as parent
ENTITY e1; a1: STRING;	UML Class with name E1 UML Attribute with name a1, type is the Datatype of the used simple type, cardinality is [1]
ENTITY e1; a1: OPTIONAL STRING;	UML Class with name E1 · UML multi-valued Attribute with name a1, type is the Datatype of the used simple type, cardinality is [0..1]
ENTITY e1; a1: LIST [3:3] OF REAL;	UML Class with name E1 · UML multi-valued Attribute with name a1, type is the Datatype of the used simple type, cardinality is [3], is ordered
ENTITY e1 ENTITY e2; a1: e1;	UML Class with name E1 UML Class with name E2, UML Association with only one specified AssociationEnd: Role name is a1, Role type is Class E1, cardinality is [1]

ENTITY e1; ENTITY e2; a1: OPTIONAL e1;	UML Class with name E1 UML Class with name E2 UML Association with only one specified AssociationEnd: Role name is a1, Role type is Class E1, cardinality is [0..1]
ENTITY e1; ENTITY e2; a1: SET [1:?] OF e1;	UML Class with name E1 UML Class with name E2 UML Association with only one specified AssociationEnd: Role name is a1, Role type is Class E1, cardinality is [1..*]
ENTITY e1; ENTITY e2; a1: OPTIONAL SET [1:?] OF e1;	UML Class with name E1 UML Class with name E2 UML Association with only one specified AssociationEnd: Role name is a1, Role type is Class E1, cardinality is [0..*]
ENTITY e1; INVERSE a2: SET[1:?] OF e2 FOR a1; ENTITY e2; a1: e1;	UML Class with name E1 UML Class with name E2 UML Association with two specified AssociationEnds: Role names are a1 and a2, Role types are Class E1 and Class E2, cardinalities are [1] and [1..*]

7.6.2 Customized mapping based on domain knowledge

ISO 10303-25 is the base for the general concept of transforming EXPRESS into UML, but does not take any domain knowledge into account. So, after obtaining the UML model by applying the general mapping to the EXPRESS schema, another mapping step is taken to introduce some mapping concepts based on the domain knowledge. The intent of this map-ping is to reduce further the complexity of the overall model and to add some information originating from the application domain. These rules cannot be applied automatically to the overall model, instead they are applied manually after close examination of the model obtained from the first mapping step. To understands these concepts, some simple examples are provided here.

For EXPRESS attributes whose domain is an EXPRESS named type with multiplicity "1", inversion and containment can be applied. The role name of the newly created composition is taken from the associated class in lower case, and the multiplicity is 0..*.

Example:

```
ENTITY item_version;
   id : STRING;
   associated_item : item;
   description : OPTIONAL string_select;
END_ENTITY;

ENTITY item_version_relationship;
   relating : item_version;
   related : item_version;
   description : OPTIONAL string_select;
   relation_type : STRING;
END_ENTITY;
```

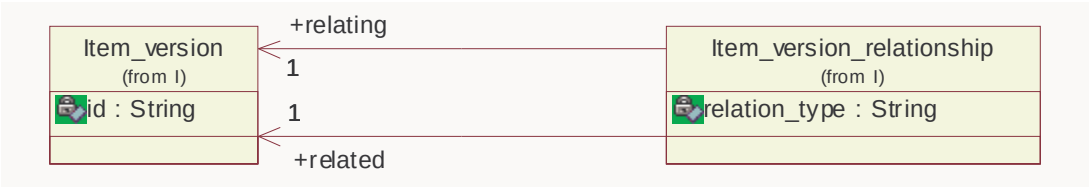


Figure 89 - UML classes related by uni-directional association

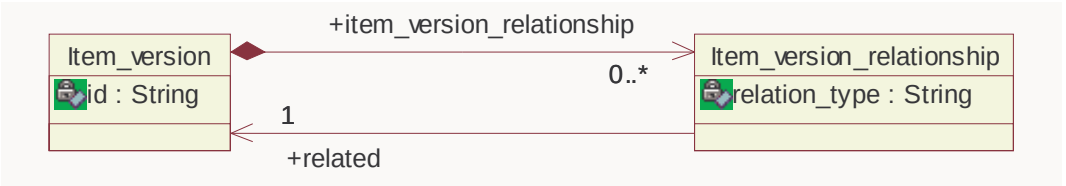


Figure 90 - UML composition modelled from uni-directional association

For EXPRESS attributes whose domain is an EXPRESS named type with multiplicity "1" and an INVERSE attribute already exists, inversion and containment can be applied. Multiplicities are taken over but the role names not.

Example:

```
ENTITY item;
   id : STRING;
   name : string_select;
   description : OPTIONAL string_select;
   INVERSE
      associated_version : SET[1:?] OF item_version FOR associated_item;
      item_classification : SET[1:?] OF specific_item_classification FOR
      associated_item;
END_ENTITY;

ENTITY item_version;
   id : STRING;
   associated_item : item;
   description : OPTIONAL string_select;
END_ENTITY;
```

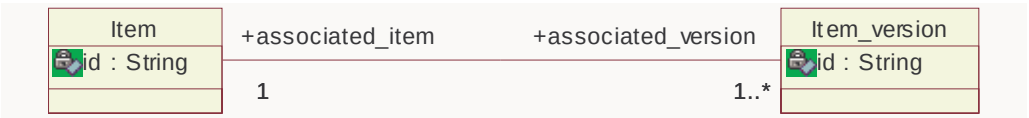


Figure 91 - UML classes related by bi-directional association

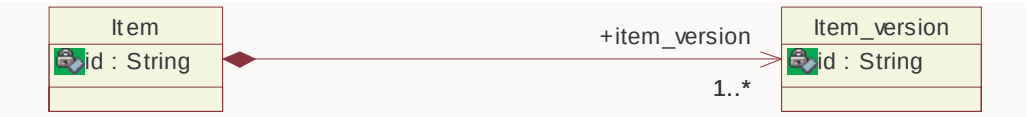


Figure 92 - UML composition modelled from bi-directional association

For EXPRESS attributes whose domain is an EXPRESS SELECT, inversion and containment can be applied for each type of the SELECT statement. This eliminates the now obsolete UML interface also, if it is not used anywhere in the model.

Example:

```
ENTITY item_instance
   ABSTRACT SUPERTYPE OF ( ONEOF(single_instance,quantified_instance,
         selected_instance,specified_instance))
   SUBTYPE OF (product_constituent);
   description : OPTIONAL string_select;
   definition : instance_definition_select;
   id : STRING;
END_ENTITY;

TYPE instance_definition_select = SELECT (
   design_discipline_item_definition,
   product_identification
   );
END_TYPE;

ENTITY design_discipline_item_definition;
   name : OPTIONAL string_select;
   id : STRING;
   associated_item_version : item_version;
   additional_context : SET[0:?] OF application_context;
   initial_context : application_context;
END_ENTITY;

ENTITY product_identification;
   associated_product_class : product_class;
   name : OPTIONAL string_select;
   version_id : OPTIONAL STRING;
   id : STRING;
   description : OPTIONAL string_select;
   INVERSE
      associated_design : SET[0:1] OF product_design FOR product;
END_ENTITY;
```

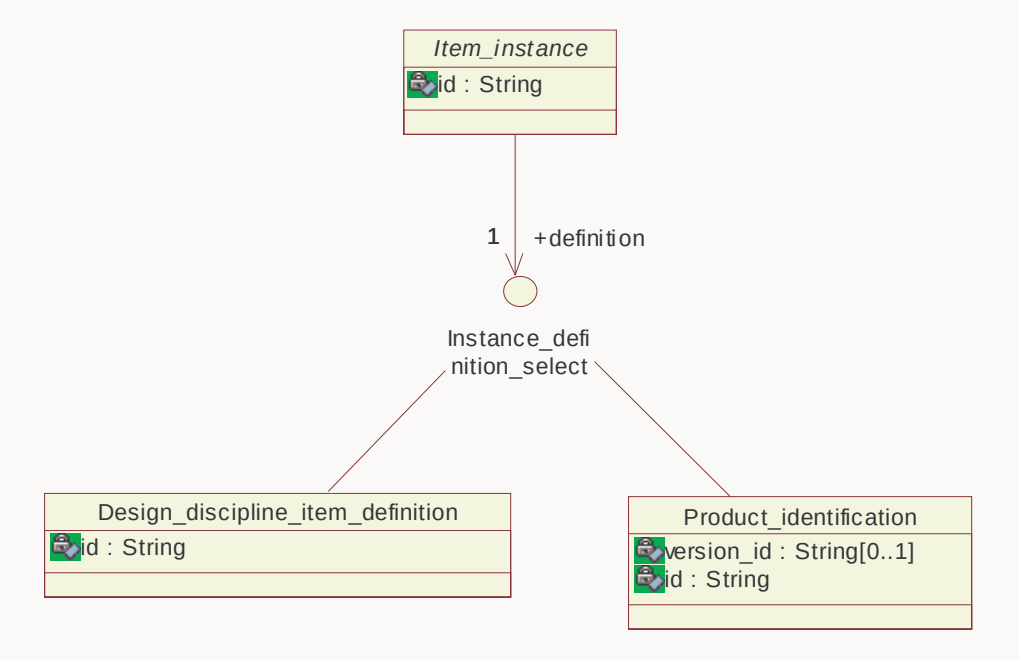


Figure 93 - UML classes related by SELECT statement

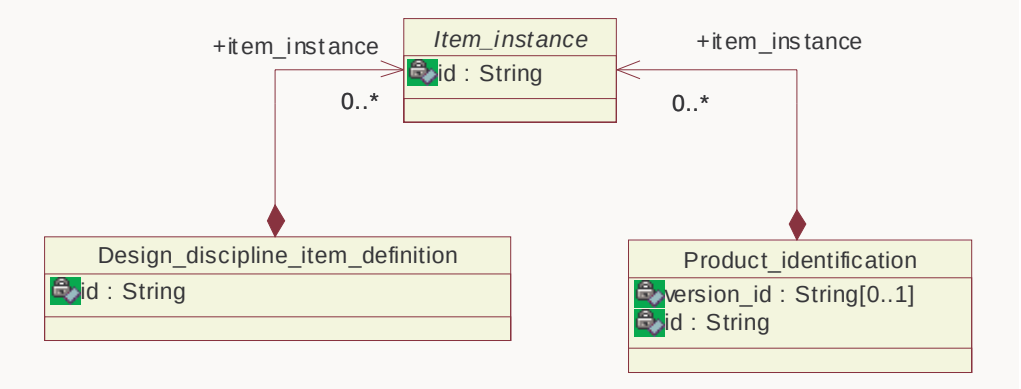


Figure 94 - UML composition modelled from SELECT statement

The following attributes of the elements were mapped as compositions:

EXPRESS Entity	EXPRESS Attribute	Attribute Type
activity_element	associated_activity	activity
activity_method_assignment	assigned_method	activity_method
activity_relationship	relating	activity
alias_identification	is_applied_to	alias_select
approval	status	approval_status

approval_relationship	relating	approval
class_category_association	associated_product_class	product_class
class_condition_association	associated_product_class	product_class
class_inclusion_association	associated_product_class	product_class
class_specification_association	associated_product_class	product_class
class_structure_relationship	relating	product_class
classification_association	associated_classification	general_classification
complex_product_relationship	relating	complex_product
component_placement	placed_component	product_component
configuration	configured_element	configured_item_select
date_and_person_assignment	assigned_date_and_person	date_and_person_organization
date_and_person_organization	person_or_organization	person_organization_select
date_time_assignment	assigned_date_time	date_time
design_constraint_association	is_constraining	complex_product
design_constraint_relationship	relating	design_constraint
design_discipline_item_definition	associated_item_version	item_version
document_assignment	is_assigned_to	documented_element_select
document_representation	associated_document_version	document_version
document_structure	relating	document_representation
document_version	associated_document	document
document_version_relationship	relating	document_version
effectivity_assignment	assigned_effectivity	effectivity
element_delivery	deliverable_element	activity_element
external_file_id_and_location	location	document_location_property
external_model	is_defined_as	digital_file
general_classification_hierarchy	super_classification	general_classification
geometric_model_relationship	relating	geometric_or_external_model_select
instance_placement	placed_instance	single_instance
item_definition_instance_relationship	relating	design_discipline_item_definition
item_definition_relationship	relating	design_discipline_item_definition
item_function_association	associated_item	design_discipline_item_definition
item_instance	definition	instance_definition_select
item_instance_relationship	relating	item_instance
item_version	associated_item	item

item_version_relationship	relating	item_version
manufacturing_configuration	configured_element	item_instance
material_property_association	described_material	material
person_in_organization	associated_person	person
person_organization_assignment	assigned_person_organization	person_organization_select
physical_assembly_relationship	physical_assembly	physical_instance
physical_instance_test_result	tested_instance	physical_instance
process_operation_definition_relationship	relating	process_operation_definition
process_operation_input_or_output	operation	process_operation_occurrence
process_operation_occurrence_relationship	relating	process_operation_occurrence
process_operation_resource_assignment	operation	process_operation_occurrence
process_plan_relationship	relating	process_plan
product_design	design	item_version
product_identification	associated_product_class	product_class
product_structure_relationship	relating	complex_product
project_relationship	relating	project
property_value_association	describing_property_value	property_value_representation
property_value_representation	specified_value	property_value
shape_description_association	is_defining_shape_for	shape_information_select
shape_element	composition	item_shape
shape_element_relationship	relating	shape_element
simple_property_association	described_element	simple_property_select
specific_document_classification_hierarchy	super_classification	specific_document_classification
specific_item_classification_hierarchy	super_classification	specific_item_classification
specification_category_hierarchy	super_category	specification_category
specification_inclusion	if_condition	specification_operand_select

7.7 Informational PIM

In this section the transformations of Chapter 7.6 are applied to the EXPRESS PIM Equivalence model model in Chapter 7.5. Where applicable the design was adapted by the proposed modelling constructs described in Chapter 7.6.2.

Additionally, some new classes were created and put into a package called "PLM_Base". This package realizes two modeling concepts of the PIM. Firstly, it introduces the concept of identifying instances by an unique identifier. This identifier must be unique throughout a session as defined by the computational model in Chapter 8. Secondly, it defines a container concept to establish a correct handling of the data passed to and from the computational model.

All classes and interfaces are listed with their packages, base classes, attributes, compositions and associations. Additionally the classes and their members are described textually. The text of all descriptions (except for the "PLM Base") are reproduced from ISO 10303-214 with permission of ISO. The copyright remains with ISO.

The PIM Informational Model has the following package hierarchy:

Package PLM_services

- Package PLM_base
- Package Part_identification
- Package Part_structure
- Package Document_and_file_management
- Package Shape_definition and_transformation
- Package Classification
- Package Properties
- Package Alias_identification
- Package Authorization
- Package Configuration_management
- Package Change_and_work_management
- Package Process_planning
- Package Multi_language_support

7.7.1 Package PLM_base

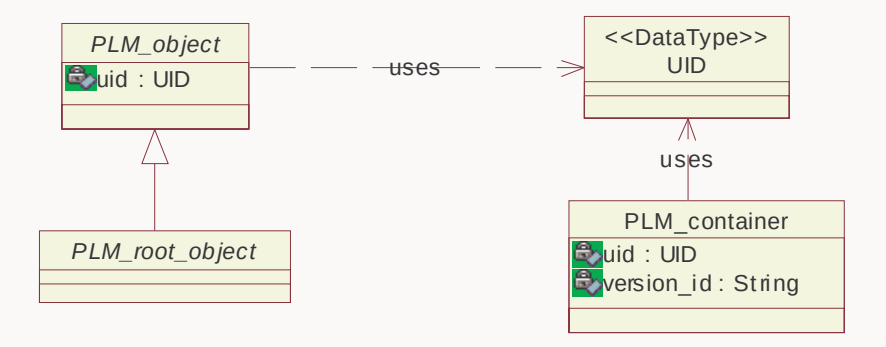


Figure 95 - PLM base

7.7.1.1 Class PLM_container

The PLM_container class is introduced to ensure that data is only handled by the computational model in a valid way.

Base Class

- none

Attributes

- uid : UID [1] — The uid uniquely identifies an object throughout a complete session defined by the computational model. After each session the uid is invalid.
- version_id : String [1] — The version_id specifies the version of the underlying PLM specification. The version for this specification shall be "1.1".

Compositions

- activity : Activity [0..*]
- classification_system : Classification_system [0..*]
- classification_attribute : Classification_attribute [0..*]
- complex_product : Complex_product (ABS) [0..*]
- address : Address [0..*]
- application_context : Application_context [0..*]
- data_environment : Data_environment [0..*]
- activity_method : Activity_method [0..*]
- approval_status : Approval_status [0..*]
- axis2_placement_3d : Axis2_placement_3d [0..*]
- cartesian_coordinate_space : Cartesian_coordinate_space (ABS) [0..*]
- cartesian_point : Cartesian_point [0..*]
- accuracy : Accuracy [0..*]
- design_constraint : Design_constraint [0..*]
- direction : Direction [0..*]
- date_time : Date_time [0..*]
- descriptive_specification : Descriptive_specification [0..*]
- document_content_property : Document_content_property [0..*]
- document : Document [0..*]
- document_file : Document_file (ABS) [0..*]
- document_format_property : Document_format_property [0..*]
- document_location_property : Document_location_property [0..*]
- document_creation_property : Document_creation_property [0..*]
- document_type_property : Document_type_property [0..*]
- duration : Duration [0..*]
- document_size_property : Document_size_property [0..*]

- item : Item [0..*]
- item_shape : Item_shape [0..*]
- language : Language [0..*]
- effectivity : Effectivity [0..*]
- event_reference : Event_reference [0..*]
- external_library_reference : External_library_reference [0..*]
- material : Material [0..*]
- organization : Organization [0..*]
- person : Person [0..*]
- physical_instance : Physical_instance [0..*]
- general_classification : General_classification [0..*]
- geometric_model : Geometric_model [0..*]
- rectangular_size : Rectangular_size [0..*]
- specific_document_classification : Specific_document_classification [0..*]
- specific_item_classification : Specific_item_classification [0..*]
- specification : Specification [0..*]
- process_operation_definition : Process_operation_definition [0..*]
- process_operation_occurrence : Process_operation_occurrence [0..*]
- process_plan : Process_plan [0..*]
- product_class : Product_class [0..*]
- project : Project [0..*]
- specification_expression : Specification_expression [0..*]
- unit : Unit [0..*]
- work_request : Work_request [0..*]
- work_order : Work_order [0..*]
- property_value : Property_value (ABS) [0..*]
- property : Property (ABS) [0..*]
- specification_category : Specification_category [0..*]
- transformation : Transformation (ABS) [0..*]

Associations

- none

7.7.1.2 Class PLM_object (ABS)

The abstract PLM_object class is introduced to provide a mechanism of binding a unique identifier to each PLM class instance. These identifiers must be valid and unique throughout a complete session defined by the computational model. After each session the identifiers may are invalid.

Base Class

- none

Attributes

- uid : UID [1] — The uid uniquely identifies an object throughout a complete session defined by the computational model. After each session the uid is invalid.

Compositions

- none

Associations

- none

7.7.1.3 Class PLM_root_object (ABS)

The abstract class PLM_root_object is defined to distinguish between types which can be directly inserted into PLM_container instances and types which are contained in the container through PLM_root_object instances.

Base Class

- PLM_object (ABS)

Attributes

- none

Compositions

- none

Associations

- none

7.7.1.4 Datatypes

Datatype Boolean

Datatype Double

Datatype Integer

Datatype String

Datatype UID

7.7.2 Package Part_identification

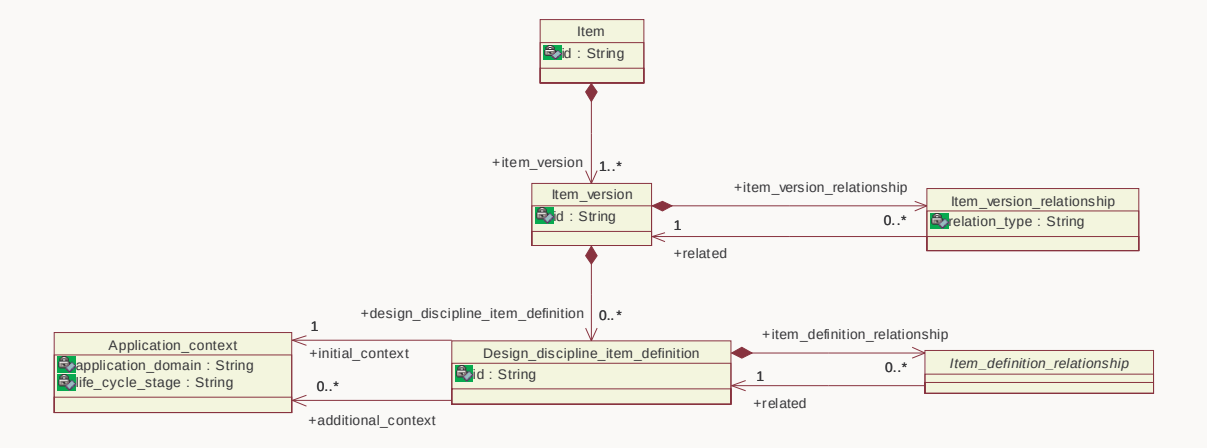


Figure 96 - Part identification

7.7.2.1 Class Application_context

An Application_context is a shared universe of discourse.

Base Class

- PLM_root_object (ABS)

Attributes

- application_domain : String [1] The application_domain is the identification of the applications for which an object may be relevant. Where applicable the following values shall be used:
  - 'assembly study': The object may be relevant for an assembly study;
  - 'digital mock-up': The object may be relevant for digital mock-up;
  - 'electrical design': The object may be relevant for the electrical design;
  - 'mechanical design': The object may be relevant for the mechanical design;
  - 'preliminary design': The object may be relevant for the preliminary design;
  - 'process planning': The object may be relevant for the process planning.
- life_cycle_stage : String [1] The life_cycle_stage is the specification of the general stage in the product life cycle to which the concerned items belong. Where applicable the following values shall be used:
  - 'design': The concerned item belongs to the design phase of the life cycle;
  - 'manufacturing': The concerned item belongs to the manufacturing phase of the life cycle;
  - 'recycling': The concerned item belongs to the recycling phase of the life cycle.

Compositions

- description : String_select [0..1]
  The description specifies additional information about the Application_context.

Associations

- none

7.7.2.2 Class Design_discipline_item_definition

A Design_discipline_item_definition is a view of an Item_version. This view is relevant for the requirements of one or more life cycle stages and application domains and collects product data of the Item_version.

Base Class

- PLM_object (ABS)

Attributes

- id : String [1] The id specifies the identifier of the Design_discipline_item_definition.

Compositions

- item_instance : Item_instance (ABS) [0..*]
  The item_instance specifies the item_instance that is defined by this Design_discipline_item_definition.
- item_definition_relationship : Item_definition_relationship (ABS) [0..*]
  The item_definition_relationship specifies the Item_definition_relationship that relates the first of the two Design_discipline_item_definition objects.
- name : String_select [0..1] The name specifies the word or group of words used to refer to the Design_discipline_item_definition.
- document_assignment : Document_assignment [0..*]
  The document_assignment specifies the object that provides information for this Design_discipline_item_definition.
- item_function_association : Item_function_association [0..*]
  The item_function_association specifies the Item_function_association which this Design_discipline_item_definition is associated with.
- alias_identification : Alias_identification [0..*]
  The Alias_identification specifies the Alias_identification that is applied to this Design_discipline_item_definition.
- simple_property_association : Simple_property_association (ABS) [0..*]
  The simple_property_association specifies the assigned simple property values.
- Item_definition_instance_relationship : Item_definition_instance_relationship (ABS) [0..*]
  The Item_definition_instance_relationship specifies the Item_definition_instance_relationship which this Design_discipline_item_definition is part of. If the Design_discipline_item_definition is an Assembly_definition, the relationship shall be a Assembly_component_relationship. If the Design_discipline_item_definition is an Collection_definition, the relationship shall be a Collected_item_association.

Associations

- initial_context : Application_context [1]
  The initial_context specifies the Application_context in which this view of the Item_version has been designed primarily.
- additional_context : Application_context [0..*]
  The additional_context specifies the set of Application_context objects in which this view of the Item_version is also relevant. The additional_context shall not contain the Application_context that is referenced as the 'initial_context'.

7.7.2.3 Class Item

An Item is either a single object or a unit in a group of objects. It collects the information that is common to all versions of the object. An Item shall always be classified as 'part', 'tool', or 'raw material' using a Specific_item_classification. Additionally, if an Assembly_definition exists for at least one version of the Item, the Item shall be classified as being an 'assembly' using Specific_item_classification.

Base Class

- PLM_root_object (ABS)

Attributes

- id : String [1] The id specifies the identifier of the Item. For the id, an owner shall be specified by a Person_organization_assignment with role 'id owner'. The id shall be unique within the scope of the organization that is specified by the Person_organization_assignment with the role 'id owner'.

Compositions

- item_version : Item_version [1..*]
  The item_version specifies the Item_version that is associated with this Item.
- description : String_select [0..1]
  The description specifies additional information about the Item.
- name : String_select [1] The name specifies the word or group of words used to refer to the Item.
- alias_identification : Alias_identification [0..*]
  The Alias_identification specifies the Alias_identification that is applied to this Item.
- document_assignment : Document_assignment [0..*]
  The document_assignment specifies the object that provides information for this Item.

Associations

- none

7.7.2.4 Class Item_definition_relationship (ABS)

An Item_definition_relationship is a relationship between two Design_discipline_item_definition objects.

Base Class

- PLM_object (ABS)

Attributes

- none

Compositions

- document_assignment : Document_assignment [0..*]
  The document_assignment specifies the object that provides information for this Item_definition_relationship.
- simple_property_association : Simple_property_association (ABS) [0..*]
  The simple_property_association specifies the assigned simple property values.

Associations

- related : Design_discipline_item_definition [1]
  The related specifies the second of the Design_discipline_item_definition objects that are part of the relationship.

7.7.2.5 Class Item_version

An Item_version is a version of an Item and serves as the collector of the data characterizing a physically realizable object in various application contexts.

Base Class

- PLM_object (ABS)

Attributes

- id : String [1] The id specifies the identifier of the Item_version. The id shall be unique within the scope of the associated Item.

Compositions

- item_version_relationship : Item_version_relationship [0..*]
  The item_version_relationship specifies the item_version_relationship that relates the first of the two Item_version objects.
- description : String_select [0..1]The description specifies additional information about the Item_version.
- product_design : Product_design [0..1]
  The product_design specifies the Product_design for which the Item_version meets the requirements.
- design_discipline_item_definition : Design_discipline_item_definition [0..*]
  The design_discipline_item_definition specifies the Design_discipline_item_definition that is a view for this Item_version.
- alias_identification : Alias_identification [0..*]
  The alias_identification specifies the Alias_identification that is applied to this Item_version.
- document_assignment : Document_assignment [0..*]
  The document_assignment specifies the object that provides information for this Item_version.

Associations

- none

7.7.2.6 Class Item_version_relationship

An Item_version_relationship is a relationship between two Item_version objects.

Base Class

- PLM_object (ABS)

Attributes

- relation_type : String [1] The relation_type specifies the meaning of the relationship. Where applicable the following values shall be used:
  - 'derivation': The application object defines a deriving relationship where the related Item_version is based on the relating Item_version which is an earlier version of the same or of a different Item;
  - 'hierarchy': The application object defines a hierarchical relationship where the related Item_version is a subordinate version of the relating Item_version;
  - 'sequence': The application object defines a version sequence where the relating Item_version is the preceding version of the related Item_version that is the following version. For a given Item_version there shall be at most one Item_version_relationship of this relation_type referring to this Item_version as 'relating' and at most one Item_version_relationship of this relation_type referring as 'related';
  - 'supplied item': The application object defines a relationship between two Item_version objects representing the same object in different organizational contexts.

Compositions

- description : String_select [0..1] The description specifies additional information about the Item_version_relationship.
- change : Change [0..*] The change specifies the change for which this object references a modified object and the corresponding original object.

Associations

- related : Item_version [1] The related specifies the second of the two Item_version objects related by the Item_version_relationship.

7.7.3 Package Part_structure

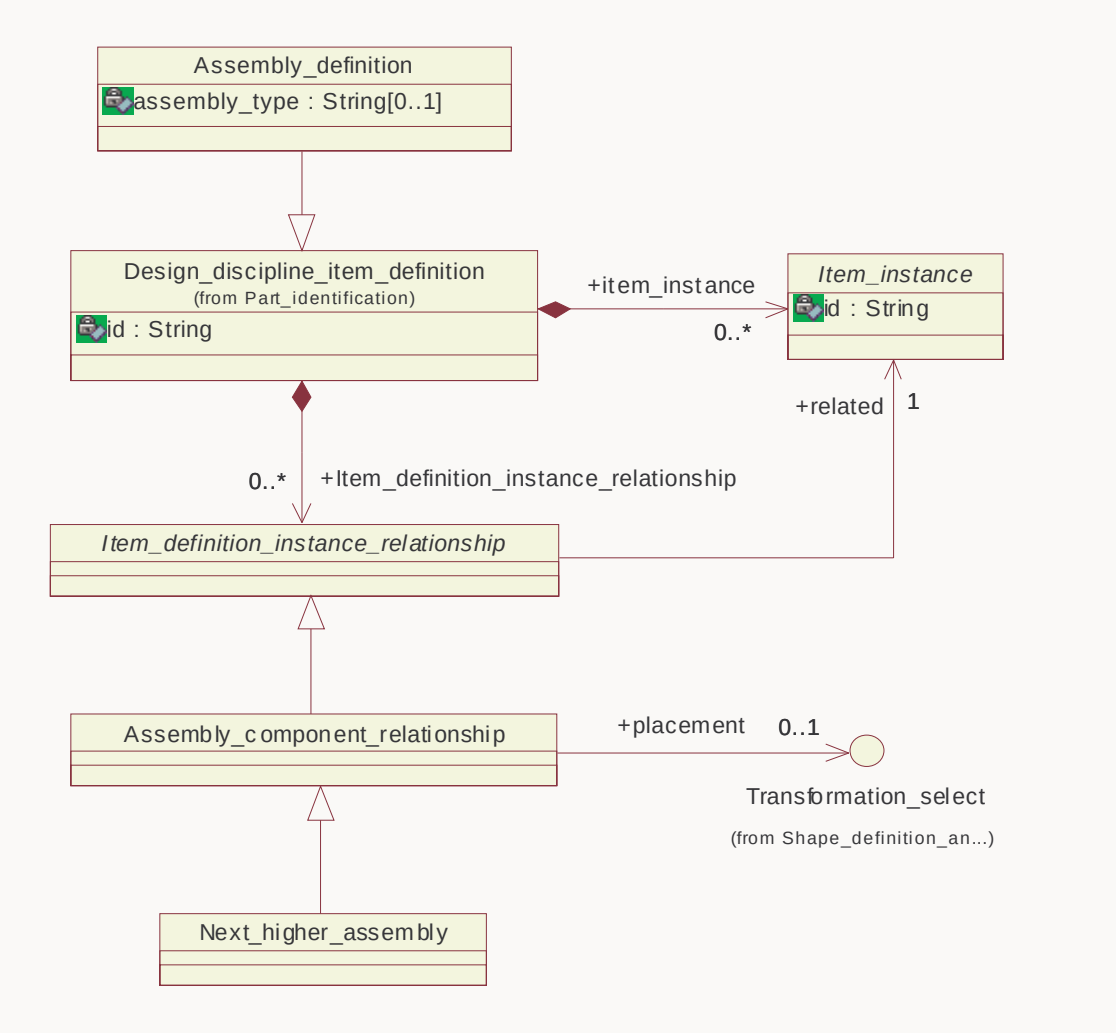


Figure 97 - Part structure - Assembly

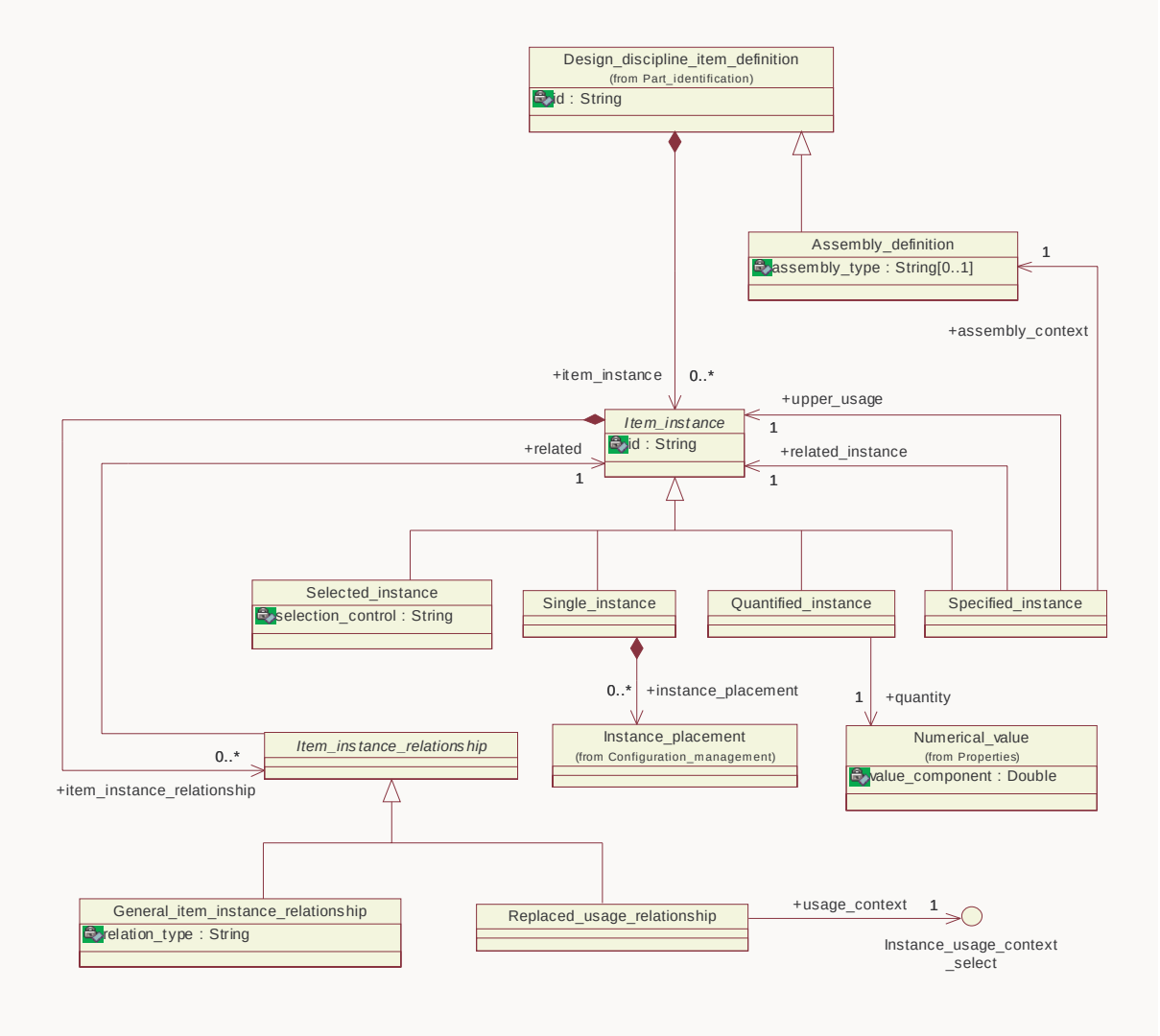


Figure 98 - Part structure - Item instance

7.7.3.1 Class Assembly_component_relationship

An Assembly_component_relationship is the relation between an Assembly_definition and an Item_instance representing a constituent of the assembly. The Assembly_definition and the Design_discipline_item_definition that serves as 'definition' of the Item_instance shall share at least one Application_context.

Base Class

- Item_definition_instance_relationship (ABS)

Attributes

- none

Compositions

- none

Associations

- placement : Transformation_select [0..1]
  The placement specifies the Geometric_model_relationship_with_transformation or the Template_instance that specify the transformation information which is used to locate and orient the constituent in the coordinate space of the Assembly_definition. In the case of a Template_instance, the scale factor shall be omitted or set to 1.0.

7.7.3.2 Class Assembly_definition

An Assembly_definition is a definition of an Item_version that contains other subordinate objects.

Base Class

- Design_discipline_item_definition

Attributes

- assembly_type : String [0..1] The assembly_type specifies the kind of the Assembly_definition.

Compositions

- none

Associations

- none

7.7.3.3 Class Collected_item_association

A Collected_item_association is a mechanism to associate Item_instance objects with a Collection_definition.

Base Class

- Item_definition_instance_relationship (ABS)

Attributes

- none

Compositions

- none

Associations

- none

7.7.3.4 Class Collection_definition

A Collection_definition is the definition of an Item_version that serves as a collector for Item_instance objects that are mounted in the same vehicle but may not be assembled together.

Base Class

- Design_discipline_item_definition

Attributes

- none

Compositions

- purpose : String_select [0..1] The purpose specifies the rationale behind the Collection_definition.

Associations

- none

7.7.3.5 Class General_item_definition_instance_relationship

A General_item_definition_instance_relationship is a relationship between a Design_discipline_item_definition and an Item_instance whose meaning is defined by the attribute 'relation_type'.

Base Class

- Item_definition_instance_relationship (ABS)

Attributes

- relation_type : String [1] The relation_type specifies the meaning of the relationship.

Compositions

- description : String_select [0..1] The description specifies additional information about the General_item_definition_instance_relationship.

Associations

- none

7.7.3.6 Class General_item_definition_relationship

A General_item_definition_relationship is a relationship between two Design_discipline_item_definition objects whose meaning is defined by the attribute 'relation_type'.

Base Class

- Item_definition_relationship (ABS)

Attributes

- relation_type : String [1] The relation_type specifies the meaning of the relationship.

Compositions

- description : String_select [0..1] The description specifies additional information about the General_item_definition_relationship.

Associations

- none

7.7.3.7 Class General_item_instance_relationship

A General_item_instance_relationship is a relationship between two Item_instance objects whose meaning is defined by the attribute 'relation_type'.

Base Class

- Item_instance_relationship (ABS)

Attributes

- relation_type : String [1] The relation_type specifies the meaning of the relationship.

Compositions

- description : String_select [0..1]The description specifies additional information about the General_item_instance_relationship.

Associations

- none

7.7.3.8 Class Item_definition_instance_relationship (ABS)

An Item_definition_instance_relationship is a relationship between a Design_discipline_item_definition and an Item_instance.

Base Class

- PLM_object (ABS)

Attributes

- none

Compositions

- document_assignment : Document_assignment [0..*]
  The document_assignment specifies the object that provides information for this Item_definition_instance_relationship.
- simple_property_association : Simple_property_association (ABS) [0..*]
  The simple_property_association specifies the assigned simple property values.

Associations

- related : Item_instance (ABS) [1]
  The related specifies the Item_instance that is part of the Item_definition_instance_relationship.

7.7.3.9 Class Item_instance (ABS)

An Item_instance is the occurrence of an object in a product structure that is defined either by a Design_discipline_item_definition or by a Product_identification.

Base Class

- PLM_object (ABS)

Attributes

- id : String [1] The id specifies the identifier of the Item_instance.

Compositions

- item_instance_relationship : Item_instance_relationship (ABS) [0..*]
  The item_instance_relationship specifies the item_instance_relationship that relates the first of the two Item_instance objects.
- description : String_select [0..1]
  The description specifies additional information about the Item_instance.
- manufacturing_configuration : Manufacturing_configuration (ABS) [0..*]
  The Manufacturing_configuration specifies the Manufacturing_configuration that controls this Item_instance.
- configuration : Configuration [0..*]
  The configuration specifies the configuration that controls this Item_instance for its valid usage.
- alias_identification : Alias_identification [0..*]
  The Alias_identification specifies the Alias_identification that is applied to this Item_instance.
- document_assignment : Document_assignment [0..*]
  The document_assignment specifies the object that provides information for this Item_instance.
- simple_property_association : Simple_property_association (ABS) [0..*]
  The simple_property_association specifies the assigned simple property values.

Associations

- none

7.7.3.10 Class Item_instance_relationship (ABS)

An Item_instance_relationship is a relationship between two Item_instance objects.

Base Class

- PLM_object (ABS)

Attributes

- none

Compositions

- document_assignment : Document_assignment [0..*]
  The document_assignment specifies the object that provides information for this Item_instance_relationship.
- simple_property_association : Simple_property_association (ABS) [0..*]
  The simple_property_association specifies the assigned simple property values.

Associations

- related : Item_instance (ABS) [1]
  The related specifies the second of the two objects related by the Item_instance_relationship.

7.7.3.11 Class Make_from_relationship

A Make_from_relationship is a relationship between a Design_discipline_item_definition which provides the definition of a raw material, or of a semi-finished item and a Design_discipline_item_definition which provides the definition of an object manufactured out of that material, or semi-finished item.

Base Class

- Item_definition_relationship (ABS)

Attributes

- none

Compositions

- description : String_select [0..1]The description specifies additional information about the Make_from_relationship.

Associations

- none

7.7.3.12 Class Next_higher_assembly

A Next_higher_assembly is a relationship where the attribute 'related' specifies a constituent of an assembly and the attribute 'relating' specifies the immediate parent assembly of the constituent.

Base Class

- Assembly_component_relationship

Attributes

- none

Compositions

- none

Associations

- none

7.7.3.13 Class Physical_assembly_relationship

A Physical_assembly_relationship is a mechanism to relate one Physical_instance as a component to another Physical_instance that plays the role of an assembly.

Base Class

- PLM_object (ABS)

Attributes

- none

Compositions

- document_assignment : Document_assignment [0..*]
  The document_assignment specifies the object that provides information for this Physical_assembly_relationship.

Associations

- physical_component : Physical_instance [1]
  The physical_component specifies the Physical_instance that serves as a component in the physical structure.
- is_realization_of : Item_instance (ABS) [1]
  The is_realization_of specifies the Item_instance the physical component is the realization of.

7.7.3.14 Class Quantified_instance

A Quantified_instance is the identification of the quantified occurrence of an object that is defined either as a Design_discipline_item_definition or as a Product_identification.

Base Class

- Item_instance (ABS)

Attributes

- none

Compositions

- none

Associations

- quantity : Numerical_value [1] The quantity specifies a Numerical_value specifying the quantity of occurrences.

7.7.3.15 Class Replaced_definition_relationship

A Replaced_definition_relationship is a relationship between two Design_discipline_item_definition objects where the relating Design_discipline_item_definition is replaced by the related Design_discipline_item_definition.

Base Class

- Item_definition_relationship (ABS)

Attributes

- none

Compositions

- change : Change [0..*] The change specifies the change for which this object references a modified object and the corresponding original object.
- description : String_select [0..1]The description specifies additional information about the Replaced_definition_relationship.

Associations

- none

7.7.3.16 Class Replaced_usage_relationship

A Replaced_usage_relationship is a relationship between two Item_instance objects where the relating Item_instance is replaced by the related Item_instance.

Base Class

- Item_instance_relationship (ABS)

Attributes

- none

Compositions

- description : String_select [0..1]The description specifies additional information about the Replaced_usage_relationship.

Associations

- usage_context : Instance_usage_context_select [1]
  The usage_context specifies the object that identifies the context in which the replacement is applicable. In the case, where the usage_context refers to a Process_operation_input_or_output, the 'relating' Item_instance shall be referred to as 'element' by the Process_operation_input_or_output. In the case, where the usage_context refers to an Item_definition_instance_relationship, the 'relating' Item_instance shall be referred to as 'related' by the Item_definition_instance_relationship. In the case, where the usage_context refers to a Product_structure_relationship, the 'relating' Item_instance shall be referred to as 'related' by the Product_structure_relationship.

7.7.3.17 Class Selected_instance

A Selected_instance is the identification of the occurrence of an object that is either defined as a Design_discipline_item_definition or as a Product_identification and whose quantity depends on certain constraints.

Base Class

- Item_instance (ABS)

Attributes

- selection_control : String [1] The selection_control specifies the constraint that has to be evaluated for the Selected_instance.

Compositions

- none

Associations

- selected_quantity : Value_with_unit (ABS) [1]
  The selected_quantity specifies the quantity of the part, tool or raw material foreseen as Selected_instance. The selected_quantity shall be of type Value_limit or Value_range.

7.7.3.18 Class Single_instance

A Single_instance is one particular occurrence of an object that is defined either as a Design_discipline_item_definition or as a Product_identification.

Base Class

- Item_instance (ABS)

Attributes

- none

Compositions

- instance_placement : Instance_placement [0..*]
  The instance_placement specifies the instance_placement which this Single_instance is placed with.

Associations

- none

7.7.3.19 Class Specified_instance

A Specified_instance is a mechanism to identify a certain Item_instance in a multi level assembly structure that reuses partial decompositions.

Base Class

- Item_instance (ABS)

Attributes

- none

Compositions

- none

Associations

- assembly_context : Assembly_definition [1]
  The assembly_context specifies an Assembly_definition object in which the instance identified by this mechanism is used.
- related_instance : Item_instance (ABS) [1]
  The related_instance specifies the Item_instance that is to be identified.
- upper_usage : Item_instance (ABS) [1]
  The upper_usage specifies the Item_instance in which the related_instance is used. This Item_instance shall be the immediate upper level instance or another Specified_instance.

7.7.3.20 Class Tool_part_relationship

A Tool_part_relationship is a relationship between two Design_discipline_item_definition objects. It establishes a relationship between an item (related) and a tool (relating) that is used to produce the item.

Base Class

- Item_definition_relationship (ABS)

Attributes

- none

Compositions

- used_technology_description : String_select [0..1]
  The used_technology_description specifies the technology that is used to manufacture the part using this tool and, possibly, the reasons for the use of a particular technology.

Associations

- placement : Transformation_target_select [0..1]
  The placement specifies the relative position of the Item representing the part with respect to the local coordinate system of the Item representing the tool.

7.7.3.21 Interfaces

Interface Instance_usage_context_select

This empty interface is realized by the following classes:

- Product_structure_relationship
- Item_definition_instance_relationship (ABS)
- Process_operation_input_or_output

Interface Item_information_select

This empty interface is realized by the following classes:

- Product_component
- Physical_instance
- Design_discipline_item_definition
- Item_instance (ABS)

Interface Product_constituent_select

This empty interface is realized by the following classes:

- Product_function
- Product_component
- Item_instance (ABS)

7.7.4 Package Document_and_file_management

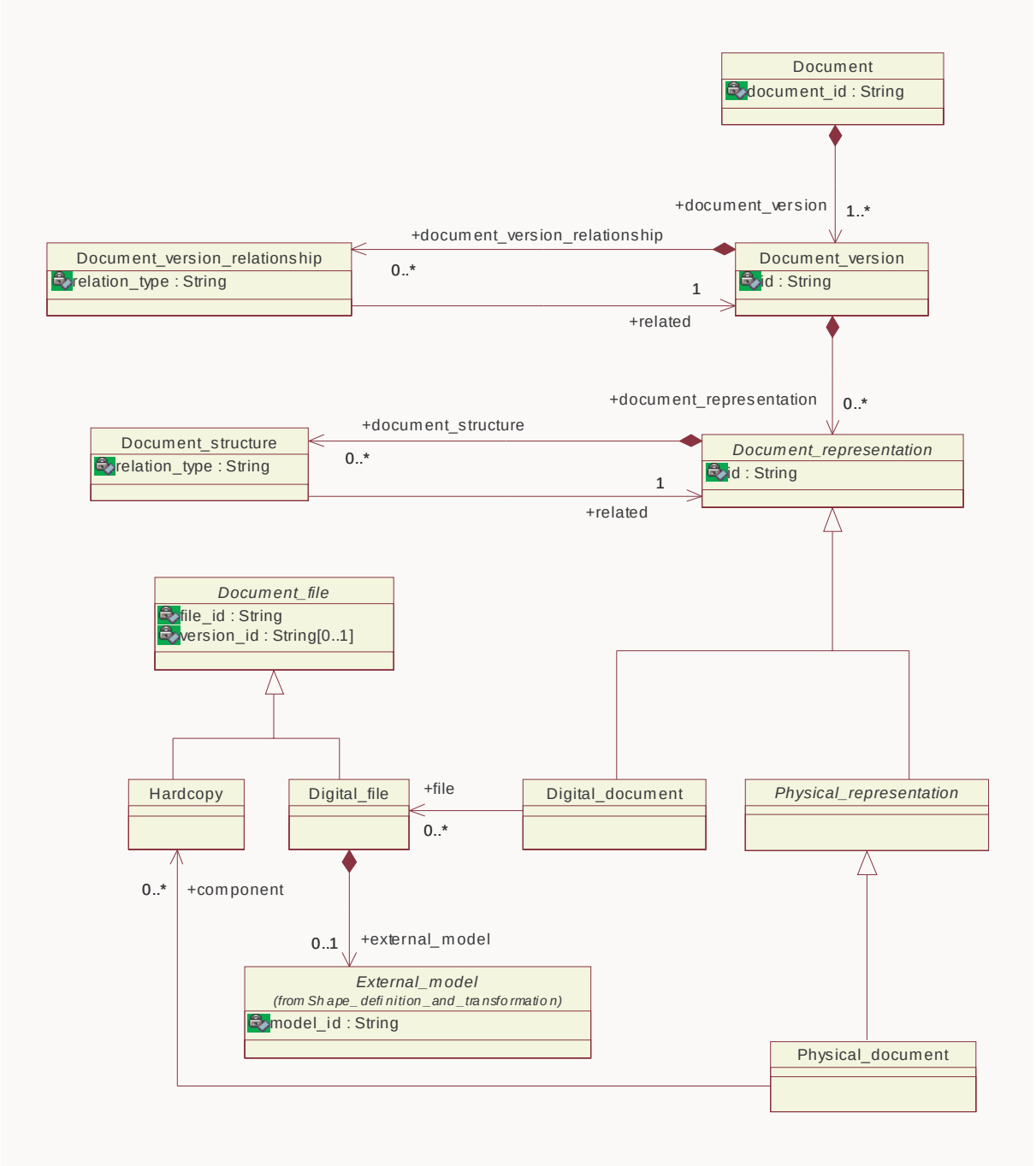


Figure 99 - Document and file management

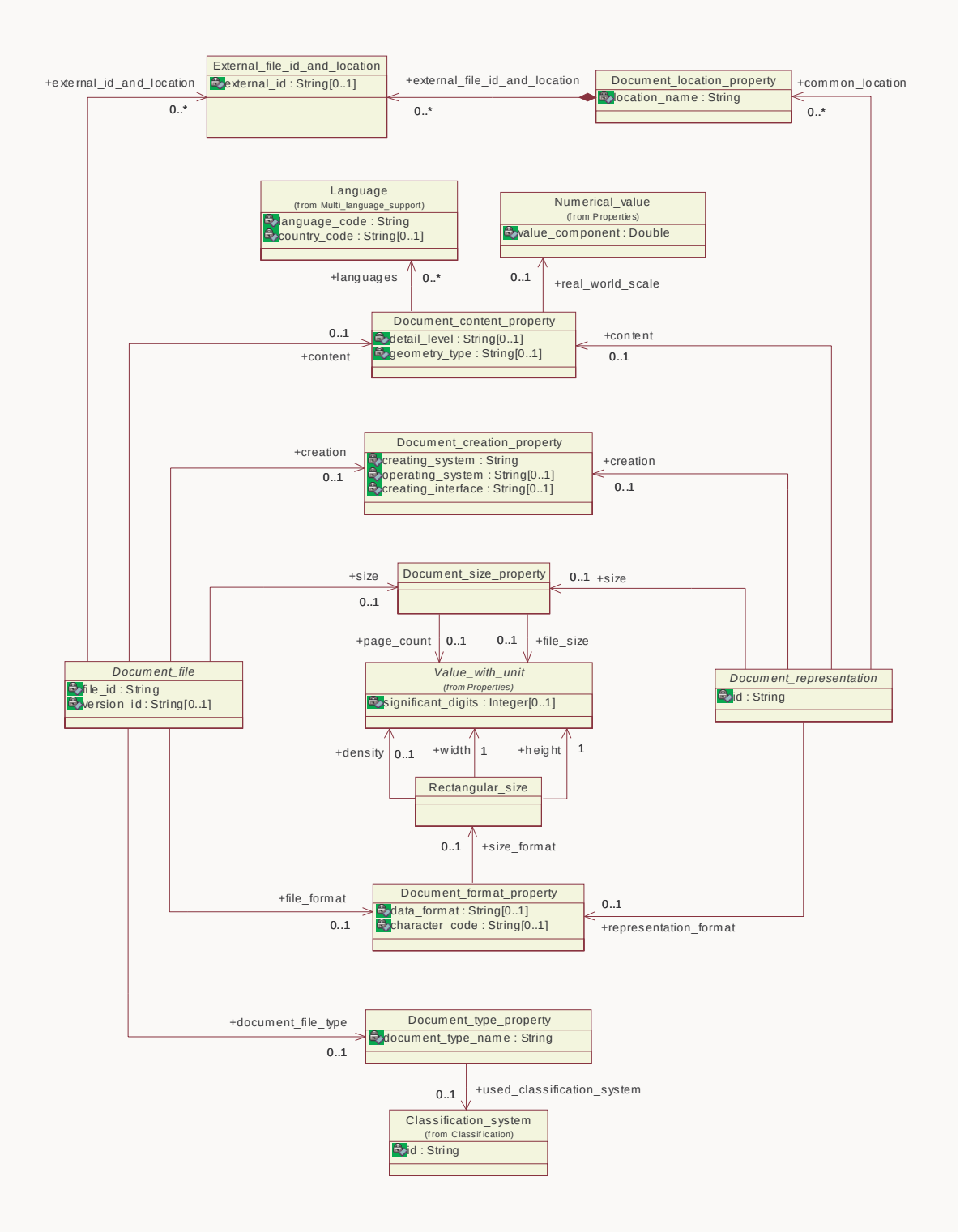


Figure 100 - Document properties

7.7.4.1 Class Digital_document

A Digital_document is a piece of product data that is archived in a digital format.

Base Class

- Document_representation (ABS)

Attributes

- none

Compositions

- none

Associations

- file : Digital_file [0..*] The file specifies a computer interpretable realization of the Digital_document.

7.7.4.2 Class Digital_file

A Digital_file contains computer interpretable data.

Base Class

- Document_file (ABS)

Attributes

- none

Compositions

- external_model : External_model (ABS) [0..1]
  The external_model specifies the externally defined geometry information contained in this Digital_file.

Associations

- none

7.7.4.3 Class Document

A Document is a logical document that serves as the identifier for a container for some product data.

Base Class

- PLM_root_object (ABS)

Attributes

- document_id : String [1] The document_id specifies the identifier of the Document.

Compositions

- document_version : Document_version [1..*]
  The document_version specifies the document_version of this logical document.
- name : String_select [1] The name specifies the word or group of words by which the Document is referred to.
- description : String_select [0..1]The description specifies additional information about the Document.
- alias_identification : Alias_identification [0..*]
  The Alias_identification specifies the Alias_identification that is applied to this Document.

Associations

- none

7.7.4.4 Class Document_assignment

A Document_assignment is a mechanism to associate a document with an object, where the assigned document provides information about the object it is associated to.

Base Class

- PLM_object (ABS)

Attributes

- role : String [1] The role specifies the meaning of the Document_assignment:
  - 'additional information': The assigned document provides information that is relevant for the associated object, but is not a description of the associated object itself;
  - 'behavior': The assigned document specifies information about the behavior of the associated object;
  - 'description': The assigned document provides textual information for the associated object it-self;
  - 'informative': The assigned document may or may not be considered;
  - 'mandatory': The associated object shall conform to the content of the assigned document;
  - 'mathematical description': The assigned document specifies the associated object by providing the algorithmic specification of its behavior.

Compositions

- none

Associations

- assigned_document : Assigned_document_select [1]
  The assigned_document specifies the Document, a Document_version, a Document_representation, or a Document_file that is used to provide information.

7.7.4.5 Class Document_content_property

A Document_content_property specifies characteristics precising the content of a Document_file or of a Document_representation. At least one of the optional attributes shall be specified for each instance of this object.

Base Class

- PLM_root_object (ABS)

Attributes

- detail_level : String [0..1] The detail_level specifies the level of detail that the Document_file or the Document_representation provides. Where applicable the following values shall be used:
  - 'rough 3d shape': 3D shape model without edge rounds and fillets;
  - 'rounded edges': 3D shape model with edge rounds and fillets.
- geometry_type : String [0..1] The geometry_type specifies the kind or kinds of geometry that an object contains. Where applicable the following values shall be used:
  - '3D wireframe model': The document contains a 3D shape model in wireframe representation;
  - '2D shape': The document contains a 2D shape model or contours only;
  - 'surface model': The document contains a 3D shape model in surface representation;
  - 'closed volume': The document contains a 3D shape model in closed body topological surface representation;
  - 'solid model': The document contains a 3D shape model in advanced boundary representation;
  - 'solid and surface model': The document contains a 3D shape model in surface and advanced boundary representation;
  - 'assembly': The document contains an assembly structure with reference to the assembled components and their transformation matrices;
  - 'assembly with mating elements': The document contains an assembly structure including the mating components only, such as screws or rivets, with exact positioning information. This assembly representation is intended to be overlayed with the assembly structure for the main components;
  - '2D drawing': The document contains a technical drawing without 3D shape representation;
  - 'drawing derived from 3D data': The document contains a technical drawing that has been derived from a 3D shape model;
  - 'drawing related to 3D data': The document contains a technical drawing that visualizes a 3D shape model and possibly establishes associative links to the 3D shape model.

Compositions

- none

Associations

- languages : Language [0..*] The languages specifies which language or languages are used in the characterized objects.
- real_world_scale : Numerical_value [0..1]The real_world_scale specifies the scale that is used in the Document_file or in the Document_representation the Document_content_property is referred by.

7.7.4.6 Class Document_creation_property

A Document_creation_property specifies characteristics of Document_file or of Document_representation objects. It specifies the context of the creation of the object. At least one of the optional attributes shall be specified for each instance of this object.

Base Class

- PLM_root_object (ABS)

Attributes

- creating_system : String [1] The creating_system specifies the computer application or the machine which is used to create the object that is characterized.
- operating_system : String [0..1] The operating_system specifies the operating system that is used to execute the computer application that created the characterized object.
- creating_interface : String [0..1] The creating_interface specifies the computer application used to create the Document_file or Document_representation object.

Compositions

- none

Associations

- none

7.7.4.7 Class Document_file (ABS)

A Document_file is one of potentially more files on a computer system or in actual stacks of paper that make up a Document_representation.

Base Class

- PLM_root_object (ABS)

Attributes

- file_id : String [1] The file_id specifies the identifier which is used to locate the file either on a computer system or in a repository of paper documents.
- version_id : String [0..1] The version_id specifies the identification of the version that distinguishes one Document_file object from other versions of Document_file objects with the same file_id.

Compositions

- simple_property_association : Simple_property_association (ABS) [0..*]
  The simple_property_association specifies the assigned simple property values.

Associations

- creation : Document_creation_property [0..1]
  The creation specifies further details of the context of the creation of the Document_file.
- content : Document_content_property [0..1]
  The content characterizes the content of the Document_file.
- file_format : Document_format_property [0..1]
  The file_format specifies the characteristics of the Document_file that specify the format of the object.

- size : Document_size_property [0..1]
  The size specifies characteristics for the size of the Document_file.
- external_id_and_location : External_file_id_and_location [0..*]
  The external_id_and_location specifies alternatives of the identifier and location of the Document_file.
- document_file_type : Document_type_property [0..1]
  The document_file_type specifies the format of the Document_file. It shall only be specified, if the Document_file does not participate in a Document.

7.7.4.8 Class Document_format_property

A Document_format_property specifies characteristics of a Document_file or of a Document_representation that specify the format of the object. At least one of the optional attributes shall be specified for each instance of this object.

Base Class

- PLM_root_object (ABS)

Attributes

- data_format : String [0..1]
  The data_format specifies the convention that was used to structure the information in the characterized object. Where applicable the following values shall be used:
  - 'DXF': The document contains data in Drawing Exchange File format;
  - 'IGES': The document contains data in Initial Graphics Exchange Specification format;
  - 'ISO 10303-203': The document contains data in ISO 10303-203 format;
  - 'ISO 10303-214': The document contains data in ISO 10303-214 format;
  - 'TIFF CCITT GR4': The document contains data in TIFF CCITT GR4 format;
  - 'VDAFS': The document contains data in VDAFS format;
  - 'VOXEL': The document contains data in VOXEL format.
- character_code : String [0..1]
  The character_code specifies the character code that is used in the characterized object. Where applicable the following values shall be used:
  - 'binary': The document contains data in binary format;
  - 'IEC 61286': The coded character set used to encode the document data according to IEC 61286;
  - 'ISO 646': The coded character set used to encode the document data according to ISO 646;
  - 'ISO 3098-1': The coded character set used to encode the document data is according to ISO 3098-1;
  - 'ISO 6937': The coded character set used to encode the document data is according to ISO/IEC 6937;
  - 'ISO 8859-1': The coded character set used to encode the document data according to ISO 8859-1;
  - 'ISO 10646': The coded character set used to encode the document data according to ISO/IEC 10646.

Compositions

- none

Associations

- size_format : Rectangular_size [0..1]
  The size_format specifies the dimensions of a physical presentation of the object the size_format is provided for.

7.7.4.9 Class Document_location_property

A Document_location_property specifies where a Document_file or a Document_representation can be found in a digital or physical data storage system.

Base Class

- PLM_root_object (ABS)

Attributes

- location_name : String [1] The location_name specifies the location, where the object that refers to the Document_location_property, can be found. 'C:\mpbs{}programs' and '/usr/local/bin' are examples for a location_name.

Compositions

- external_file_id_and_location : External_file_id_and_location [0..*]
  The external_file_id_and_location specifies the Document_file that is stored in this Document_location_property.

Associations

- none

7.7.4.10 Class Document_representation (ABS)

A Document_representation is one of potentially more alternative representations of a Document_version.

Base Class

- PLM_object (ABS)

Attributes

- id : String [1] The id specifies the identifier of the Document_representation.

Compositions

- document_structure : Document_structure [0..*]
  The document_structure specifies the document_structure that relates the first of the two Document_representation objects.
- description : String_select [0..1]The description specifies additional information about the Document_representation.
- alias_identification : Alias_identification [0..*]
  The Alias_identification specifies the Alias_identification that is applied to this Document_representation.

- simple_property_association : Simple_property_association (ABS) [0..*]
  The simple_property_association specifies the assigned simple property values.

Associations

- content : Document_content_property [0..1]
  The content specifies characteristics of the content of the Document_representation.
- size : Document_size_property [0..1]
  The size specifies the size of the represented document.
- representation_format : Document_format_property [0..1]
  The representation_format specifies the format of the document represented by Document_representation.
- common_location : Document_location_property [0..*]
  The common_location specifies the location of a Document_representation, where all its constituents can be found.
- creation : Document_creation_property [0..1]
  The creation specifies further details of the creation of the Document_representation.

7.7.4.11 Class Document_size_property

A Document_size_property specifies the size of a Document_file or of a Document_representation object. At least one of the optional attributes shall be specified for each instance of this object.

Base Class

- PLM_root_object (ABS)

Attributes

- none

Compositions

- none

Associations

- page_count : Value_with_unit (ABS) [0..1]
  The page_count specifies the number of pages of the application object the Document_size_property is referred by. The page_count shall only be used in cases where the Document_size_property is referred by a Hardcopy or a Physical_representation.
- file_size : Value_with_unit (ABS) [0..1]
  The file_size specifies the Value_with_unit that represents the size of a digitally stored document. The file_size shall only be applied in cases where the Document_size_property is referred by a Digital_document or a Document_file.

7.7.4.12 Class Document_structure

A Document_structure is a relationship between two Document_representation objects.

Base Class

- PLM_object (ABS)

Attributes

- relation_type : String [1] The relation_type specifies the meaning of the relationship. Where applicable the following values shall be used:
  - 'addition': The application object specifies that the related document provides supplementary or collateral information with regard to the information provided by the relating document;
  - 'copy': The application object defines a relationship where the related Document_representation is a copy of the relating Document_representation;
  - 'decomposition': The application object defines a relationship where the related Document_representation is one of potentially more sub documents of the relating Document_representation;
  - 'derivation': The application object defines a relationship where the related Document_representation is derived from the relating Document_representation;
  - 'peer': The application object specifies that the related document provides required information with regard to that provided by the relating document. The peer document is essential for a complete understanding;
  - 'reference': The application object defines a relationship where the related document is referenced from the relating;
  - 'sequence': The application object defines a logical sequence where the related Document_representation comes after the relating Document_representation;
  - 'substitution': The application object defines a relationship where the related Document_representation replaces the relating Document_representation;
  - 'translation': The Document_structure specifies that the related document is generated through a translation process from the relating document.

Compositions

- description : String_select [0..1]The description specifies additional information about the Document_structure.

Associations

- related : Document_representation (ABS) [1]
  The related specifies the second of the two objects related by the Document_structure.

7.7.4.13 Class Document_type_property

A Document_type_property specifies the kind of a Document_file.

Base Class

- PLM_root_object (ABS)

Attributes

- document_type_name : String [1]
  The document_type_name specifies the word or the group of words that describe the kind of object the characteristics are provided for. Where applicable the following values shall be used:
  - 'geometry': The document represents a shape model;

- 'NC data': The document represents numerical control data;
- 'FE data': The document represents finite element data;
- 'sample data': The document represents measured data;
- 'process plan': The document represents process planning data;
- 'check plan': The document represents quality control planning data;
- 'drawing': The document represents a technical drawing.

Compositions

- alias_identification : Alias_identification [0..*]
  The Alias_identification specifies the Alias_identification that is applied to this Document_type_property.

Associations

- used_classification_system : Classification_system [0..1]
  The used_classification_system specifies the Classification_system the document_type_name is defined in.

7.7.4.14 Class Document_version

A Document_version is a release of a Document.

Base Class

- PLM_object (ABS)

Attributes

- id : String [1] The id specifies the identifier of the Document_version. The id shall be unique within the scope of the associated Document.

Compositions

- document_version_relationship : Document_version_relationship [0..*]
  The document_version_relationship specifies the document_version_relationship that relates the first of the two Document_version objects.
- description : String_select [0..1]The description specifies additional information about the Document_version.
- document_representation : Document_representation (ABS) [0..*]
  The document_representation specifies the document_representation that represents this version of the logical document.
- alias_identification : Alias_identification [0..*]
  The Alias_identification specifies the Alias_identification that is applied to this Document_version.

Associations

- none

7.7.4.15 Class Document_version_relationship

A Document_version_relationship is a relationship between two Document_version objects.

Base Class

- PLM_object (ABS)

Attributes

- relation_type : String [1] The relation_type specifies the meaning of the relationship. Where applicable the following values shall be used:
  - 'derivation': The application object defines a deriving relationship where the related Document_version is based on the relating Document_version which is an earlier version of the same or of a different Document;
  - 'hierarchy': The application object defines a hierarchical relationship where the related Document_version is a sub version of the relating Document_version;
  - 'sequence': The application object defines a version sequence where the relating Document_version is the preceding version and the related Document_version is the following version.;
  - 'supplied document': The application object defines a relationship between two Document_version objects representing the same object in different organizational contexts.

Compositions

- description : String_select [0..1] The description specifies additional information about the Document_version_relationship.

Associations

- related : Document_version [1] The related specifies the second of the two objects related by the Document_version_relationship.

7.7.4.16 Class External_file_id_and_location

An External_file_id_and_location specifies the location of a file in an external storage system.

Base Class

- PLM_object (ABS)

Attributes

- external_id : String [0..1] The external_id specifies the identifier of a document in an external storage system.

Compositions

- none

Associations

- none

7.7.4.17 Class Hardcopy

A Hardcopy is the actual stack of paper consisting of one or more sheets, on which some product data is written, printed or plotted.

Base Class

- Document_file (ABS)

Attributes

- none

Compositions

-

Associations

- none

7.7.4.18 Class Named_size

A Named_size is the definition of the size of a Document_file or of a Document_representation where the size is specified by a standardized identifier.

Base Class

- Rectangular_size

Attributes

- size : String [1] — The size specifies the size of the object. If the size differs from the dimensions specified by the inherited 'width' and 'height' attributes the size is overridden.

Compositions

- none

Associations

- referenced_standard : Classification_system [0..1]
  The referenced_standard specifies a standard according to which the size is specified.

7.7.4.19 Class Physical_document

A Physical_document is a piece of product data that is archived in a non-digital form.

Base Class

- Physical_representation (ABS)

Attributes

- none

Compositions

- none

Associations

- component : Hardcopy [0..*] The component specifies the physical realization of the Physical_document.

7.7.4.20 Class Physical_representation (ABS)

A Physical_representation is a physically realizable representation of a Document_version.

Base Class

- Document_representation (ABS)

Attributes

- none

Compositions

- none

Associations

- none

7.7.4.21 Class Rectangular_size

A Rectangular_size is the definition of the planar size of an object.

Base Class

- PLM_root_object (ABS)

Attributes

- none

Compositions

- none

Associations

- density : Value_with_unit (ABS) [0..1]
  The density specifies the resolution of the object if it is a raster picture.
- height : Value_with_unit (ABS) [1]
  The height specifies the size of the object in vertical direction.
- width : Value_with_unit (ABS) [1]The width specifies the size of the object in horizontal direction.

7.7.4.22 Interfaces

Interface Assigned_document_select

This empty interface is realized by the following classes:

- Document_version

- Document_representation (ABS)
- Document_file (ABS)
- Document

7.7.5 Package Shape_definition_and_transformation

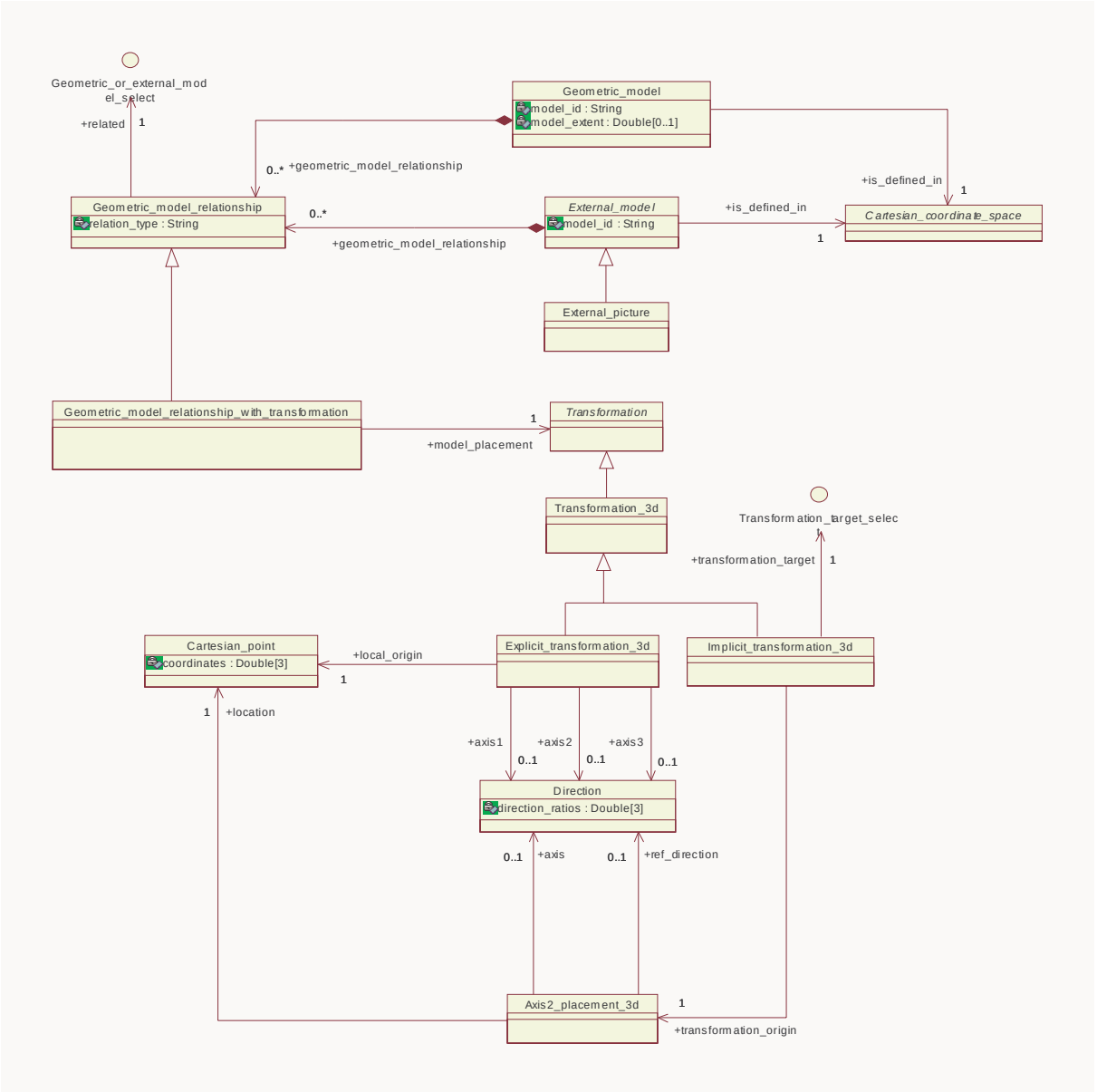


Figure 101 - Shape definition and transformation

7.7.5.1 Class Accuracy

An Accuracy is the information about the geometrical accuracy of the product data contained in a model.

Base Class

- PLM_root_object (ABS)

Attributes

- accuracy_value : Double [1] The accuracy_value specifies a numerical value defining the Accuracy.
- accuracy_type : String [1] The accuracy_type specifies the kind of accuracy that is applied. Where applicable the following values shall be used:
  - 'angular accuracy': A kind of accuracy that specifies the maximum value for the absolute angle between two curve tangents or two surface normals for which the creating system assumes curve tangents or surface normals being identical;
  - 'curvature accuracy': A kind of accuracy that specifies the value for the term under which a system can assume that the two radii of curvature R1 and R2 are identical. The curvature accuracy value is used to determine the accuracy range for curvature continuous curve or surface connections;
  - 'distance accuracy': A kind of accuracy that specifies the distance under which two points can be considered as having the same location. The distance accuracy value defined for a Geometric_model is valid for all geometric elements of the Geometric_model.

Compositions

- description : String_select [0..1]The description specifies additional information about the Accuracy.

Associations

- is_defined_for : Accuracy_select [1..*]
  The is_defined_for specifies the geometry to which the Accuracy is assigned.

7.7.5.2 Class Axis2_placement_3d

Axis2_placement_3d is a geometric_representation_item that specifies the location and orientation in three dimensional space of two mutually perpendicular axes.

Base Class

- PLM_root_object (ABS)

Attributes

- none

Compositions

- none

Associations

- ref_direction : Direction [0..1] The ref_direction can be used to determine the direction of the local X axis.
- axis : Direction [0..1] The axis defines the exact direction of the local Z axis.

- location : Cartesian_point [1] The location defines the spatial position of the reference point and origin of the associated placement coordinate system.

7.7.5.3 Class Cartesian_coordinate_space (ABS)

Cartesian_coordinate_space is a coordinate space in which geometric and annotation elements may be defined. It is either two-dimensional or three-dimensional. An origin for coordinate values is implicitly defined. The units applicable to the coordinate values of elements defined in the Cartesian_coordinate_space are specified.

Base Class

- PLM_root_object (ABS)

Attributes

- none

Compositions

- none

Associations

- unit_of_values : Unit [0..*] The unit_of_values specifies the various units in which any values are expressed. The same length unit is applied to each coordinate direction. Only one unit of a kind shall be specified. In the case where geometric elements are defined in the Cartesian_coordinate_space, there shall be at least two units specified, the length unit and the plane angle unit.

7.7.5.4 Class Cartesian_coordinate_space_2d

A Cartesian_coordinate_space_2d is a two-dimensional coordinate space. Any two-dimensional geometric and annotation element shall be defined in a Cartesian_coordinate_space_2d.

Base Class

- Cartesian_coordinate_space (ABS)

Attributes

- none

Compositions

- none

Associations

- none

7.7.5.5 Class Cartesian_coordinate_space_3d

A Cartesian_coordinate_space_3d is a three-dimensional coordinate space. Any three-dimensional geometric data shall be defined in a Cartesian_coordinate_space_3d.

Base Class

- Cartesian_coordinate_space (ABS)

Attributes

- none

Compositions

- none

Associations

- none

7.7.5.6 Class Cartesian_point

A Cartesian_point is a point that is defined by its coordinates in a rectangular Cartesian coordinate system.

Base Class

- PLM_root_object (ABS)

Attributes

- coordinates : Double [3] The coordinates specify the 3 coordinates of the point.

Compositions

- none

Associations

- none

7.7.5.7 Class Direction

A Direction in a 3-dimensional space is expressed as a vector.

Base Class

- PLM_root_object (ABS)

Attributes

- direction_ratios : Double [3] The direction_ratios specify the 3 ratios of the direction vector components.

Compositions

- none

Associations

- none

7.7.5.8 Class Explicit_transformation_3d

A geometric relationship between external models can be defined explicitly by using an Explicit_transformation_3d that has a local origin and a rotation matrix.

Base Class

- Transformation_3d

Attributes

- none

Compositions

- none

Associations

- axis3 : Direction [0..1] The axis3 is the Z axis direction of the transformation target.
- axis2 : Direction [0..1] The axis2 is the Y axis direction of the transformation target.
- axis1 : Direction [0..1] The axis1 is the X axis direction of the transformation target.
- local_origin : Cartesian_point [1]
  The local_origin is the required translation specified as a cartesian point. The actual translation included in the transformation is from the geometric origin to the local origin.

7.7.5.9 Class External_geometric_model

An External_geometric_model is the identification of a model that contains geometry in a 3D context only.

Base Class

- External_model (ABS)

Attributes

- model_extent : String [0..1] The model_extent specifies the radius of a sphere that contains all elements of the model and whose centre is at the origin of the Cartesian_coordinate_space of the External_geometric_model. The model_extent is specified using a length unit.

Compositions

- none

Associations

- none

7.7.5.10 Class External_model (ABS)

An External_model is the identification of a model that is described in a Digital_file and by the Cartesian_coordinate_space that is needed to further process the externally described information.

Base Class

- PLM_object (ABS)

Attributes

- model_id : String [1] The model_id specifies the identifier of the External_model.

Compositions

- geometric_model_relationship : Geometric_model_relationship [0..*]
  The geometric_model_relationship specifies the geometric_model_relationship that relates the first of the two External_model objects.
- description : String_select [0..1]The description specifies additional information about the External_model.

Associations

- is_defined_in : Cartesian_coordinate_space (ABS) [1]
  The is_defined_in specifies the Cartesian_coordinate_space that defines the context for the externally described geometry. If the External_model is an External_picture, the context shall be a Cartesian_coordinate_space_2d.

7.7.5.11 Class External_picture

An External_picture is the identification of a model that is described by a two dimensional image.

Base Class

- External_model (ABS)

Attributes

- none

Compositions

- none

Associations

- none

7.7.5.12 Class Geometric_model

A Geometric_model is a representation of geometry. A Geometric_model that does not reference any Detailed_geometric_model_element objects through one of the subtypes directly shall either reference at least one Template_instance as 'additional_element' or shall reference Axis_placement objects exclusively.

Base Class

- PLM_root_object (ABS)

Attributes

- model_id : String [1] The model_id specifies the identifier of the Geometric_model.

- model_extent : Double [0..1] The model_extent specifies the radius of a sphere that contains all elements of the model and whose centre is at the origin of the Cartesian_coordinate_space of the Geometric_model. The model_extent is specified using a length unit.

Compositions

- description : String_select [0..1]
  The description specifies additional information about the Geometric_model.
- geometric_model_relationship : Geometric_model_relationship [0..*]
  The geometric_model_relationship specifies the geometric_model_relationship that relates the first of the two Geometric_model objects.

Associations

- is_defined_in : Cartesian_coordinate_space (ABS) [1]
  The is_defined_in specifies the Cartesian_coordinate_space in which the Geometric_model is defined. The specified Cartesian_coordinate_space serves also as the reference coordinate space for the transformation of Template_instance objects used as additional elements in the Geometric_model.

7.7.5.13 Class Geometric_model_relationship

A Geometric_model_relationship is a relationship between two models. The models may be either of type Geometric_model or of type External_model.

Base Class

- PLM_object (ABS)

Attributes

- relation_type : String [1] The relation_type specifies the meaning of the relationship.

Compositions

- description : String_select [0..1]
  The description specifies additional information about the Geometric_model_relationship.

Associations

- related : Geometric_or_external_model_select [1]
  The related specifies the second of the two model objects related by the Geometric_model_relationship.

7.7.5.14 Class Geometric_model_relationship_with_transformation

A Geometric_model_relationship_with_transformation is a relationship between two model objects with the additional information about a geometric Transformation. This Transformation defines the location and orientation of the related model relative to the relating model.

Base Class

- Geometric_model_relationship

Attributes

- none

Compositions

- none

Associations

- model_placement : Transformation (ABS) [1]
  The model_placement specifies the geometric Transformation that places and orients the related model relative to the relating model.

7.7.5.15 Class Geometrical_relationship

A Geometrical_relationship is the relationship between two Design_discipline_item_definition objects specifying two parts that are geometrically related.

Base Class

- Item_definition_relationship (ABS)

Attributes

- none

Compositions

- description : String_select [0..1]The description specifies additional information about the Geometrical_relationship.

Associations

- definition_placement : Transformation_select [1]
  The definition_placement specifies the Geometric_model_relationship_with_transformation or the Template_instance that has the Transformation to be applied to the relating Design_discipline_item_definition in order to define the location and the orientation of the related Design_discipline_item_definition. Translation, rotation, and mirroring, i.e., inversion, is included; scaling is not included. In the case of a Template_instance, the scale factor shall be omitted or set to 1.0.

7.7.5.16 Class Implicit_transformation_3d

A geometric relationship between external models can be defined implicitly by using an Implicit_transformation_3d that has two reference points to specify origin and target of the transformation.

Base Class

- Transformation_3d

Attributes

- none

Compositions

- none

Associations

- transformation_origin : Axis2_placement_3d [1]
  The transformation_origin specifies the origin of the transformation.
- transformation_target : Transformation_target_select [1]
  The transformation_target specifies the target of the transformation.

7.7.5.17 Class Item_shape

An Item_shape is the definition of the shape of a Design_discipline_item_definition, an Item_instance or of a Physical_instance.

Base Class

- PLM_root_object (ABS)

Attributes

- none

Compositions

- shape_element : Shape_element [0..*]
  The shape_element specifies the shape_element that is part of this Item_shape.
- shape_description_association : Shape_description_association [0..*]
  The shape_description_association specifies the shape_description_association that is associated with this Item_shape.
- document_assignment : Document_assignment [0..*]
  The document_assignment specifies the object that provides information for this Item_shape.
- description : String_select [0..1]The description specifies additional information about the Item_shape.
- simple_property_association : Simple_property_association (ABS) [0..*]
  The simple_property_association specifies the assigned simple property values.

Associations

- described_object : Item_information_select [1]
  The described_object specifies the object whose shape the Item_shape defines.

7.7.5.18 Class Material

A Material is the substance out of which an item is or can be made.

Base Class

- PLM_root_object (ABS)

Attributes

- material_name : String [1] The material_name specifies the word or group of words by which the Material is referred to.

Compositions

- document_assignment : Document_assignment [0..*]
  The document_assignment specifies the object that provides information for this Material.
- material_property_association : Material_property_association [0..*]
  The material_property_association specifies the material_property_association in which a property value is assigned to this Material.

Associations

- described_element : Item_property_select [1..*]
  The described_element specifies the objects the material information is provided for.

7.7.5.19 Class Shape_description_association

A Shape_description_association is a mechanism to associate the definition of a shape or of a portion of a shape with a geometric representation.

Base Class

- PLM_object (ABS)

Attributes

- role : String [1] The role specifies the function performed by the referenced model. Where applicable the following values shall be used:
  - 'detailed representation': The geometry in the referenced model provides a detailed representation of the shape;
  - 'idealized representation': The geometry in the referenced model provides a simplified representation of the shape, e.g., for analysis purposes.

Compositions

- none

Associations

- defining_geometry : Shape_definition_select [1]
  The defining_geometry specifies the Geometric_model or the External_model that contains the shape information.

7.7.5.20 Class Shape_element

A Shape_element is a portion of shape that has to be identified explicitly to be associated with other information.

Base Class

- PLM_object (ABS)

Attributes

- element_name : String [0..1] The element_name specifies the word or group of words by which the Shape_element is referred to.

Compositions

- change : Change [0..*] The change specifies the change for which this object references a modified object and the corresponding original object.
- document_assignment : Document_assignment [0..*]
  The document_assignment specifies the object that provides information for this Shape_element.
- shape_description_association : Shape_description_association [0..*]
  The shape_description_association specifies the shape_description_association that is associated with this Shape_element.
- shape_element_relationship : Shape_element_relationship [0..*]
  The shape_element_relationship specifies the shape_element_relationship that relates the first of the two Shape_element objects.
- description : String_select [0..1]The description specifies additional information about the Shape_element.
- simple_property_association : Simple_property_association (ABS) [0..*]
  The simple_property_association specifies the assigned simple property values.

Associations

- none

7.7.5.21 Class Shape_element_relationship

A Shape_element_relationship is a relationship between two Shape_element objects.

Base Class

- PLM_object (ABS)

Attributes

- relation_type : String [1] The relation_type specifies the meaning of the relationship.

Compositions

- document_assignment : Document_assignment [0..*]
  The document_assignment specifies the object that provides information for this Shape_element_relationship.
- shape_description_association : Shape_description_association [0..*]
  The shape_description_association specifies the shape_description_association that is associated with this Shape_element_relationship.
- description : String_select [0..1]The description specifies additional information about the Shape_element_relationship.
- simple_property_association : Simple_property_association (ABS) [0..*]
  The simple_property_association specifies the assigned simple property values.

Associations

- related : Shape_element [1] The related specifies the second of the two Shape_element objects related by a Shape_element_relationship.

7.7.5.22 Class Transformation (ABS)

A Transformation is a geometric transformation composed of translation and rotation. Scaling is not included.

Base Class

- PLM_root_object (ABS)

Attributes

- none

Compositions

- none

Associations

- none

7.7.5.23 Class Transformation_3d

A Transformation_3d is the definition of a geometric transformation in 3D space.

Base Class

- Transformation (ABS)

Attributes

- none

Compositions

- none

Associations

- none

7.7.5.24 Interfaces

Interface Accuracy_select

This empty interface is realized by the following classes:

- Geometric_model
- External_geometric_model

Interface Geometric_or_external_model_select

This empty interface is realized by the following classes:

- Geometric_model
- External_model (ABS)

Interface Shape_definition_select

This empty interface is realized by the following classes:

- Geometric_model
- External_geometric_model

Interface Transformation_select

This empty interface is realized by the following class:

- Geometric_model_relationship_with_transformation

Interface Transformation_target_select

This empty interface is realized by the following classes:

- Explicit_transformation_3d
- Axis2_placement_3d

7.7.6 Package Classification

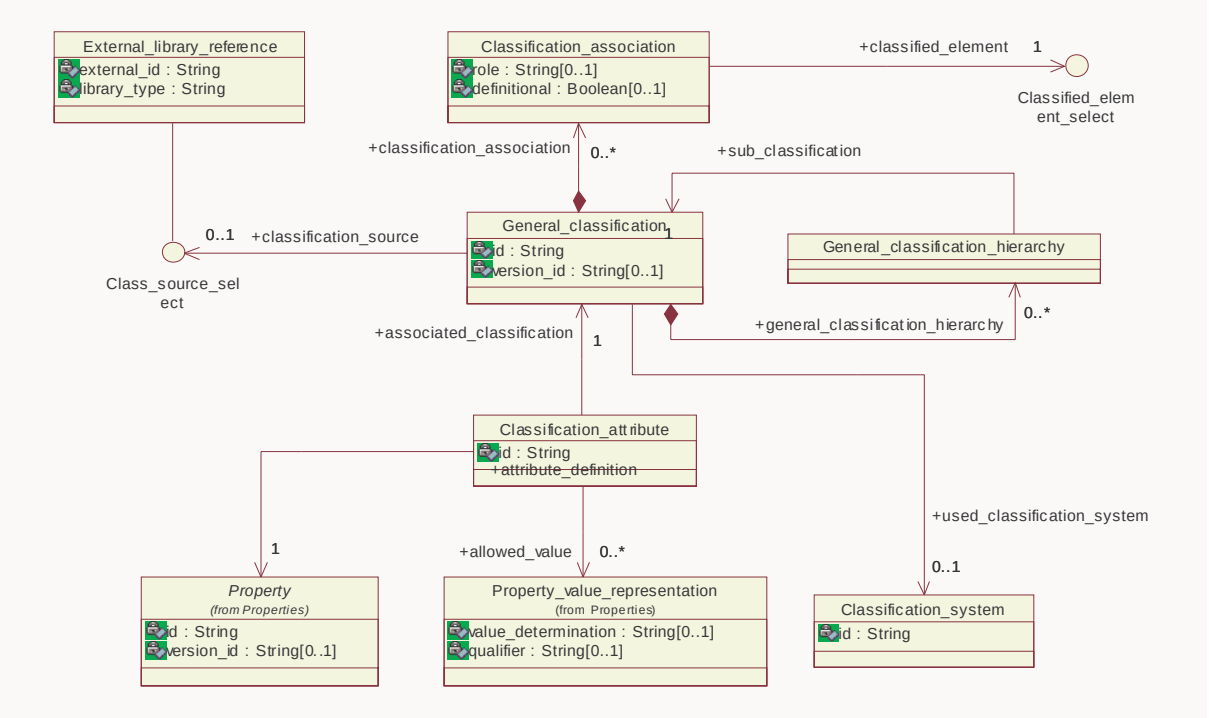


Figure 102 - Classification - General

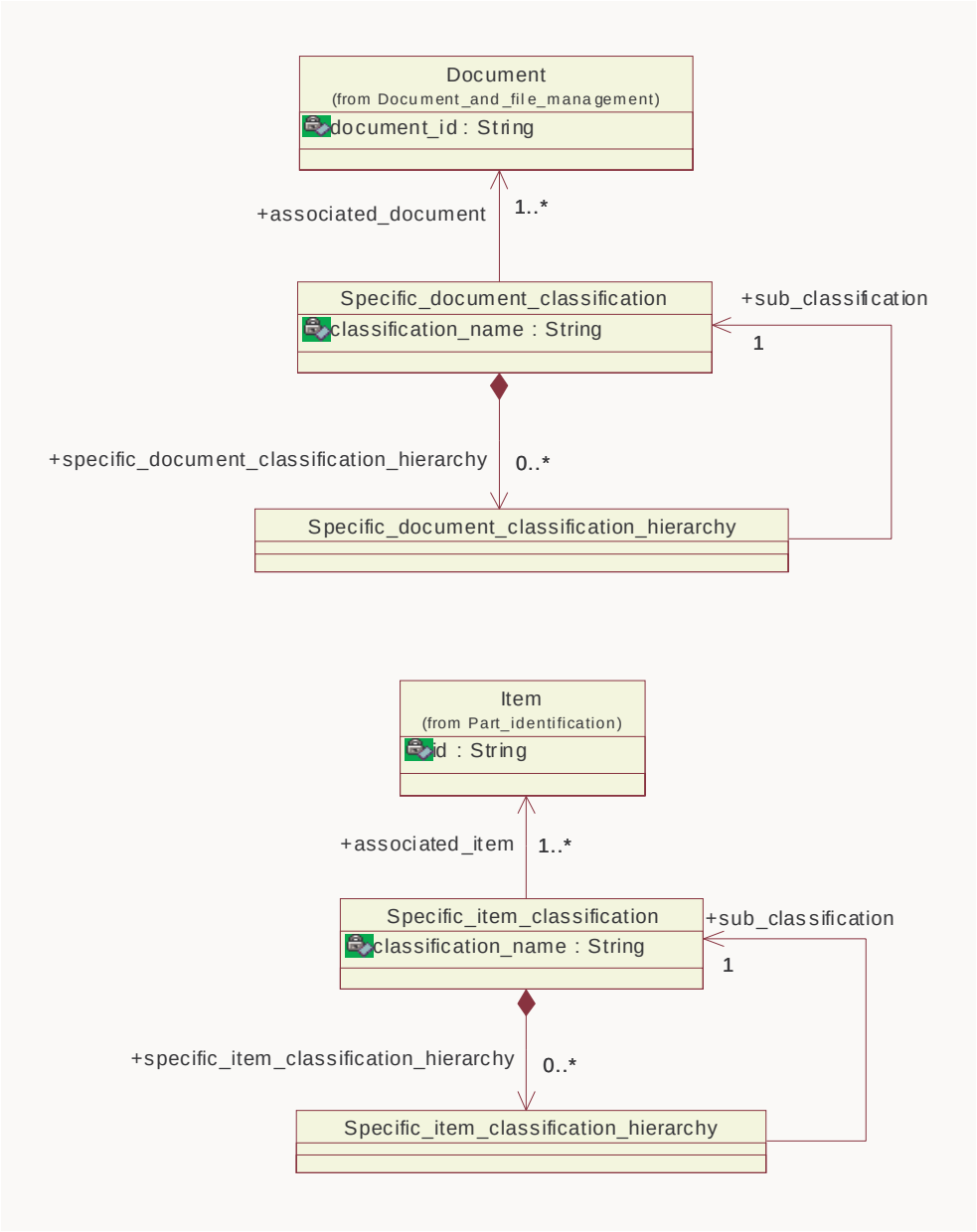


Figure 103 - Classification - Item and document

7.7.6.1 Class Classification_association

A Classification_association associates a General_classification with an object.

Base Class

- PLM_object (ABS)

Attributes

- role : String [0..1] The role specifies the relationship between the General_classification and the associated object. Where applicable the following values shall be used:
  - 'electromagnetic compatibility': The associated object is the classification that categorizes the classified element in respect of its ability to comply with requirements concerning electromagnetic interference;
  - 'environmental conditions': The associated object is the classification that categorizes the classified element with respect to its ability to comply with environmental impact requirements.
- definitional : Boolean [0..1] The definitional specifies whether a General_classification serves as definition. A value of 'true' indicates that the General_classification is definitional. The 'associated_classification' does not take precedence over the descriptions of the 'classified_element' made using Property_value or Geometric_model objects.

Compositions

- none

Associations

- classified_element : Classified_element_select [1]
  The classified_element specifies the object that is classified.

7.7.6.2 Class Classification_attribute

A Classification_attribute is a characteristic used to classify an object associated with the corresponding General_classification. The definition attribute of each 'allowed_value' shall refer to the property identified within 'attribute_definition'.

Base Class

- PLM_root_object (ABS)

Attributes

- id : String [1] The id specifies the identifier of the Classification_attribute that shall be unique within the scope of the associated General_classification.

Compositions

- alias_identification : Alias_identification [0..*]
  The Alias_identification specifies the Alias_identification that is applied to this Classification_attribute.
- document_assignment : Document_assignment [0..*]
  The document_assignment specifies the object that provides information for this Classification_attribute.
- description : String_select [0..1]The description specifies additional information about the Classification_attribute.
- name : String_select [0..1] The name specifies the word or group of words by which the Classification_attribute is referred to.

Associations

- attribute_definition : Property (ABS) [1]
  The attribute_definition specifies the Property that characterizes the allowed values.
- allowed_value : Property_value_representation [0..*]
  The allowed_value specifies the set of Property_value_representation objects that represent characteristic values of the Classification_attribute.
- associated_classification : General_classification [1]
  The associated_classification specifies the General_classification the Classification_attribute is a characteristic of.

7.7.6.3 Class Classification_system

A Classification_system is the scheme used to define the categorization of an item.

Base Class

- PLM_root_object (ABS)

Attributes

- id : String [1] The id specifies the identifier of the Classification_system.

Compositions

- description : String_select [0..1]The description specifies additional information about the Classification_system.
- document_assignment : Document_assignment [0..*]
  The document_assignment specifies the object that provides information for this Classification_system.
- alias_identification : Alias_identification [0..*]
  The Alias_identification specifies the Alias_identification that is applied to this Classification_system.

Associations

- none

7.7.6.4 Class External_library_reference

An External_library_reference is a mechanism to refer to an entry in an external library other than ISO 13584.

Base Class

- PLM_root_object (ABS)

Attributes

- external_id : String [1] The external_id specifies the unique identifier of the referenced entry in the external library.
- library_type : String [1] The library_type specifies the type of library that is used.

Compositions

- description : String_select [0..1]The description specifies additional information about the External_library_reference.

Associations

- none

7.7.6.5 Class General_classification

A General_classification is a classification of an object which characterizes all objects of the same kind; such a classification is independent from the application of the classified object.

Base Class

- PLM_root_object (ABS)

Attributes

- id : String [1] The id specifies the identifier of the General_classification.
- version_id : String [0..1] The version_id specifies the identification of a particular version of the General_classification.

Compositions

- description : String_select [0..1]The description specifies additional information about the General_classification.
- general_classification_hierarchy : General_classification_hierarchy [0..*]
  The General_classification_hierarchy specifies the General_classification_hierarchy for which this General_classification is the higher level, and that includes the sub class.
- classification_association : Classification_association [0..*]
  The Classification_association specifies the Classification_association for which this General_classification object provides classification information.
- alias_identification : Alias_identification [0..*]
  The Alias_identification specifies the Alias_identification that is applied to this General_classification.
- document_assignment : Document_assignment [0..*]
  The document_assignment specifies the object that provides information for this General_classification.

Associations

- classification_source : Class_source_select [0..1]
  The classification_source specifies the External_library_reference or the Plib_class_reference that contains the specification of the General_classification.
- used_classification_system : Classification_system [0..1]
  The used_classification_system specifies the Classification_system that contains the information about the definition of the classification and how to interpret the name of the General_classification.

7.7.6.6 Class General_classification_hierarchy

A General_classification_hierarchy defines a hierarchical relationship between two instances of General_classification.

Base Class

- PLM_object (ABS)

Attributes

- none

Compositions

- none

Associations

- sub_classification : General_classification [1]
  The sub_classification specifies the lower level of General_classification in a General_classification_hierarchy that is included in the super class.

7.7.6.7 Class Specific_document_classification

A Specific_document_classification is a classification of a Document with respect to specific criteria. The specific criteria are covered in the 'classification_name' attribute.

Base Class

- PLM_root_object (ABS)

Attributes

- classification_name : String [1]The classification_name provides classification information. Where applicable the following values shall be used:
  - 'catalogue': The assigned document is the catalogue in which the associated object is listed;
  - 'manual': The assigned document is the handbook that is supplied for the associated object;
  - 'specification': The assigned document specifies the considerations that lead to the design finally chosen for the associated object.

Compositions

- specific_document_classification_hierarchy : Specific_document_classification_hierarchy [0..*]
  The Specific_document_classification_hierarchy specifies the Specific_document_classification_hierarchy for which this Specific_document_classification is the higher level, and that is included in the sub class.
- description : String_select [0..1]The description specifies additional information about the Specific_document_classification.

Associations

- associated_document : Document [1..*]
  The associated_document the associated_document specifies the Document with which a particular Specific_document_classification is associated.

7.7.6.8 Class Specific_document_classification_hierarchy

A Specific_document_classification_hierarchy is used to build up hierarchical structures of Specific_document_classification_hierarchy objects.

Base Class

- PLM_object (ABS)

Attributes

- none

Compositions

- none

Associations

- sub_classification : Specific_document_classification [1]
  The sub_classification specifies the lower level of Specific_document_classification in Specific_document_classification_hierarchy that is included in the super class.

7.7.6.9 Class Specific_item_classification

A Specific_item_classification is a classification of an Item with respect to specific criteria. The specific criteria are covered in the 'classification_name' attribute.

Base Class

- PLM_root_object (ABS)

Attributes

- classification_name : String [1]The classification_name provides high level classification information. Where applicable the following values shall be used:
  - 'application control': This type of classification is used to indicate that an Item shall be considered under certification aspects; these aspects may be specified further by the - - 'description' attribute;
  - 'assembly': This type of classification shall be used for any Item that has an Assembly_definition provided for at least one of its versions, i.e., it is decomposed further;
  - 'collection': This type of classification shall be used for any Item that has a Collection_definition provided for at least one of its versions;
  - 'completely knocked down': This type of classification is used to indicate that an Item is used in a production site that has assembling facilities only;
  - 'detail': This type of classification shall be used for any Item that has no Assembly_definition provided for any of its versions, i.e., it is not further decomposed;

- 'in process': This type of classification is used to indicate that the Item identifies an intermediate object in a manufacturing process;
- 'part': The Item plays the role of a part;
- 'prototype': This type of classification is used to indicate that the Item identifies a prototype and is not intended for serial production;
- 'raw material': The Item plays the role of raw material;
- 'regulated': This type of classification is used to indicate that for an Item certain regulations have to be considered;
- 'safety': This type of classification is used to indicate that an Item is relevant for safety purposes;
- 'service': This type of classification is used to indicate that an Item is relevant for service purposes;
- 'tool': The Item plays the role of a tool.

Compositions

- specific_item_classification_hierarchy : Specific_item_classification_hierarchy [0..*]
  The Specific_item_classification_hierarchy specifies the Specific_item_classification_hierarchy for which this Specific_item_classification is the higher level, and that includes the sub class.
- description : String_select [0..1]The description specifies additional information about the Specific_item_classification.
- document_assignment : Document_assignment [0..*]
  The document_assignment specifies the object that provides information for this Specific_item_classification.

Associations

- associated_item : Item [1..*] The associated_item specifies the Item with which a particular Specific_item_classification is associated.

7.7.6.10 Class Specific_item_classification_hierarchy

A Specific_item_classification_hierarchy is used to build up hierarchical structures of Specific_item_classification.

Base Class

- PLM_object (ABS)

Attributes

- none

Compositions

- none

Associations

- sub_classification : Specific_item_classification [1]
  The sub_classification specifies the lower level of Specific_item_classification in a Specific_item_classification_hierarchy that is included in the super class.

7.7.6.11 Interfaces

Interface Class_source_select

This empty interface is realized by the following class:

- External_library_reference

Interface Classified_element_select

This empty interface is realized by the following classes:

- Approval_status
- Work_request
- Work_order
- Project
- Activity_method
- Activity
- Specification_category
- Product_identification
- Product_class
- Design_constraint
- Complex_product (ABS)
- Document_version
- Document_representation (ABS)
- Document_file (ABS)
- Document
- Item_version
- Item
- Design_discipline_item_definition
- Item_instance (ABS)
- Process_plan
- Process_operation_occurrence
- Process_operation_definition
- Property_value_association (ABS)
- Property (ABS)
- Shape_element
- Material

7.7.7 Package Properties

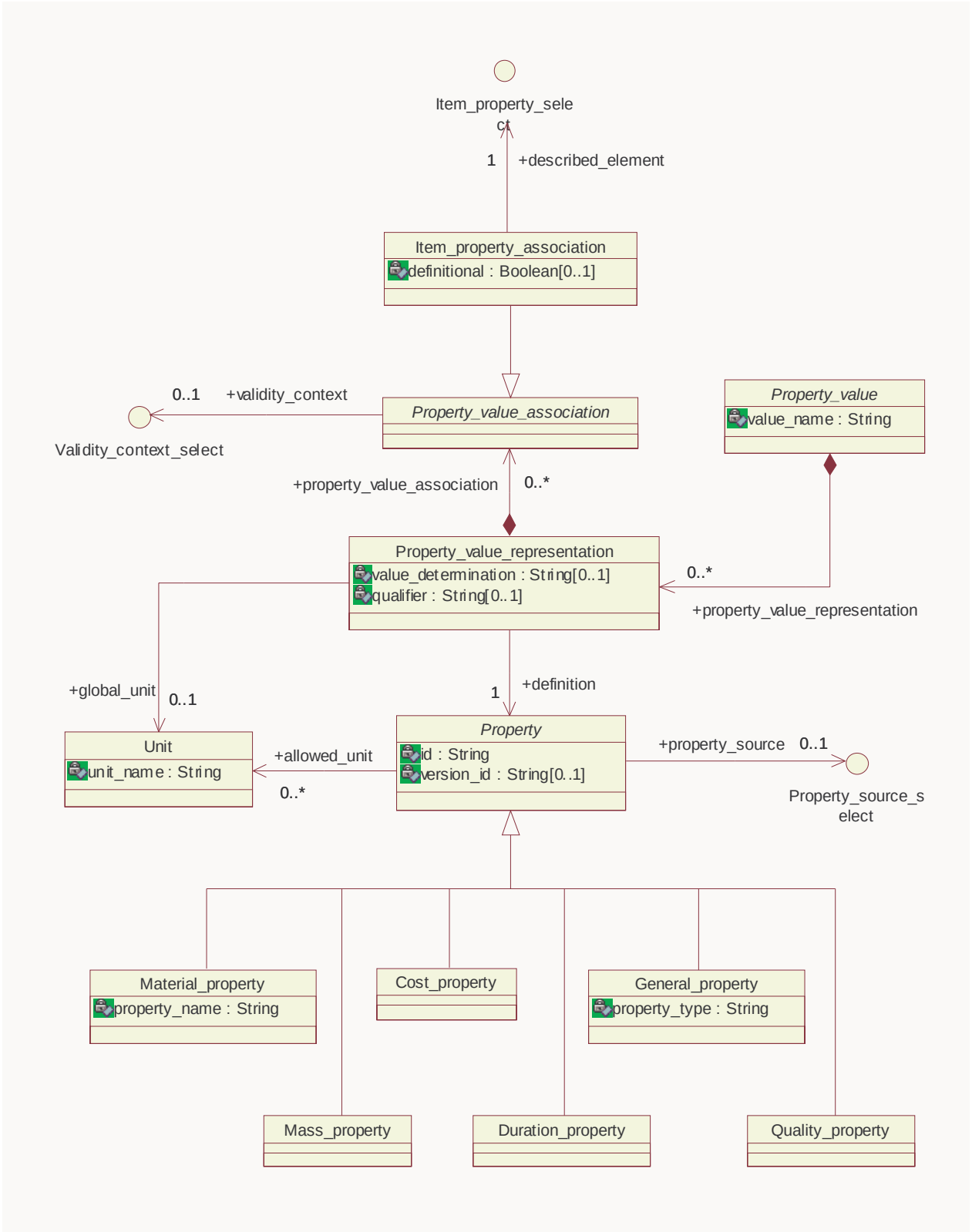

Item_property_sele
ct
1
+described_element
Item_property_association
definitional : Boolean[0..1]
0..1
+validity_context
Validity_context_select
Property_value_association
Property_value
value_name : String
+property_value_association
0..*
Property_value_representation
value_determination : String[0..1]
qualifier : String[0..1]
0..*
+property_value_representation
+global_unit
0..1
1
+definition
Unit
unit_name : String
+allowed_unit
0..*
Property
id : String
version_id : String[0..1]
+property_source
0..1
Property_source_s
elect
Material_property
property_name : String
Cost_property
General_property
property_type : String
Mass_property
Duration_property
Quality_property

Figure 104 Property

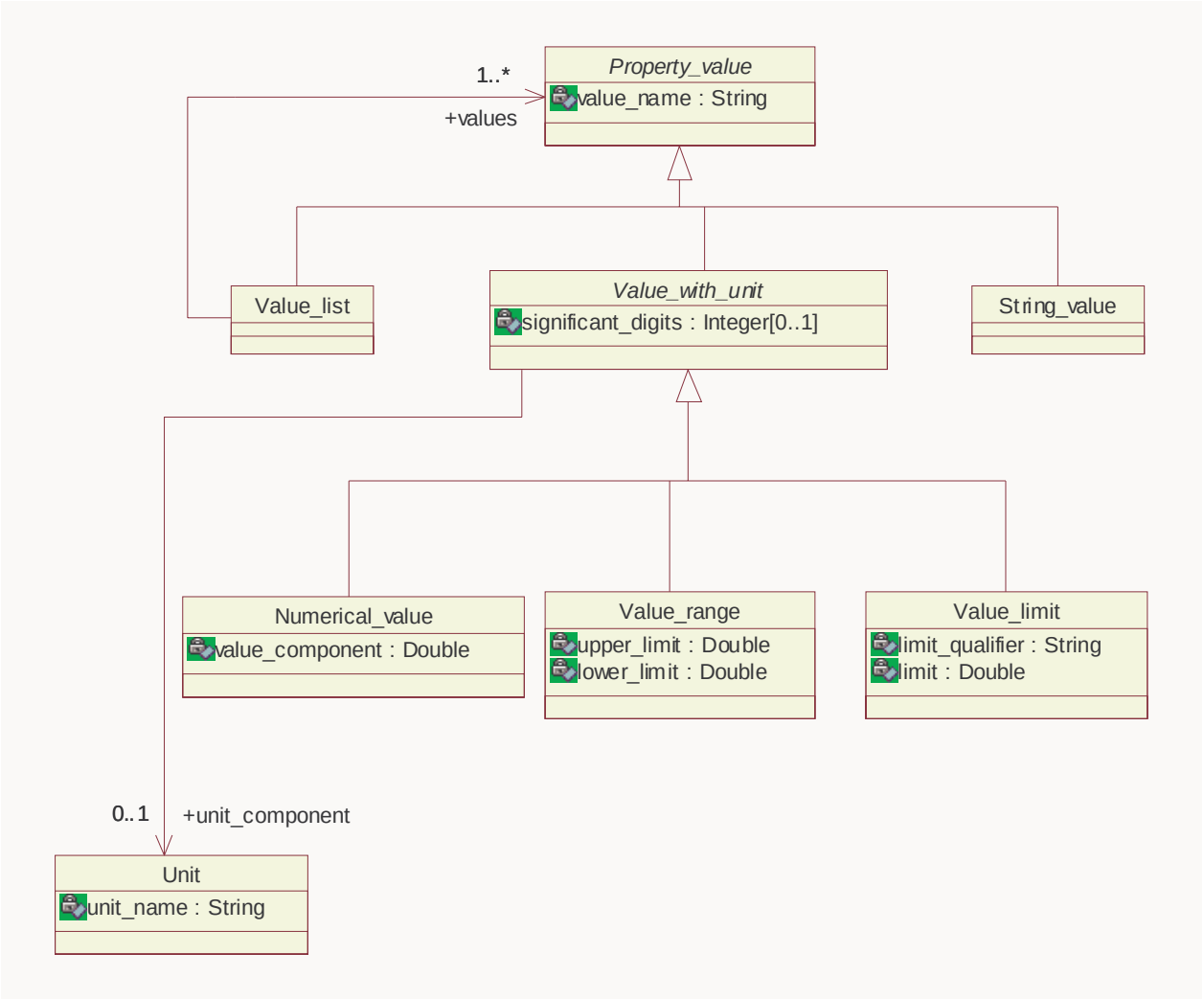


Figure 105 Property value

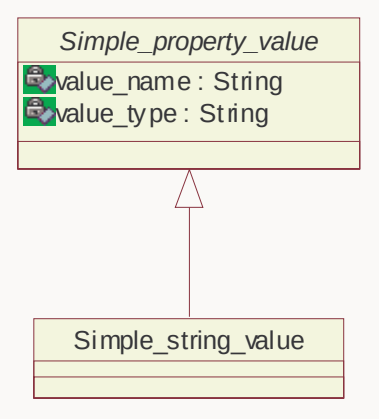


Figure 106 Simple property

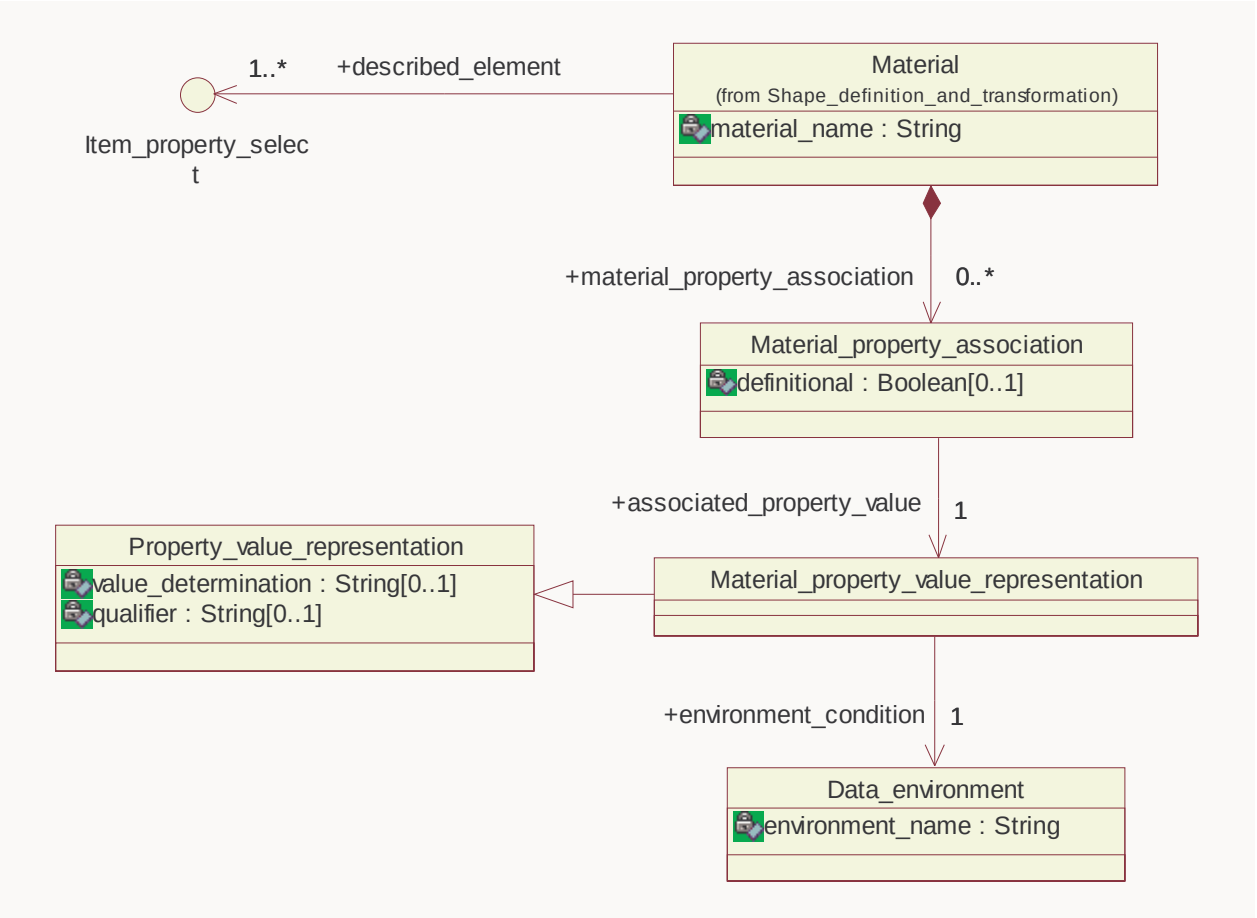


Figure 107 Material property

7.7.7.1 Class Cost_property

A Cost_property is a property that specifies costs.

Base Class

- Property (ABS)

Attributes

- none

Compositions

- none

Associations

- none

7.7.7.2 Class Data_environment

A Data_environment is the specification of the conditions under which a Material_property_value_representation is valid.

Base Class

- PLM_root_object (ABS)

Attributes

- environment_name : String [1] The environment_name specifies the word or group of words by which the Data_environment is referred to.

Compositions

- description : String_select [0..1]The description specifies additional information about the Data_environment.

Associations

- none

7.7.7.3 Class Duration_property

A Duration_property is a property that specifies a period of time during which a given object is used or will last.

Base Class

- Property (ABS)

Attributes

- none

Compositions

- none

Associations

- none

7.7.7.4 Class General_property

A General_property is the definition of a property that is specified by the attribute 'property_type'.

Base Class

- Property (ABS)

Attributes

- property_type : String [1] The property_type specifies the kind of property the General_property defines. Where applicable the following values shall be used:
  - 'overall axle distance': The overall axle distance is the distance between the first front axle and the rear most axle of the vehicle combination;
  - 'positioning': The General_property is the definition of a Model_property_value that provides an a geometric model for a Product_component or an Item_instance for the purpose of placement;
  - 'theoretical wheelbase': The theoretical wheelbase is the distance between the resolved weight lines of front and rear axle combinations;
  - 'track': The track is the distance between the centre of the tires mounted on an axle of a vehicle;
  - 'wheel space': The wheel space is the distance between the perpendicular lines constructed to the longitudinal median plane of the vehicle from two points that represent the wheels situated at the same side of the axle that is of interest.

Compositions

- none

Associations

- none

7.7.7.5 Class Item_property_association

An Item_property_association is a mechanism to associate a property value with an object.

Base Class

- Property_value_association (ABS)

Attributes

- definitional : Boolean [0..1] The definitional specifies whether the associated Property_value_representation object may be used to distinguish the described_element from others of the same kind. A value of 'true' indicates that the associated Property_value_representation distinguishes it from others.

Compositions

- none

Associations

- described_element : Item_property_select [1]
  The described_element specifies the object that is characterized by the Property_value.

7.7.7.6 Class Mass_property

A Mass_property is a quantity of matter that an object consists of.

Base Class

- Property (ABS)

Attributes

- none

Compositions

- none

Associations

- none

7.7.7.7 Class Material_property

A Material_property is a characteristic that depends on material aspects.

Base Class

- Property (ABS)

Attributes

- property_name : String [1] The property_name specifies the kind of Material_property.

Compositions

- none

Associations

- none

7.7.7.8 Class Material_property_association

A Material_property_association is an object that associates a Material object with a Material_property_value_representation object.

Base Class

- PLM_object (ABS)

Attributes

- definitional : Boolean [0..1] The definitional specifies whether the associated_property_value may be used to distinguish the described_material from others of the same kind. A value of 'true' indicates that the Material_property_value_representation distinguishes the 'described_element' from others.

Compositions

- none

Associations

- associated_property_value : Material_property_value_representation [1]
  The associated_property_value specifies the associated Material_property_value_representation.

7.7.7.9 Class Material_property_value_representation

A Material_property_value_representation is the representation of a characteristic of a material.

Base Class

- Property_value_representation

Attributes

- none

Compositions

- none

Associations

- environment_condition : Data_environment [1]
  The environment_condition specifies the environmental conditions in which the defined Material_property_value_representation is applicable.

7.7.7.10 Class Numerical_value

A Numerical_value is a quantity expressed with a numerical value and a unit.

Base Class

- Value_with_unit (ABS)

Attributes

- value_component : Double [1] The value_component specifies the quantity of the Numerical_value.

Compositions

- none

Associations

- none

7.7.7.11 Class Property (ABS)

A Property is the definition of a particular quality.

Base Class

- PLM_root_object (ABS)

Attributes

- id : String [1] The id specifies the identifier of the Property.
- version_id : String [0..1] The version_id specifies the identification of a particular version of a Property.

Compositions

- description : String_select [0..1]The description specifies additional information about the Property.
- alias_identification : Alias_identification [0..*]
  The Alias_identification specifies the Alias_identification that is applied to this Property.
- document_assignment : Document_assignment [0..*]
  The document_assignment specifies the object that provides information for this Property.

Associations

- property_source : Property_source_select [0..1]
  The property_source specifies the External_library_reference or Plib_property_reference object that defines this kind of property.
- allowed_unit : Unit [0..*] The allowed_unit specifies the unit or set of units that are accepted.

7.7.7.12 Class Property_value (ABS)

A Property_value is the numerical or textual value of a Property_value_representation.

Base Class

- PLM_root_object (ABS)

Attributes

- value_name : String [1] The value_name specifies the word or group of words by which the Property_value is referred to.

Compositions

- property_value_representation : Property_value_representation [0..*]
  The property_value_representation specifies the property_value_representation that is qualified by this Property_value, by a Value_with_unit, a String_value, or an arbitrary aggregate thereof.

Associations

- none

7.7.7.13 Class Property_value_association (ABS)

A Property_value_association is a mechanism to assign a Property_value_representation to an object.

Base Class

- PLM_object (ABS)

Attributes

- none

Compositions

- description : String_select [0..1]The description specifies additional information about the Property_value_association.

Associations

- validity_context : Validity_context_select [0..1]
  The validity_context specifies the context in which a Property_value_association is applicable.

7.7.7.14 Class Property_value_representation

A Property_value_representation is the representation of Property.

Base Class

- PLM_object (ABS)

Attributes

- value_determination : String [0..1]The value_determination specifies information on how the Property_value_representation shall be interpreted. Where applicable the following values shall be used:
  - 'calculated': The value has been calculated;
  - 'designed': The value represents a value intended by the design;
  - 'estimated': The value has been estimated;
  - 'measured': The value has been measured;
  - 'required': The value represents a requirement;
  - 'set point': The value is used as the initialization value.
- qualifier : String [0..1] The qualifier specifies the kind of the Property_value_representation. The following values shall be used:
  - 'nominal': The value is the nominal value;
  - 'specified': The value is specified;
  - 'typical': The value is a typical value.

Compositions

- property_value_association : Property_value_association (ABS) [0..*]
  The property_value_association specifies the property_value_association which this object is assigned to.

Associations

- definition : Property (ABS) [1] The definition specifies the Property that the Property_value_representation characterizes. If the Property_value_representation is a Material_property_value_representation, the definition shall specify a Material_property.
- global_unit : Unit [0..1] The global_unit specifies a unit that is valid for all Property_value that are referenced as 'specified_value' by the Property_value_representation.

7.7.7.15 Class Quality_property

A Quality_property is a property that enables to provide information about the level of quality of products or processes.

Base Class

- Property (ABS)

Attributes

- none

Compositions

- none

Associations

- none

7.7.7.16 Class Recyclability_property

A Recyclability_property is information concerning the ability to reuse objects or components of objects after their primarily intended usage.

Base Class

- Property (ABS)

Attributes

- none

Compositions

- none

Associations

- none

7.7.7.17 Class simple_property_association (ABS)

A simple_property_association holds a name and a type. The value is added by one of its sub-types.

Base Class

- PLM_object (ABS)

Attributes

- value_name : String [1] — The value_name specifies the word or group of words by which the Property_value is referred to.
- value_type : String [1] — The property_type specifies the kind of property the General_property defines. Where applicable the following values shall be used:
  - 'cost': The cost of an object;
  - 'duration': The duration specifies a period of time during which a given object is used or will last;
  - 'mass': The mass is the quantity of matter that an object consists of;
  - 'overall axle distance': The overall axle distance is the distance between the first front axle and the rear most axle of the vehicle combination;
  - 'positioning': The General_property is the definition of a Model_property_value that provides an a geometric model for a Product_component or an Item_instance for the purpose of placement;
  - 'quality': The quality of products or processes;
  - 'recyclability': The recyclability is the ability to reuse objects or components of objects after their primarily intended usage;
  - 'theoretical wheelbase': The theoretical wheelbase is the distance between the resolved weight lines of front and rear axle combinations;
  - 'track': The track is the distance between the centre of the tyres mounted on an axle of a vehicle;
  - 'wheel space': The wheel space is the distance between the perpendicular lines constructed to the longitudinal median plane of the vehicle from two points that represent the wheels situated at the same side of the axle that is of interest.

Compositions

- none

Associations

- none

7.7.7.18 Class Simple_string_value

A Simple_string_value represents a sequence of one or more alphanumeric characters.

Base Class

- Simple_property_value (ABS)

Attributes

- none

Compositions

- value_specification : String_select [1]
  The value_specification specifies the string represented by the Simple_string_value.

Associations

- none

7.7.7.19 Class String_value

A String_value represents a sequence of one or more alphanumeric characters.

Base Class

- Property_value (ABS)

Attributes

- none

Compositions

- value_specification : String_select [1]
  The value_specification specifies the string represented by the String_value.

Associations

- none

7.7.7.20 Class Unit

A Unit is a quantity chosen as a standard in terms of which other quantities may be expressed.

Base Class

- PLM_root_object (ABS)

Attributes

- unit_name : String [1] The unit_name specifies the term representing the kind of unit.

Compositions

- none

Associations

- none

7.7.7.21 Class Value_limit

A Value_limit is a qualified numerical value representing either the lower limit or the upper limit of a particular physical characteristic.

Base Class

- Value_with_unit (ABS)

Attributes

- limit_qualifier : String [1] The limit_qualifier specifies the kind of limit.
- limit : Double [1] The limit specifies the value of the limit.

Compositions

- none

Associations

- none

7.7.7.22 Class Value_list

A Value_list is an ordered collection of Property_value objects.

Base Class

- Property_value (ABS)

Attributes

- none

Compositions

- none

Associations

- values : Property_value (ABS) [1..*]
  The values specifies the ordered collection of Property_value objects that together are provided as a Property_value.

7.7.7.23 Class Value_range

A Value_range is a pair of numerical values representing the range in which the value shall lie.

Base Class

- Value_with_unit (ABS)

Attributes

- upper_limit : Double [1] The upper_limit specifies the maximum acceptable value that is constrained by the Value_range.
- lower_limit : Double [1] The lower_limit specifies the minimum acceptable value that is constrained by the Value_range.

Compositions

- none

Associations

- none

7.7.7.24 Class Value_with_unit (ABS)

A Value_with_unit is either a single numerical measure, or a range of numerical measures with upper, lower, or upper and lower bounds.

Base Class

- Property_value (ABS)

Attributes

- significant_digits : Integer [0..1] The significant_digits specifies the number of decimal digits that are relevant for the use of the Value_with_unit. If present, the numerical measure or range may be specified using more digits than the significant digits but shall not be specified using less digits.

Compositions

- none

Associations

- unit_component : Unit [0..1] The unit_component specifies the unit in which the Value_with_unit is expressed.

7.7.7.25 Interfaces

Interface Item_property_select

This empty interface is realized by the following classes:

- Product_structure_relationship
- Product_identification
- Product_class
- Physical_instance
- Design_constraint
- Complex_product (ABS)
- Document_representation (ABS)
- Document_file (ABS)
- Item_definition_relationship (ABS)
- Design_discipline_item_definition
- Item_instance_relationship (ABS)
- Item_instance (ABS)

- Item_definition_instance_relationship (ABS)
- Shape_element_relationship
- Shape_element
- Item_shape

Interface Property_source_select

This empty interface is realized by the following class:

- External_library_reference

Interface Validity_context_select

This empty interface is realized by the following classes:

- Organization
- Product_identification
- Product_class

7.7.8 Package Alias_identification

Figure 2?22 Alias identification

7.7.8.1 Class Alias_identification

An Alias_identification is a mechanism to associate an object with an additional identifier that is used to identify the object of interest in a different context, either in another Organization, or in some other context. The scope of the Alias_identification shall be specified either by the attribute 'alias_scope' or by the attribute 'description'.

Base Class

- PLM_object (ABS)

Attributes

- alias_id : String [1] The alias_id specifies the identifier used in the context specified by the alias_scope, or by the description.
- alias_version_id : String [0..1] The alias_version_id specifies the version of the object as known in the context of the Alias_identification.

Compositions

- description : String_select [0..1]The description specifies the type of the Alias_identification.

Associations

- alias_scope : Organization [0..1]The alias_scope specifies the Organization in which the Alias_identification is valid.

7.7.9 Package Authorization

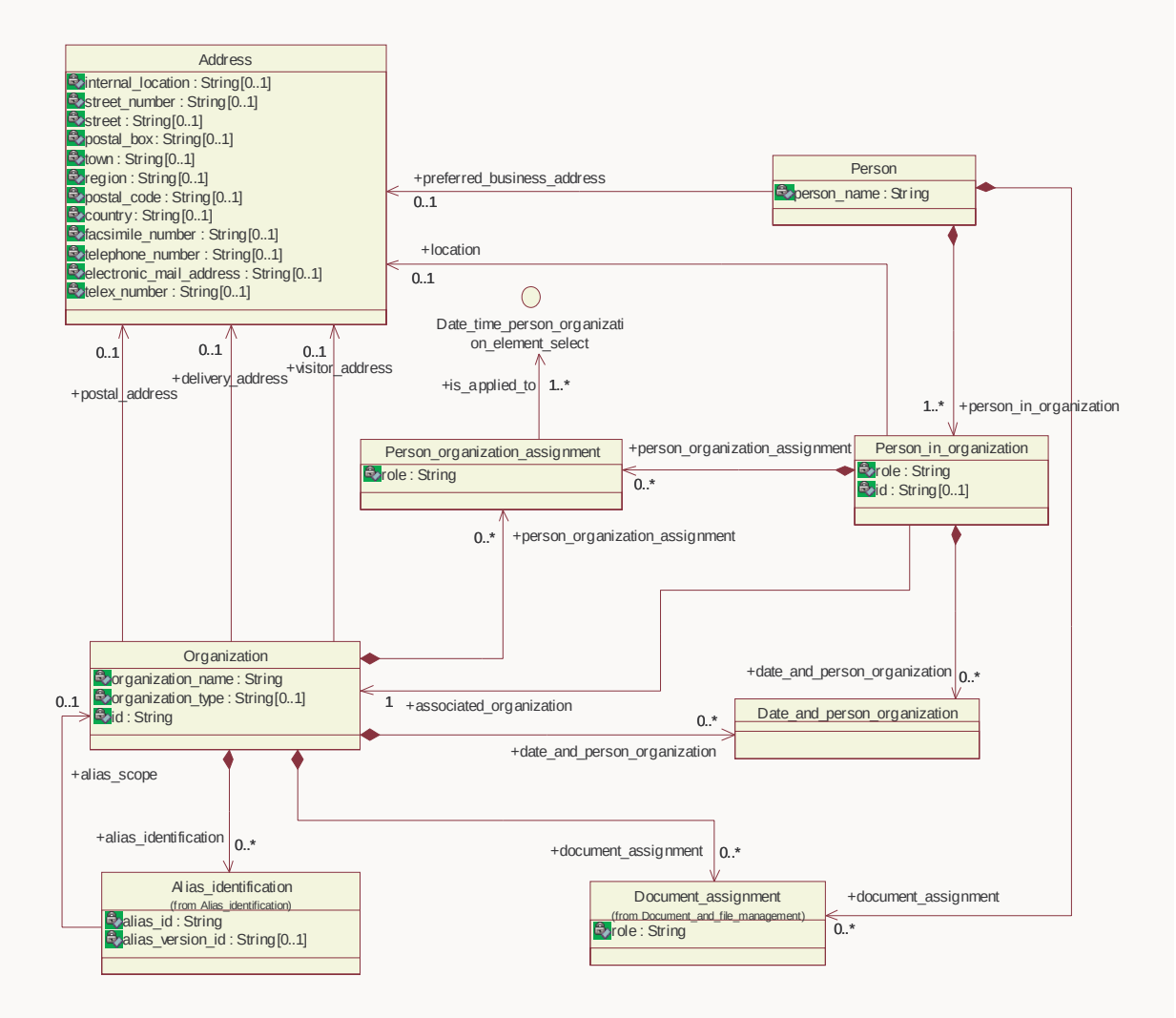


Figure 108 Authorization - Person and organization

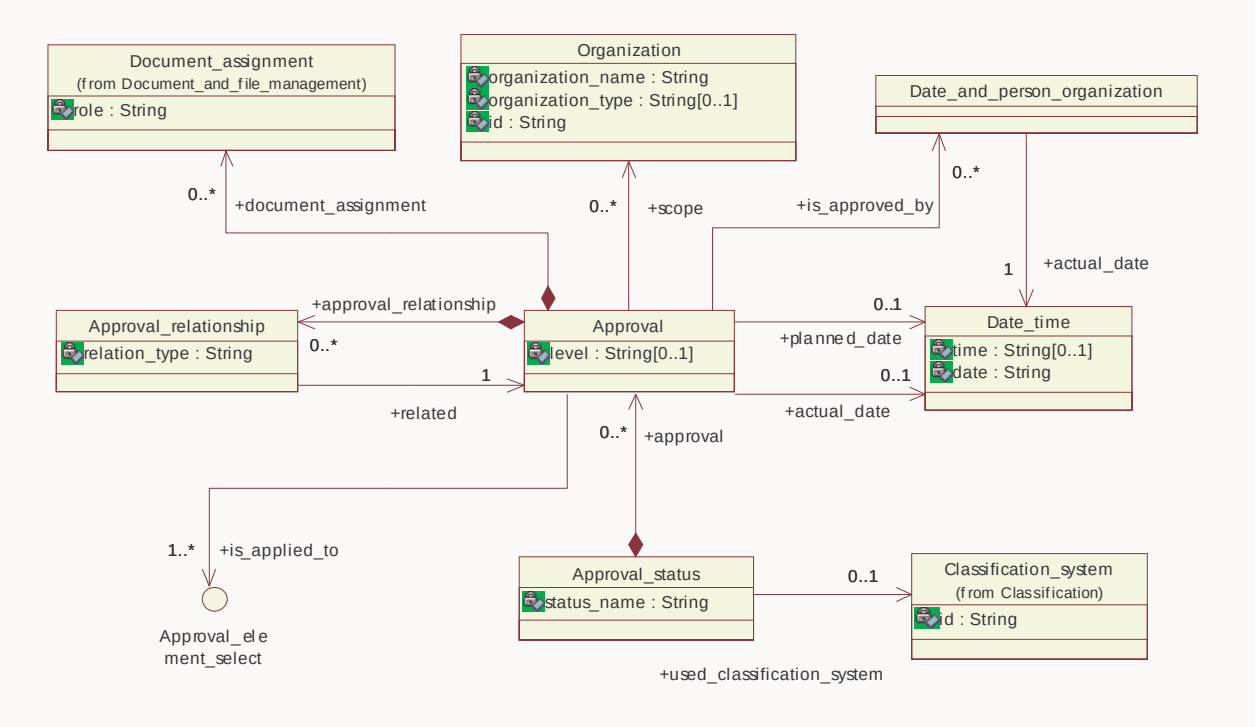


Figure 109 Authorization - Approval

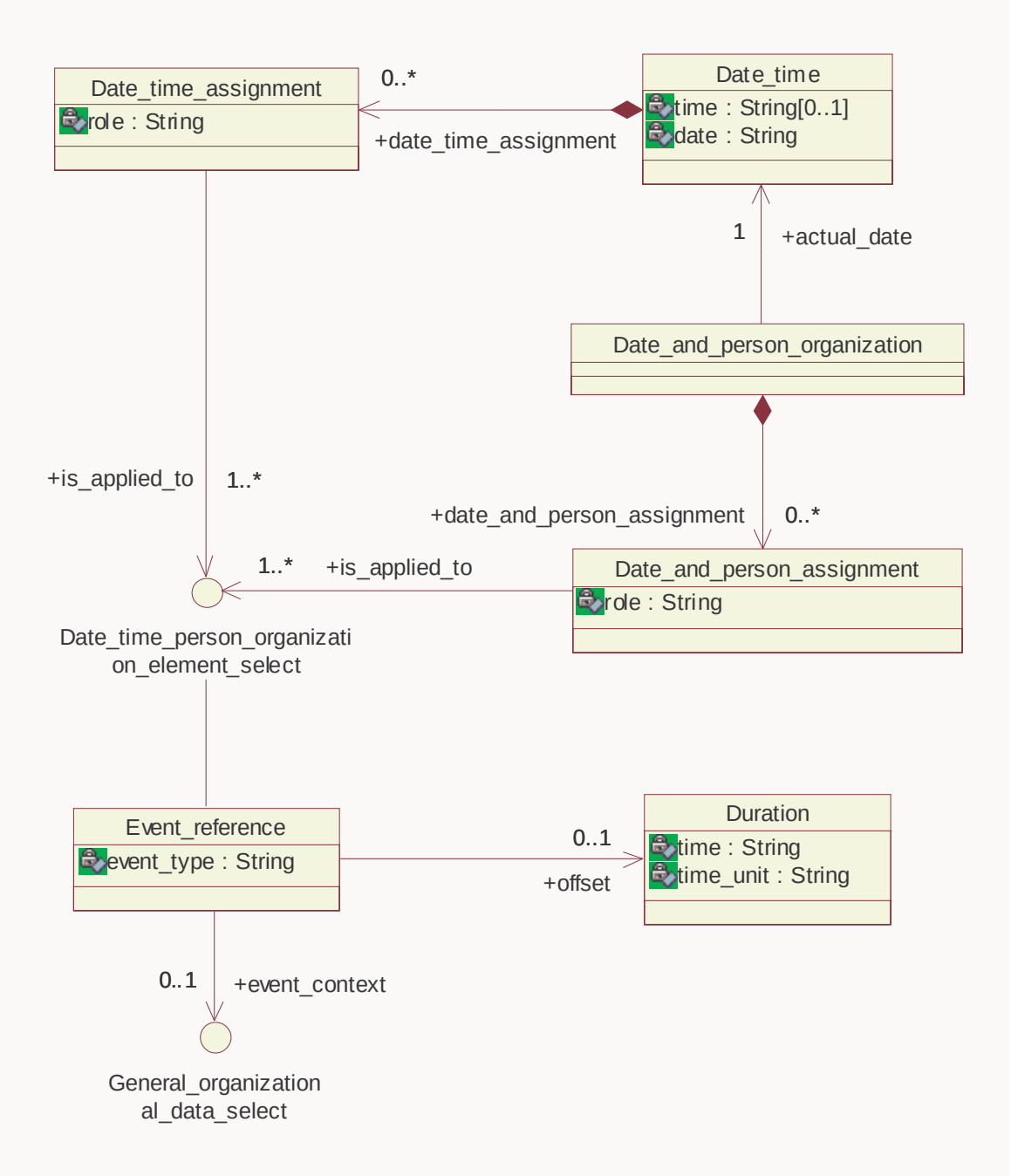


Figure 110 Authorization - Date and time

7.7.9.1 Class Address

An Address contains information about how a person or an organization can be contacted.

Base Class

- PLM_root_object (ABS)

Attributes

- internal_location : String [0..1] The internal location.
- street_number : String [0..1] The street number.
- street : String [0..1] The street.
- postal_box : String [0..1] The postal box.
- town : String [0..1] The town.
- region : String [0..1] The region.
- postal_code : String [0..1] The postal code.
- country : String [0..1] The country.
- facsimile_number : String [0..1]The fax number.
- telephone_number : String [0..1]The telephone number.
- electronic_mail_address : String [0..1]
  The e-mail address.
- telex_number : String [0..1] The telex number.

Compositions

- none

Associations

- none

7.7.9.2 Class Approval

An Approval is a judgement concerning the quality of those product data that are subject of the Approval. An Approval represents a statement made by technical personnel or management personnel whether certain requirements are met. The absence of approval information does not imply any approval status by default.

Base Class

- PLM_object (ABS)

Attributes

- level : String [0..1] The level represents the aspect for which the object subject to approval, by reference as 'is_applied_to', is endorsed. Where applicable the following values shall be used:
  - 'disposition': The referenced object is approved for series production;
  - 'equipment order': The referenced object has reached a status in which changes are subject to a defined change process and tools and other equipment required for production may be ordered;
  - 'planning': The referenced object is technically complete and has reached a status sufficiently stable so that other designs may be based on it.

Compositions

- approval_relationship : Approval_relationship [0..*]
  The Approval_relationship specifies the Approval_relationship that relates the first of the two Approval objects.
- document_assignment : Document_assignment [0..*]
  The document_assignment specifies the object that provides information for this Approval.

Associations

- scope : Organization [0..*] The scope specifies the set of Organization objects for which the Approval is valid.
- actual_date : Date_time [0..1] The actual_date specifies the date when the Approval actually became valid. If this attribute is absent, the approval has not yet occurred, i.e., it is pending.
- planned_date : Date_time [0..1]The planned_date specifies the date when the Approval is or was supposed to be performed.
- is_approved_by : Date_and_person_organization [0..*]
  The is_approved_by specifies personnel responsible for the Approval and the dates of the Approval.
- is_applied_to : Approval_element_select [1..*]
  The is_applied_to specifies the objects to which the Approval is assigned.

7.7.9.3 Class Approval_relationship

An Approval_relationship is a relationship between two Approval objects.

Base Class

- PLM_object (ABS)

Attributes

- relation_type : String [1] The relation_type specifies the meaning of the relationship. Where applicable the following values shall be used:
  - 'decomposition': The Approval_relationship defines a relationship where the related Approval is one of the components into which the relating Approval is broken down with no implication of 'sequence' or 'dependency';
  - 'dependency': The Approval_relationship defines a relationship where the issuing of the related Approval is dependent on the issuing of the relating Approval;
  - 'precedence': the Approval_relationship defines a relationship where the related Approval has higher priority than the relating Approval;
  - 'sequence': The Approval_relationship defines a relationship where the relating Approval shall be completed before the related Approval is given.

Compositions

- description : String_select [0..1]The description specifies additional information about the Approval_relationship.

Associations

- related : Approval [1] The related specifies the second of the two Approval objects related by the Approval_relationship.

7.7.9.4 Class Approval_status

An Approval_status is the state of acceptance of some product data.

Base Class

- PLM_root_object (ABS)

Attributes

- status_name : String [1] The status_name specifies the terms characterizing the Approval_status.

Compositions

- approval : Approval [0..*] The Approval indicates the approval that is applied to the level of acceptance of this Approval_status, for the specified 'level'.
- alias_identification : Alias_identification [0..*]
  The Alias_identification specifies the Alias_identification that is applied to this Approval_status.

Associations

- used_classification_system : Classification_system [0..1]
  The used_classification_system specifies the Classification_system that contains the informa-tion about how to interpret the Approval_status.

7.7.9.5 Class Date_and_person_assignment

A Date_and_person_assignment is an object that associates a Date_and_person_organization with product data. This assignment provides additional in-formation for the associated object.

Base Class

- PLM_object (ABS)

Attributes

- role : String [1] The role specifies the relationship between the date or time and the person or organization in the Date_and_person_assignment. Where applicable the following values shall be used:
  - 'creation': The assignment specifies that the referenced object has been created by the given person or organization at the given date and time;
  - 'update': The assignment specifies that the referenced object has been altered by the given person or organization at the given date and time.

Compositions

- description : String_select [0..1]The description specifies additional information about the Date_and_person_assignment.

Associations

- is_applied_to : Date_time_person_organization_element_select [1..*]
  The is_applied_to specifies the set of objects with which the Date_and_person_assignment is associated.

7.7.9.6 Class Date_and_person_organization

A Date_and_person_organization is a Person_in_organization or an Organization associ-ated with a Date_time or an Event_reference.

Base Class

- PLM_object (ABS)

Attributes

- none

Compositions

- date_and_person_assignment : Date_and_person_assignment [0..*]
  The Date_and_person_assignment specifies the Date_and_person_assignment for this Date_and_person_organization.

Associations

- actual_date : Date_time [1] The actual_date specifies the date and an optional time of day component of a Date_and_person_organization, or alternatively a discrete point in time as an Event_reference.

7.7.9.7 Class Date_time

A Date_time is the specification of a date and an optional time of day.

Base Class

- PLM_root_object (ABS)

Attributes

- time : String [0..1] The time specifies a moment of occurrence measured by hour, minute, and second.
- date : String [1] The date specifies the calendar time, defined according to the Gregorian calendar, conveying information about the year, the month, and the day in no specific order. The representation of a date shall be complete, i.e., millennium, century, and year-within-century data shall be included.

Compositions

- date_time_assignment : Date_time_assignment [0..*]
  The Date_time_assignment specifies the Date_time_assignment which this Date_time is assigned to.

Associations

- none

7.7.9.8 Class Date_time_assignment

A Date_time_assignment is an association of point in time specified as a Date_time or an Event_reference with product data.

Base Class

- PLM_object (ABS)

Attributes

- role : String [1] The role specifies the action associated with the Date_time_assignment. Where applicable the following values shall be used:
  - 'classification date': The assignment specifies that the specified object is classified at the given date and time. This value shall only be used, if the Date_time_assignment refers to instances of Classification_association as 'is_applied_to';
  - 'creation': The assignment specifies that the referenced object was created at the given date and time;
  - 'installation': The assignment specifies that the referenced object was mounted in a product at the given date and time;
  - 'production': The assignment specifies that the referenced object was produced at the given date and time;
  - 'registration': The assignment specifies that the referenced object was determined at the given date and time;
  - 'update': The assignment specifies that the referenced object was altered at the given date and time.

Compositions

- description : String_select [0..1]The description specifies additional information about the Date_time_assignment.

Associations

- is_applied_to : Date_time_person_organization_element_select [1..*]
  The is_applied_to specifies the set of objects of product data with which the Date_time_assignment is associated.

7.7.9.9 Class Duration

A Duration is the definition of a period of time.

Base Class

- PLM_root_object (ABS)

Attributes

- time : String [1] The time specifies the extend of the Duration.
- time_unit : String [1] The time_unit specifies the unit in which the time is specified.

Compositions

- none

Associations

- none

7.7.9.10 Class Event_reference

An Event_reference is the definition of a point in time established relative to an event.

Base Class

- PLM_root_object (ABS)

Attributes

- event_type : String [1] The event_type specifies the kind of event that serves as reference.

Compositions

- description : String_select [0..1]The description specifies additional information about the Event_reference.

Associations

- event_context : General_organizational_data_select [0..1]
  The event_context specifies the piece of product data the Event_reference refers to.
- offset : Duration [0..1] The offset specifies the amount of time before or after the defined event that shall be used to calculate the actual point in time.

7.7.9.11 Class Organization

An Organization is a group of people involved in a particular business process.

Base Class

- PLM_root_object (ABS)

Attributes

- organization_name : String [1] The organization_name specifies the word or group of words used to refer to the Organization.
- organization_type : String [0..1]The organization_type specifies the type of the Organization. Where applicable the following values shall be used:
  - 'company': The organization_type specifies that the Organization is a company;
  - 'department': The organization_type specifies that the Organization is a department;
  - 'plant': The organization_type specifies that the Organization is a plant.
- id : String [1] The id specifies the identifier of the Organization.

Compositions

- document_assignment : Document_assignment [0..*]
  The document_assignment specifies the object that provides information for this Organization.
- person_organization_assignment : Person_organization_assignment [0..*]
  The Person_organization_assignment specifies the Person_organization_assignment that concerns this Organization
- date_and_person_organization : Date_and_person_organization [0..*]
  The Date_and_person_organization specifies the Date_and_person_organization which this Organization is part of.
- alias_identification : Alias_identification [0..*]
  The Alias_identification specifies the Alias_identification that is applied to this Organization.

Associations

- postal_address : Address [0..1] The postal_address specifies the address where letter mail is delivered.
- delivery_address : Address [0..1]
  The delivery_address specifies the address where goods are delivered.
- visitor_address : Address [0..1]The visitor_address specifies the address where the organization receives visitors.

7.7.9.12 Class Organization_relationship

A Organization_relationship is the relationship between two Organization objects.

Base Class

- PLM_object (ABS)

Attributes

- relation_type : String [1] The relation_type specifies the intention of the Organization_relationship. Where applicable the following values shall be used:
  “hierarchy”: The related Organization is a sub organization of the relating Organization.
  “legal succession”: The related Organization is the legal successor of the relating Organization.
  “reorganization”: The related Organization is the successor of the relating Organization due to a organizational transfer of responsibility.

Compositions

- description : String [1] The description specifies additional information about the Organization_relationship.

Associations

- relating : Organization The relating specifies the first of the two Organization objects related by an Organization_relationship.
- related : Organization The related specifies the second of the two Organization objects related by an Organization_relationship

7.7.9.13 Class Person

A Person is an individual human being who has some relationship to product data. The Person shall always be identified in the context of one or more organizations.

Base Class

- PLM_root_object (ABS)

Attributes

- person_name : String [1] The person_name specifies the word or group of words used to refer to the Person.

Compositions

- person_in_organization : Person_in_organization [1..*]
  The Person_in_organization specifies the person_in_organization which this Person is assigned to.
- document_assignment : Document_assignment [0..*]
  The document_assignment specifies the object that provides information for this Person.

Associations

- preferred_business_address : Address [0..1]
  The preferred_business_address specifies the location of the office of the Person.

7.7.9.14 Class Person_in_organization

A Person_in_organization is the specification of a Person in the context of an Organization.

Base Class

- PLM_object (ABS)

Attributes

- role : String [1] The role specifies the relationship between the Person and the Organization.
- id : String [0..1] The id specifies an identifier of the person. The identifier shall be unique within the scope of the 'associated_organization'.

Compositions

- person_in_organization_relationship : Person_in_organization_relationship [0..*]
  The Person_in_organization_relationship specifies the Person_in_organization_relationship that concerns this Person_in_organization.
- person_organization_assignment : Person_organization_assignment [0..*]
  The Person_organization_assignment specifies the Person_organization_assignment that concerns this Person_in_organization.
- date_and_person_organization : Date_and_person_organization [0..*]
  The Date_and_person_organization specifies the Date_and_person_organization which this Person_in_organization is part of.

Associations

- location : Address [0..1] The location specifies the relevant address of the Person_in_organization.
- associated_organization : Organization [1]
  The associated_organization specifies the Organization with which the Person is associated.

7.7.9.15 Class Person_organization_assignment

A Person_organization_assignment is an object that associates an Organization or a Person_in_organization with product data.

Base Class

- PLM_object (ABS)

Attributes

- role : String [1] The role specifies the responsibility of the assigned Person or Organization with respect to the object that it is applied to. Where applicable the following values shall be used:
  - 'author': The referenced object has been created by the assigned Person or Organization. The author holds the copyright;
  - 'classification officer': The assigned Person or Organization is formally responsible for the classification of the referenced object;
  - 'creator': The referenced object has been created by the assigned Person or Organization;
  - 'custodian': The assigned Person or Organization is responsible for the existence and integrity of the referenced object;
  - 'customer': The assigned Person or Organization acts as a purchaser or consumer of the referenced object;
  - 'design supplier': The assigned Person or Organization is the one who delivers the data de-scribing the referenced object;
  - 'editor': The assigned Person or Organization is responsible for making any changes to any at-tribute of the referenced object;
  - 'id owner': The assigned Person or Organization is the one responsible for the designation of an identifier;
  - 'location': The assigned Organization is the place where the referenced object can be found or where it takes place;
  - 'manufacturer': The assigned Person or Organization is the one who produces the actual (physical) object;
  - 'owner': The assigned Person or Organization owns the referenced object, and has final say over its disposition and any changes to it;
  - 'supplier': The assigned Person or Organization is the one who delivers the actual (physical) object (e.g., a dealer);
  - 'wholesaler': The assigned Person or Organization is the one who is in the sales chain between the manufacturer and the supplier.

Compositions

- description : String_select [0..1] The description specifies additional information about the Person_organization_assignment.

Associations

- is_applied_to : Date_time_person_organization_element_select [1..*]
  The is_applied_to specifies the object with which the Person_organization_assignment is associated.

7.7.9.16 Class Person_in_organization_relationship

A Person_in_organization_relationship is a mechanism which allows to specify a relationship between two persons in an organization.

Base Class

- PLM_object (ABS)

Attributes

- relation_type : String [1] The relation_type specifies the meaning of the Person_in_organization_relationship. Where applicable the following values shall be used:
  “successor”: The related Person_in_organization is the successor of the relating Person_in_organization.

Compositions

- description : String [1] The description specifies additional information about the Person_in_organization_relationship.

Associations

- relating : Person_in_organizationThe relating specifies the first of the two Person_in_orgnaization objects related by an Person_in_organization_relationship.
- related : Person_in_organizationThe related specifies the second of the two Person_in_organization objects related by an Person_in_organization_relationship

7.7.9.17 Interfaces

Interface Approval_element_select

This empty interface is realized by the following classes:

- Work_request
- Work_order
- Project
- Activity_method_assignment
- Activity_element
- Activity
- General_classification
- Classification_system
- Classification_association

- Specification_inclusion
- Specification_expression
- Specification_category
- Specification
- Product_structure_relationship
- Product_class
- Physical_instance_test_result
- Physical_instance
- Manufacturing_configuration (ABS)
- Design_constraint
- Configuration
- Complex_product (ABS)
- Class_structure_relationship
- Class_specification_association
- Class_inclusion_association
- Class_condition_association
- Class_category_association
- Document_version
- Document_representation (ABS)
- Document_file (ABS)
- Document
- Item_version
- Item_definition_relationship (ABS)
- Design_discipline_item_definition
- Physical_assembly_relationship
- Item_instance_relationship (ABS)
- Item_instance (ABS)
- Item_definition_instance_relationship (ABS)
- Process_plan
- Property_value_association (ABS)
- Property (ABS)
- Material
- Geometric_model

Interface Date_time_person_organization_element_select

This empty interface is realized by the following classes:

- Person_in_organization
- Event_reference
- Approval_status
- Work_request
- Work_order
- Project
- Activity_method_assignment
- Activity_element
- Activity
- General_classification
- Classification_system
- Classification_association
- Specification_inclusion
- Specification_expression
- Specification_category
- Specification
- Product_structure_relationship
- Product_identification
- Product_class
- Physical_instance_test_result
- Physical_instance
- Manufacturing_configuration (ABS)
- Design_constraint
- Configuration
- Complex_product_relationship
- Complex_product (ABS)
- Class_structure_relationship
- Class_specification_association
- Class_inclusion_association
- Class_condition_association
- Class_category_association
- Document_version

- Document_representation (ABS)
- Document_file (ABS)
- Document
- Item_version_relationship
- Item_version
- Item_definition_relationship (ABS)
- Item
- Design_discipline_item_definition
- Physical_assembly_relationship
- Item_instance_relationship (ABS)
- Item_instance (ABS)
- Item_definition_instance_relationship (ABS)
- Process_plan
- Process_operation_resource_assignment
- Process_operation_occurrence
- Process_operation_definition
- Property_value_association (ABS)
- Property (ABS)
- Material
- Geometric_model

Interface Event_or_date_select

This empty interface is realized by the following classes:

- Event_reference
- Date_time

Interface General_organizational_data_select

This empty interface is realized by the following classes:

- Person_in_organization
- Approval_status
- Work_request
- Work_order
- Project
- Activity_method_assignment
- Activity_element

- Activity
- General_classification
- Classification_system
- Classification_association
- Specification_inclusion
- Specification_expression
- Specification_category
- Specification
- Product_structure_relationship
- Product_identification
- Product_class
- Physical_instance_test_result
- Physical_instance
- Manufacturing_configuration (ABS)
- Design_constraint
- Configuration
- Complex_product_relationship
- Complex_product (ABS)
- Class_structure_relationship
- Class_specification_association
- Class_inclusion_association
- Class_condition_association
- Class_category_association
- Document_version
- Document_representation (ABS)
- Document_file (ABS)
- Document
- Item_version_relationship
- Item_version
- Item_definition_relationship (ABS)
- Item
- Design_discipline_item_definition
- Physical_assembly_relationship
- Item_instance_relationship (ABS)

- Item_instance (ABS)
- Item_definition_instance_relationship (ABS)
- Process_plan
- Process_operation_resource_assignment
- Process_operation_occurrence
- Process_operation_definition
- Property_value_association (ABS)
- Property (ABS)
- Material
- Geometric_model

Interface Period_or_date_select

This empty interface is realized by the following classes:

- Event_reference
- Duration
- Date_time

7.7.10 Package Configuration_management

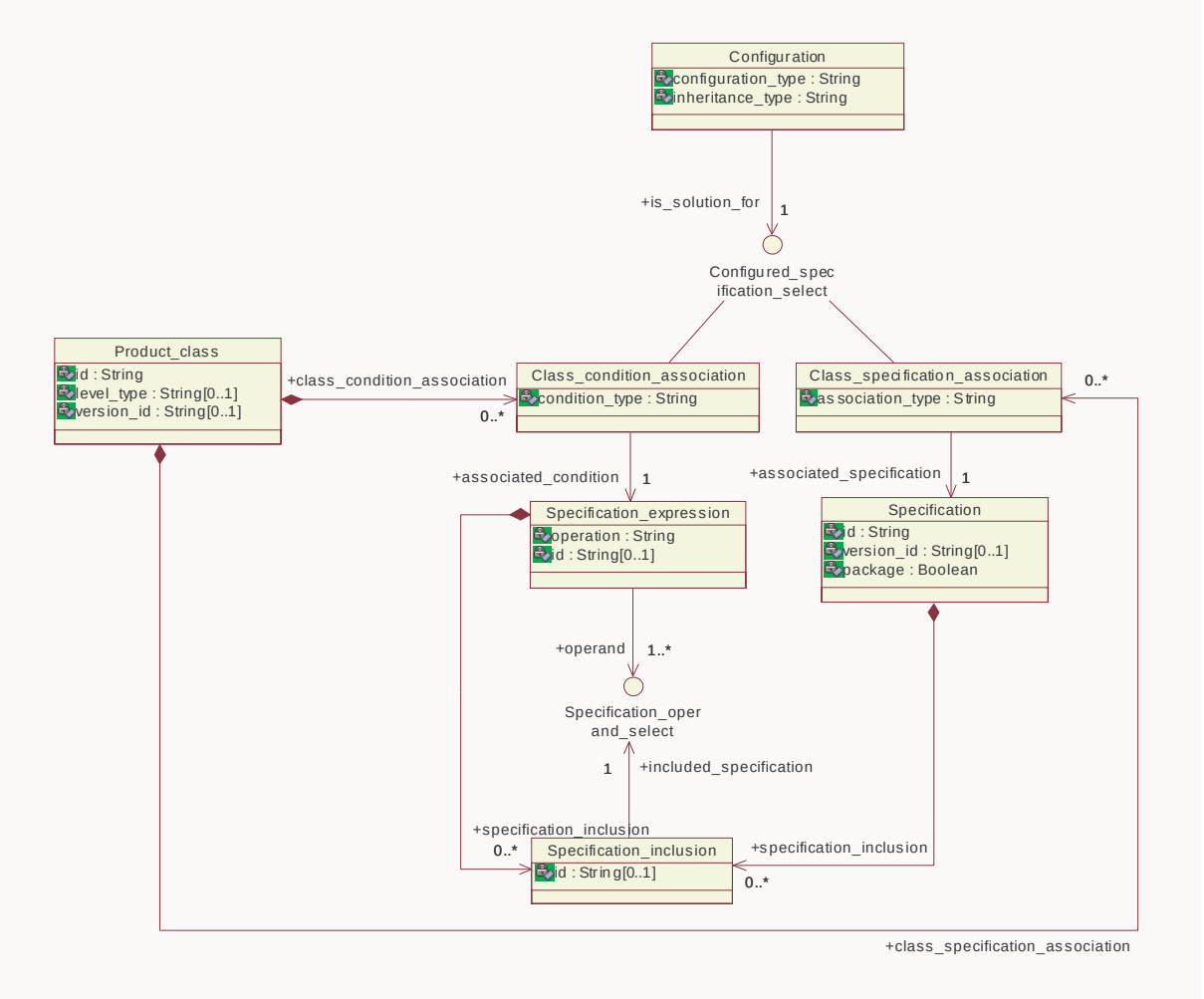


Figure 111 - Configuration management - Product class condition and specification

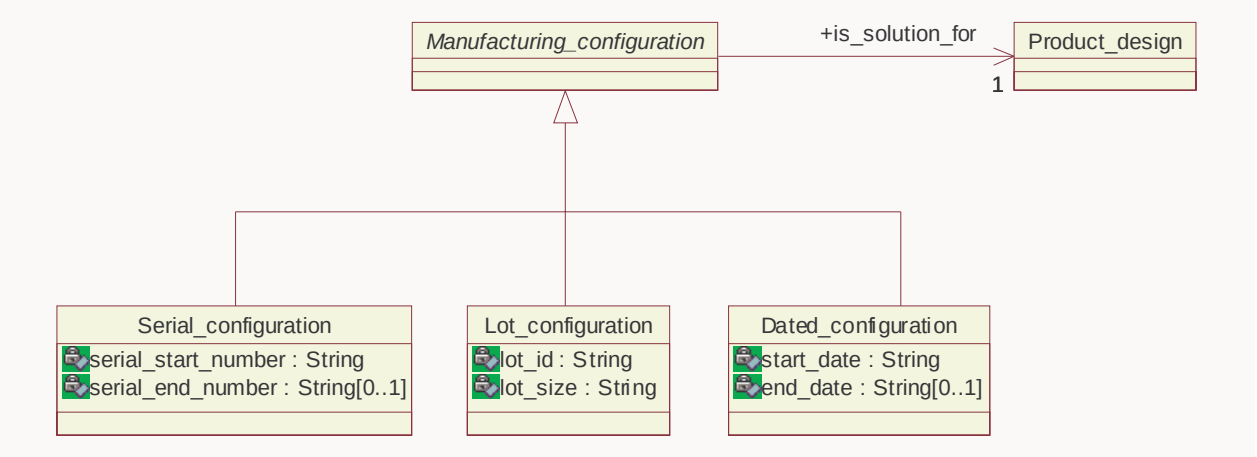


Figure 112 - Configuration management - manufacturing configuration

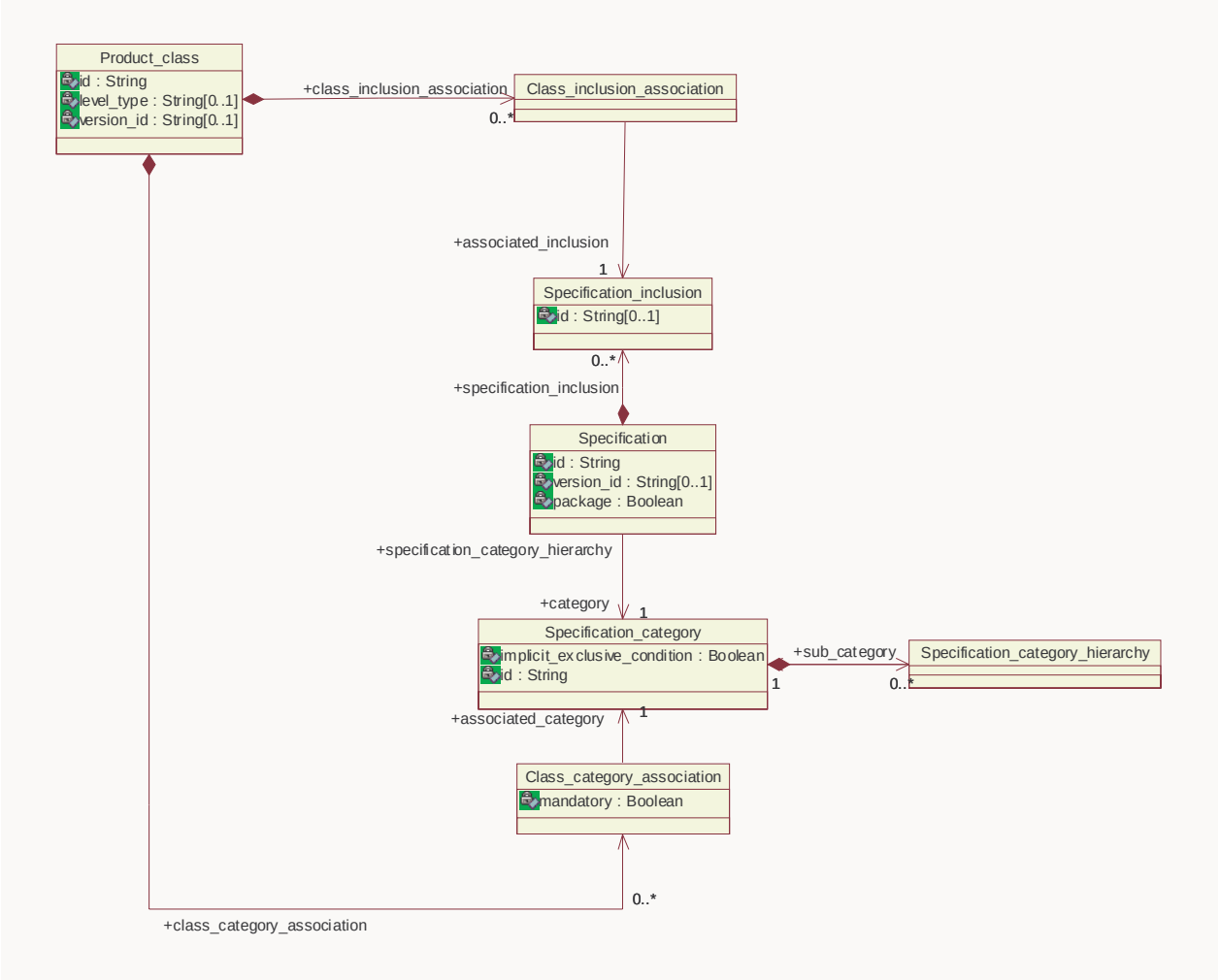


Figure 113 - Change management - specification category and inclusion

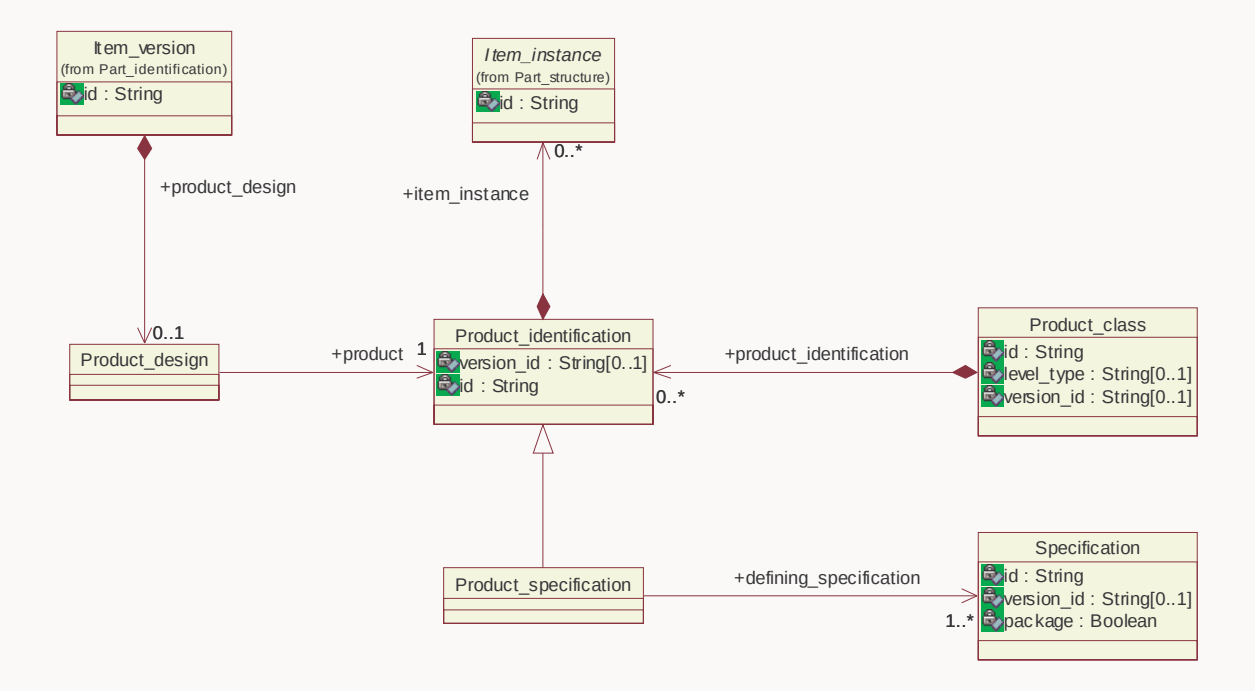


Figure 114 - Change management - Product identification

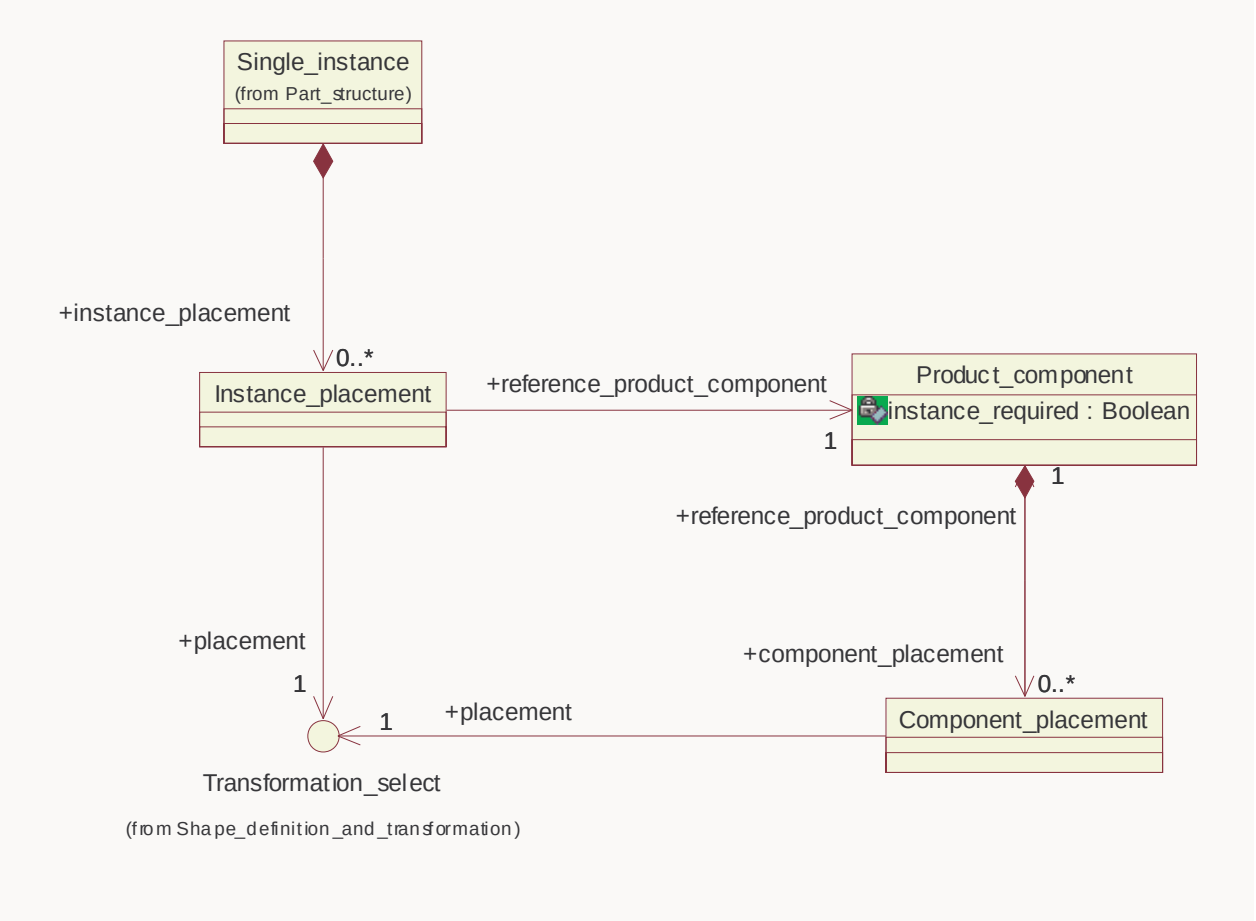


Figure 115 - Configuration management - Component and instance placement

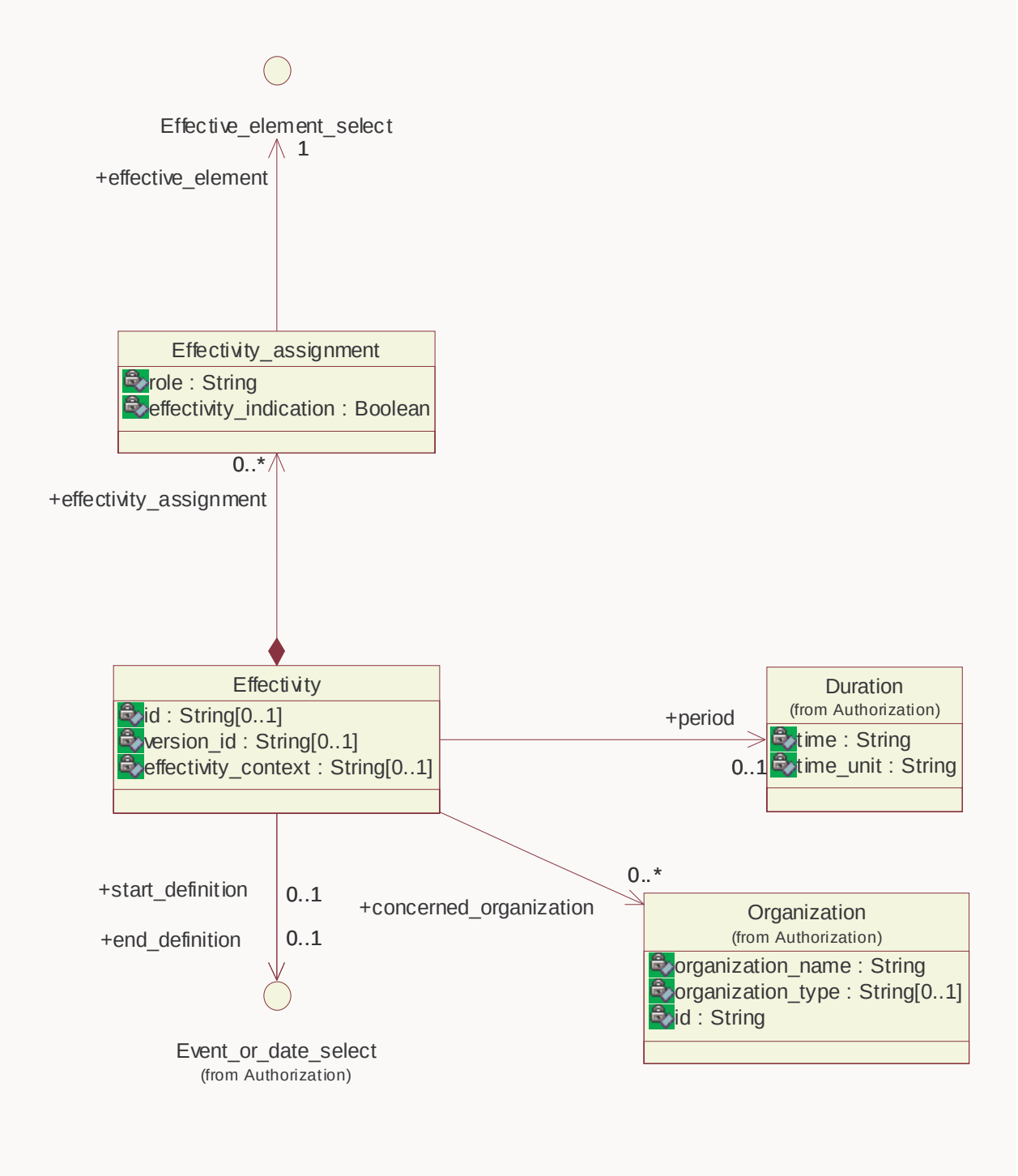


Figure 116 - Configuration management - Effectivity

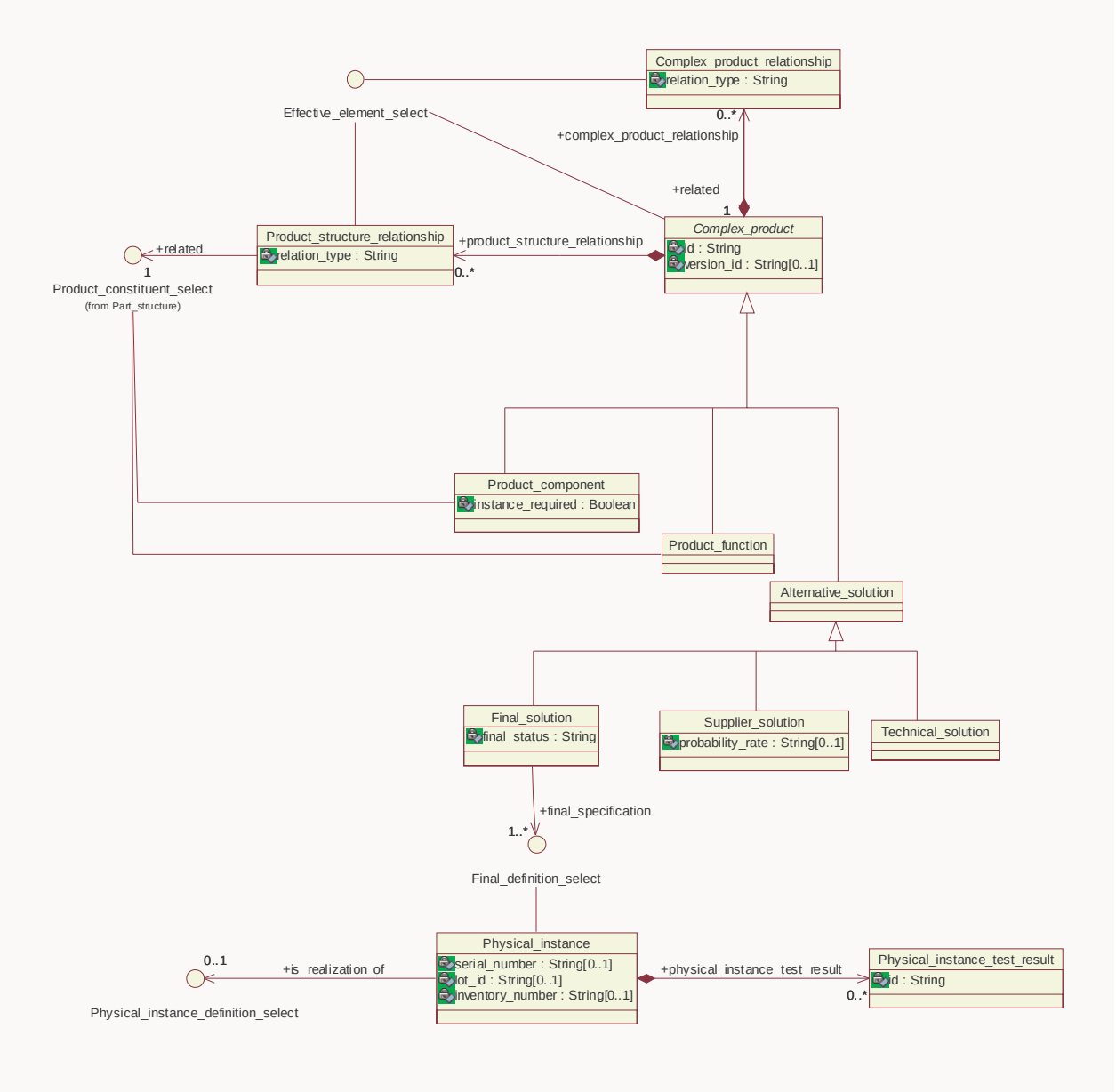


Figure 117 - Configuration management - Complex product

7.7.10.1 Class Alternative_solution

An Alternative_solution is the identification of one of potentially many mutually exclusive implementations of a Product_function or of a Product_component.

Base Class

- Complex_product (ABS)

Attributes

- none

Compositions

- configuration : Configuration [0..*]
  The configuration specifies the configuration that controls this Alternative_solution for its valid usage.

Associations

- base_element : Complex_product_select [1]
  The base_element specifies the object, for which the Alternative_solution provides a design alternative. All Alternative_solution objects for the same base_element are mutually exclusive.

7.7.10.2 Class Class_category_association

A Class_category_association is the association of a Specification_category with a Product_class. Additionally, this assignment specifies if the usage of one or more Specification objects belonging to this Specification_category, is mandatory or optional for all products of that Product_class.

Base Class

- PLM_object (ABS)

Attributes

- mandatory : Boolean [1] The mandatory specifies whether the Specification objects referring to the associated Specification_category have to be used or may be used (optional) for products within the referenced Product_class. A value of 'true' indicates that the usage is mandatory.

Compositions

- none

Associations

- associated_category : Specification_category [1]
  The associated_category specifies the Specification_category that is associated with the Product_class.

7.7.10.3 Class Class_condition_association

A Class_condition_association is the association of a Specification_expression with a Product_class.

Base Class

- PLM_object (ABS)

Attributes

- condition_type : String [1] The condition_type specifies the meaning of the association. Where applicable the following values shall be used:

- 'design case': The Specification_expression specifies a condition when a given object has to be designed and verified. This value of the condition_type is for information only and shall not be interpreted when querying design cases or usage cases. For such a query, the value of the attribute 'configuration_type' of Configuration shall be evaluated;
- 'identification': The Specification_expression specifies a condition that enables to distinguish the associated Product_class from other Product_class objects. This value is not applicable for a top level node in a hierarchy of Product_class objects. This identification is part of the identification of all sub classes of this Product_class;
- 'part usage': The Specification_expression specifies a condition for the usage of the components of an Alternative_solution, the usage of an Item_instance or for the application of a Process_plan or a Process_operation_occurrence in the products of the associated Product_class. In this case, the Class_condition_association shall be referenced by at least one Configuration object;
- 'validity': The Specification_expression specifies a condition that is used to verify a Product_specification for the associated Product_class. That means that the Specification_expression evaluates to 'true' if the set of Specification objects is valid; otherwise it evaluates to 'false' with the meaning that the specified object is invalid for the Product_class.
It is valid for all products belonging to the 'associated_product_class' in case of the condition types 'identification' and 'validity'.

Compositions

- description : String_select [0..1]The description specifies additional information about the Class_condition_association.

Associations

- associated_condition : Specification_expression [1]
  The associated_condition specifies the Specification_expression that is assigned to the Product_class.

7.7.10.4 Class Class_inclusion_association

A Class_inclusion_association is the assignment of a Specification_inclusion to a Prod-uct_class. This assignment contains the information that a particular Specification_inclusion applies for all products of that Product_class.

Base Class

- PLM_object (ABS)

Attributes

- none

Compositions

- description : String_select [0..1]The description specifies additional information about the Class_inclusion_association.

Association

- associated_inclusion : Specification_inclusion [1]
  The associated_inclusion specifies the Specification_inclusion that is associated with the Product_class.

7.7.10.5 Class Class_specification_association

A Class_specification_association is an association of a Specification with a Product_class. This Specification serves as a potential characteristic of all products belonging to the Product_class.

Base Class

- PLM_object (ABS)

Attributes

- association_type : String [1] The association_type specifies the kind of availability of a particular Specification in a Product_class.

Compositions

- none

Associations

- associated_specification : Specification [1]
  The associated_specification specifies the Specification that is associated with the Product_class.

7.7.10.6 Class Class_structure_relationship

A Class_structure_relationship is an association between a Product_class object and either a Product_component or a Product_function object.

Base Class

- PLM_object (ABS)

Attributes

- relation_type : String [1] The relation_type specifies the meaning of the relationship. Where applicable the following values shall be used:
  - 'functionality': The related Product_function is an element of the functional structure of the relating Product_class. This relation type shall only be used if the related object is a Product_function;
  - 'realization': The related Product_component fulfils, partially or fully, the requirements identified with the relating Product_class. This relation type shall only be used if the related object is a Product_component.

Compositions

- description : String_select [0..1]The description specifies additional information about the Class_structure_relationship.

Associations

- related : Product_function_component_select [1]
  The related specifies the Product_component or Product_function object related by the Class_structure_relationship.

7.7.10.7 Class Complex_product (ABS)

A Complex_product is an object with the capability that it can be realized by, decomposed into or specialized as Product_constituent objects in a functional, logical, or physical way.

Base Class

- PLM_root_object (ABS)

Attributes

- id : String [1] The id specifies the identifier of the Complex_product.
- version_id : String [0..1] The version_id identifies a version of the concept represented by a Complex_product.

Compositions

- product_structure_relationship : Product_structure_relationship [0..*]
  The product_structure_relationship specifies the product_structure_relationship where this Complex_product is decomposed functionally, logically, or physically into or realized by the related Product_constituent.
- design_constraint_association : Design_constraint_association [0..*]
  The design_constraint_association specifies the design_constraint_association so that the De-sign_constraint.affects this object.
- complex_product_relationship : Complex_product_relationship [0..*]
  The complex_product_relationship specifies the complex_product_relationship that relates the first of the two Complex_product objects.
- alias_identification : Alias_identification [0..*]
  The Alias_identification specifies the Alias_identification that is applied to this Complex_product.
- document_assignment : Document_assignment [0..*]
  The document_assignment specifies the object that provides information for this Complex_product.
- simple_property_association : Simple_property_association (ABS) [0..*]
  The simple_property_association specifies the assigned simple property values.

Associations

- none

7.7.10.8 Class Complex_product_relationship

A Complex_product_relationship is a relationship between two Complex_product objects.

Base Class

- PLM_object (ABS)

Attributes

- relation_type : String [1] The relation_type specifies the meaning of the relationship. Where applicable the following values shall be used:

- 'derivation': the Complex_product_relationship defines a relationship where the related Complex_product is derived from the relating Complex_product;
- 'replacement': The Complex_product_relationship defines a relationship where the related Complex_product is used in place of the relating Complex_product;
- 'version hierarchy': the Complex_product_relationship defines a relationship where the related Complex_product is a sub version of the relating Complex_product;
- 'version sequence': the Complex_product_relationship defines a relationship where the relat-ing Complex_product is the preceding version and the related Complex_product is the following version.

Compositions

- description : String_select [0..1]The description specifies additional information about the Complex_product_relationship.

Associations

- related : Complex_product (ABS) [1]
  The related specifies the second of the two objects related by the Complex_product_relationship.

7.7.10.9 Class Component_placement

A Component_placement is the information pertaining to the placement of a Product_component, which is defined in its own Cartesian_coordinate_space, in the coordinate space of a reference Product_component.

Base Class

- PLM_object (ABS)

Attributes

- none

Compositions

- none

Associations

- reference_product_component : Product_component [1]
  The reference_product_component specifies the high level Product_component that is defined in the reference coordinate space. A Model_property_association shall be assigned to the reference_product_component to define this reference coordinate space.
- placement : Transformation_select [1]
  The placement specifies the Geometric_model_relationship_with_transformation or the Template_instance that defines the position of the 'placed_component' relatively to the 'reference_product_component'. In the case of Template_instance, the scale shall be omitted or set to 1.0.

7.7.10.10 Class Configuration

A Configuration is the association of a Class_condition_association or a Class_specification_association object with a design or with a process in order to define a valid usage of it in the context of a certain Product_class.

Base Class

- PLM_object (ABS)

Attributes

- configuration_type : String [1] The configuration_type specifies the valid usage of a Configuration object that is applied to the application object as configured_element. The following values shall be used:
  - 'design': The object referenced as 'configured_element' has to be designed and verified before it can actually be used in a given context. This context is specified by the Class_condition_association and Class_specification_association objects referenced as the 'is_solution_for'.
  - 'usage': The object referenced as the 'configured_element' is controlled by a Configuration. The Class_condition_association and Class_specification_association objects specify the us-age cases and are referenced as the 'is_solution_for'.
- inheritance_type : String [1] The inheritance_type specifies whether or not an inheritance scheme for the configuration in-formation in a hierarchical structure is applied to the application object referenced as the configured_element. The levels within such a hierarchy are defined through Product_structure_relationship objects or the attribute 'base_element' of Alternative_solution. The following values shall be used:
  - 'exception': No inheritance scheme is applicable and all required configuration information must be attached locally at the application object. The value indicates that the configuration in-formation may be inconsistent to the structural levels above it or that it is, on purpose, contradictory to it. Such a condition implies that an inheritance scheme shall not continue beyond this point in the product structure tree;
  - 'inherited': A scheme for inheritance of configuration information applies. The complete con-figuration information shall be collected from the different levels in the structure by evaluation of results. The results shall be evaluated using the logical AND to combine configuration informa-tion starting at the referenced configured_element and using the logical OR to combine alternatives. In addition, this evaluation shall consider related effectivity information. 'inherited' only applies for objects for which the same value of 'configuration_type' is defined;
  - 'local': No inheritance scheme is applicable and all required configuration information must be attached locally at the application object. Nevertheless, any potentially inherited configuration information of a higher level shall be consistent, i.e., be a subset of the locally defined configuration information.

Compositions

- none

Associations

- is_solution_for : Configured_specification_select [1]
  The is_solution_for specifies the characteristic or combination of characteristics for which the object referenced as the configured_element provides a solution or which is needed to control a process operation. These characteristics are defined by a

Class_specification_association and combinations of characteristics are defined by a Class_condition_association where the attribute 'condition type' is 'part usage'.

7.7.10.11 Class Dated_configuration

A Dated_configuration is a Manufacturing_configuration that applies onwards from a given date, or between a start and an end date.

Base Class

- Manufacturing_configuration (ABS)

Attributes

- start_date : String [1] The start_date specifies the first date when the Dated_configuration is valid.
- end_date : String [0..1] The end_date specifies the date and time when the validity of the 'configured_element' is not defined any longer.

Compositions

- none

Associations

- none

7.7.10.12 Class Descriptive_specification

A Descriptive_specification is a textual description of an object.

Base Class

- PLM_root_object (ABS)

Attributes

- id : String [0..1] The id specifies the identifier of the Descriptive_specification.

Compositions

- description : String_select [1] The description specifies the Descriptive_specification.
- document_assignment : Document_assignment [0..*]
  The document_assignment specifies the object that provides information for this Descriptive_specification.

Associations

- none

7.7.10.13 Class Design_constraint

A Design_constraint is a requirement that has to be considered in the design process of a Complex_product. This constraint may be geometry based.

Base Class

- PLM_root_object (ABS)

Attributes

- constraint_id : String [1] The constraint_id specifies the identifier of the Design_constraint.

Compositions

- design_constraint_relationship : Design_constraint_relationship [0..*]
  The design_constraint_relationship specifies the design_constraint_relationship that relates the first of the two Design_constraint objects.
- description : String_select [0..1]The description specifies additional information about the Design_constraint.
- name : String_select [0..1] The name specifies the word or group of words by which the Design_constraint is referred to.
- document_assignment : Document_assignment [0..*]
  The document_assignment specifies the object that provides information for this Design_constraint.
- simple_property_association : Simple_property_association (ABS) [0..*]
  The simple_property_association specifies the assigned simple property values.

Associations

- is_valid_for : Product_class [0..*]The is_valid_for specifies the set of Product_class objects that are affected by the Design_constraint.

7.7.10.14 Class Design_constraint_association

A Design_constraint_association is a mechanism to associate a Design_constraint with an object that is subject to the constraint indicated.

Base Class

- PLM_object (ABS)

Attributes

- none

Compositions

- name : String_select [0..1] The name specifies the word or group of words by which the Design_constraint_association is referred to.

Associations

- is_based_on : Design_constraint [1]The is_based_on specifies the Design_constraint that represents the constraint.

7.7.10.15 Class Design_constraint_relationship

A Design_constraint_relationship is a relationship between two Design_constraint objects.

Base Class

- PLM_object (ABS)

Attributes

- relation_type : String [1] The relation_type specifies the meaning of the relationship.

Compositions

- description : String_select [0..1] The description specifies additional information about the Design_constraint_relationship.

Associations

- related : Design_constraint [1] The related specifies the second of the two Design_constraint objects related by the Design_constraint_relationship.

7.7.10.16 Class Design_constraint_version

A Design_constraint_version is a particular version of a Design_constraint.

Base Class

- Design_constraint

Attributes

- version_id : String [1] The version_id specifies the identification of a particular version of a Design_constraint. The version_id shall be unique within the scope of a Design_constraint.

Compositions

- none

Associations

- none

7.7.10.17 Class Effectivity

An Effectivity is the identification of the valid use of an aspect of product data tracked by date or event.

Base Class

- PLM_root_object (ABS)

Attributes

- id : String [0..1] The id specifies the identifier of the Effectivity.
- version_id : String [0..1] The version_id specifies the identification of a particular version of the Effectivity.
- effectivity_context : String [0..1] The effectivity_context specifies the life cycle stage for which the Effectivity is valid.

Compositions

- effectivity_assignment : Effectivity_assignment [0..*]
  The effectivity_assignment specifies the effectivity_assignment which this Effectivity is assigned to.
- description : String_select [0..1]The description specifies additional information about the Effectivity.

Associations

- end_definition : Event_or_date_select [0..1]
  The end_definition specifies the end of the period. The bound specified by the end_definition is excluded from the interval of effectivity.
- start_definition : Event_or_date_select [0..1]
  The start_definition specifies the start of the period. The bound specified by the start_definition is included in the interval of effectivity.
- period : Duration [0..1] The period specifies the period of time in which the Effectivity is defined, either starting at the point in time specified by 'start_definition' or ending at the point in time specified by 'end_definition'. period shall be specified with a positive value.
- concerned_organization : Organization [0..*]
  The concerned_organization specifies the set of Organization objects in which the Effectivity is valid.

7.7.10.18 Class Effectivity_assignment

An Effectivity_assignment associates an Effectivity with the object whose effectivity is controlled by the associated Effectivity. The association of an Effectivity to product data does not imply any statement concerning the effectivity outside of the specified interval. The same applies in the absence of any assigned effectivity, i.e. no statement concerning the effectivity is implied.

Base Class

- PLM_object (ABS)

Attributes

- role : String [1]
  The role specifies the relationship between the Effectivity and the object that has an effectivity assigned to it. Where applicable the following values shall be used:
  - 'actual': The actual period during which the Effectivity lasted;
  - 'planned': The period associated with the Effectivity defines a planned period of time during which the associated object is or was supposed to be effective;
  - 'required': The associated object must be kept effective for this period.
- effectivity_indication : Boolean [1]
  The effectivity_indication specifies whether the assigned_effectivity defines a period of effectivity (value equal 'TRUE') or a period of ineffectivity (value equal 'FALSE') for the effective_element. In the first case, use of the effective_element is or was valid during the considered period.

Compositions

- none

Associations

- effective_element : Effective_element_select [1]
  The effective_element specifies the object that has an Effectivity assigned to it.

7.7.10.19 Class Final_solution

A Final_solution is the specification of a set of additional sensual characteristics that can be applied to an Item_instance that represents a neutral part in order to finalize its definition.

Base Class

- Alternative_solution

Attributes

- final_status : String [1] The final_status specifies the level of completion between the neutral part and the final part.

Compositions

- none

Associations

- final_specification : Final_definition_select [1..*]
  The final_specification specifies the means of finalization that is applied to the neutral part and which may be objects of type Descriptive_specification, Physical_instance, or Design_discipline_item_definition.

7.7.10.20 Class Instance_placement

An Instance_placement is the information pertaining to the placement of a Single_instance, which is defined in its own Cartesian_coordinate_space, in the coordinate space of a reference Product_component.

Base Class

- PLM_object (ABS)

Attributes

- none

Compositions

- none

Associations

- reference_product_component : Product_component [1]
  The reference_product_component specifies the Product_component that specifies indirectly the reference coordinate space. A Model_property_association shall be assigned to the reference_product_component to define this reference coordinate space.
- placement : Transformation_select [1]
  The placement specifies the Geometric_model_relationship_with_transformation or the

Template_instance that defines the position of the 'placed_instance' relatively to the 'reference_product_component'. In the case of Template_instance, the scale shall be omitted or set to 1.0.

7.7.10.21 Class Item_function_association

An Item_function_association is a mechanism to relate a Product_function and a Design_discipline_item_definition.

Base Class

- PLM_object (ABS)

Attributes

- association_type : String [1] The association_type specifies the kind of association.

Compositions

- description : String_select [0..1]The description specifies additional information about the Item_function_association.

Associations

- associated_function : Product_function [1]
  The associated_function specifies the associated Product_function.

7.7.10.22 Class Lot_configuration

A Lot_configuration is a Manufacturing_configuration that applies to a given production batch of the product that is related with the object referred to as 'is_solution_for'.

Base Class

- Manufacturing_configuration (ABS)

Attributes

- lot_id : String [1] The lot_id specifies the identification of the batch for which the Lot_configuration applies.
- lot_size : String [1] The lot_size specifies the size of the batch for which the Lot_configuration applies.

Compositions

- none

Associations

- none

7.7.10.23 Class Manufacturing_configuration (ABS)

A Manufacturing_configuration is the association of a Product_design with an Item_instance.

Base Class

- PLM_object (ABS)

Attributes

- none

Compositions

- none

Associations

- concerned_organization : Organization [0..*]
  The concerned_organization specifies the Organization in which the Manufacturing_configuration is valid. The case where the concerned_organization is an empty set means that the Manufacturing_configuration regards any organization that may consider the 'configured_element'.
- is_solution_for : Product_design [1]The is_solution_for specifies the design for which an Item_instance is configured.

7.7.10.24 Class Physical_instance

A Physical_instance is the denomination of a physically realized object. A Physical_instance may be identified by a serial number. A lot id may be provided additionally to the serial number.

Base Class

- PLM_root_object (ABS)

Attributes

- serial_number : String [0..1] The serial_number is an identifier that distinguishes one Physical_instance from another.
- lot_id : String [0..1] The lot_id specifies the identifier of the lot the Physical_instance is part of.
- inventory_number : String [0..1]The inventory_number specifies an alphanumerical string to identify an item in the detailed list of articles, such as goods and chattels, found in the possession of a person or enterprise.

Compositions

- physical_instance_test_result : Physical_instance_test_result [0..*]
  The physical_instance_test_result specifies the physical_instance_test_result for which this Physical_instance was the subject of the test activity.
- description : String_select [0..1]The description specifies additional information about the Physical_instance.
- physical_assembly_relationship : Physical_assembly_relationship [0..*]
  The physical_assembly_relationship specifies the physical_assembly_relationship for which this Physical_instance serves as the assembly in the physical structure.
- document_assignment : Document_assignment [0..*]
  The document_assignment specifies the object that provides information for this Physical_instance.
- alias_identification : Alias_identification [0..*]
  The Alias_identification specifies the Alias_identification that is applied to this Physical_instance.

- simple_property_association : Simple_property_association (ABS) [0..*]
  The simple_property_association specifies the assigned simple property values.

Associations

- is_realization_of : Physical_instance_definition_select [0..1]
  The is_realization_of specifies the Product_identification or the Design_discipline_item_definition that collects the information defining the Physical_instance.

7.7.10.25 Class Physical_instance_test_result

A Physical_instance_test_result is a mechanism to associate a Physical_instance with measurements made on this Physical_instance.

Base Class

- PLM_object (ABS)

Attributes

- id : String [1] The id specifies the identifier of the Physical_instance_test_result.

Compositions

- description : String_select [0..1]The description specifies additional information about the Physical_instance_test_result.
- Document_assignment : Document_assignment [0..*]
  The document_assignment specifies the object that provides information for this Physical_instance_test_result.

Associations

- test_result : Property_value_representation [0..*]
  The test_result specifies the characteristics that were determined by the performed test.
- test_activity : Test_activity_select [0..1]
  The test_activity specifies the Activity or the Process_operation_occurrence that has lead to the test result.

7.7.10.26 Class Product_class

A Product_class is the identification of a set of similar products to be offered to the market. Product_class objects that are related to each other by a Product_class_relationship do not inherit any characteristics from each other.

Base Class

- PLM_root_object (ABS)

Attributes

- id : String [1] The id specifies the identifier of the Product_class that shall be unique.
- level_type : String [0..1] The level_type specifies the level or category of this Product_class in a hierarchical structure of Product_class objects. The level_type shall only be used if and only if the

level_type is specified in the the context of the unit of functionality 'specification_control' (UoF S7).

- version_id : String [0..1] The version_id specifies the identification of a particular version of a Product_class.

Compositions

- product_identification : Product_identification [0..*]
  The product_identification specifies the product_identification of the product that belongs to this Product_class.
- description : String_select [0..1]The description specifies additional information about the Product_class.
- name : String_select [0..1] The name specifies the word or group of words by which the Product_class is referred to.
- class_structure_relationship : Class_structure_relationship [0..*]
  The class_structure_relationship specifies the class_structure_relationship that relates this Product_class.
- class_specification_association : Class_specification_association [0..*]
  The class_specification_association specifies the class_specification_association that is valid for this Product_class.
- class_inclusion_association : Class_inclusion_association [0..*]
  The class_inclusion_association specifies the class_inclusion_association that is valid for this Product_class.
- class_condition_association : Class_condition_association [0..*]
  The class_condition_association specifies the class_condition_association that is valid for this Product_class.
- class_category_association : Class_category_association [0..*]
  The class_category_association specifies the class_category_association that is valid for this Product_class.
- document_assignment : Document_assignment [0..*]
  The document_assignment specifies the object that provides information for this Product_class.
- alias_identification : Alias_identification [0..*]
  The Alias_identification specifies the Alias_identification that is applied to this Product_class.
- simple_property_association : Simple_property_association (ABS) [0..*]
  The simple_property_association specifies the assigned simple property values.

Associations

- none

7.7.10.27 Class Product_component

A Product_component is an element in a conceptual product structure.

Base Class

- Complex_product (ABS)

Attributes

- instance_required : Boolean [1]The instance_required specifies if the existence of a corresponding Item_instance is required for the various Alternative_solution objects of that Product_component. A value of 'true' indicates that a corresponding Item_instance is required.

Compositions

- description : String_select [0..1]The description specifies additional information about the Product_component.
- name : String_select [0..1] The name specifies the word or group of words by which the Product_component is referred to.
- configuration : Configuration [0..*]
  The configuration specifies the configuration that controls this Product_component for its valid usage.
- component_placement : Component_placement [0..*]
  The component_placement specifies the component_placement that is positioned with respect to this Product_component.

Associations

- is_relevant_for : Application_context [0..*]
  The is_relevant_for specifies the Application_context objects in which the Product_component has to be considered.
- is_influenced_by : Class_category_association [0..*]
  The is_influenced_by specifies the Specification_category objects that impact the design of a solution for the Product_component in the context of the Product_class objects that are referred to by the Class_category_association objects.

7.7.10.28 Class Product_design

A Product_design is a mechanism to associate an Item_version with its corresponding Product_identification.

Base Class

- PLM_object (ABS)

Attributes

- none

Compositions

- none

Associations

- product : Product_identification [1]
  The product specifies the Product_identification that represents the requirements.

7.7.10.29 Class Product_function

A Product_function is a behaviour or an action expected from a product.

Base Class

- Complex_product (ABS)

Attributes

- none

Compositions

- description : String_select [0..1]The description specifies additional information about the Product_function.
- name : String_select [0..1] The name specifies the word or group of words by which the Product_function is referred to.
- configuration : Configuration [0..*]
  The configuration specifies the configuration that controls this Product_function for its valid usage.

Associations

- is_relevant_for : Application_context [0..*]
  The is_relevant_for specifies the Application_context objects in which the Product_function has to be considered.

7.7.10.30 Class Product_identification

A Product_identification identifies a manufacturable object, or expected as so. A Product_identification is defined with respect to the Product_class it is a member of.

Base Class

- PLM_object (ABS)

Attributes

- version_id : String [0..1] The version_id specifies the identification of a particular version of a Product_identification.
- id : String [1] The id specifies the identifier of the Product_identification.

Compositions

- description : String_select [0..1]The description specifies additional information about the Product_identification.
- name : String_select [0..1] The name specifies the word or group of words by which the Product_identification is referred to.
- item_instance : Item_instance (ABS) [0..*]
  The item_instance specifies the item_instance for which this Product_identification serves as a definition.
- document_assignment : Document_assignment [0..*]
  The document_assignment specifies the object that provides information for this Product_identification.
- simple_property_association : Simple_property_association (ABS) [0..*]
  The simple_property_association specifies the assigned simple property values.

Associations

- none

7.7.10.31 Class Product_specification

A Product_specification is a Product_identification for which one or more additional Specification objects enhance the characterization provided for the associated Product_class.

Base Class

- Product_identification

Attributes

- none

Compositions

- none

Associations

- defining_specification : Specification [1..*]
  The defining_specification specifies the set of Specification objects necessary to discriminate the Product_specification within its Product_class.

7.7.10.32 Class Product_structure_relationship

A Product_structure_relationship is an association between a Complex_product and a Product_constituent, in which the Product_constituent is a functional, logical, or physical component or a realization of the Complex_product.

Base Class

- PLM_object (ABS)

Attributes

- relation_type : String [1]
  The relation_type specifies the meaning of the relationship. Where applicable the following values shall be used:
  - 'decomposition': The related Product_constituent is one of potentially more components of the relating Complex_product. This relation type shall only be used for Complex_product and Product_constituent of the same type;
  - 'functionality': The related Product_constituent is an element of the functional structure of the relating Complex_product. This relation type shall only be used with a Complex_product of type Alternative_solution or Product_component and with a Product_constituent of type Product_function;
  - 'occurrence': The related Product_constituent is an occurrence defined by the relating Complex_product. This relation type shall only be used if related Product_constituent is of type Product_component;
  - 'realization': The related Product_constituent is a means for fulfilling, either partially or fully, the requirements identified with the relating Complex_product. This relation type shall be used only when the Complex_product and the Product_constituent are of different types;

- 'specialization': The related Product_constituent fulfils the requirements of the relating Complex_product in a more specific way than defined for the relating Complex_product. This relation type shall only be used for Product_constituent and Complex_product of the same type.

Compositions

- description : String_select [0..1]The description specifies additional information about the Product_structure_relationship.
- document_assignment : Document_assignment [0..*]
  The document_assignment specifies the object that provides information for this Product_structure_relationship.
- simple_property_association : Simple_property_association (ABS) [0..*]
  The simple_property_association specifies the assigned simple property values.

Associations

- related : Product_constituent_select [1]The related specifies the Product_constituent that is a functional, logical, or physical component or a realization of the relating Complex_product.

7.7.10.33 Class Serial_configuration

A Serial_configuration is a Manufacturing_configuration that applies onwards from a given serial number of the product that is considered within the object referred to as 'is_solution_for'.

Base Class

- Manufacturing_configuration (ABS)

Attributes

- serial_start_number : String [1]The serial_start_number specifies the serial number of that instance of the product that is the first instance for which the Serial_configuration applies.
- serial_end_number : String [0..1]The serial_end_number specifies the serial number of that instance of the product that is the last instance for which the Serial_configuration applies.

Compositions

- none

Associations

- none

7.7.10.34 Class Specification

A Specification is a characteristic of a product. A Specification discriminates one product from other members of the same Product_class. A Specification refers to a Specification_category that completes the semantics of the Specification.

Base Class

- PLM_root_object (ABS)

Attributes

- id : String [1] The id specifies the identifier of the Specification that shall be unique within the scope of a Specification_category.
- version_id : String [0..1] The version_id specifies the identification of a particular version of a Specification.
- package : Boolean [1] The package specifies whether this Specification represents a package of Specification objects or not. Such a Specification combines those Specification objects that shall be offered to the market as a set. In the case where package is 'true', there shall be exactly one Specification_inclusion per Product_class considered, that refers to this Specification as 'if_condition'. The Specification objects that are members of the package, shall be specified as included_specification.

Compositions

- specification_inclusion : Specification_inclusion [0..*]
  The specification_inclusion specifies the specification_inclusion for which this Specification serves as the condition for the inclusion.
- description : String_select [0..1]The description specifies additional information about the Specification.
- name : String_select [0..1] The name specifies the word or group of words by which the Specification is referred to.
- alias_identification : Alias_identification [0..*]
  The Alias_identification specifies the Alias_identification that is applied to this Specification.
- document_assignment : Document_assignment [0..*]
  The document_assignment specifies the object that provides information for this Specification.

Associations

- category : Specification_category [1]
  The category specifies the Specification_category that completes the semantics of the Specification.

7.7.10.35 Class Specification_category

A Specification_category is the definition of a set of Specification objects serving the same purpose.

Base Class

- PLM_root_object (ABS)

Attributes

- implicit_exclusive_condition : Boolean [1]
  The implicit_exclusive_condition specifies whether the Specification objects within the Specification_category are mutually exclusive for the production of one particular product. A value of 'true' indicates that the referenced objects are mutually exclusive for the production of the particular product.
- id : String [1] The id specifies the identifier of the Specification_category that shall be unique.

Compositions

- specification_category_hierarchy : Specification_category_hierarchy [0..*]
  The specification_category_hierarchy specifies the specification_category_hierarchy for which this Specification_category is the higher level.
- description : String_select [1] The description specifies information about the Specification_category.
- document_assignment : Document_assignment [0..*]
  The document_assignment specifies the object that provides information for this Specification_category.
- alias_identification : Alias_identification [0..*]
  The Alias_identification specifies the Alias_identification that is applied to this Specification_category.

Associations

- none

7.7.10.36 Class Specification_category_hierarchy

A Specification_category_hierarchy is used to build up hierarchical structures of Specification_category objects.

Base Class

- PLM_object (ABS)

Attributes

- none

Compositions

- none

Associations

- sub_category : Specification_category [1]
  The sub_category is the lower level of Specification_category in Specification_category_hierarchy.

7.7.10.37 Class Specification_expression

A Specification_expression is a combination of Specification objects formed by Boolean operations.

Base Class

- PLM_root_object (ABS)

Attributes

- operation : String [1] The operation specifies the kind of Boolean operation. Four kinds of operations are permitted:
  - 'and': All of the identified Specification objects shall be used;
  - 'or': A subset or all of the identified Specification objects shall be used;

- 'oneof': Exactly one of the identified Specification objects shall be used;
- 'not': The identified Specification shall not be used.

- id : String [0..1] The id specifies the identifier of the Specification_expression.

Compositions

- specification_inclusion : Specification_inclusion [0..*]
  The specification_inclusion specifies the specification_inclusion for which this Specification_expression serves as the condition for the inclusion.
- description : String_select [0..1]
  The description specifies additional information about the Specification_expression.

Associations

- operand : Specification_operand_select [1..*]
  The operand specifies the operands of the Boolean operation that are either Specification objects or other Specification_expression objects.

7.7.10.38 Class Specification_inclusion

A Specification_inclusion is the representation of the statement that specifies that the application of a Specification or of a Specification_expression implies the inclusion of an additional Specification or Specification_expression.

Base Class

- PLM_object (ABS)

Attributes

- id : String [0..1] The id specifies the identifier of the Specification_inclusion.

Compositions

- description : String_select [0..1]The description specifies additional information about the Specification_inclusion.

Associations

- included_specification : Specification_operand_select [1]
  The included_specification specifies the Specification or the Specification_expression objects that are to be included. The included_specification shall not reference a Specification_expression with an operation of type 'or' or 'oneof', except for negating expressions, i.e., as participants in an expression preceded by a 'not' operator. Expressions of operator 'not' shall not be nested within each other.

7.7.10.39 Class Supplier_solution

A Supplier_solution is an alternative solution provided by a particular supplier.

Base Class

- Alternative_solution

Attributes

- probability_rate : String [0..1] The probability_rate specifies the share that is assigned to the supplier in the context of the base element.

Compositions

- none

Associations

- supplier : Organization [1] The supplier specifies the Organization that acts as supplier for the Supplier_solution.

7.7.10.40 Class Technical_solution

A Technical_solution is an alternative solution where the functional requirements are fulfilled in a certain technical way.

Base Class

- Alternative_solution

Attributes

- none

Compositions

- description : String_select [1] The description specifies additional information about the Technical_solution.

Associations

- none

7.7.10.41 Interfaces

Interface Complex_product_select

This empty interface is realized by the following classes:

- Product_function
- Product_component
- Alternative_solution

Interface Configured_specification_select

This empty interface is realized by the following classes:

- Class_specification_association
- Class_condition_association

Interface Effective_element_select

This empty interface is realized by the following classes:

- Classification_system

- Specification_inclusion
- Specification_expression
- Specification_category
- Specification
- Product_structure_relationship
- Product_identification
- Product_class
- Design_constraint
- Configuration
- Complex_product_relationship
- Complex_product (ABS)
- Class_structure_relationship
- Class_specification_association
- Class_inclusion_association
- Class_condition_association
- Class_category_association
- Document_version
- Document_representation (ABS)
- Document_file (ABS)
- Document
- Item_version
- Item_definition_relationship (ABS)
- Item
- Item_instance_relationship (ABS)
- Item_instance (ABS)
- Item_definition_instance_relationship (ABS)
- Process_plan
- Process_operation_resource_assignment
- Process_operation_occurrence_relationship
- Process_operation_occurrence
- Process_operation_definition_relationship
- Process_operation_definition
- Property_value_association (ABS)
- Property (ABS)

- Material
- Geometric_model

Interface Final_definition_select

This empty interface is realized by the following classes:

- Physical_instance
- Descriptive_specification
- Design_discipline_item_definition

Interface Physical_instance_definition_select

This empty interface is realized by the following classes:

- Product_identification
- Design_discipline_item_definition

Interface Product_function_component_select

This empty interface is realized by the following classes:

- Product_function
- Product_component

Interface Specification_operand_select

This empty interface is realized by the following classes:

- Specification_expression
- Specification

Interface Test_activity_select

This empty interface is realized by the following classes:

- Activity
- Process_operation_occurrence

7.7.11 Package Change_and_work_management

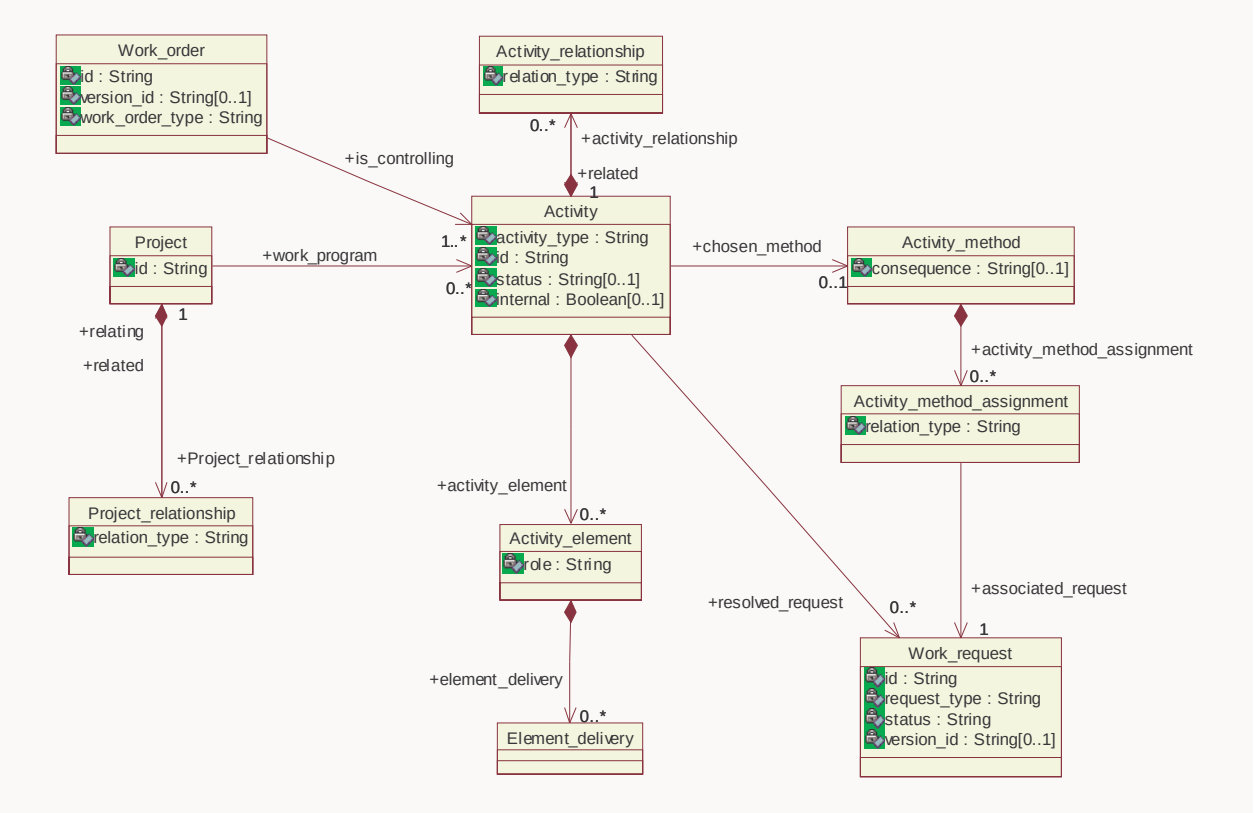


Figure 118 - Change management

7.7.11.1 Class Activity

An Activity is the fact of achieving or accomplishing an action.

Base Class

- PLM_root_object (ABS)

Attributes

- activity_type : String [1] — The activity_type specifies the purpose of the Activity. Where applicable the following values shall be used:
  - 'amendment': An Activity to add information to product data;
  - 'analysis': An Activity to determine the behaviour of an element under certain physical circumstances;
  - 'cancellation': An Activity to delete an element from the bill of material or to cancel the whole bill of material;
  - 'delivery change': An Activity to change the delivery schedule of an element;
  - 'design change': An Activity to change the design of an item or an assembly; this might include changes to the geometry or to properties of the object;

- 'design': An Activity concerning the development of a design of an item;
- 'mock-up creation': An Activity to create an experimental model or replica of an item;
- 'prototype building': An Activity to manufacture a preliminary version of an item;
- 'rectification': An Activity to correct the data, documentation or structure associated with an item;
- 'restructuring': An Activity to create a new structure or position within a bill of material without changing the data associated with the items in the bill of material;
- 'spare part creation': An Activity to design a spare part or to classify an item as a spare part;
- 'stop notice': An Activity to stop the manufacturing process of an item;
- 'testing': An Activity to test an item;
- 'work definition': An Activity to manage several sub-activities related to this Activity by an Activity_relationship with a 'relation_type' of value 'decomposition'.

- id : String [1] The id specifies the identifier of the Activity.
- status : String [0..1] The status specifies the level of completion of the Activity.
- internal : Boolean [0..1] The internal specifies whether the activity is carried out within the organization that initiated the activity. A value of 'true' indicates that the activity is carried out within this particular organization.

Compositions

- activity_relationship : Activity_relationship [0..*]
  The Activity_relationship specifies the Activity_relationship that relates the first of the two Activity objects.
- activity_element : Activity_element [0..*]
  The Activity_element specifies the Activity_element that belongs to this Activity.
- description : String_select [0..1]The description specifies additional information about the Activity.
- document_assignment : Document_assignment [0..*]
  The document_assignment specifies the object that provides information for this Activity.
- simple_property_association : Simple_property_association (ABS) [0..*]
  The simple_property_association specifies the assigned simple property values.

Associations

- chosen_method : Activity_method [0..1]
  The chosen_method specifies the Activity_method used to carry out the Activity.
- actual_start_date : Date_time [0..1]
  The actual_start_date specifies the date when the Activity actually started.
- planned_start_date : Event_or_date_select [0..1]
  The planned_start_date specifies the date when the Activity is or was supposed to be started.
- planned_end_date : Period_or_date_select [0..1]
  The planned_end_date specifies the date when the Activity is or was supposed to be finished.
- actual_end_date : Date_time [0..1]
  The actual_end_date specifies the date when the Activity actually finished.

- requestor : Date_and_person_organization [0..1]
  The requestor specifies the Person or Organization that requested the Activity and the date the request was submitted.
- supplying_organization : Organization [0..*]
  The supplying_organization specifies the set of Organization objects that carry out the work.
- concerned_organization : Organization [0..*]
  The concerned_organization specifies the set of Organization objects that are affected by the result of the Activity.
- resolved_request : Work_request [0..*]
  The resolved_request specifies the set of Work_request objects that are resolved by the Activity.

7.7.11.2 Class Activity_element

An Activity_element is an item of work that is part of an Activity.

Base Class

- PLM_object (ABS)

Attributes

- role : String [1]
  The role specifies the function that is performed by the Activity_element in the context of the concerned Activity. Where applicable the following values shall be used:
  - 'control': The referenced element is an object that has immediate influence on the Activity performed;
  - 'input': The referenced element serves as initial data for the Activity;
  - 'output': The referenced element is a result of the Activity.

Compositions

- element_delivery : Element_delivery [0..*]
  The Element_delivery specifies the Element_delivery which this Activity_element is subject to.
- document_assignment : Document_assignment [0..*]
  The document_assignment specifies the object that provides information for this Activity_element.

Associations

- element : Activity_element_select [1]The element specifies the piece of product data that is under work.

7.7.11.3 Class Activity_method

An Activity_method is a procedure that may be used to solve a request.

Base Class

- PLM_root_object (ABS)

Attributes

- consequence : String [0..1] The consequence specifies the expected positive or negative effects of the application of a particular Activity_method.

Compositions

- activity_method_assignment : Activity_method_assignment [0..*]
  The activity_method_assignment specifies the activity_method_assignment for which this activity_method is recommended or shall not be chosen.
- name : String_select [1] The name specifies the word or group of words by which the Activity_method is referred to.
- document_assignment : Document_assignment [0..*]
  The document_assignment specifies the object that provides information for this Activity_method.

Associations

- description : String_select [1] The description specifies additional information that defines the Activity_method in terms of either the nature of the Activity_method or in terms of the specific procedure steps required to implement it.

7.7.11.4 Class Activity_method_assignment

An Activity_method_assignment is an object that associates an Activity_method with a Work_request. The associated Activity_method serves as a recommended or non-recommended method to resolve the tasks specified in the Work_request.

Base Class

- PLM_object (ABS)

Attributes

- relation_type : String [1] The relation_type specifies whether the specified Activity_method may be used or not. Where applicable the following values shall be used:
  - 'non recommended method': The specified Activity_method shall not be used in order to accomplish the specified Work_request;
  - 'recommended method': The specified Activity_method may be used in order to accomplish the specified Work_request.

Compositions

- simple_property_association : Simple_property_association (ABS) [0..*]The simple_property_association specifies the assigned simple property values.

Associations

- associated_request : Work_request [1]
  The associated_request identifies the Work_request that the recommended or non-recommended method applies to.

7.7.11.5 Class Activity_relationship

An Activity_relationship is a relationship between two Activity objects.

Base Class

- PLM_object (ABS)

Attributes

- relation_type : String [1] The relation_type specifies the meaning of the relationship. Where applicable the following values shall be used:
  - 'alternative': The application object defines a relationship where the related Activity may be used alternatively instead of the relating Activity;
  - 'decomposition': The application object defines a relationship where the related Activity is one of potentially more sub-activities into which the relating Activity is broken down;
  - 'derivation': The application object defines a relationship where the related Activity is derived from the relating Activity
  - 'exclusiveness': The application object defines a relationship where the relating and the related Activity shall not have any overlap in time of execution;
  - 'precedence': The application object defines a relationship where the related Activity has higher priority than the relating Activity;
  - 'sequence': The application object defines a relationship where the relating Activity shall be completed before the related Activity starts;
  - 'simultaneity': The application object defines a relationship that establishes that both the relating and related Activity are considered as occurring during the same time period or shall be performed together in order to ensure consistency and enhance efficiency.

Compositions

- description : String_select [0..1]The description specifies additional information about the Activity_relationship.

Associations

- related : Activity [1] The related specifies the second of the two Activity objects related by an Activity_relationship.

7.7.11.6 Class Change

A Change is a mechanism to collect the Model_change objects and the Property_change objects that describe the differences between the two objects referenced by the specified relationship object.

Base Class

- PLM_object (ABS)

Attributes

- none

Compositions

- description : String_select [0..1]The description specifies additional information about the Change.

- document_assignment : Document_assignment [0..*]
  The document_assignment specifies the object that provides information for this Change.

Associations

- none

7.7.11.7 Class Element_delivery

An Element_delivery is the specification of the expected delivery of an Activity_element.

Base Class

- PLM_object (ABS)

Attributes

- none

Compositions

- none

Associations

- quantity : Value_with_unit (ABS) [1]
  The quantity specifies the number of objects referred by the Activity_element to be delivered.
- destination : Organization [1] The destination specifies the Organization the Activity_element is to be delivered to.

7.7.11.8 Class Project

A Project is an identified program of work.

Base Class

- PLM_root_object (ABS)

Attributes

- id : String [1] The id specifies the identifier of the Project.

Compositions

- Project_relationship : Project_relationship [0..*]
  The Project_relationship specifies the Project_relationship that relates the first of the two Project objects.
- description : String_select [0..1]The description specifies additional information about the Project.
- name : String_select [1] The name specifies the word or group of words by which the Project is referred to.
- document_assignment : Document_assignment [0..*]The document_assignment specifies the object that provides information for this Project.

Associations

- planned_end_date : Period_or_date_select [0..1]
  The planned_end_date specifies either the date when the Project is or was supposed to be finished or the planned duration of the Project.
- work_program : Activity [0..*] The work_program specifies the Activity objects that are carried out within the Project.
- planned_start_date : Event_or_date_select [0..1]The planned_start_date specifies the date when the Project is or was supposed to be started.
- actual_end_date : Date_time [0..1]The actual_end_date specifies the date when the Project was actually finished.
- actual_start_date : Date_time [0..1]The actual_start_date specifies the date when the Project was actually started.

7.7.11.9 Class Project_assignment

A Project_assignment is a relationship between a Project and the objects the work carried out by that project is applied to.

Base Class

- PLM_object (ABS)

Attributes

- role : String [1] The role specifies the meaning of the relationship.

Compositions

none.

Associations

- is_applied_to : Project_information_select [0..*]The is_applied_to specifies the set of objects that the work carried out by a Project applies to.
- project : Project[1] The project this relationship is assigned to.

7.7.11.10 Class Project_relationship

A Project_relationship is a relationship between two Project objects.

Base Class

- PLM_object (ABS)

Attributes

- relation_type : String [1] The relation_type specifies the meaning of the relationship. Where applicable the following values shall be used:
  - 'decomposition': The application object defines a relationship where the related Project is one of potentially more components into which the relating Project is broken down;
  - 'dependency': The related Project is dependent upon the relating Project;
  - 'sequence': The application object defines a relationship where the relating Project shall be completed before the related Project starts;
  - 'succession': The related Project is the successor of the relating Project.

Compositions

- description : String_select [0..1]The description specifies additional information about the Project_relationship.

Associations

- related : Project [1] The related specifies the second of the two Project objects related by a Project_relationship.

7.7.11.11 Class Work_order

A Work_order is the authorization for one or more Activity objects to be performed.

Base Class

- PLM_root_object (ABS)

Attributes

- id : String [1] The id specifies the identifier of the Work_order.
- version_id : String [0..1] The version_id specifies the identification of a particular version of a Work_order.
- work_order_type : String [1] The work_order_type specifies the kind of the Work_order. Where applicable the following values shall be used:
  - 'design deviation permit': An authorization for a deviation from the approved design data;
  - 'design release': An authorization for the design of a product or of an item or to create a bill of material;
  - 'management resolution': An authorization by a committee, such as the board of directors, to design or change an item;
  - 'manufacturing release': An authorization for the manufacturing process of a product or of an item;
  - 'production deviation permit': An authorization for a deviation from the approved manufacturing process.

Compositions

- description : String_select [0..1]The description specifies additional information about the Work_order.
- document_assignment : Document_assignment [0..*]
  The document_assignment specifies the object that provides information for this Work_order.

Associations

- is_controlling : Activity [1..*] The is_controlling specifies the Activity objects that are controlled by this particular Work_order.

7.7.11.12 Class Work_request

A Work_request is the solicitation for some work to be done.

Base Class

- PLM_root_object (ABS)

Attributes

- id : String [1] — The id specifies the identifier of the Work_request.
- request_type : String [1] — The request_type specifies the intention of the Work_request. Where applicable the following values shall be used:
  - 'change of standard': A request to translate a change to a standard into action;
  - 'cost reduction': A request aimed at reducing the engineering and manufacturing costs of an item;
  - 'customer rejection': A request resulting from a rejection by a customer;
  - 'customer request': A request for an activity that is necessary to solve the request of a customer;
  - 'durability improvement': A request aimed at extending the life time of an item;
  - 'government regulation': A request resulting from legal requirements;
  - 'procurement alignment': A request to adjust the purchasing process of different items;
  - 'production alignment': A request to adjust the manufacturing process of different items;
  - 'production relief': A request aimed at achieving a simpler assembly and production process;
  - 'production requirement': A request for an activity that is necessary from a production point of view;
  - 'quality improvement': A request aimed at increasing the quality of an item;
  - 'security reason': A request for an activity that is necessary from a security point of view;
  - 'standardization': A request to unify variants of an item;
  - 'supplier request': A request for an activity necessary to solve the request of a supplier;
  - 'technical improvement': A request aimed at improving the technical aspects of an item;
  - 'tool improvement': A request aimed at increasing the useful life of a tool.
- status : String [1] — The status specifies the stage of the Work_request. Where applicable the following values shall be used:
  - 'in work': The request is being developed;
  - 'issued': The request has been completed and reviewed, and immediate action takes place;
  - 'proposed': The request has been completed and is awaiting review and authorization;
  - 'resolved': The request is resolved; the actions as defined by the request have been completed and no further work is required.
- version_id : String [0..1] — The version_id specifies the identification of a particular version of a Work_request.

Compositions

- description : String_select [0..1]
  The description specifies additional information about the Work_request.
- document_assignment : Document_assignment [0..*]
  The document_assignment specifies the object that provides information for this Work_request.

Associations

- notified_person : Date_and_person_organization [1..*]
  The notified_person specifies the personnel that shall be informed about the Work_request and the date when the personnel or organization shall be informed.
- scope : Activity_element_select [0..*]
  The scope specifies the objects that are subject to the Work_request.

- requestor : Date_and_person_organization [1]
  The requestor specifies the person or organization who issued the Work_request and the date when this person or organization issued the Work_request.

7.7.11.13 Class Work_request_relationship

A Work_request_relationship is the relationship between two Work_request objects.

Base Class

- PLM_object (ABS)

Attributes

- description : String [1] — The description specifies additional information about the Work_request_relationship.
- relating : Work_request — The relating specifies the first of the two Work_Request objects related by an Work_request_relationship.
- related : Work_request — The related specifies the second of the two Work_Request objects related by an Work_request_relationship
- relation_type : String [1] — The relation_type specifies the intention of the Work_request. Where applicable the following values shall be used:

7.7.11.14 Interfaces

Interface Activity_element_select

This empty interface is realized by the following classes:

- Activity_method
- Specification_inclusion
- Specification_expression
- Specification_category
- Specification
- Product_structure_relationship
- Product_identification
- Product_class
- Physical_instance
- Manufacturing_configuration (ABS)
- Design_constraint
- Configuration
- Complex_product (ABS)
- Class_structure_relationship
- Class_specification_association
- Class_inclusion_association

- Class_condition_association
- Class_category_association
- Document_version
- Document_representation (ABS)
- Document_file (ABS)
- Document
- Item_version
- Item_definition_relationship (ABS)
- Item
- Design_discipline_item_definition
- Physical_assembly_relationship
- Item_instance_relationship (ABS)
- Item_instance (ABS)
- Item_definition_instance_relationship (ABS)
- Process_plan
- Process_operation_occurrence
- Process_operation_definition
- Property_value_association (ABS)
- Property (ABS)
- Material
- Geometric_model

Interface Project_information_select

This empty interface is realized by the following Classes:

- Product_identification
- Product_class
- Physical_instance
- Complex_product (ABS)
- Document_version
- Document
- Item_version
- Item

7.7.12 Package Process_planning

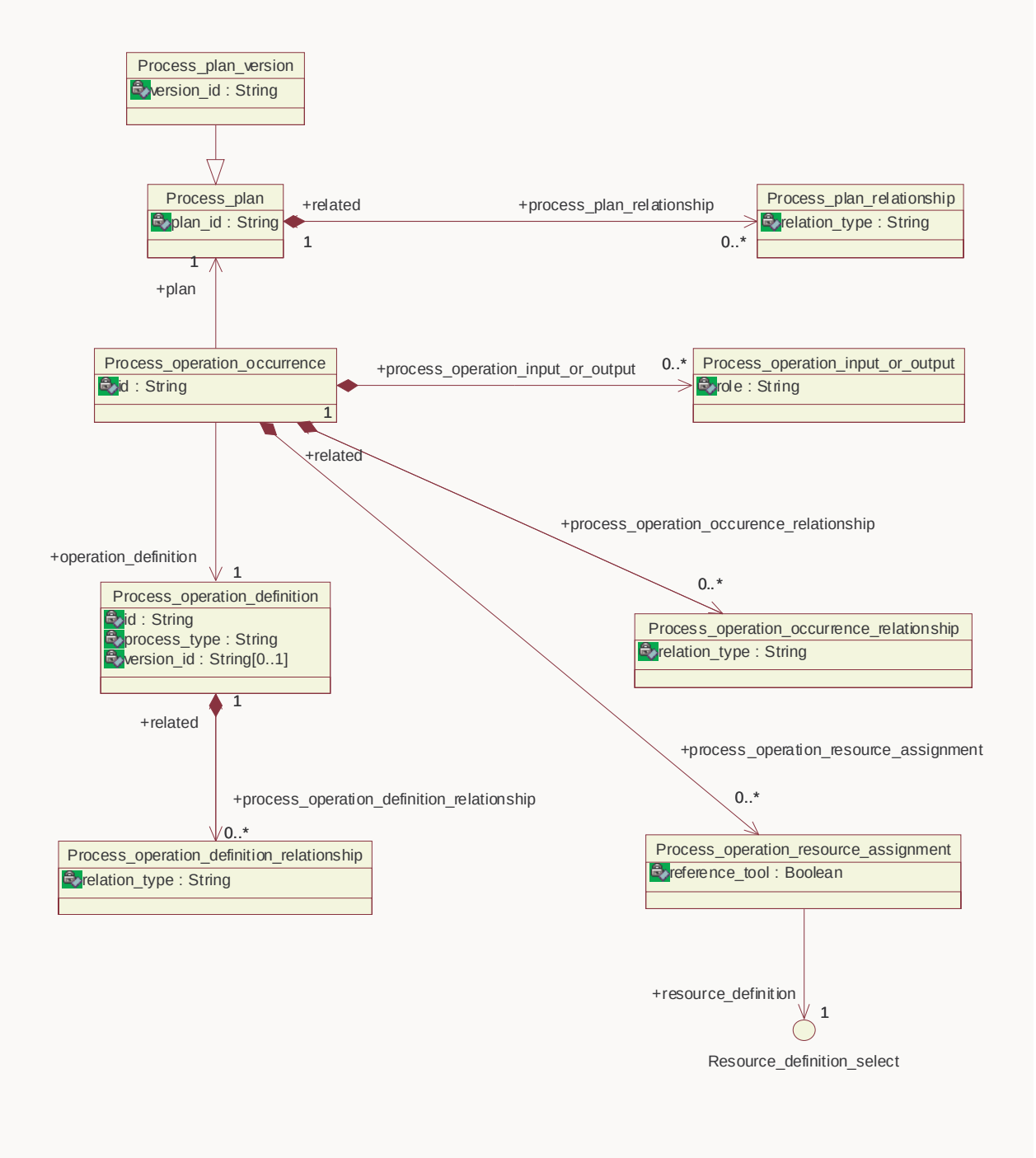


Figure 119 - Process planning

7.7.12.1 Class Process_operation_definition

A Process_operation_definition is the specification of an activity that may be included in a Process_plan. A Process_operation_definition characterizes a manufacturing or control operation.

Base Class

- PLM_root_object (ABS)

Attributes

- id : String [1] The id specifies the identifier of the Process_operation_definition that shall be unique within the scope of the associated Process_plan_version.
- process_type : String [1] The process_type specifies the type of the Process_operation_definition.
- version_id : String [0..1] The version_id specifies the identification of a particular version of a Process_operation_definition.

Compositions

- process_operation_definition_relationship : Process_operation_definition_relationship [0..*]
  The process_operation_definition_relationship specifies the process_operation_definition_relationship that relates the first of the two Process_operation_definition objects.
- description : String_select [0..1]The description specifies additional information about the Process_operation_definition.
- name : String_select [0..1] The name specifies the word or group of words by which the Process_operation_definition is referred to.
- simple_property_association : Simple_property_association (ABS) [0..*]
  The simple_property_association specifies the assigned simple property values.

Associations

- none

7.7.12.2 Class Process_operation_definition_relationship

A Process_operation_definition_relationship is a relationship between two Process_operation_definition objects.

Base Class

- PLM_object (ABS)

Attributes

- relation_type : String [1] The relation_type specifies the meaning of the relationship. Where applicable the following values shall be used:
  - 'alternative': The application object defines a relationship where the related Process_operation_definition may be used alternatively instead of the relating Process_operation_definition;
  - 'substitution': The application object defines a relationship where the related Process_operation_definition replaces the relating Process_operation_definition;
  - 'version association': The application object defines a relationship where the related

Process_operation_definition is a version of the relating Process_operation_definition. In this case, only the related Process_operation_definition shall specify a version_id.;
- 'version sequence': The application object defines a relationship where the relating Process_operation_definition is the preceding version and the related Process_operation_definition is the following version. In this case, both Process_operation_definition objects shall specify a version_id.

Compositions

- none

Associations

- related : Process_operation_definition [1]
  The related specifies the second of the two objects related by the Process_operation_definition_relationship.

7.7.12.3 Class Process_operation_input_or_output

A Process_operation_input_or_output is the input or expected result of a Process_operation_definition.

Base Class

- PLM_object (ABS)

Attributes

- role : String [1] The role specifies whether the identified element plays the role of an input or an output for the operation.

Compositions

- description : String_select [0..1]The description specifies additional information about the Process_operation_input_or_output.

Associations

- concerned_shape : Shape_element [0..*]The concerned_shape specifies the set of Shape_element objects that are affected by the Process_operation_occurrence.
- placement : Transformation (ABS) [0..1]
  The placement specifies the geometrical Transformation between the local coordinate system of the element acting as Process_operation_input_or_output, and the reference coordinate system. The reference coordinate system is either the coordinate system of the reference tool, if present, for the concerned Process_operation_occurrence or, if no reference tool is present, the coordinate system of the Process_operation_occurrence itself.
- element : Process_operation_input_or_output_select [1]
  The element specifies the element that plays the role of the input or the output for the operation.

7.7.12.4 Class Process_operation_occurrence

A Process_operation_occurrence is the usage of a Process_operation_definition in a Process_plan. This association states that the Process_operation_definition is part of the Process_plan.

Base Class

- PLM_root_object (ABS)

Attributes

- id : String [1] The id specifies the identifier of the Process_operation_occurrence.

Compositions

- process_operation_resource_assignment : Process_operation_resource_assignment [0..*]
  The process_operation_resource_assignment specifies the process_operation_resource_assignment that is associated with this Process_operation_occurrence.
- process_operation_occurence_relationship : Process_operation_occurrence_relationship [0..*]
  The process_operation_occurence_relationship specifies the process_operation_occurence_relationship that relates the first of the two Process_operation_occurrence objects.
- process_operation_input_or_output : Process_operation_input_or_output [0..*]
  The process_operation_input_or_output specifies the process_operation_input_or_output that s associated with this Process_operation_occurrence.
- configuration : Configuration [0..*]
  The configuration specifies the configuration that controls this Process_operation_occurrence for its valid usage.
- document_assignment : Document_assignment [0..*]
  The document_assignment specifies the object that provides information for this Process_operation_occurrence.
- simple_property_association : Simple_property_association (ABS) [0..*]
  The simple_property_association specifies the assigned simple property values.

Associations

- operation_definition : Process_operation_definition [1]The operation_definition specifies the Process_operation_definition that defines the Process_operation_occurrence in a Process_plan.
- is_defined_in : Cartesian_coordinate_space (ABS) [0..1]
  The is_defined_in specifies the Cartesian_coordinate_space of the Process_operation_occurrence for the case where none of the tools associated by Process_operation_input_or_output plays the role of a reference tool defining the reference coordinate space.
- plan : Process_plan [1] The plan specifies the Process_plan to which the Process_operation_occurrence is assigning a Process_operation_definition.

7.7.12.5 Class Process_operation_occurrence_relationship

A Process_operation_occurrence_relationship is a relationship between two Process_operation_occurrence objects.

Base Class

- PLM_object (ABS)

Attributes

- relation_type : String [1] The relation_type specifies the meaning of the relationship. Where applicable the following values shall be used:
  - 'decomposition': The application object defines a relationship where the related Process_operation_occurrence is one of the components of the relating Process_operation_occurrence;
  - 'exclusiveness': The application object defines a relationship where the relating and the related Process_operation_occurrence shall not have any overlap in time of execution;
  - 'sequence': The application object defines a relationship where the relating Process_operation_occurrence shall be completed before the related Process_operation_occurrence starts;
  - 'simultaneity': The application object defines a relationship where the relating and the related Process_operation_occurrence are considered as occurring during the same time period;
  - 'substitution': The application object defines a relationship where the related Process_operation_occurrence replaces of the relating Process_operation_occurrence.

Compositions

- description : String_select [0..1] The description specifies additional information about the Process_operation_occurrence_relationship.
- change : Change [0..*] The change specifies the change for which this object references a modified object and the corresponding original object.

Associations

- cycle_time : Duration [0..1] The cycle_time specifies the interval of time within which both Process_operation_occurrence objects have to take place in order to be declared as simultaneous.
- waiting_time : Property_value (ABS) [0..1] The waiting_time specifies the time which shall elapse, at least, between the completion of the relating Process_operation_occurrence and the start of the related Process_operation_occurrence. The referenced shall have a definition that is a Duration_property.
- related : Process_operation_occurrence [1] The related specifies the second of the two Process_operation_occurrence objects related by a Process_operation_occurrence_relationship.

7.7.12.6 Class Process_operation_resource_assignment

A Process_operation_resource_assignment is a mechanism to associate a resource with a Process_operation_occurrence.

Base Class

- PLM_object (ABS)

Attributes

- reference_tool : Boolean [1] The reference_tool specifies whether or not the resource identified by the Process_operation_resource_assignment plays the role of the reference tool for the occurrence of an operation.

Compositions

- reason : String_select [0..1] The reason specifies the rationale behind the use of the resource for a particular Process_operation_occurrence.
- simple_property_association : Simple_property_association (ABS) [0..*]
  The simple_property_association specifies the assigned simple property values.

Associations

- placement : Transformation (ABS) [0..1]
  The placement specifies the geometrical Transformation between the local coordinate system of the Process_operation_resource_assignment and the reference coordinate system.
- resource_definition : Resource_definition_select [1]
  The resource_definition specifies the tool that is used to perform the operation.

7.7.12.7 Class Process_plan

A Process_plan is the manufacturing planning information, necessary to realize or produce a particular version of an Item.

Base Class

- PLM_root_object (ABS)

Attributes

- plan_id : String [1] The plan_id specifies the identifier of the Process_plan that shall be unique within the scope of an organization.

Compositions

- process_plan_relationship : Process_plan_relationship [0..*]
  The process_plan_relationship specifies the process_plan_relationship that relates the first of the two Process_plan objects.
- description : String_select [0..1]The description specifies additional information about the Process_plan.
- name : String_select [0..1] The name specifies the word or group of words by which the Process_plan is referred to.
- configuration : Configuration [0..*]
  The configuration specifies the configuration that controls this Process_plan for its valid usage.
- document_assignment : Document_assignment [0..*]
  The document_assignment specifies the object that provides information for this Process_plan.
- simple_property_association : Simple_property_association (ABS) [0..*]
  The simple_property_association specifies the assigned simple property values.

Associations

- produced_output : Item_version [0..*]
  The produced_output specifies the set of Item_version objects that are produced by the operations of the Process_plan.

7.7.12.8 Class Process_plan_relationship

A Process_plan_relationship is the relationship between two Process_plan objects.

Base Class

- PLM_object (ABS)

Attributes

- relation_type : String [1] The relation_type specifies the meaning of the relationship. Where applicable the following values shall be used:
  - 'alternative': The application object defines a relationship where the related Process_plan may be used alternatively to the relating Process_plan;
  - 'version association': The application object defines a relationship where the related Process_plan is a version of the relating Process_plan. In this case, the related Process_plan shall be a Process_plan_version;
  - 'version sequence': The application object defines a relationship where the relating Process_plan is the preceding version and the related Process_plan is the following version. In this case, both Process_plan objects shall be of type Process_plan_version.

Compositions

- description : String_select [0..1]The description specifies additional information about the Process_plan_relationship.
- change : Change [0..*] The change specifies the change for which this object references a modified object and the cor-responding original object.

Associations

- related : Process_plan [1] The related specifies the second of the two Process_plan objects related by a Process_plan_relationship.

7.7.12.9 Class Process_plan_version

A Process_plan_version is a particular version of a Process_plan.

Base Class

- Process_plan

Attributes

- version_id : String [1] The version_id specifies the identification of a particular version of a Process_plan.

Compositions

- none

Associations

- none

7.7.12.10 Class Process_property_association

A Process_property_association is a mechanism to assign a property value to process related objects.

Base Class

- Property_value_association (ABS)

Attributes

- none

Compositions

- none

Associations

- described_element : Process_property_select [1]
  The described_element specifies the object that is described by the property value.

7.7.12.11 Class Process_state

A Process_state is a view of an in-process-item definition of a particular version of an Item. It characterizes a state of the Item_version that occurs before the state identified by the 'related_item_definition'. The identifier of a Process_state shall be unique within the context of the Item_version and of the Process_plan_version.

Base Class

- Design_discipline_item_definition

Attributes

- none

Compositions

- none

Associations

- related_item_definition : Design_discipline_item_definition [1]
  The related_item_definition specifies the Design_discipline_item_definition that defines the final item that the in-process-item is a preliminary stage of.

7.7.12.12 Interfaces

Interface Process_operation_input_or_output_select

This empty interface is realized by the following classes:

- Design_discipline_item_definition
- Item_instance (ABS)
- Assembly_component_relationship

Interface Process_property_select

This empty interface is realized by the following classes:

- Activity_method_assignment
- Activity
- Process_plan
- Process_operation_resource_assignment
- Process_operation_occurrence
- Process_operation_definition

Interface Resource_definition_select

This empty interface is realized by the following classes:

- Product_component
- Physical_instance
- Descriptive_specification
- Design_discipline_item_definition
- Item_instance (ABS)

7.7.13 Package Multi_language_support

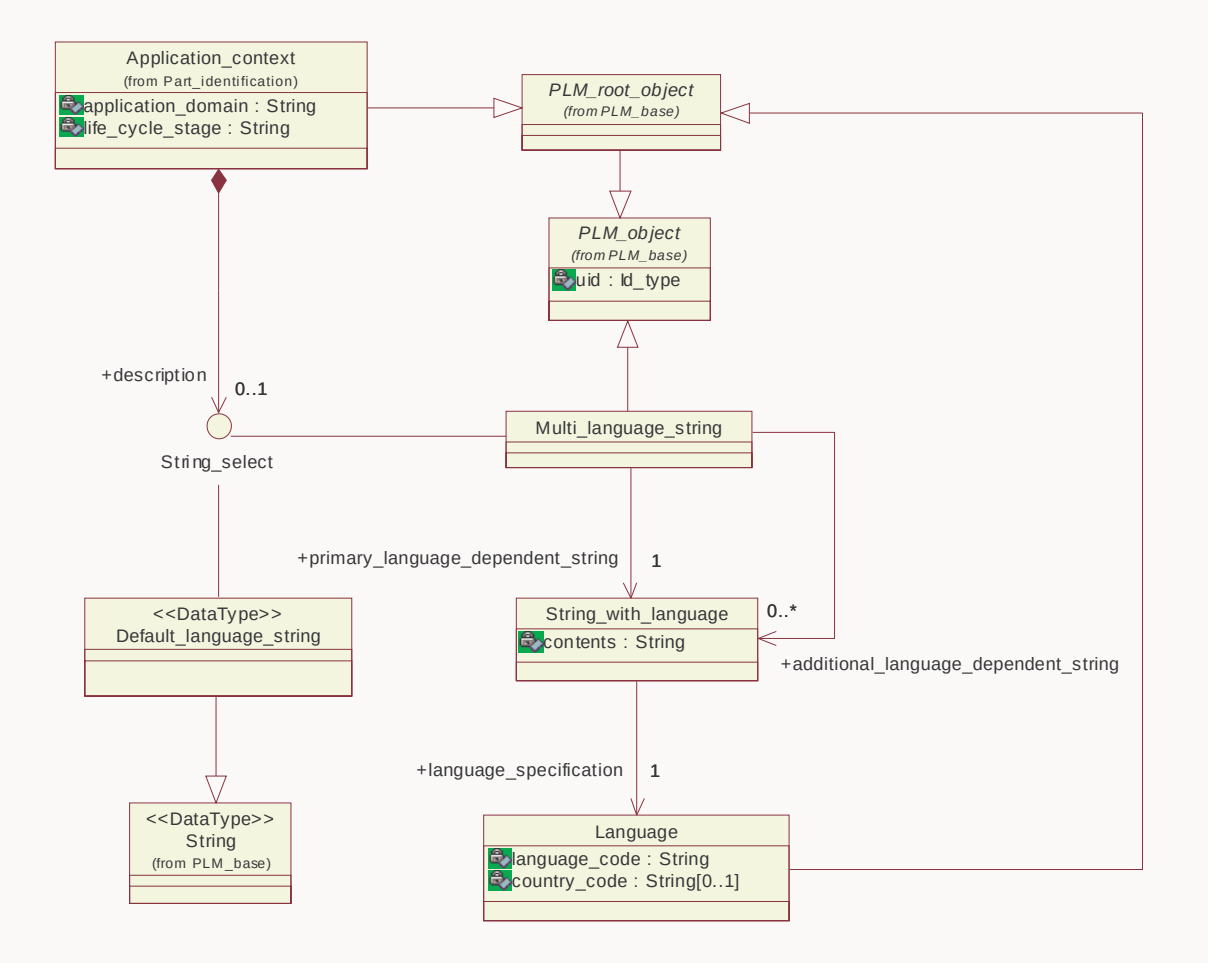


Figure 120 - Multi language support

7.7.13.1 Class Language

A Language is a specification of the language in which an information is given.

Base Class

- PLM_root_object (ABS)

Attributes

- language_code : String [1] The language_code specifies the language of the text information in the Alpha-3 bibliographic code specified in ISO 639-2.
- country_code : String [0..1] The country_code specifies the country, as addition to the language, according to the alpha-2 code specified in ISO 3166-1.

Compositions

- none

Associations

- none

7.7.13.2 Class Multi_language_string

A Multi_language_string represents text information, expressed in one or more languages, that is associated with objects.

Base Class

- PLM_object (ABS)

Attributes

- none

Compositions

- none

Associations

- primary_language_dependent_string : String_with_language [1]
  The primary_language_dependent_string specifies the String_with_language that represents the text information in the original language.
- additional_language_dependent_string : String_with_language [0..*]
  The additional_language_dependent_string specifies the String_with_language objects that represent the text information in a particular language.

7.7.13.3 Class String_with_language

A String_with_language represents text information in a specific language together with an identification of the language used.

Base Class

- none

Attributes

- contents : String [1] The contents is textual information stored in the language identified by the language attribute.

Compositions

- none

Association

- language_specification : Language [1]
  The language_specification specifies the Language in which the contents is given.

7.7.13.4 Interfaces

Interface String_select

This empty interface is realized by the following class:

- Multi_language_string

7.7.13.5 Datatypes

Datatype Default_language_string

8 Computational Viewpoint

8.1 Overview

The computational viewpoint captures the functional aspects of the model described in Chapter 7.7. There are many different use-cases for the platform independent data model. The main usage of STEP ISO 10303-214:214 [8] is the exchange of engineering data, but nowadays some companies think about using STEP as a company wide data model for all information exchange process.

To support a wide range of use cases the data model must be enriched by functional elements. Those elements should support an effective and easy to use interface for handling the data model.

The Computational Viewpoint provides the necessary life cycle functionality to create, read, update and possibly to delete instances of the data model defined in the Informational Viewpoint. Especially, it defines a mechanism to query and traverse instances of the Informational Viewpoint. Therefore, the Computational Viewpoint is dependent on the Informational Viewpoint.

8.2 PLM Connector

A PLM connector has a similar functional model as the connector defined in the J2EE Connector Architecture specification. The PLM connector uses four specific object types: PLM_resource_adapter, PLM_object_factory, PLM_connection_factory, and PLM_connection and the data types URL, UID, Query, PLM_container, PLM_message, PLM_property, and PLM_properties_descriptor. The types PLM_container and UID are defined in the Informational Viewpoint. The type URL is used to model URL's. All operations of all interfaces can throw PLM_exception objects.

8.3 PLM_property_descriptor and PLM_properties_descriptor

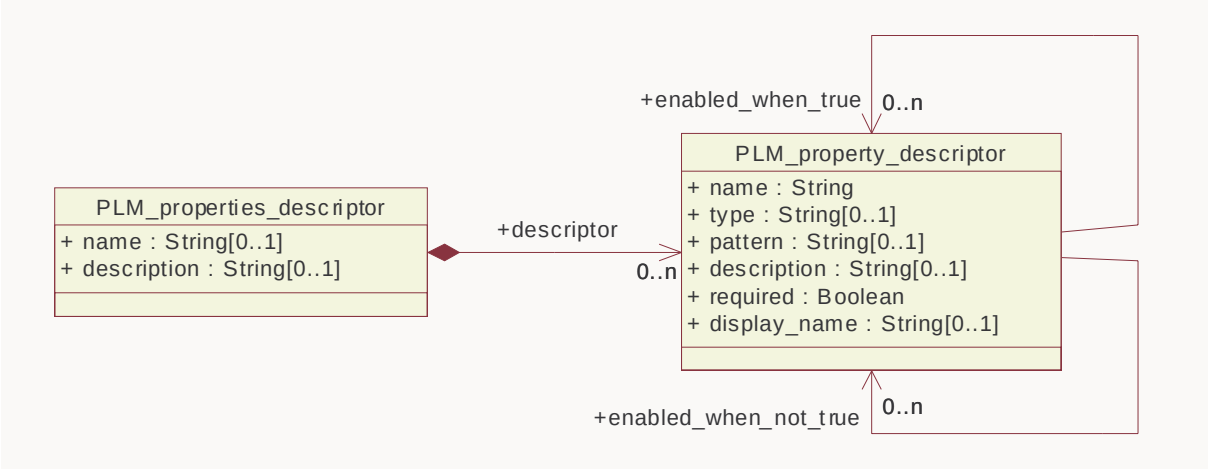


Figure 1 PLM_properties_descriptor and PLM_property_descriptor

Some of the operations defined in the computational model use parameters of type PLM_property. The supported values of those parameters are implementation specific. Each operation with a parameter of type PLM_property has a corresponding operation which a client can use to obtain descriptions of the actual supported variants of values of the

properties parameter. One supported variant of values is described by an instance of type PLM_properties_descriptor. A PLM_properties_descriptor has an attribute name that contains the name of the variant, an attribute description which contains a description of the variant and a list of PLM_property_descriptor. Each element of the PLM_properties_descriptor list describes one PLM_property of the variant.

A PLM_property_descriptor describes one PLM_property instance. The attribute name of the PLM_property_descriptor defines the value of attribute name of the PLM_property instance. The attribute type describes the type of the PLM_property instance. The attribute pattern defines a pattern that must match valid values of attribute value of the PLM_property. The attribute description of the PLM_property_descriptor contains a description of the described PLM_property instance. The attribute required defines wether the described PLM_property instance must be present or if it is optional. The references enabled_when_true and enabled_when_not_true can select other PLM_property_descriptor instances. The selected instance must have the type Boolean and must be contained by the same PLM_properties_descriptor instance. If a PLM_property_descriptor has enabled_when_true- or enabled_when_not_true references its attribute required must not have a value of TRUE. A described PLM_property value can only be used in a properties parameter list for an operation

· if all PLM_property values described by the PLM_property_descriptors referenced by enabled_when_true are also in the properties parameter list and have the value TRUE

· and no PLM_property_value described by a PLM_property_descriptor referenced by enabled_when_not_true is in the properties parameter list and has the value TRUE.

The value of the attribute display_name can be used as display name of the described PLM_property in user interfaces.

8.3.1 Sample "login" PLM_properties_descriptors

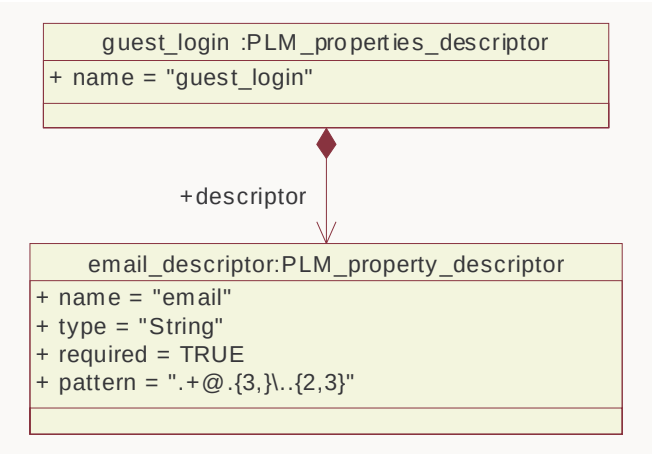


Figure 2 Sample "guest login" PLM_properties_descriptor

Figure 2 shows a sample PLM_property_descriptor instance for a guest login. A valid PLM_property instance set for this descriptor must contain one PLM_property instance with a name attribute of "email" and a value attribute which matches the pattern ".+@.{3,}\..{2,3}".

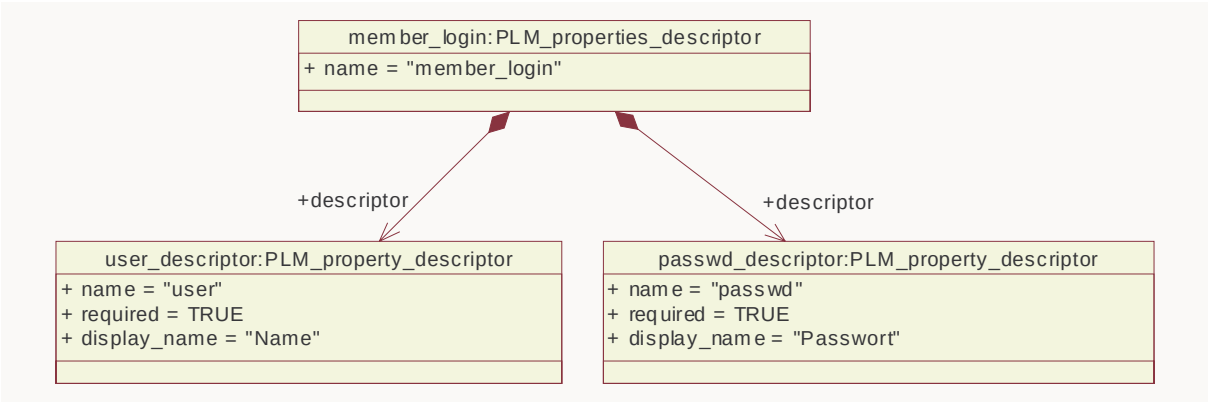


Figure 3 Figure Sample "member login" PLM_properties_descriptor

Figure 3 shows a sample PLM_property_descriptor instance for a member login. A valid PLM_property instance set for this descriptor must contain two PLM_property instances. One instance with a name attribute of "user" and an arbitrary value attribute and one instance with a name attribute of "passwd" and an arbitrary value attribute.

A set of the two sample instances, the "guest login" PLM_properties_descriptor and the "member login" PLM_properties_descriptor is an example of a result of the operation get_connection_properties_descriptors() of an implementation of interface PLM_connection_factory explained in Chapter 8.5. This result means that the operation get_connection() of the same implementation can be called with one of the two described properties parameter variants.

8.3.2 Sample "assembly export" PLM_properties_descriptor

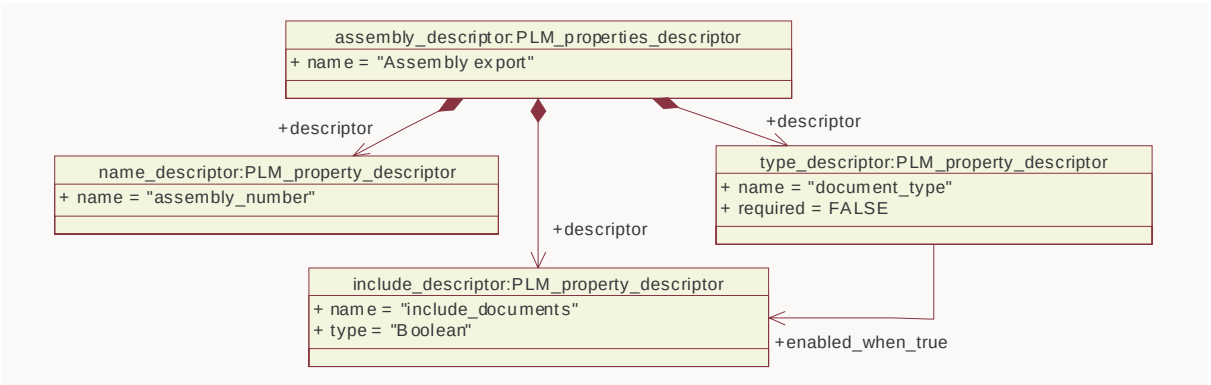


Figure 4 Sample "assembly export" PLM_properties_descriptor

Figure 4 shows a sample PLM_property_descriptor instance as result of the operation get_export_data_properties_descriptors of an implementation of interface PLM_connection explained in . This result means, that the operation export_data() of the same implementation can be called with one set of at least two PLM_property instances. These two required instances have the name attributes "assembly_number" and "include_documents". If the PLM_property with name attribute "include_documents" has the value attribute TRUE than a third PLM_property instance with a name attribute of "document_type" can be added to the properties parameter of the operation export_data().

8.4 PLM_resource_adapter Class

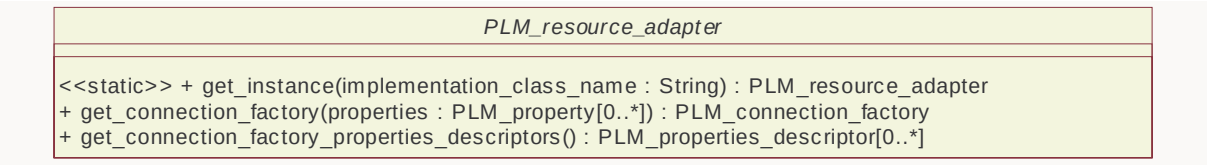


Figure 114 - The PLM_resource_adapter Class

A PLM connector vendor must provide an implementation of the abstract PLM_resource_adapter class. A client may obtain an instance of a specific PLM resource adapter class by the static member function get_instance() with the class name of the specific PLM resource adapter as parameter.

By the operation get_connection_factory() the client can obtain a PLM_connection_factory object. The value of the parameter name is the name of the PLM connection factory. The list of all supported values for this parameter can be obtained by the operation get_connection_factory_names(). In the parameter properties the client can pass specific parameters. The values and semantics of the properties parameter will be defined in the Platform Specific Models. Examples for property names are "java.naming.provider.url" and "java.naming.factory.initial" if the PLM connector implementation uses a JNDI name service.

8.5 PLM_object_factory Interface

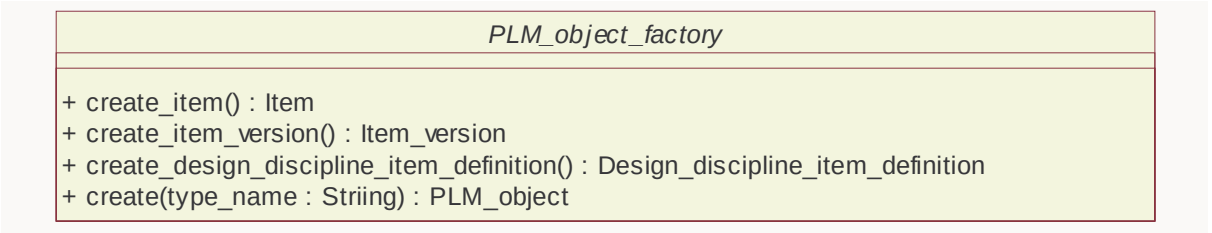


Figure 115 - The PLM_object_factory Interface (fragmentary)

The PLM_object_factory provides one specific create operation for each non abstract type of the Informational PIM which extends direct or indirect PLM_object. Therefore, this interface is directly dependent on the Information Model.

Additionally a generic create operation is provided. Allowed parameter values for the generic create operation are the names of those types for which a specific create operation in the PLM_object_factory exist. The result PLM_objects from the create operations are local objects. The operation write() from the interface PLM_connection has to be used to transfer a local object to a PLM system (create a new object in the PLM system).

8.6 PLM_connection_factory Interface

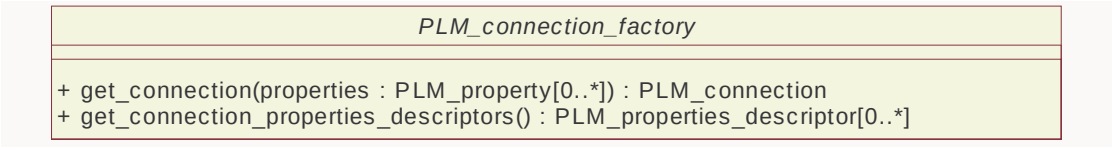


Figure 116 - The PLM_connection_factory Interface

The interface PLM_connection_factory provides the operation get_connection() which returns a PLM_connection instance. By the parameter properties the client may pass specific information to the PLM_connection_factory. This could be “user” and “password” properties. The actual properties are implementation specific, and its descriptors can be obtained by the operation get_connection_factory_properties_descriptors.

8.7 PLM_container Type

All operations in this specification use the type PLM_container as input parameter type or return type when PLM data has to be transfered. So, the PLM_container serves as a container to transfer arbitrary PLM data. The PLM_container type is defined in Chapter 7.7.1.1

8.8 PLM_connection Interface

The PLM_connection is the central interface of this specification. It's purpose is to grant access to the PLM system. To pass PLM data, it uses instances of the class PLM_container. To define the semantics of the operations, it is assumed, that all PLM data in the PLM system is instantiated as a single instance of PLM_container and the implementation of the operations works on that instance.

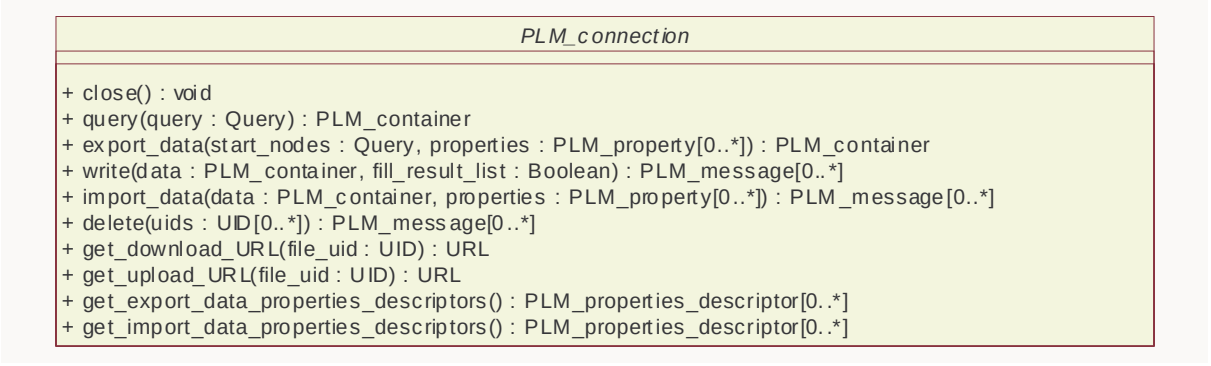


Figure 117 - The PLM_connection Interface

8.9 Query Operation

query(in query: Query): PLM_container

The operation query() expects an Query instance as its input parameter query. By applying this query to the data in a PLM system, a set of selected nodes is generated. As result of the query, a PLM_container instance is returned containing all selected nodes of the query and all nodes required to fulfill the minimum multiplicity constraints of the relationships of the selected nodes.

8.9.1 Write Operation

write(data: PLM_container, fill_result_list: Boolean): PLM_message[0..*]

The operation write() expects a PLM_container instance as an input parameter. The PLM system uses the uid-Attributes of the single nodes in the PLM_container instance to identify which nodes already exist in the PLM System and which nodes have to be created. The operation has a return value of PLM_message objects. In this return value information on manipulated objects is given. If the client ignores this information, the parameter fill_result_list shall be set to FALSE. By creating a new node, it is for a PLM system in general not feasibe to use the attributes of the parameter data set. The operation adds one Object_changed_message for each changed object. If the uid-Attribute, the id-Attribute (e.g. id, name, Document_id, File_id) or any other attribute has changed the new object uid, the new object id or the remainder_unchanged attribute of the Object_changed_message are set, accordingly. The result list is also used to inform the client, if not all objects of the data parameter were inserted in the PLM System. This information is added to the result list as Object_not_inserted_message instances. It is allowed for an implementiation of the operation write() to add extra PLM objects such as creator or creation time objects ot the PLM system. If a write operation adds additional PLM objects into the PLM system this information has to be added to the result list as Additional_objects_written_message instances.

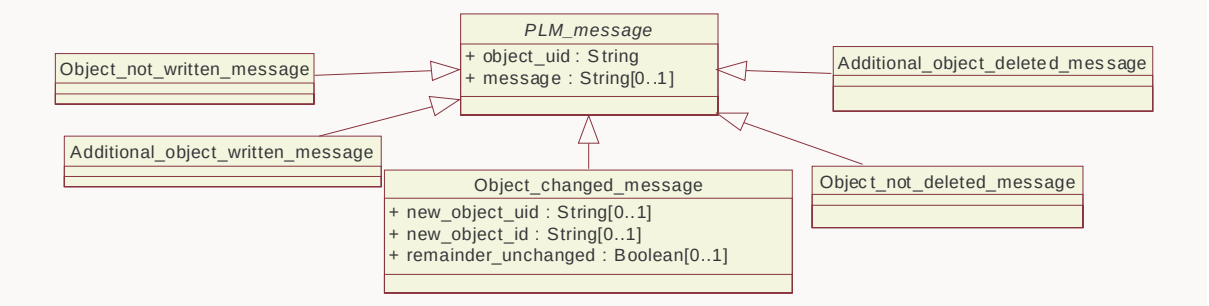


Figure 118 - Message types

All elements of the data set are transferred to the PLM system. Should one element already exist, all attribute values of the existing entity in the PLM system are replaced by the attributes values of the entity in the parameter. The relationships of an existing entity are not replaced by the relationships of the corresponding entity in the parameter. Instead, the relationships of the entity of the parameter not already existing are created.

8.9.2 Export_data Operation

export_data(start_nodes: Query, properties: PLM_property[0..*]):PLM_container

The operation export_data() expects an Query instance as its input parameter start_nodes. It is implementation specific which result this operation returns in the PLM_container. The export_data operation accepts a set of PLM_property objects as additional parameter. The allowed values and the semantic of this parameter are implementation specific, too and can be obtained by the operation get_export_data_properties_descriptors().

8.9.3 Import_data Operation

import_data(data: PLM_container, properties: PLM_property[0..*]): PLM_message[0..*]

The operation import_data() expects a PLM_container instance as an input parameter. The PLM System may transform, filter or extent the input data prior writing to its data base. The actual behavior is implementation specific. The import_data operation accepts a set of PLM_property objects as additional parameter. The allowed values and the semantic of this parameter are implementation specific, too and can be obtained by the operation get_import_data_properties_descriptors(). The return type of the import_data operation is the abstract type PLM_message.

8.9.4 Delete Operation

delete(in uids: UID[0..*]):PLM_Message[0..*]

The operation delete() expects a list of UID elements as input parameter. All objects with the given uids are deleted from the PLM system by this delete operation. Additionally, all nodes are deleted, which no longer fulfill the minimum multiplicity constraints of their type. The operation has a return value of PLM_Message objects. For each object which could not be deleted a single Object_not_deleted_message is added to this list. For each additionally deleted object an Additional_object_deleted_message instance has to be added to the result list.

8.9.5 Get_download_URL Operation

get_download_URL(in file_uid: UID): URL

The get_download_URL() operation is assigned an uid-attribute of a Digital_file object as the only parameter. As a return value, it delivers an URL to retrieve the content of a Digital_file from the PLM system.

8.9.6 Get_upload_URL Operation

get_upload_URL(in file_uid: UID): URL

The get_upload_URL() operation expects an uid-attribute of a Digital_file object as the parameter. It returns an URL which is used to upload a new content of the Digital_file to the PLM system.

8.9.7 Close Operation

close(): void

The close() operation shuts down a connection to a PLM system. After a successful call of the close operation, all subsequent calls to this connection may raise an exception.

8.9.8 Get_export_data_properties_descriptors Operation

get_export_data_properties_descriptors(): PLM_properties_descriptor[0..*]

The get_export_data_properties_descriptors() operation returns the descriptors of all supported parameter variants of the export_data() operation.

8.9.9 Get_import_data_properties_descriptors Operation

get_import_data_properties_descriptors(): PLM_properties_descriptor[0..*]

The get_import_data_properties_descriptors() operation returns the descriptors of all supported parameter variants of the import_data() operation.

8.10 PLM_exception classes

All operations of the interfaces of the Computational Viewpoint can raise exceptions derived from the abstract type PLM_exception. As actual subtypes of PLM_exception the following exceptions are defined in this specification: Authentication_exception, Authorization_exception, Session_timeout_exception and Object_uid_timeout_exception, Invalid_session_id_exception, Unsupported_pattern_exception, Unsupported_query_exception, Unsupported_operation_exception and Invalid_object_uid_exception

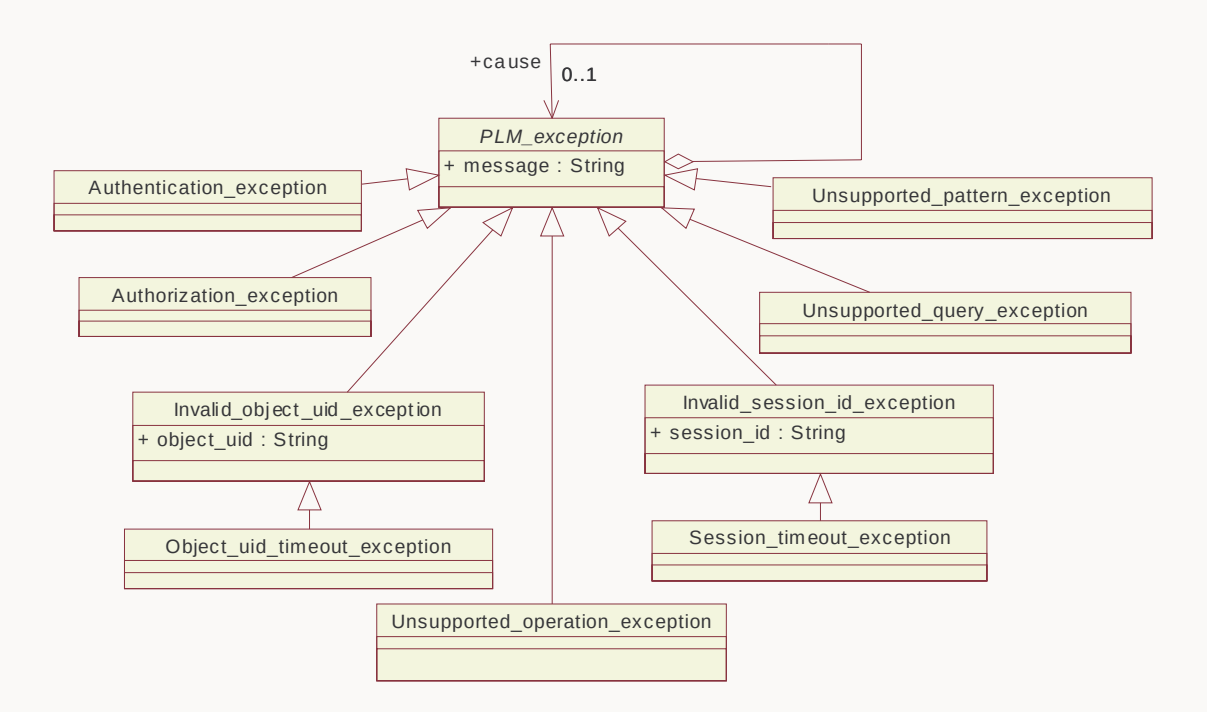


Figure 119 PLM_exception and its subtypes

8.10.1 Authentication_exception

The Authentication_exception is thrown by the operation get_connection of the interface PLM_connection_factory if the authentication of the client fails. The authentication mechanism is implementation specific.

8.10.2 Authorization_exception

The Authorization_exception is thrown by an operation if the client has not the right to perform the requested operation with the given parameters.

8.10.3 Invalid_session_id_exception

The Invalid_session_id_exception is thrown by an operation of the interface PLM_connection if a session identifier is used for that operation which is unknown to the service implementation. The transfer of session identifiers has to be defined by the platform specific models.

8.10.4 Session_timeout_exception

The Session_timeout_exception is thrown by an operation of the interface PLM_connection when the session time has expired.

8.10.5 Object_uid_timeout_exception

An object UID may expire before a session is closed. The Object_uid_timeout_exception must be thrown by an operation of the interface PLM_connection if such an expired object UID is used by a client as a parameter.

8.10.6 Invalid_object_uid_exception

The Invalid_object_uid_exception is thrown by an operation of the interface PLM_connection when a UID value of a server object is used in one parameter of the operation the associated object of which no longer exist or had never exist on the server. The UID value is returned in the attribute object_uid of the exception.

8.10.7 Unsupported_query_exception

The Unsupported_query_exception is thrown by the query and export_data operation of the interface PLM_connection if a Query value is used as parameter, that is not supported by the service implementation.

8.10.8 Unsupported_pattern_exception

The Unsupported_pattern_exception is thrown by the query() and export_data() operations of the interface PLM_connection if a pattern value is used which is not supported by the service implementation.

8.10.9 Unsupported_operation_exception

The Unsupported_operation_exception is thrown by a operation if the requested operation is not supported by the service implementation.

8.11 Query Type

The type Query is an abstract base type. It is used as parameter in the query and export operation of the PLM_connection. The type Query has to be specialized in “Queries Conformance Points” Chapter 8.12.

When a Query instance is applied to the set of PLM_objects of a server it selects a subset of these PLM_objects. The way of selecting this initial result set is specific to each specialization of the Query type. The initial result set of the Query instance has to be extended by further PLM_objects of the server until the minimal result set is selected, which contains all initially selected PLM_objects and fulfills all occurrency constraints of all selected PLM_objects. This specification defines the following rules how to extend an initial result set to fulfill the multiplicity constaints:

- If a selected PLM_object instance is a component in a composition, the result set has to be extended by the composite

instance.

- If a selected PLM_object instance is a composite in a composition and the multiplicity of the component in the composition is one, the result set has to be extended by the component instance.
- ·If a selected PLM_object instance has a reference and the multiplicity of the referenced objects is one, the result set has to be extended by the referenced instance.
- ·If a selected PLM_object instance is a composite in a composition or has a reference and the minimum multiplicity of the component respectively referenced objects is not zero and the maximum multiplicity is greater than one the result set may be extended by selecting further PLM_object instances. Either there are enough PLM_object instances selected in the result set that play the component role in the composition respectively the referenced role in the directed association to fulfill the minimum multilplicity constraint or the result set is extended by NIL objects which are used as components respectively referenced objects. NIL objects are special instances of types derived from PLM_object which can be used as helper instances to fulfill multiplicity constraints in PLM_object sets. The creation and distinction of NIL objects has to be defined in platform specific models.

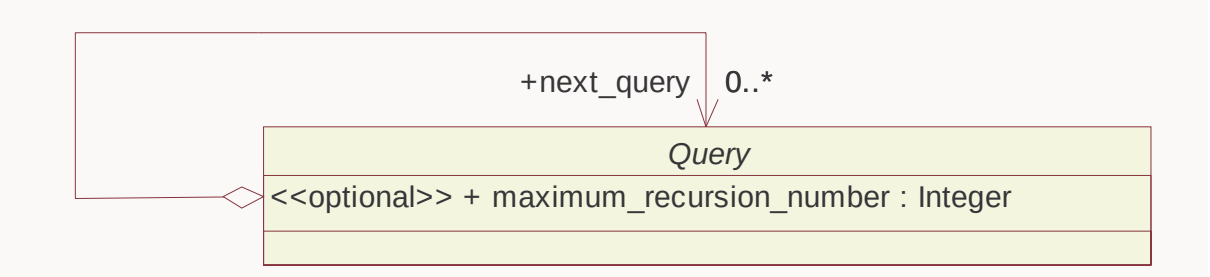


Figure 120 - Query Type

The Query type provides the possibility of concatenated batch, conditional, and recursive queries.

The concatenation of queries is realized by an association which links a Query object with a next Query object(s). The role name of the linked next Query object(s) is next_query. If a query is extended by another query to a concatenated query, the result of the concatenated query is defined as the union of the results of the two single queries. The start nodes of the second query are limited to the nodes which the PLM_connection would return as result of the first query alone. This limitation concerns only the start nodes but not the result of the second query. In the second query all links from the result nodes of the first query to arbitrary nodes in the PLM system can be evaluated and added to the result of the second query.

In general, executing queries against a tree of PLM objects as defined by the Informational viewpoint would require in a recursive tree traversal. This recursion of a Query is controlled by the attribute maximum_recursion_number. If this attribute is not set or has the value 0 a non recursive query is applied. If the attribute has a positive value n the query has n recursions. A recursion of a query instance has the same semantic as the concatenation of n equal query instances. A maximum_recursion_number with a negative value means an infinitive recursion.

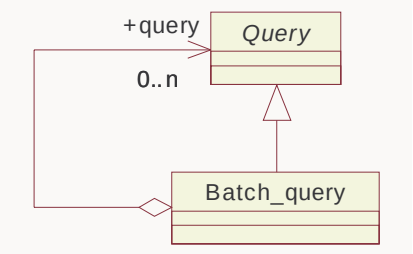


Figure 121 Class diagram of class Batch_query

The type Batch_query combines other query instances to a batch job. A Batch_query instance is evaluated by evaluating all contained query instances of the Batch_query instance independently and create one result from all objects selected by the contained queries.

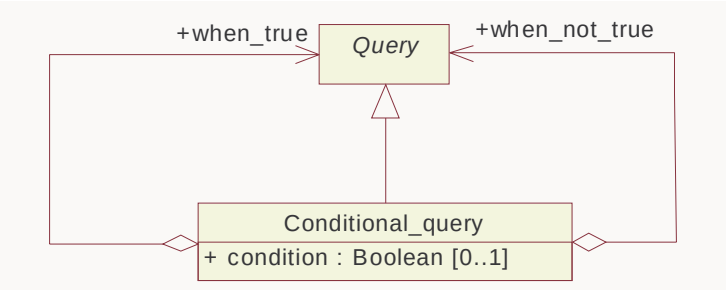


Figure 122 Class diagram of class Conditional_query

The type Conditional_query enables the execution of a query in dependency of a condition. If the attribute evaluates true, the query referenced by query is executed otherwise the query is not executed.

8.12 Generic Queries Conformance Point

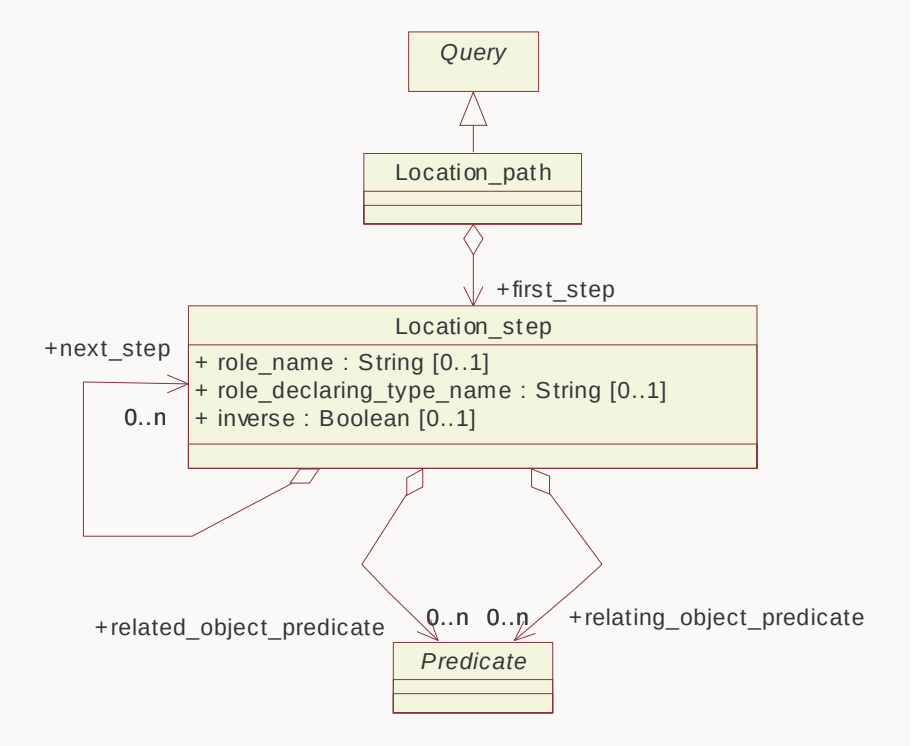


Figure 123 - The class diagram of the Generic Queries Conformance Point

The Generic Queries Conformance Point defines a toolset of classes that can be used to query arbitrary data from a PLM system. This toolset consists of the types Location_path, Location_step, Predicate and specializations of Predicate.

The PLM_container instance models PLM data as a set of direct or indirect contained nodes (instances of PLM_Object). The nodes are related by relationships. The relationship types of a node are composition or directed association. They are described in Chapter 7.7 for each node type.

To define a subset of the nodes of a PLM_container instance an instance of the abstract type Query has to be used. The type Location_path is the specialization of the Query type for the Generic Conformance Point. The Location_path is an new query tool that is de-signed to optimally implement the PLM Services needs. A Location_path consists of a tree of instances of Location_step.

The root node of the Tree is defined by the association first_step of the Location_path. By the association next_step of a Location_step instance the child nodes of this Location_step instance node in the tree are determined.

By applying a Location_path instance to a PLM_container instance each Location_step of the path in turn selects a set of nodes relative to the currently selected node-set.

The initially selected node-set is defined by all nodes that are directly or indirectly related to the PLM_container instance. The resulting selected node-set of a Location_path is the union of all selected node-sets of all Location_steps of the Location_path.

A location step consists of:

- a role name which specifies the nodes selected by the location step,
- the name of the type that declares the relationship with the role,
- a flag that indicates if the navigation direction is inverse in respect of the informational model,
- zero or more Related_object_predicate predicates which use arbitrary expressions further refining the set of nodes selected by the location step,
- zero or more Relating_object_predicate predicates which use arbitrary expressions further refining the set of nodes selected by the location step, and
- a list of location steps following directly the current location step.

The node-set selected by a location step is the node-set that results from generating an initial node-set from all nodes that are reached from the nodes in the current selected node-set by following the named relationship, and then filtering that node-set by each of the predicates in turn. If a Location_step has more than one next_step these steps result in one different selected node-set for each step.

8.12.1 Specialized Predicates for filtering of object sets

Each non-abstract spezialisation of the abstract class Predicate defines a constraint for filtering object sets. Filtering means that the algorithm is applied to each object in the set and only the objects which fit the constraint remain in the set. The following non-abstract specializations of the class Predicate are used in this specification.

8.12.1.1 Alternative_predicate

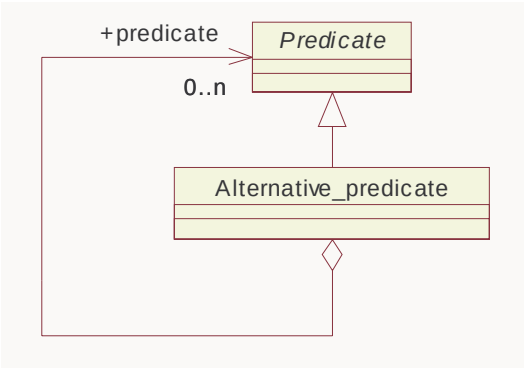


Figure 124 Class diagram of class Alternative_predicate

An object fulfill an Alternative_predicate constraint if it fulfill at least one the Predicate instances referenced by the relationship predicate of the Alternative_predicate.

8.12.1.2 Attribute_equals_predicate

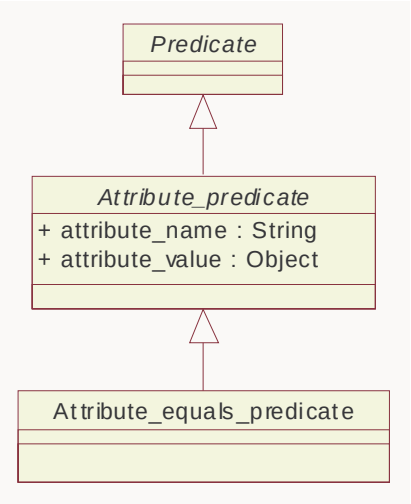


Figure 125 Class diagram of class Attribute_equals_predicate

An object fulfill an Attribute_equals_predicate if it has an attribute with the name given in the attribute attribute_name of the Attribute_equals_predicate and if that attribute has a value which is equals to the value given by the attribute attribute_value of the Attribute_equals_predicate.

8.12.1.3 Attribute_greater_than_predicate

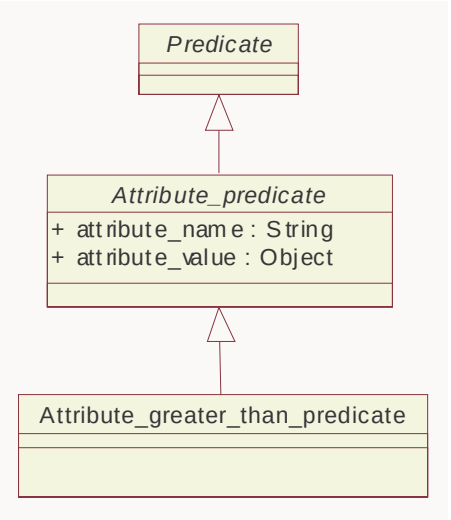


Figure 126 Class diagram of class Attribute_greater_than_predicate

An object fulfill an Attribute_greater_than_predicate if it has an attribute with the name given in the attribute attribute_name of the Attribute_greater_than_predicate and if that attribute has a value which is greater than the value given by the attribute attribute_value of the Attribute_greater_than_predicate.

8.12.1.4 Attribute_less_than_predicate

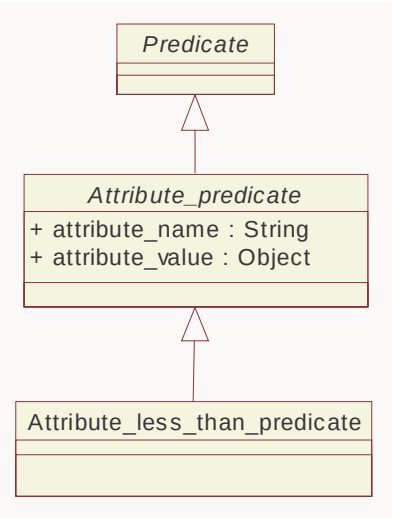


Figure 127 Class diagram of class Attribute_less_than_predicate

An object fulfill an Attribute_less_than_predicate if it has an attribute with the name given in the attribute attribute_name of the Attribute_less_than_predicate and if that attribute has a value which is less than the value given by the attribute attribute_value of the Attribute_less_than_predicate.

8.12.1.5 Attribute_pattern_predicate

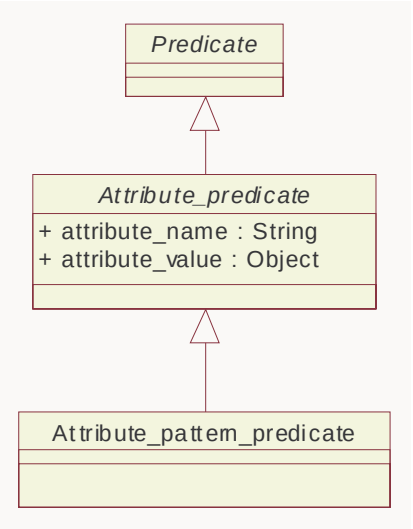


Figure 128 Class diagram of class Attribute_pattern_predicate

An object fulfill an Attribute_pattern_predicate if it has an attribute with the name given in the attribute attribute_name of the Attribute_pattern_predicate and if that attribute has a value which match the pattern given by the attribute attribute_value of the Attribute_pattern_predicate. This specification uses the pattern language defined in [XML Schema W3C Recommendation 28 October 2004].

8.12.1.6 Identifier_predicate

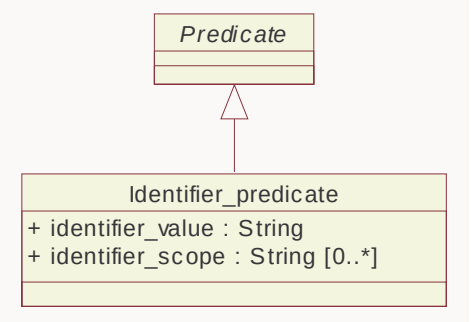


Figure 129 Class diagram of class Identifier_predicate

All classes which have a composition of type Alias_identification have also an attribute that corresponds with the attribute alias_id of the related Alias_identification. These corresponding attributes are identifying attributes and can be filtered by Identifier_predicates.

There are three alternatives how an object can fulfill the constraints of an Identifier_predicate.

1. If the attribute identifier_scope of the Identifier_predicate is not set, an object fulfill the Identifier_predicate if it has an identifier attribute and if that attribute has a value that matches the pattern given by the attribute identifier_value of the Identifier_predicate.

2. If the attribute identifier_scope of an Identifier_predicate is set, an object fulfill the Identifier_predicate if it has an Alias_identification with an value for its attribute alias_scope that is equals to the value of the attribute identifier_scope and if the attribute alias_id of the Alias_identification has a value that matches the pattern given by the attribute identifier_value of the Identifier_predicate.

3. If the attribute identifier_scope of an Identifier_predicate is set, an object fulfill the Identifier_predicate if it has an identifier attribute and if that attribute has a value that matches the pattern given by the attribute identifier_value of the Identifier_predicate and if it is referenced by the relationship is_applied_to of a Person_organization_assignment_instance and the attribute role of the Person_organization_assignment instance has the value "id owner" and the Person_organization_assignment is referenced by the composition person_organization_assignment of an Organization instance and the attribute id of the Organization instance is equals to the value of the attribute identifier_scope of the Identifier_predicate.

This specification uses the pattern language defined in [XML Schema W3C Recommendation 28 October 2004].

8.12.1.7 Relationship_predicate

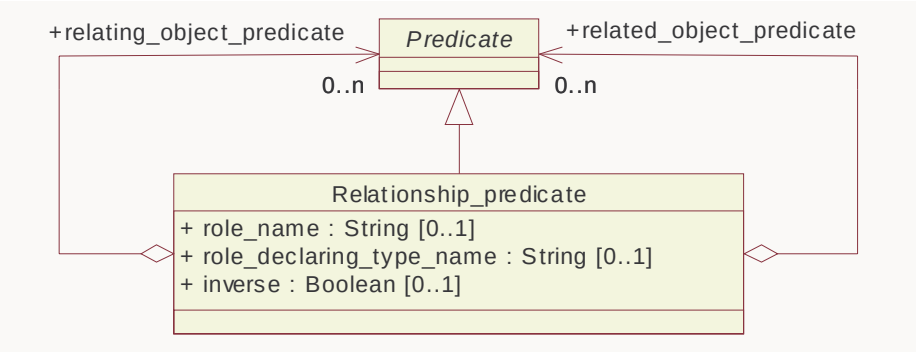


Figure 130 Class diagram of class Relationship_predicate

An object fulfill a Relationship_predicate constraint if it fulfill the following partial constraints:

- The object is related with another object that fulfill all the Predicate instances referenced by the relationship predicate of the Relationship_predicate.
- If the value of the attribute inverse of the Relationship_predicate is not true and if the attribute role_name is set, the role name of the other object in the relationship must be equals to the value of the attribute role_name of the Relationship_predicate.
- If the value of the attribute inverse of the Relationship_predicate is true and if the attribute role_name is set, the role name of this object in the relationship must be equals to the value of the attribute role_name of the Relationship_predicate.
- If the attribute role_declaring_type_name is set, the relationship must be defined in a type which name is equals to the value of the attribute role_decalring_type_name.

8.12.1.8 String_select_predicate

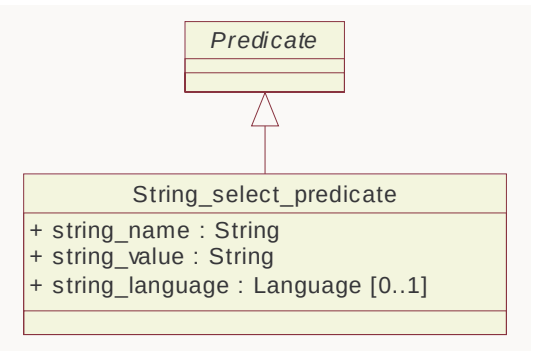


Figure 131 Class diagram of class String_select_predicate

An object fulfill a String_select_predicate if it has an attribute of type String_select with the name given in the attribute string_name of the String_select_predicate and if it fulfill one of the following constraints:

- If the attribute string_language is not set and the attribute with the name given by the attribute string_name must be a Default_language_string which value is equals to the value given by the attribute string_value of the String_select_predicate.
- If the attribute string_language is set and equals the default language of the server implementation the attribute with the name given by the attribute string_name must be an instance of Default_language_string_with a value which is equals to the value given by the attribute string_value of the String_select_predicate or an instance of Multi_language_string with a primary_language_dependent_string which value is equals to the value given by the attribute string_value of the String_select_predicate.
- If the attribute string_language is set and not equals the default language of the server implementation the attribute with the name given by the attribute string_name must be an instanceof Multi_language_string and have a additional_language_dependent_string which value is equals to the value given by the attribute string_value of the String_select_predicate.

8.12.1.9 Type_predicate

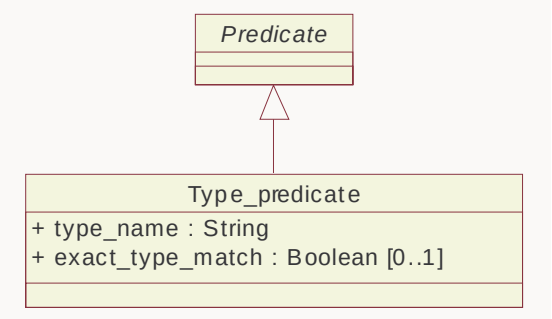


Figure 132 Class diagram of class Type_predicate

If the value of the attribute exact_type_match of an Type_predicate is TRUE, an object fulfill that Type_predicate constraint if it has exact the type specified in the attribute type_name of the Type_predicate.

If the value of the attribute exact_type_match of an Type_predicate is not TRUE, an object fulfill that Type_predicate constraint if it is an instance of the type specified in the attribute type_name of the Type_predicate or an instance of a derivation of that type.

8.12.2 Query_with_relating_type_predicate

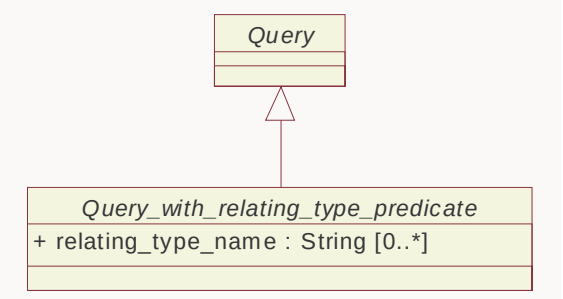


Figure 133 Class diagram of abstract class Query_with_relating_type_predicate

The abstract class Query_with_relating_type_predicate is used as base class for all queries which need an attribute relating_type_name.

8.12.3 Relationship_query

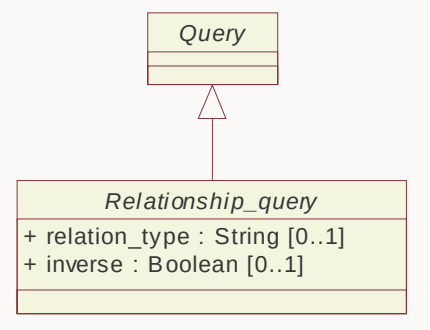


Figure 134 Class diagram of abstract class Relationship_query

The abstract class Relationship_query is used as base class for all queries which need an attribute relation_type and an attribute inverse.

8.13 XPath Queries Conformance Point

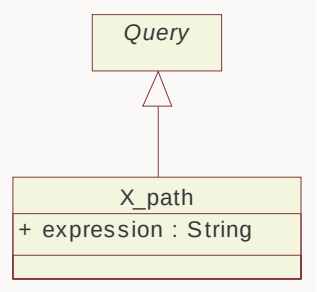


Figure 135 - The class diagram of the XPath Queries Conformance Point

The XPath conformance point defines the type X_path as specialization of the type Query. The type X_path provides the possibility to use arbitrary XPath expressions conforming to the W3C XPath specification as queries. The Web Service PSM defined in this specification defines how a PLM_container instance has to be transformed to a XML-Document. An XPath expression selects nodes in this XML-Document. These nodes (or their parent nodes in the case of non XML element nodes) have equivalent instances in the PIM that are subtypes of PLM_object. These instances are the result set of a XPath expression at the PIM level.

8.14 Specific Queries Conformance Point

The Specific Queries Conformance Point defines a set of low level specialized queries that are building blocks to fulfill the requirements of the use cases described in Chapter 7.2. The semantic of each specialized query of this conformance point is defined by an equivalent Location_path instance. The semantic of Location_path is defined in the Generic Queries Conformance Point in Chapter 8.12.

8.14.1 Common interfaces for types of start and target objects

This chapter defines some interfaces which are used to group their implementing classes for the purpose of specifying the start node types and the target node types of the queries of the Specific Queries Conformance Point.

8.14.1.1 Interface Simple_property_select

Compositions:

- simple_property_association : Simple_property_association [0..*]

Extended by:

- Item_property_select
- Process_property_select

8.14.1.2 Interface Alias_select

Compositions:

- alias_identification : Alias_identification [0..*]

Implemented By:

- Organization
- Complex_product
- Classification_attribute
- Item
- Document_type_property
- Product_class
- Document_version
- Specification_category
- Document
- Specification
- Item_version
- Classification_system

- Item_instance
- Document_representation
- Property
- General_classification
- Design_discipline_item_definition
- Physical_instance
- Approval_status

8.14.1.3 Interface Configured_item_select

Compositions:

- configuration : Configuration [0..*]

Implemented By:

- Process_operation_occurrence
- Product_function
- Product_component
- Alternative_solution
- Process_plan
- Item_instance

8.14.1.4 Interface Documented_element_select

Compositions:

- document_assignment : Document_assignment [0..*]

Implemented By:

- Shape_element_relationship
- Process_operation_occurrence
- Work_order
- Product_identification
- Organization
- Physical_instance_test_result
- Item_definition_instance_relationship
- Complex_product

- Classification_attribute
- Item
- Product_class
- Item_definition_relationship
- Specification_category
- Change
- Specific_item_classification
- Material
- Specification
- Item_version
- Activity_element
- Project
- Classification_system
- Process_plan
- Activity_method
- Approval
- Item_instance
- Descriptive_specification
- Property
- Product_structure_relationship
- Shape_element
- General_classification
- Design_discipline_item_definition
- Item_instance_relationship
- Physical_instance
- Work_request
- Item_shape
- Design_constraint
- Physical_assembly_relationship
- Activity
- Class_structure_relationship

- Person

8.14.1.5 Interface Person_organization_select

Compositions:

- date_and_person_organization : Date_and_person_organization [0..*]
- person_organization_assignment : Person_organization_assignment [0..*]

Implemented By:

- Person_in_organization
- Organization

8.14.1.6 Interface Instance_definition_select

Compositions:

- item_instance : Item_instance [0..*]

Implemented By:

- Product_identification
- Design_discipline_item_definition

8.14.1.7 Interface Shape_information_select

Compositions:

- shape_description_association : Shape_description_association [0..*]

Implemented By:

- Shape_element_relationship
- Shape_element
- Item_shape

8.14.1.8 Interface Specification_operand_select

Compositions:

- specification_inclusion : Specification_inclusion [0..*]

Implemented By:

- Specification
- Specification_expression

8.14.1.9 Interface Change_relationship_select

Compositions:

- change : Change [0..*]

Implemented By:

- Process_operation_occurrence_relationship
- Process_plan_relationship
- Shape_element
- Replaced_definition_relationship
- Item_version_relationship

8.14.2 Activity_element_query

The Activity_element_query traverses from Activity objects via Activity_element objects to Activity_element_select objects.

Parameters

- role: String [0..1]
- element_type_name: String [0..1]

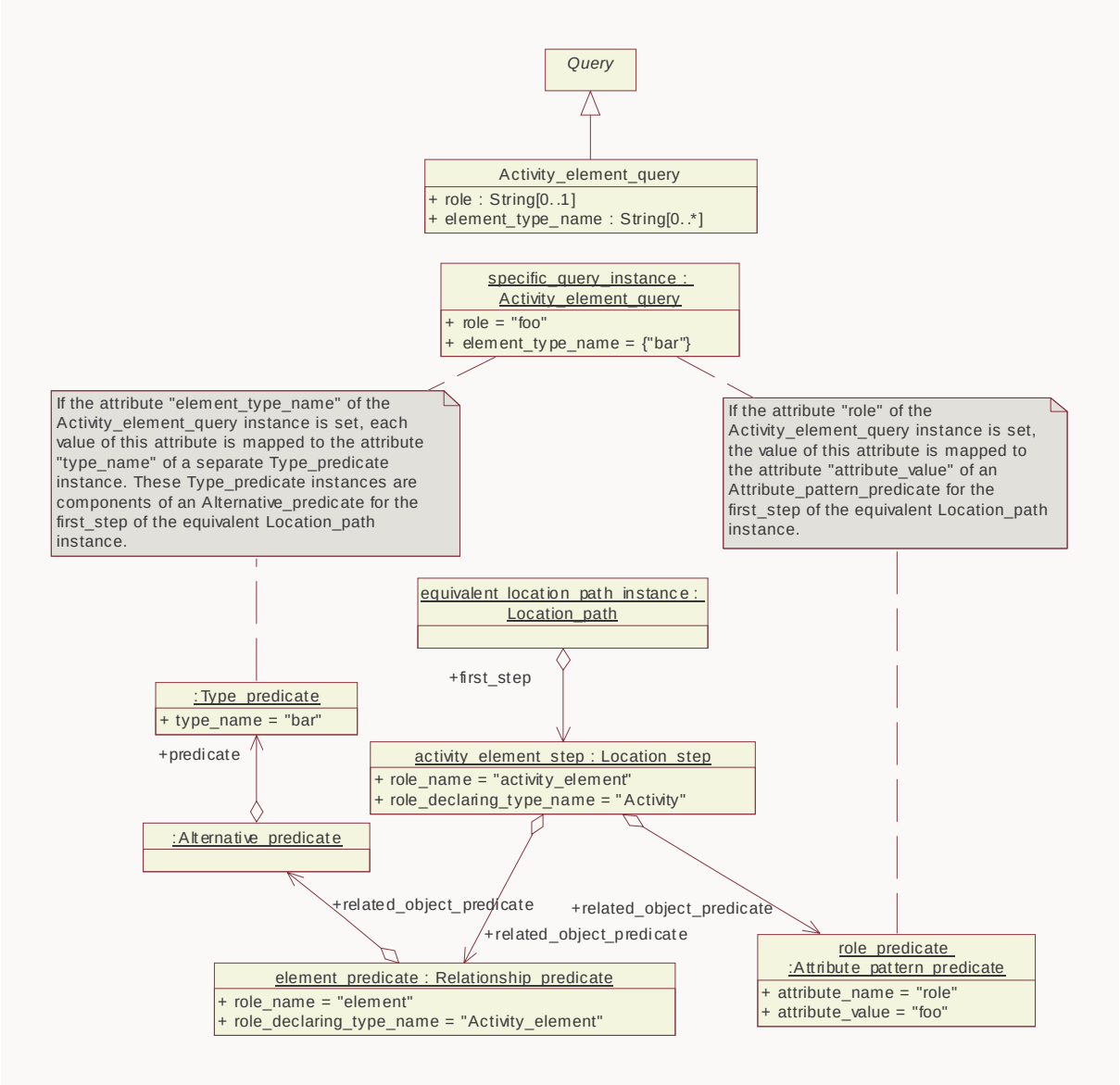


Figure 136 Definition, sample instance and equivalent Location_path instance of the Activity_element_query

8.14.3 Activity_relationship_query

The Activity_relationship_query traverses from Activity objects via Activity_relationship objects to Activity objects.

Parameters

- relation_type : String [0..1]
- maximum_recursion_number : Integer [0..1]

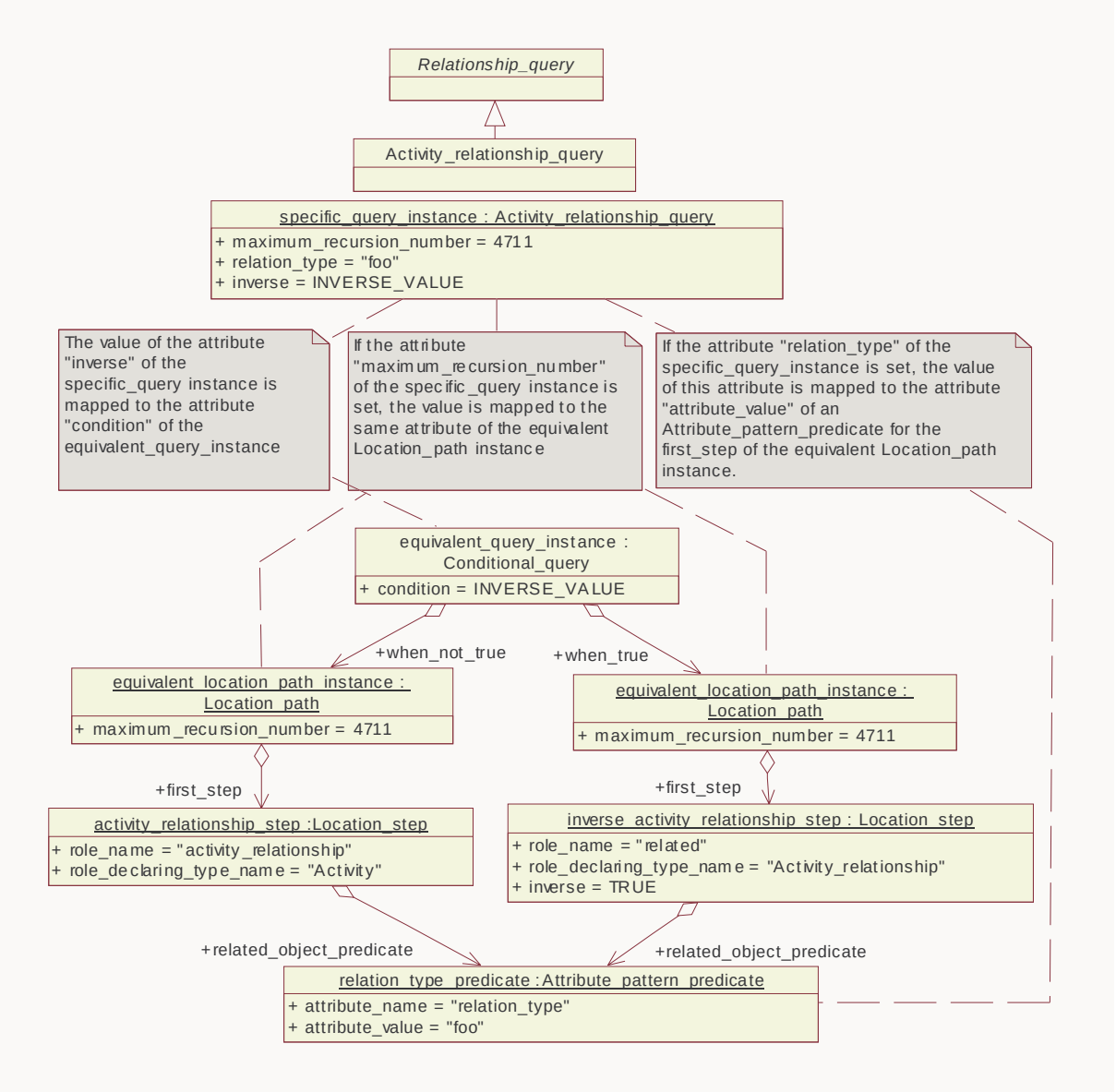


Figure 137 Definition, sample instance and equivalent Location_path instance of the Activity_relationship_query

8.14.4 Alias_identification_query

The Alias_identification_query traverses alias information from instances which implement the interface Alias_select.

Parameters

- alias_id: String [0..1]
- alias_version_id: String [0..1]

- alias_scope: String [0..1]

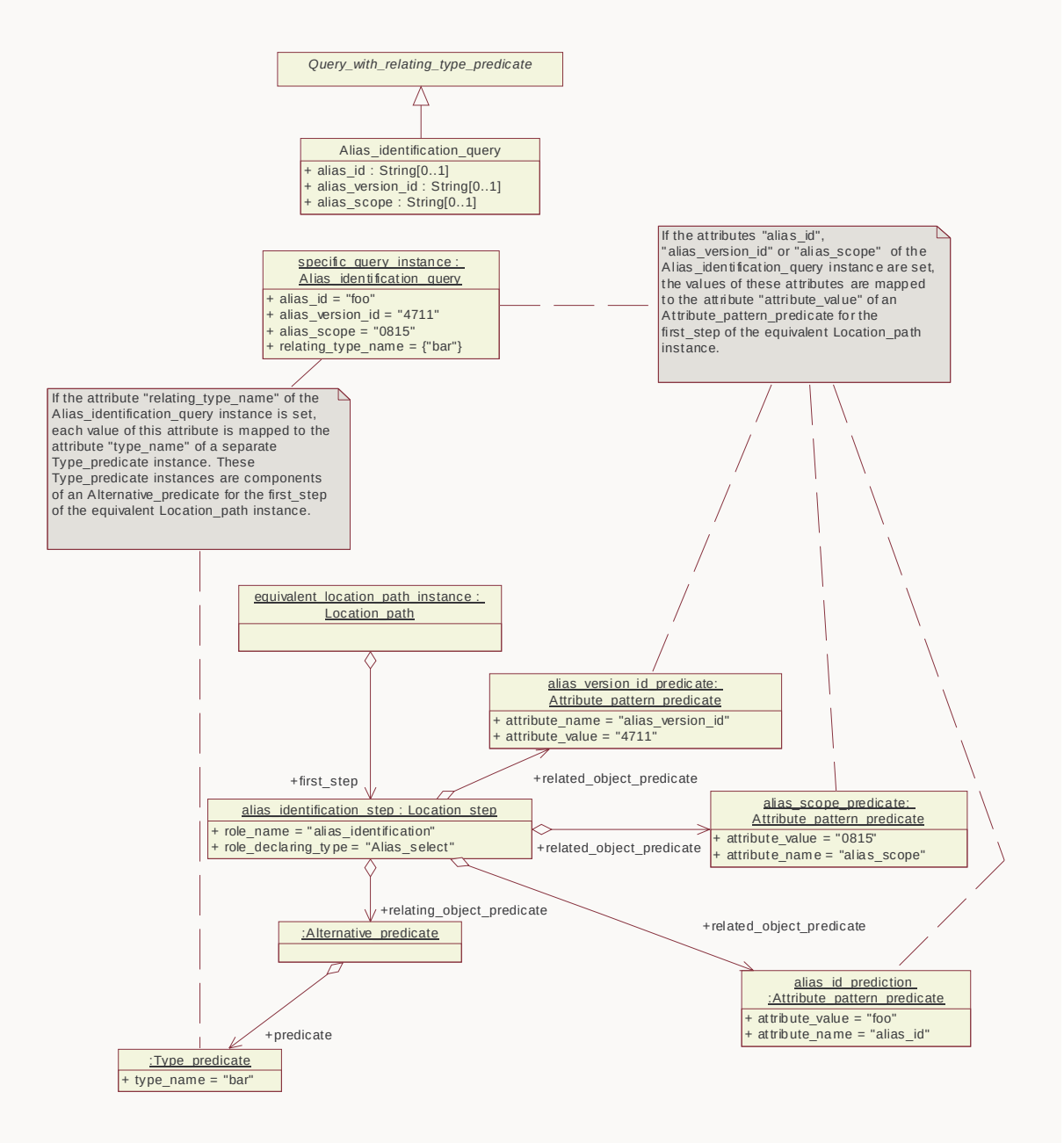


Figure 138 - Definition, sample instance and equivalent Location_path instance of the Alias_identification_query

8.14.5 Alternative_solution_query

The Alternative_solution_query traverses information from Complex_product_objects to Alternative_solution objects.

Parameters

- relating_type_name: String [0..*]

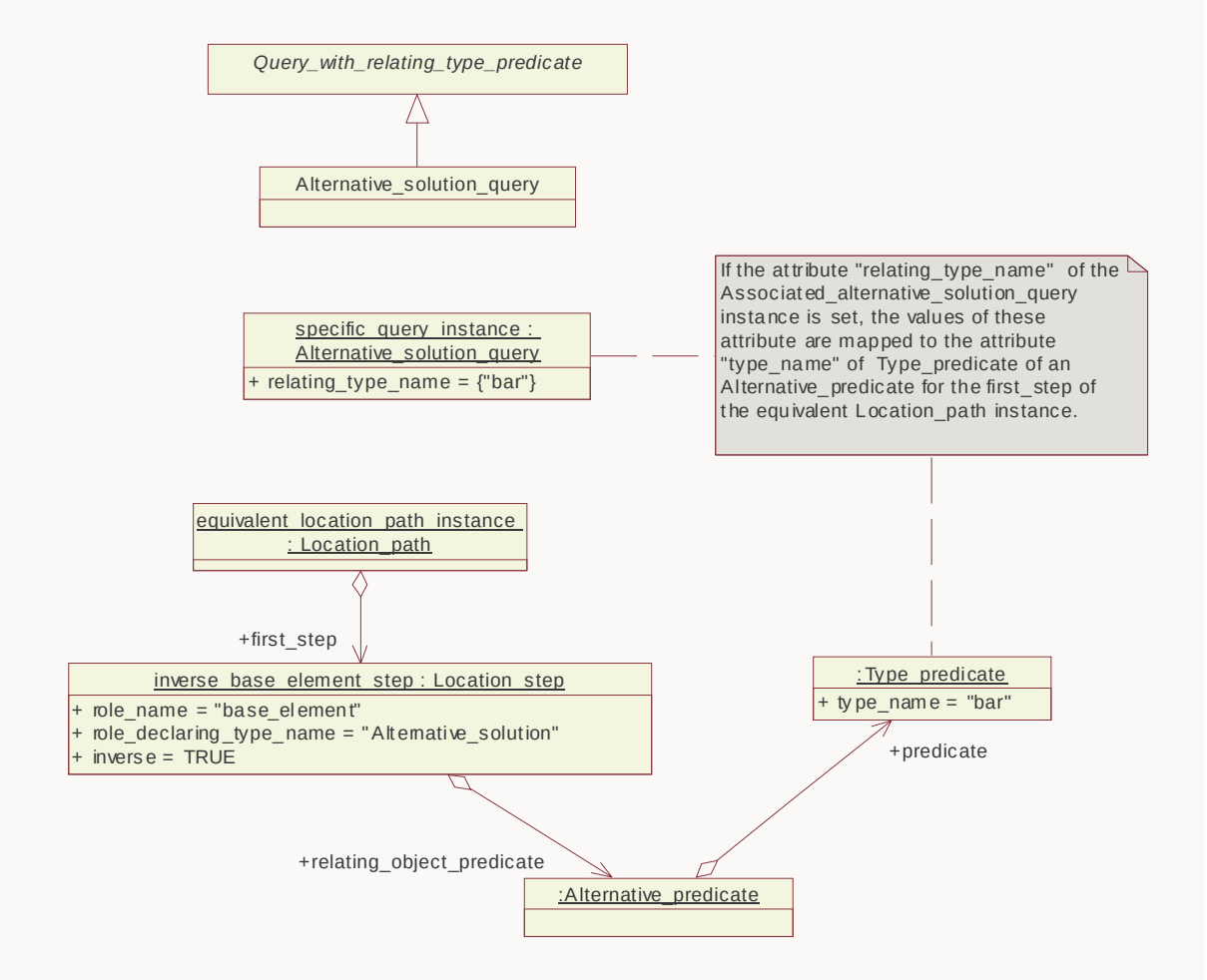


Figure 139 - Definition, sample instance and equivalent Location_path instance of the Alternative_solution_query

8.14.6 Application_context_query

The Application_context_query selects Application_context objects.

Parameters

- application_domain: String [0..1]
- life_cycle_stage: String [0..1]

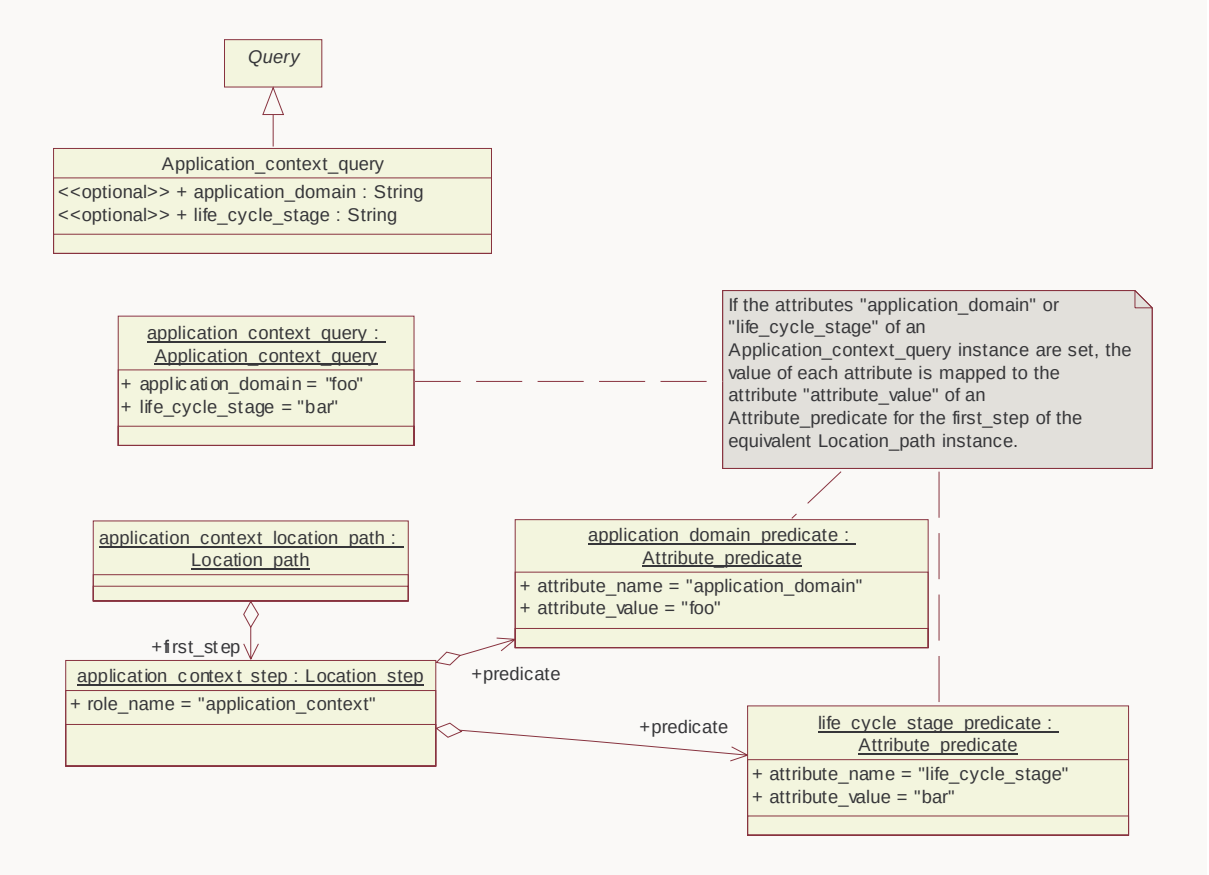


Figure 140 - Definition, sample instance and equivalent Location_path instance of the Application_context_query

8.14.7 Approval_relationship_query

The Approval_relationship_query traverses from Approval objects via Approval_relationship_objects to Approval objects.

Parameters

- relation_type: String [0..1]
- maximum_recursion_numer: Integer [0..1]
- inverse: Boolean [0..1]

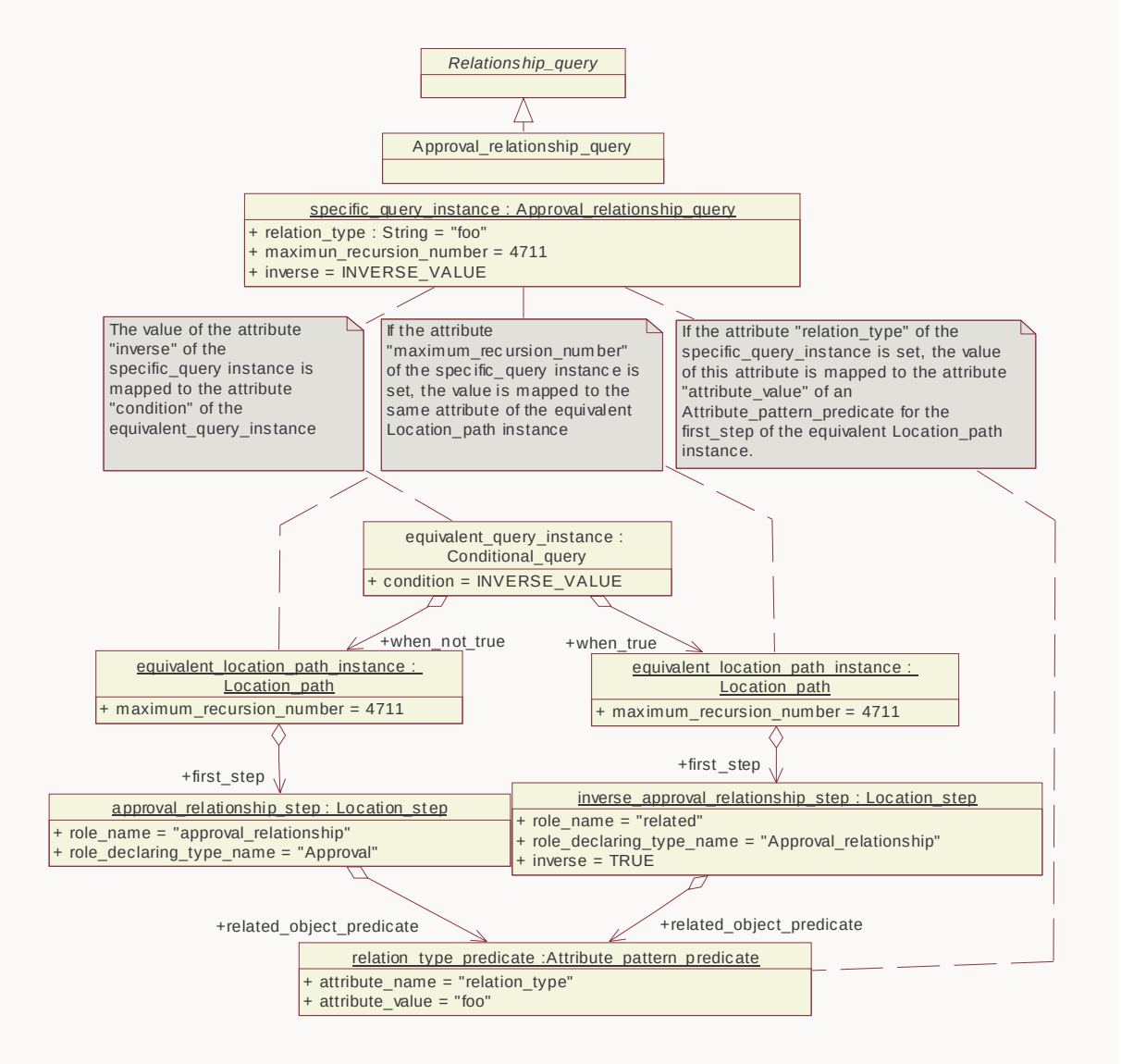


Figure 141 - Definition, sample instance and equivalent Location_path instance of the Approval_relationship_query

8.14.8 Assembly_component_placement_query

The Assembly_component_placement_query traverses from Assembly_component_relationship objects to Transformation_select objects.

Parameters

- none

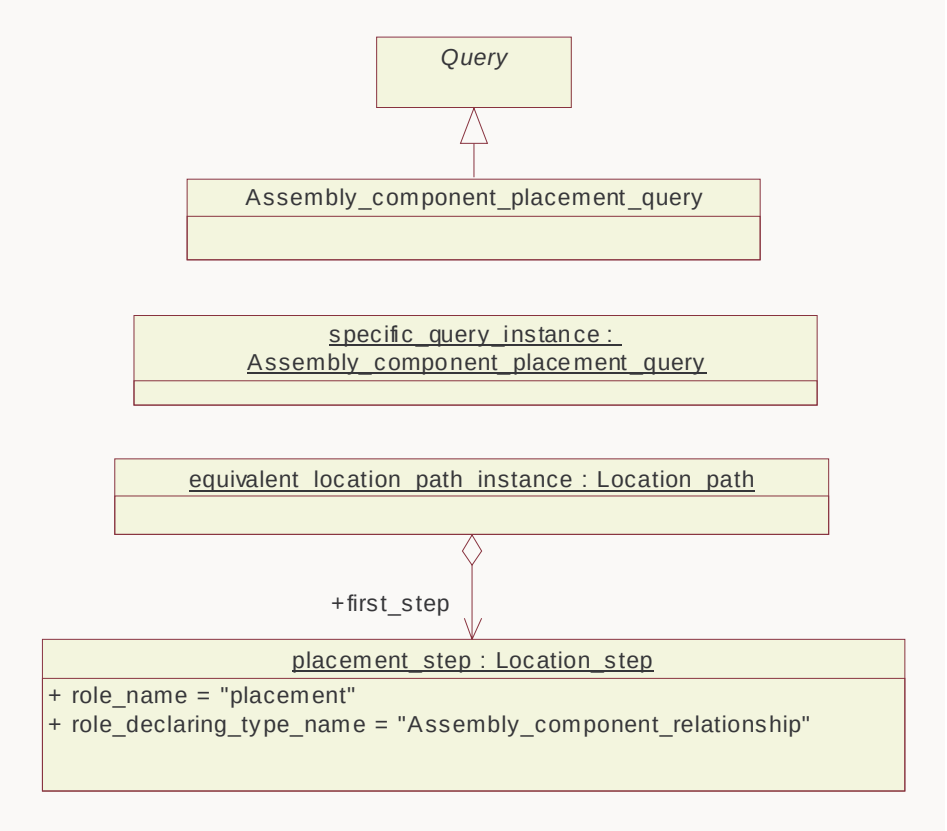


Figure 142 Definition, sample instance and equivalent Location_path instance of the Assembly_component_placememt_query

8.14.9 Assembly_structure_query

The Assembly_structure_query traverses the assembly structure from Assembly_definition objects.

Parameters

- maximum_recursion_number: Integer [0..1]limits the recursion level of the query.

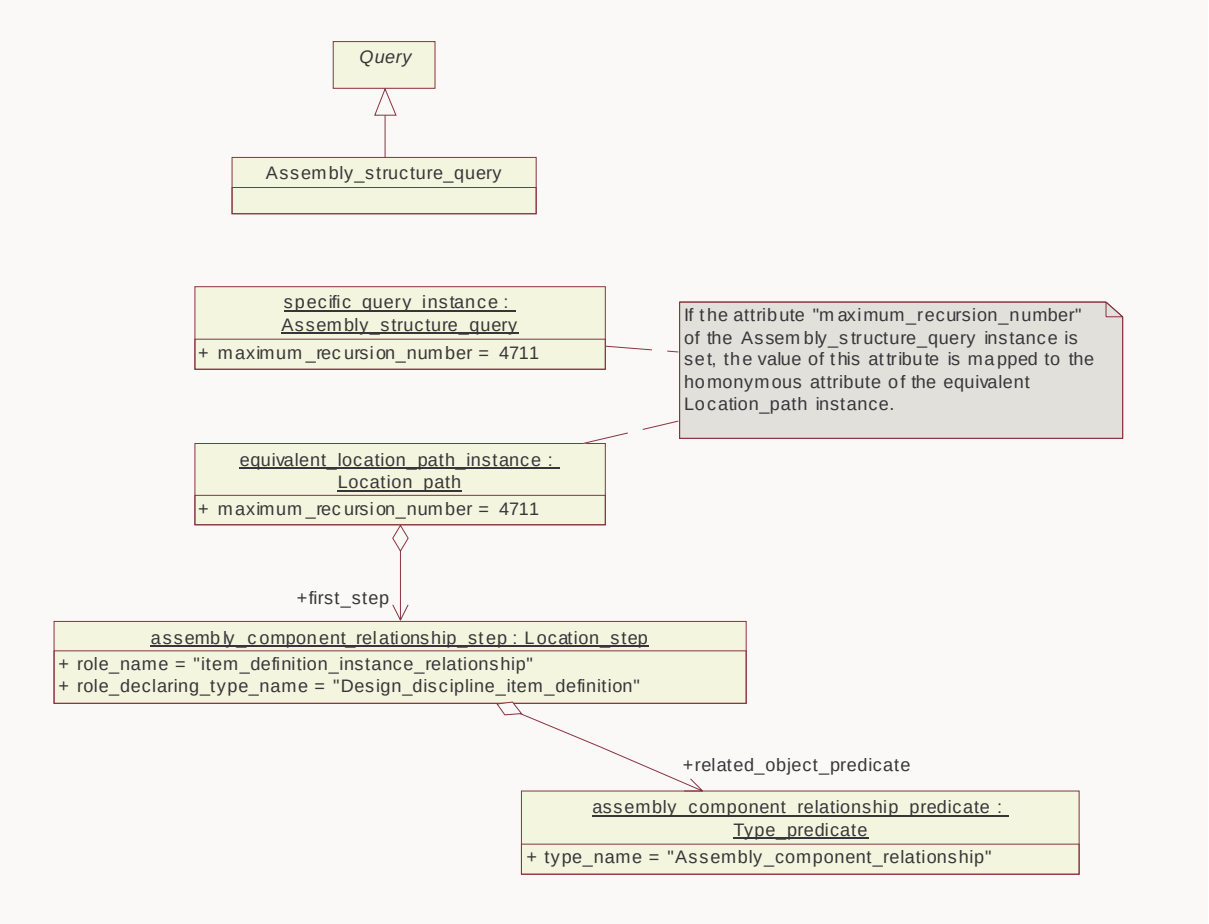


Figure 143 Definition, sample instance and equivalent Location_path instance of the Assembly_structure_query

8.14.10 Associated_activity_query

The Associated_activity_query traverses from Activity_element_select objects via Activity_element objects to Activity objects.

Parameters

- relation_type: String [0..1]
- relating_type_name: String [0..*]

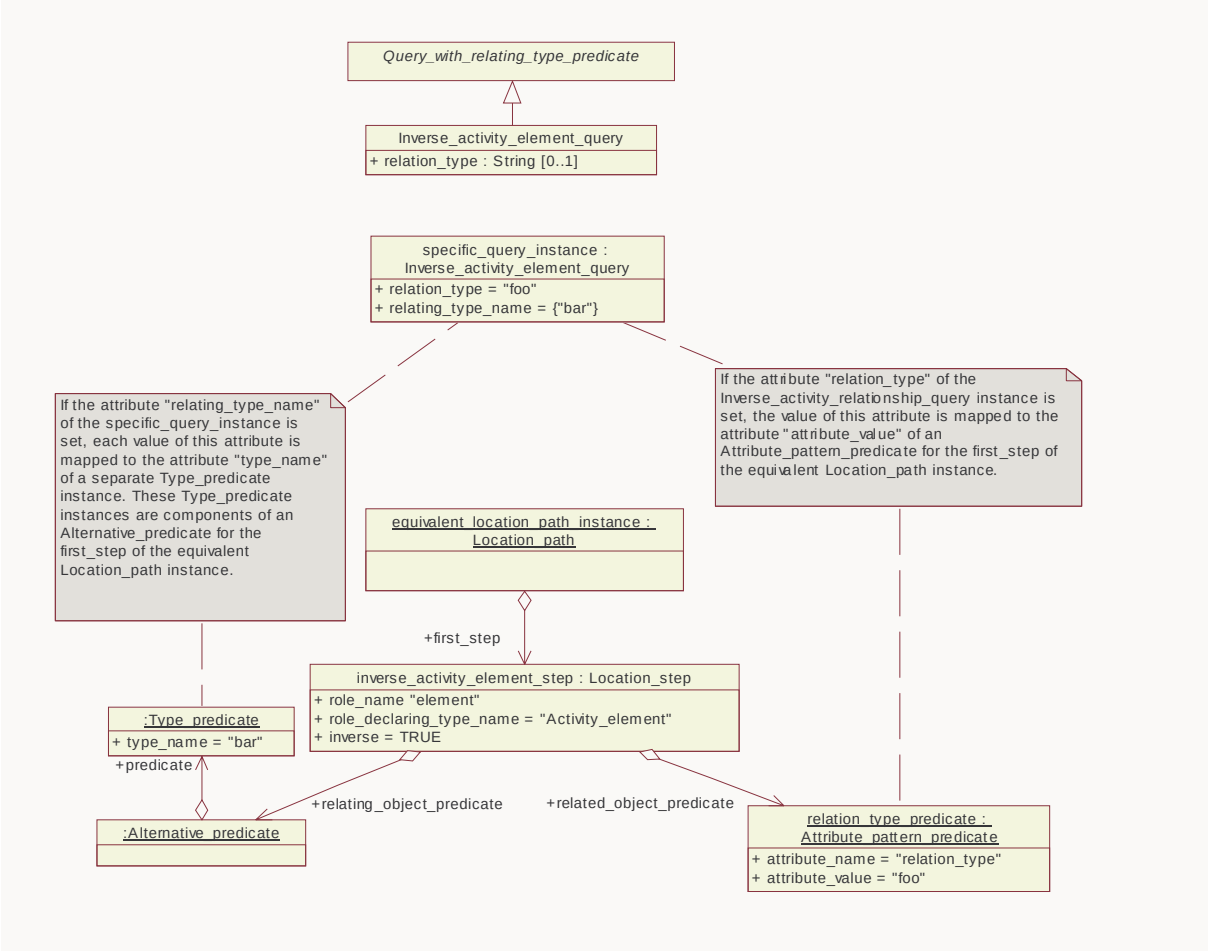


Figure 144 Definition, sample instance and equivalent Location_path instance of the Associated_activity_query

8.14.11 Associated_approval_query

The Associated_approval_query traverses from Approval_element_select objects to Approval_objects.

Parameters

- level: String [0..1]
- relating_type_name: String [0..*]

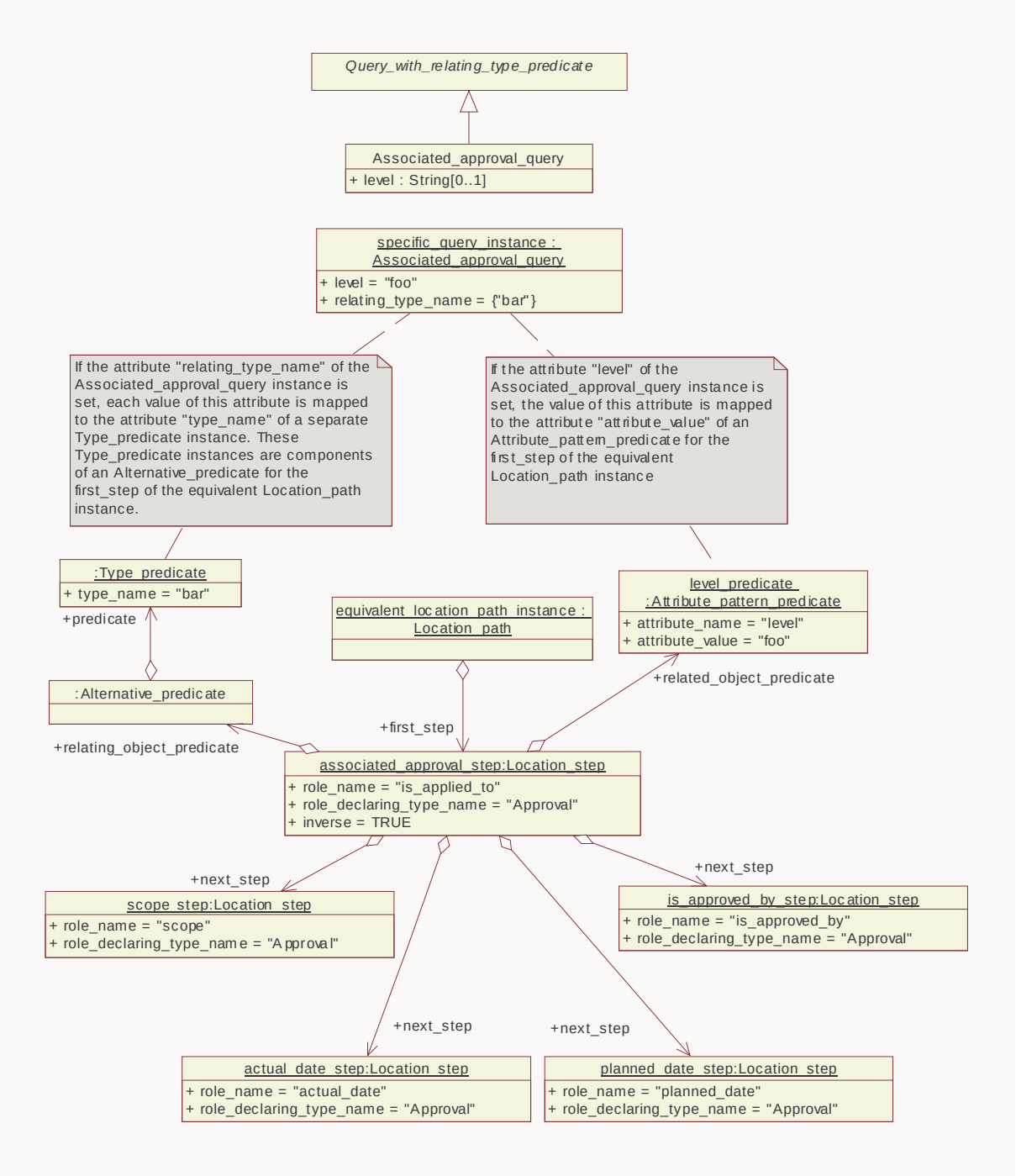


Figure 145 - Definition, sample instance and equivalent Location_path instance of the Associated_approval_query

8.14.12 Associated_classification_query

The Associated_classification_query traverses from Classified_element_select objects via Classification_association objects to General_classification objects.

Parameters

- role : String [0..1]
- relating_type_name : String [0..*]

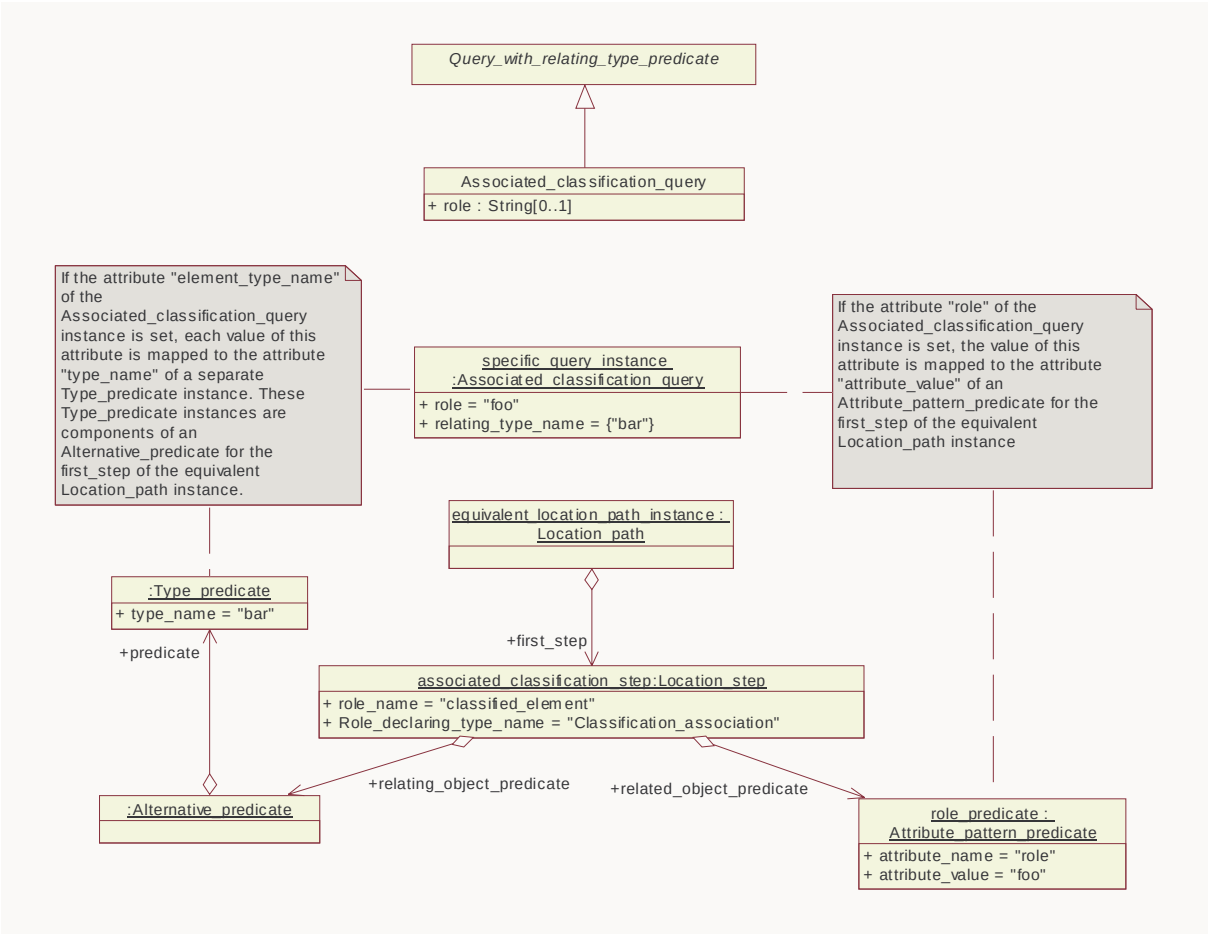


Figure 146 Definition, sample instance and equivalent Location_path instance of the Associated_classification_query

8.14.13 Associated_date_time_query

The Associated_date_time_query traverses from Date_time_person_organization_select objects objects via Date_time_assignment objects to Date_time objects..

Parameters

- role: String [0..1]
- relating_type_name: String [0..*]

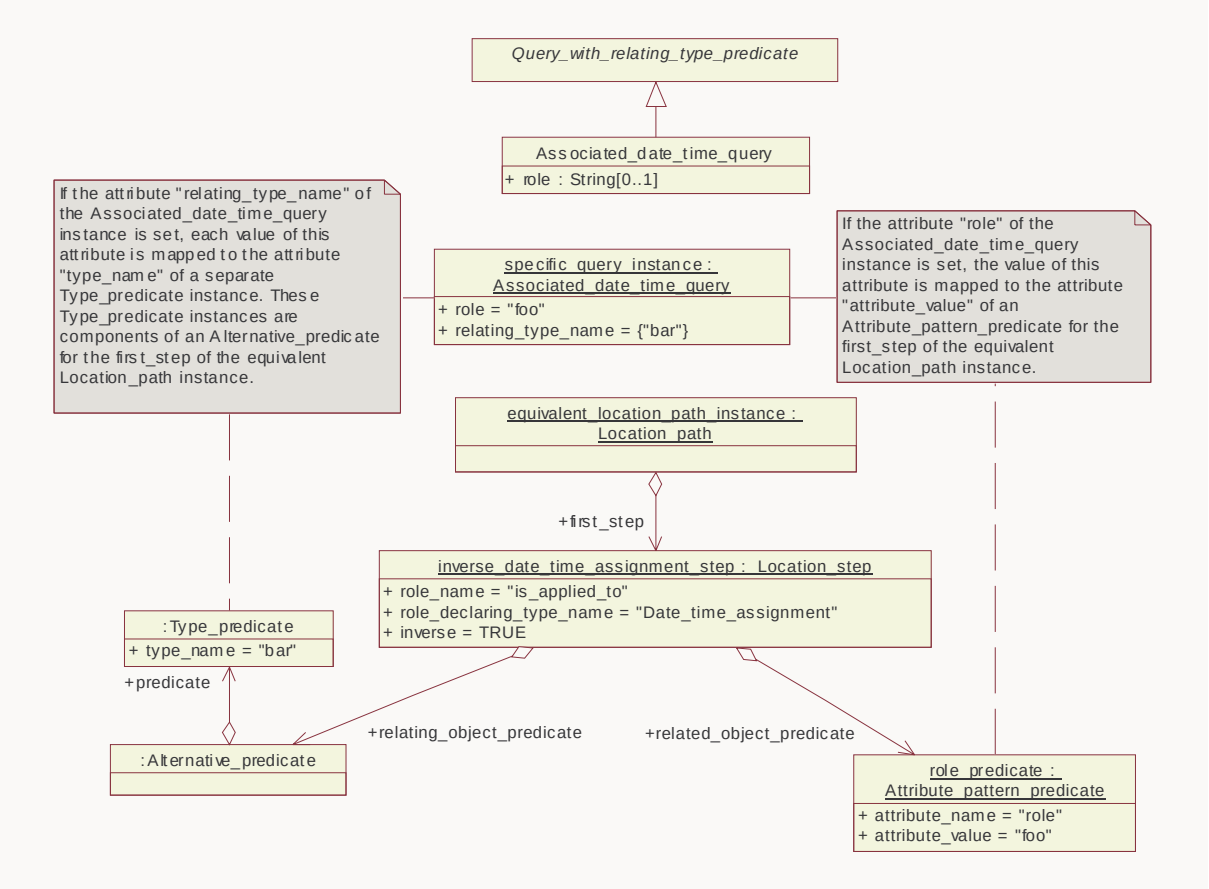


Figure 147 - Definition, sample instance and equivalent Location_path instance of the Associated_date_time_query

8.14.14 Associated_document_query

The Associated_document_query traverses from Document_element_select objects via Document_assignment objects to Assigned_document_select objects.

Parameters

- role: String [0..1]
- relating_type_name: String [0..1]

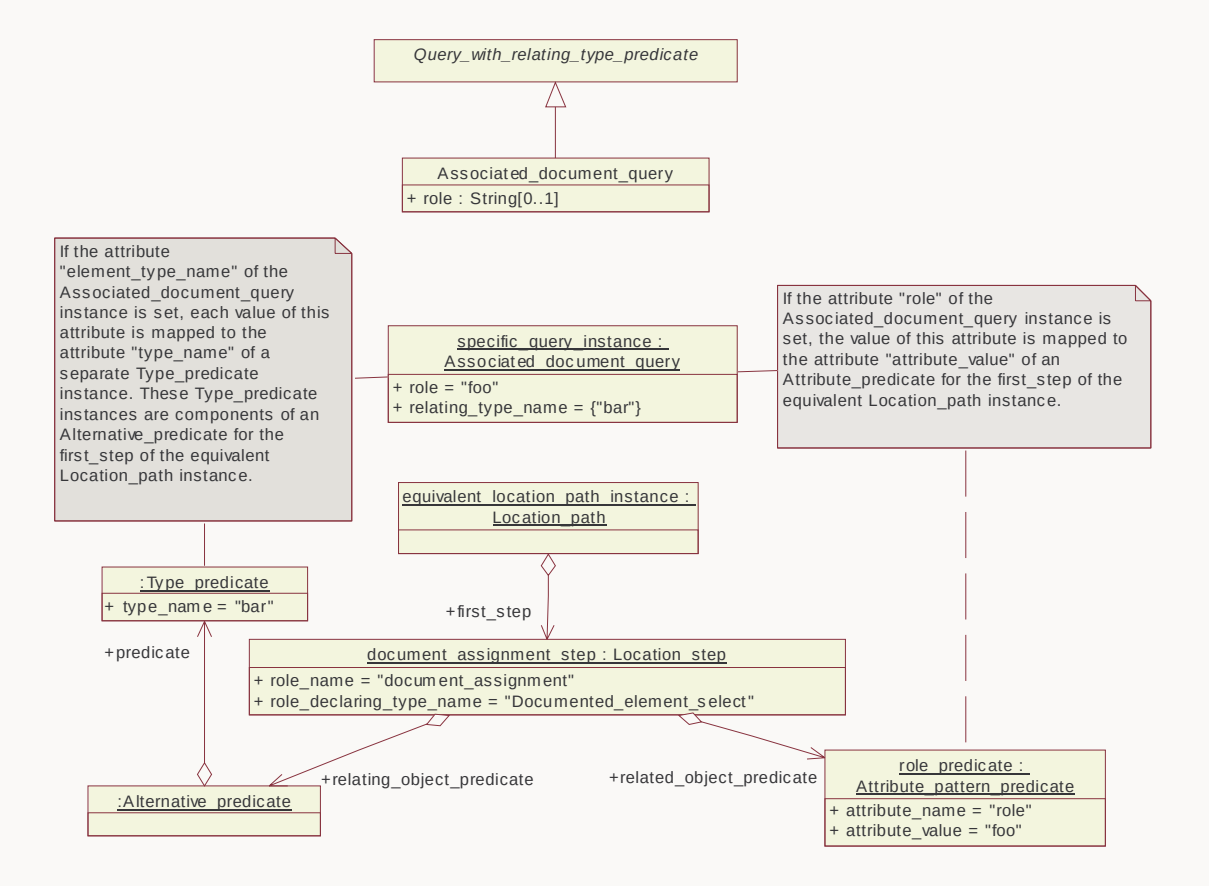


Figure 148 - Definition, sample instance and equivalent Location_path instance of the Associated_document_query

8.14.15 Associated_effectivity_query

The Associated_effectivity_query traverses from Effectivity_element_select objects via Effectivity_assignment objects to Effectivity objects.

Parameters

- role: String [0..1]
- relating_type_name: String [0..1]

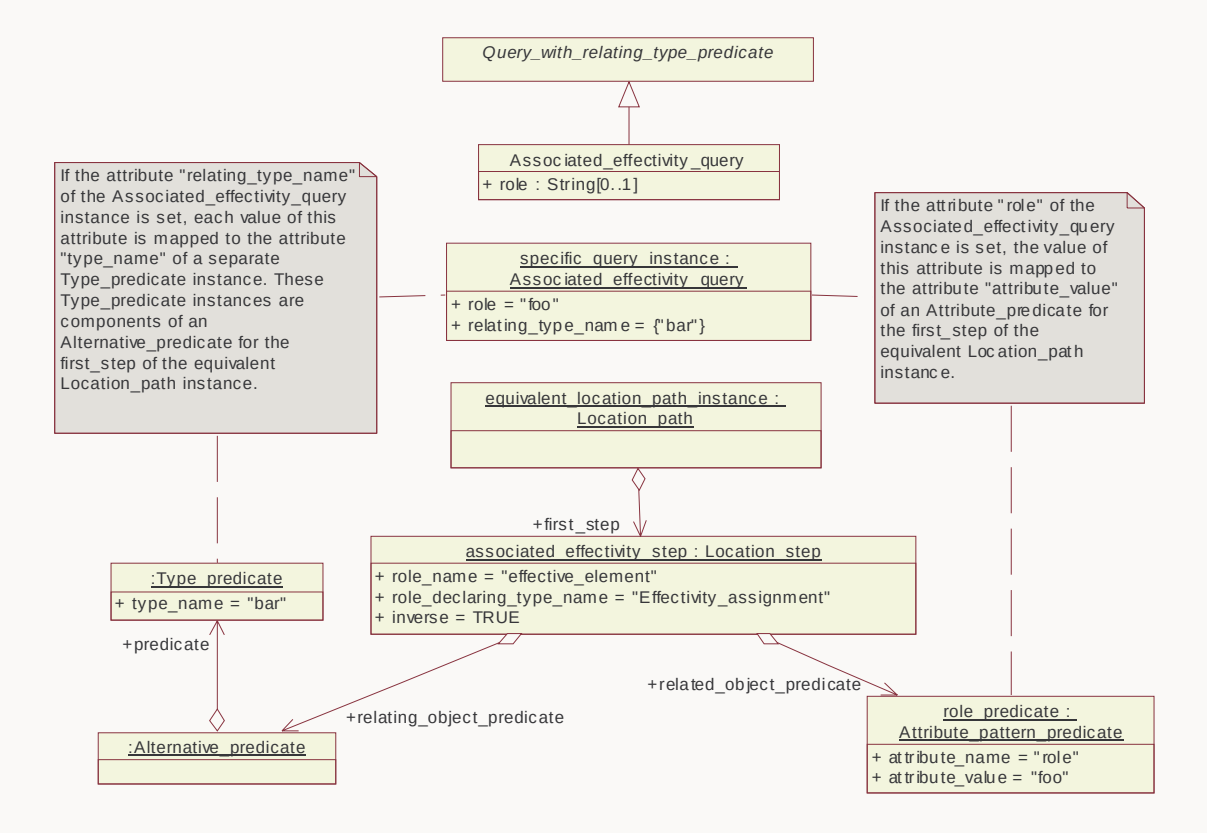


Figure 149 - Definition, sample instance and equivalent Location_path instance of the Associated_effectivity_query

8.14.16 Associated_file_query

The Associated_file_query traverses the external files and its properties from Document_representation objects.

The properties are Document_size_property, Document_format_property, Document_content_property, Document_file_id_and_location and Document_type_property.

Parameters

- none

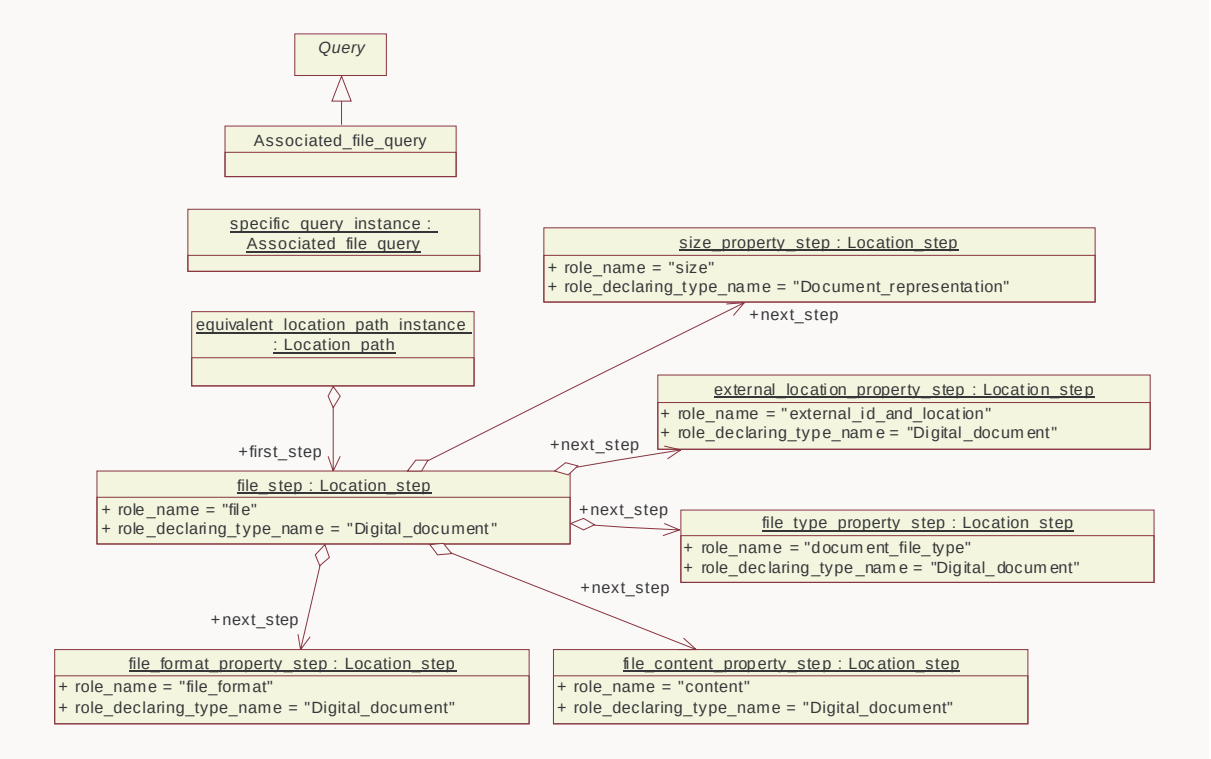


Figure 150 - Definition, sample instance and equivalent Location_path instance of the Associ-ated_file_query

8.14.17 Associated_item_property_query

The Associated_item_property_query traverses from Item_property_select objects via Property_value_association objects and Property_value_representation objects to the associated Property_value objects.

Parameters

- value_name : String [0..1]
- relating_type_name : String [0..*]

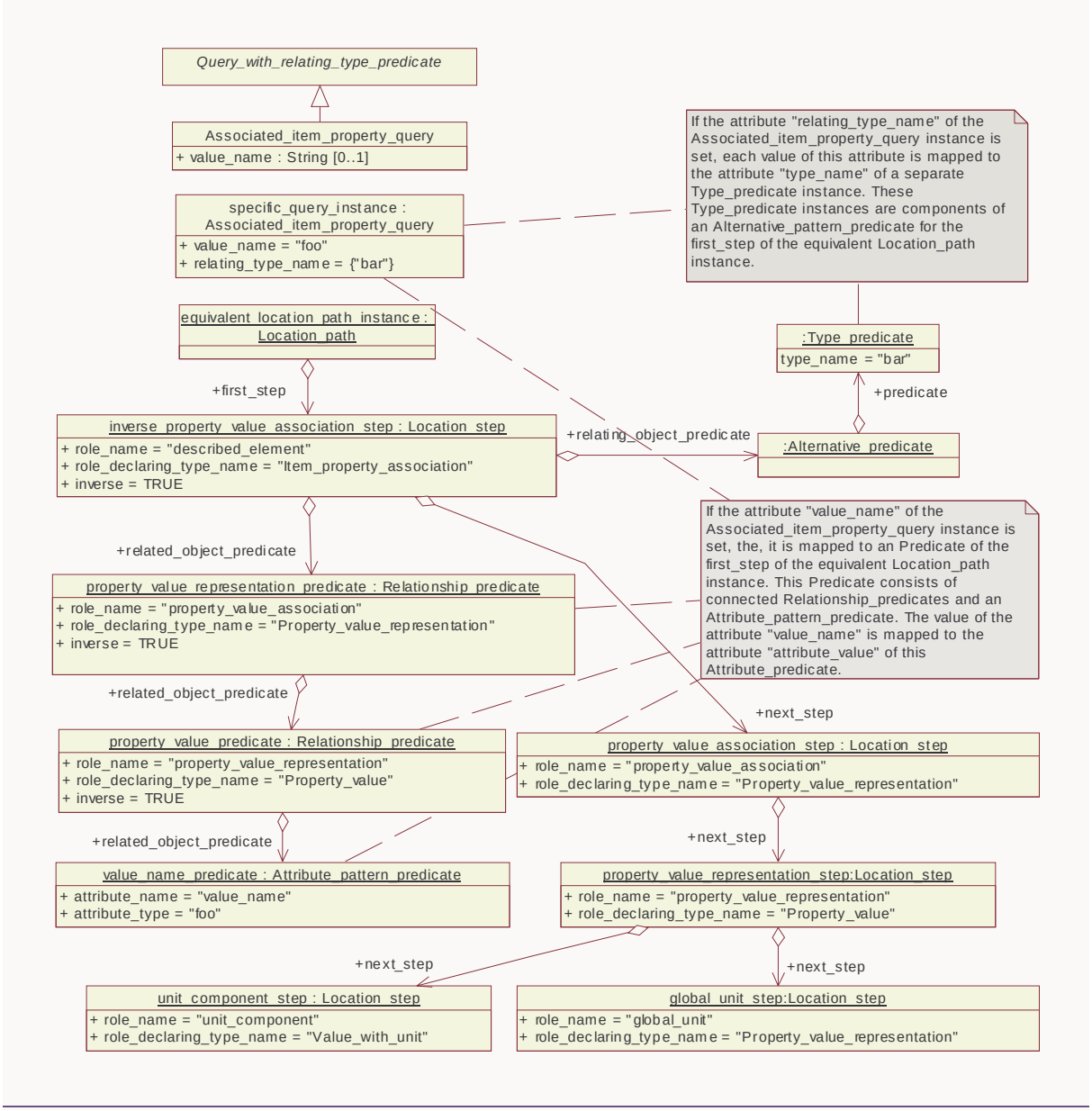


Figure 151 Definition, sample instance and equivalent Location_path instance of the Associated_item_property_query

8.14.18 Associated_person_organization_query

The Associated_organization_query traverses from Date_time_person_organization_element_select objects via Person_organization_assignment objects to Person_organization_select objects

Parameters

- role: String [0..1]

- relating_type_name: String [0..*]

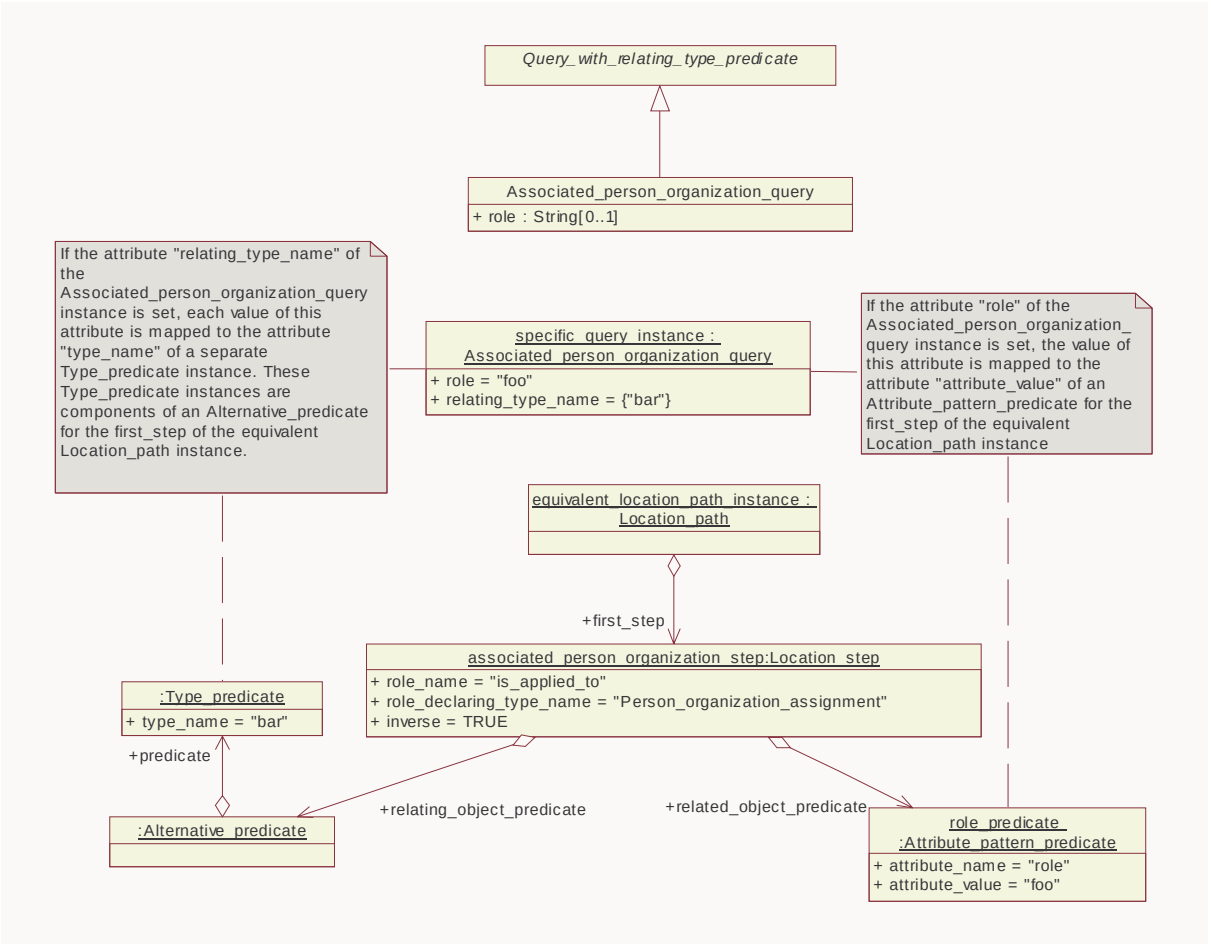


Figure 152 - Definition, sample instance and equivalent Location_path instance of the Associated_person_organization_query

8.14.19 Associated_process_property_query

The Associated_process_property_query traverses from Process_property_select objects via Property_value_association objects and Property_value_representation objects to the associated Property_value objects.

Parameters

- value_name : String [0..1]
- relating_type_name : String [0..*]

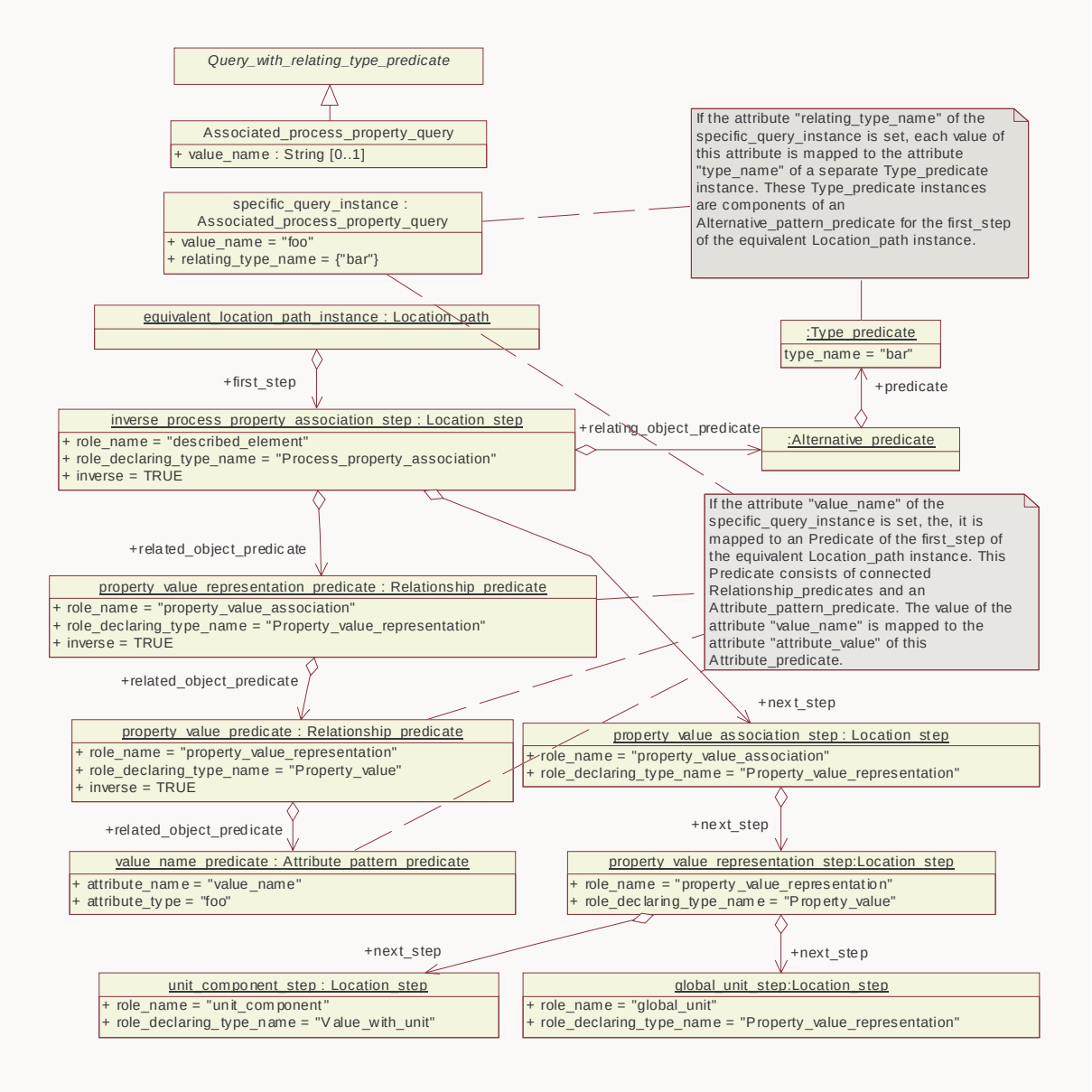


Figure 153 Definition, sample instance and equivalent Location_path instance of the Associated_process_property_query

8.14.20 Associated_project_query

The Associated_project_query traverses from Project_information_select objects via Project_assignment objects to Project objects.

Parameters

- role : String [0..1]

- relating_type_name : String [0..*]

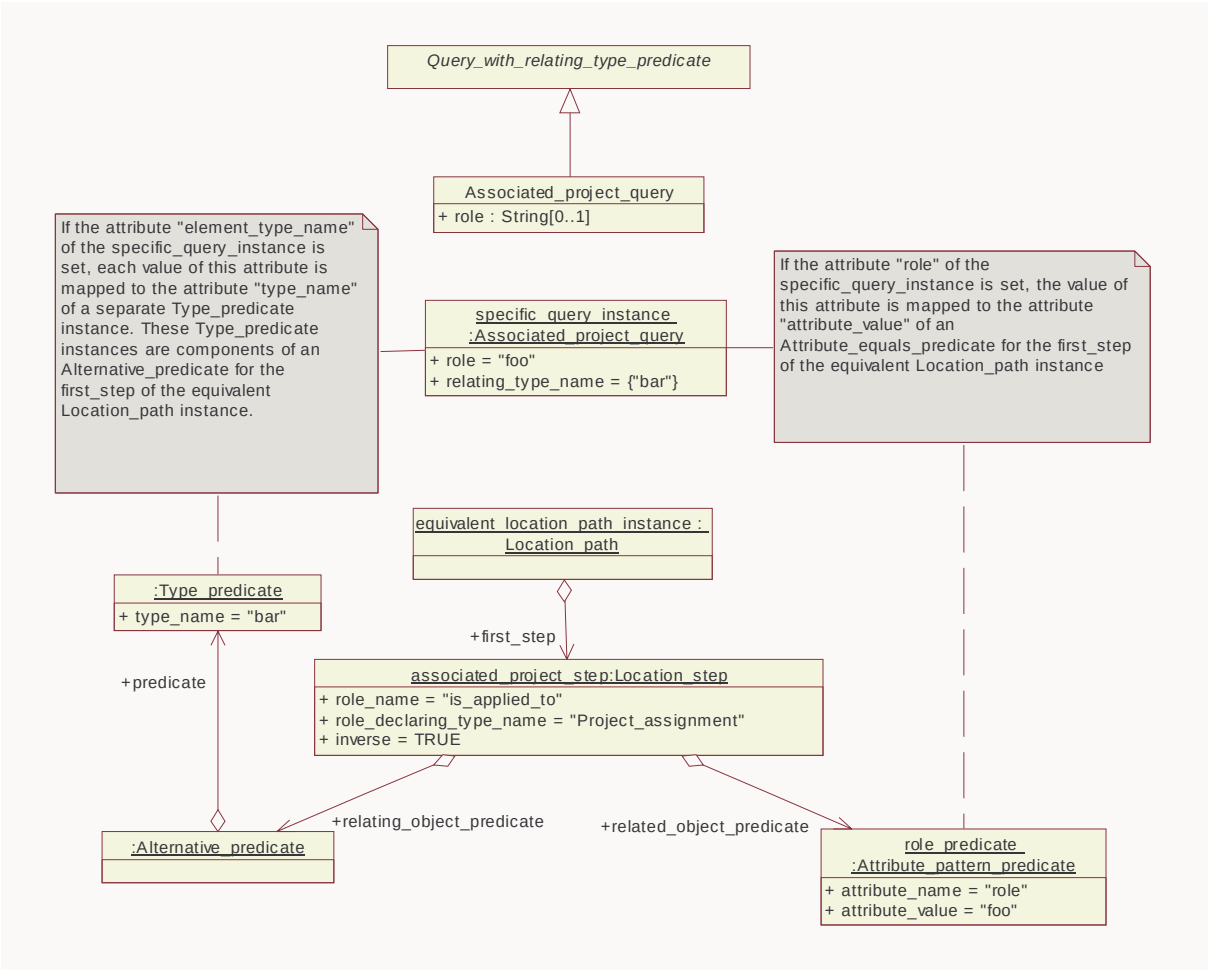


Figure 154 Definition, sample instance and equivalent Location_path instance of the Associated_project_query

8.14.21 Associated_property_query

The Associated_property_query traverses from Item_property_select and Process_property_select objects via Property_value_association objects and Property_value_representation objects to the associated Property_value objects.

The Associated_property_query is defined as a Batch_query of an Associated_item_property_query and an Associated_process_property_query.

Parameters

- value_name: String [0..1]
- relating_type_name: String [0..1]

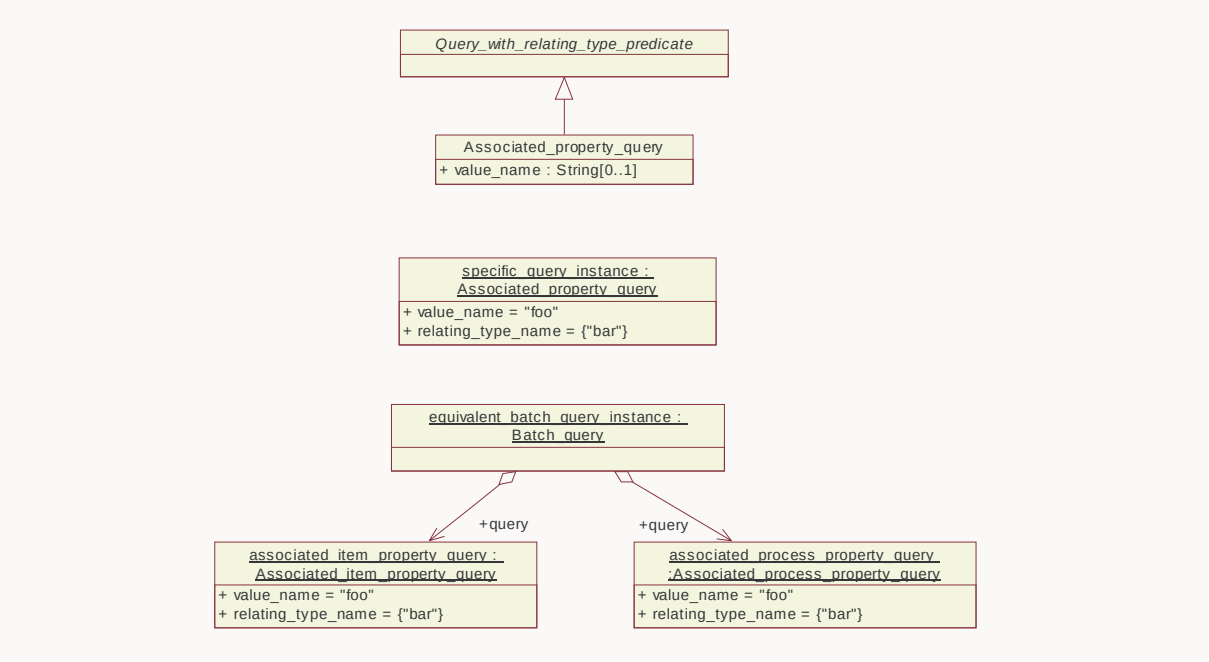


Figure 155 - Definition, sample instance and equivalent Batch_query instance of the Associated_property_query

8.14.22 Class_structure_query

The Class_structure_query traverses from Product_class objects via Class_structure_relationship objects to Product_function_component_select objects.

Parameters

- relation_type: String [0..1]

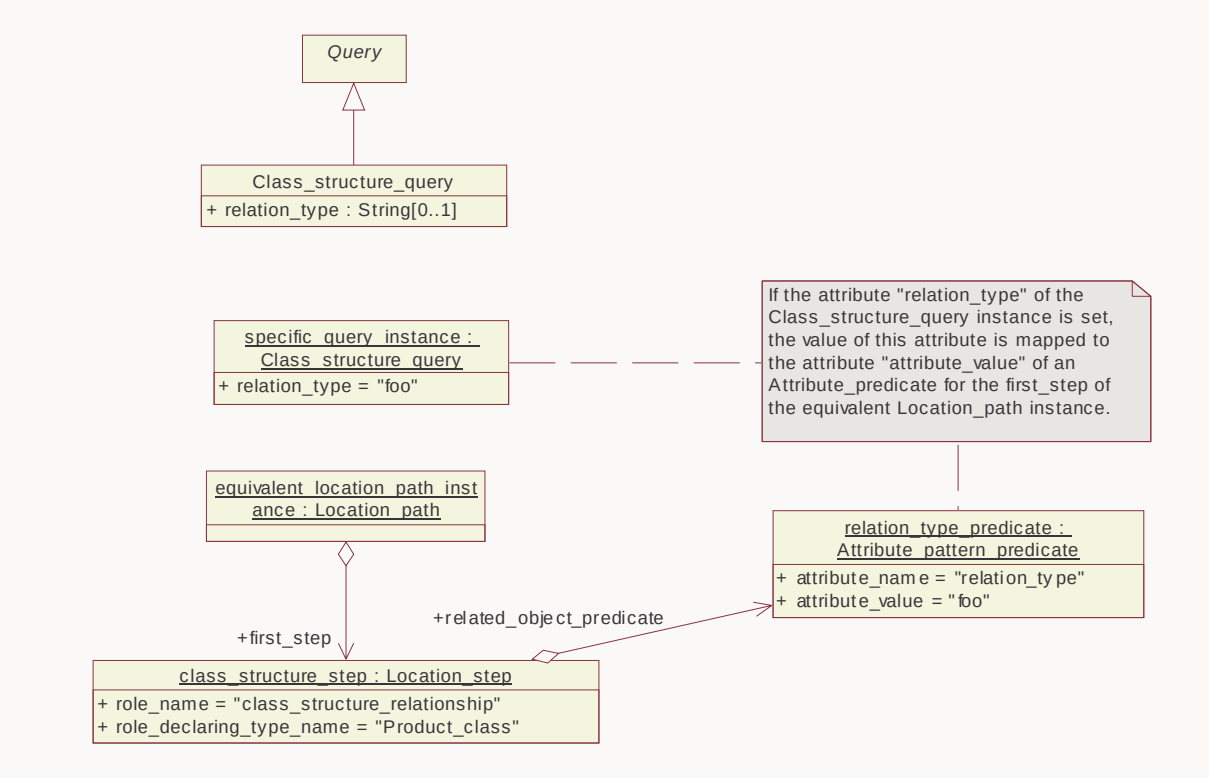


Figure 156 - Definition, sample instance and equivalent Location_path instance of the Class_structure_query

8.14.23 Complex_product_query

The Complex_product_query selects Complex_product objects by its id and version_id atrtributes.

Parameters

- id: String [0..1]
- id_scope: String [0..1]
- version_id: String [0..1]

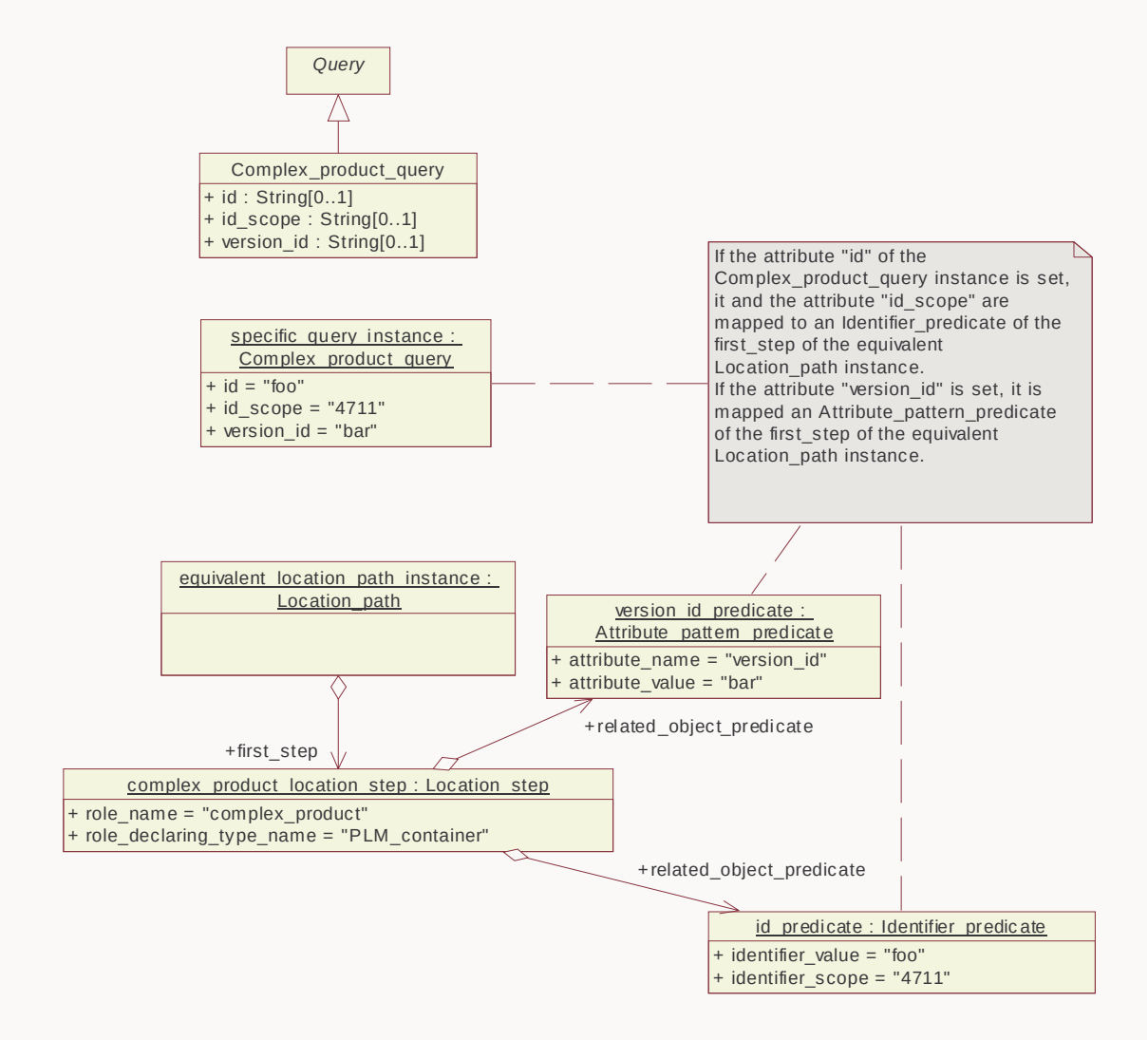


Figure 157 - Definition, sample instance and equivalent Location_path instance of the Complex_product_query

8.14.24 Configuration_query

The Configuration_query traverses from Configured_item_select objects via Configration objects to Configuration_specification_select objects.

Parameters

- configuration_type: String [0..1]
- inheritance_type: String [0..1]
- relating_type_names: String [0..*]

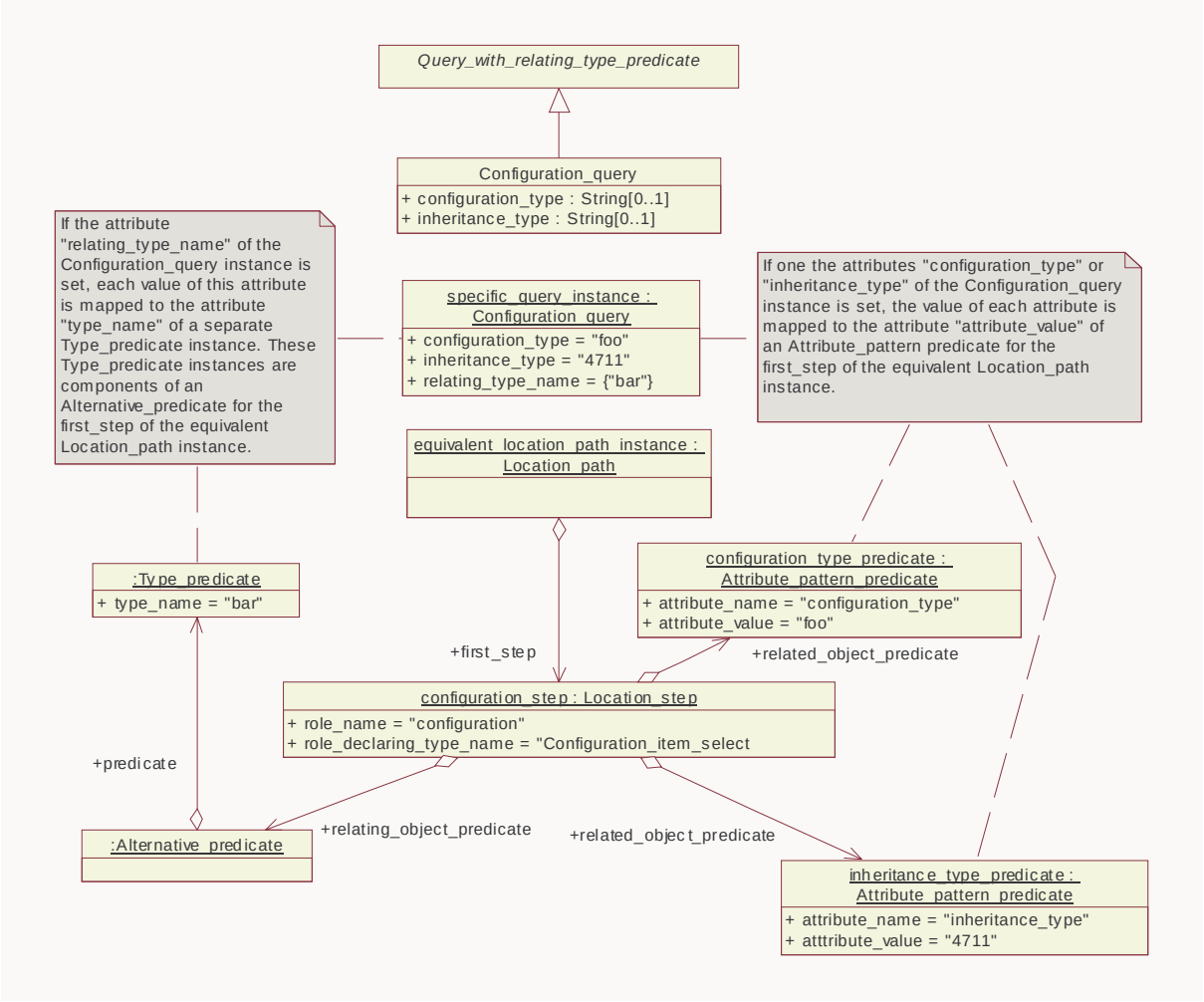


Figure 158 - Definition, sample instance and equivalent Location_path instance of the Configuration_query

8.14.25 Design_discipline_item_definition_query

The Design_discipline_item_definition_query traverses from Item_version objects to Design_discipline_item_definition objects.

Parameters

- id: String [0..1]
- application_domain: String [0..1]
  traverse only Design_discipline_item_definition objects which relates via their initial_context association to an Application_context object with an applica-tion_domain attribute of the given value
- life_cycle_stage: String [0..1]
  traverse only Design_discipline_item_definition objects which relates via their initial_context association to an Application_context object with an life_cycle_stage attribute of the given value

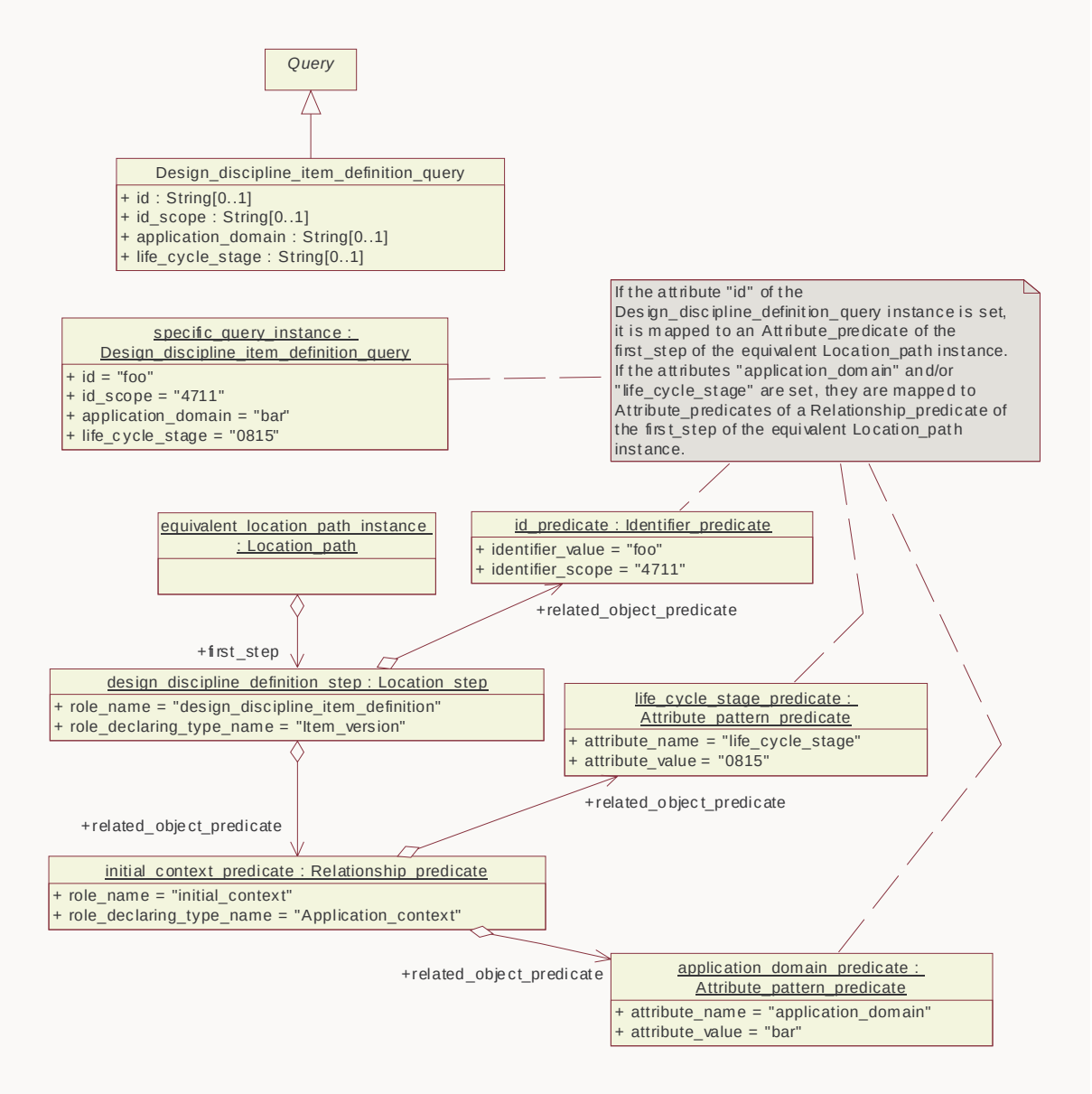


Figure 159 - Definition, sample instance and equivalent Location_path instance of the Design_discipline_item_definition_query

8.14.26 Document_classification_query

The Document_classification_query traverses from Document objects to Specific_document_classification objects.

Parameters

- classification_name: String [0..1]

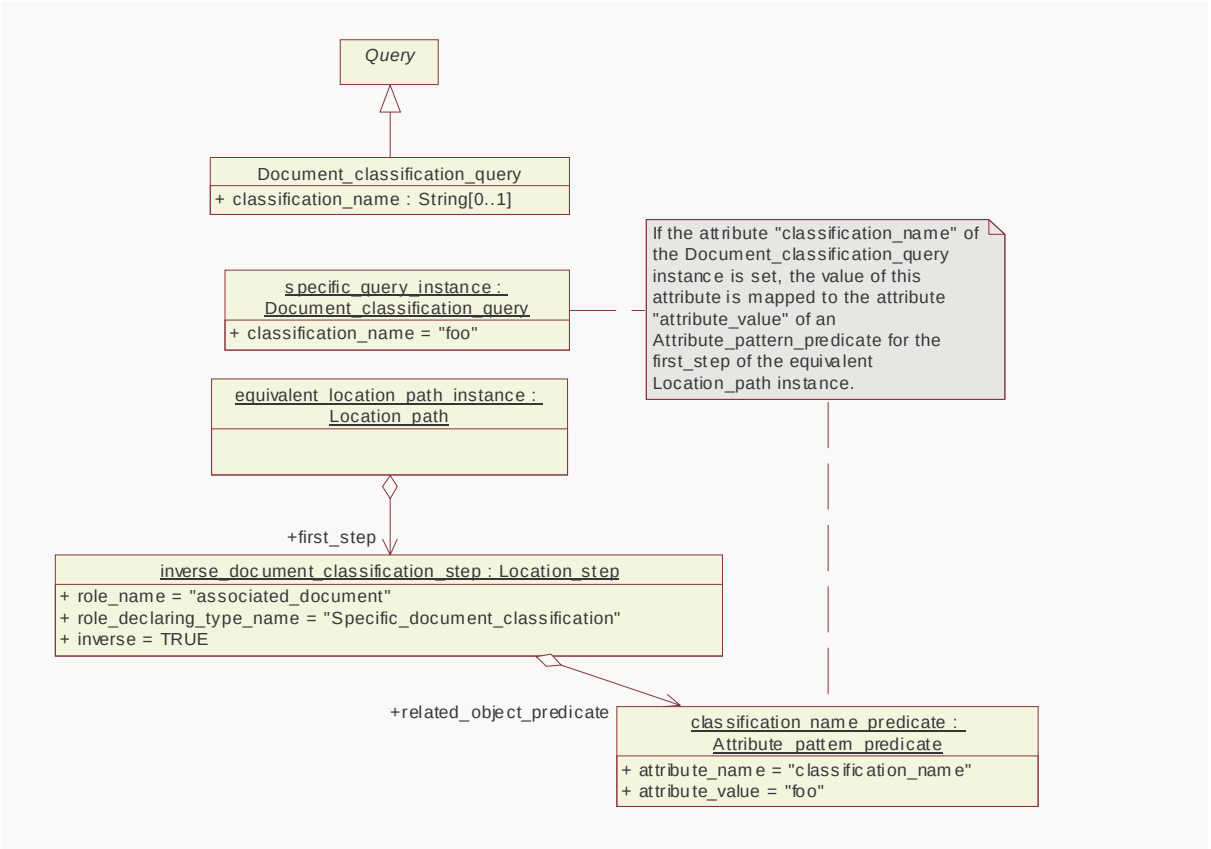


Figure 160 - Definition, sample instance and equivalent Location_path instance of the Document_classification_query

8.14.27 Document_property_query

The Document_property_query traverses the document properties from Document_representation objects.

These properties are Document_size_property, Document_format_property, Document_content_property, and Document_file_id_and_location.

Parameters

- none

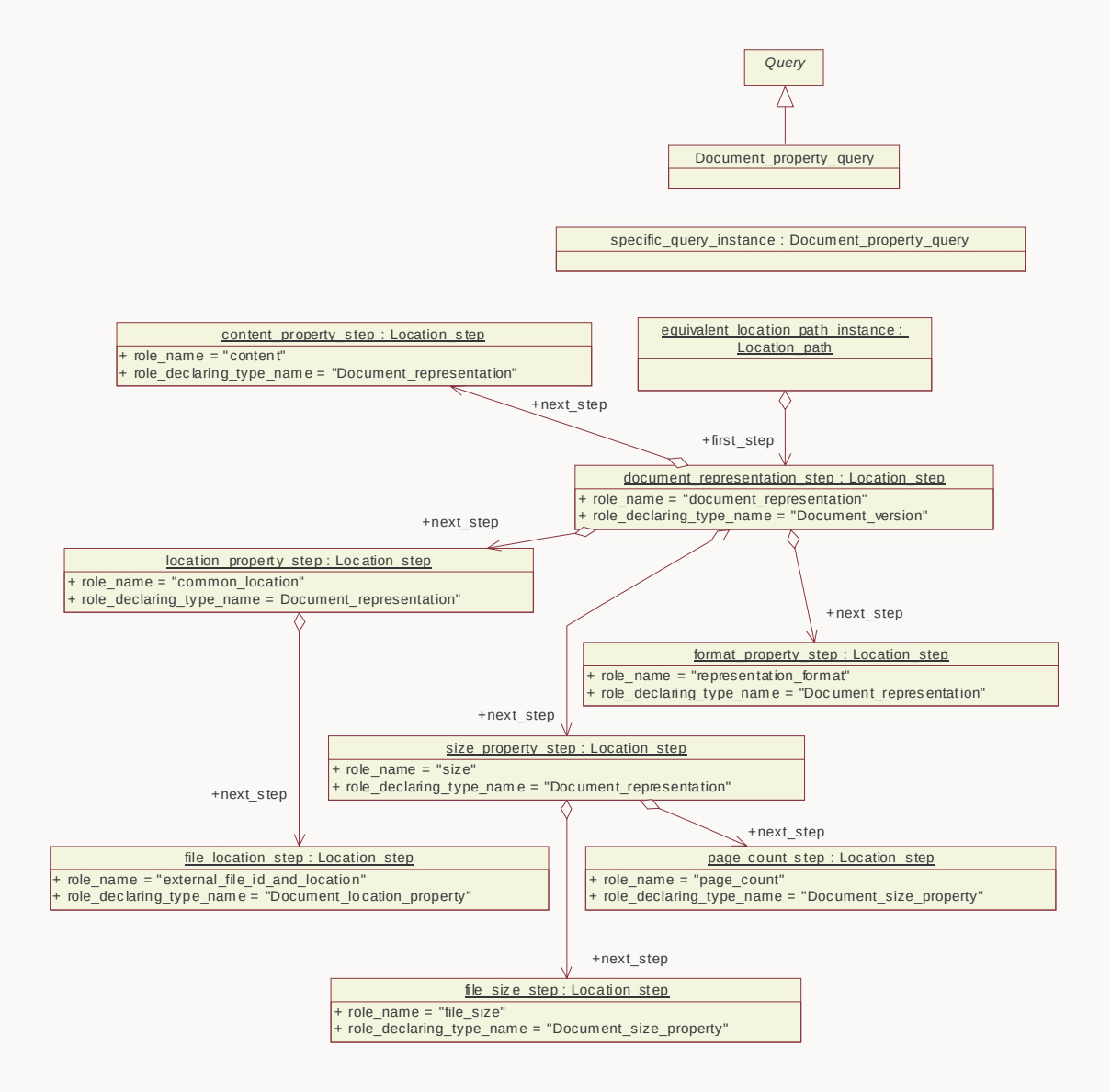


Figure 161 - Definition, sample instance and equivalent Location_path instance of the Document_property_query

8.14.28 Document_query

The Document_query selects Document objects.

Parameters

- document_id: String [0..1]
- document_id_scope: String [0..1]

- name: String [0..1]
- name_language: Language[0..1]
- version_id: String [0..1]
- classification_name: String [0..1]

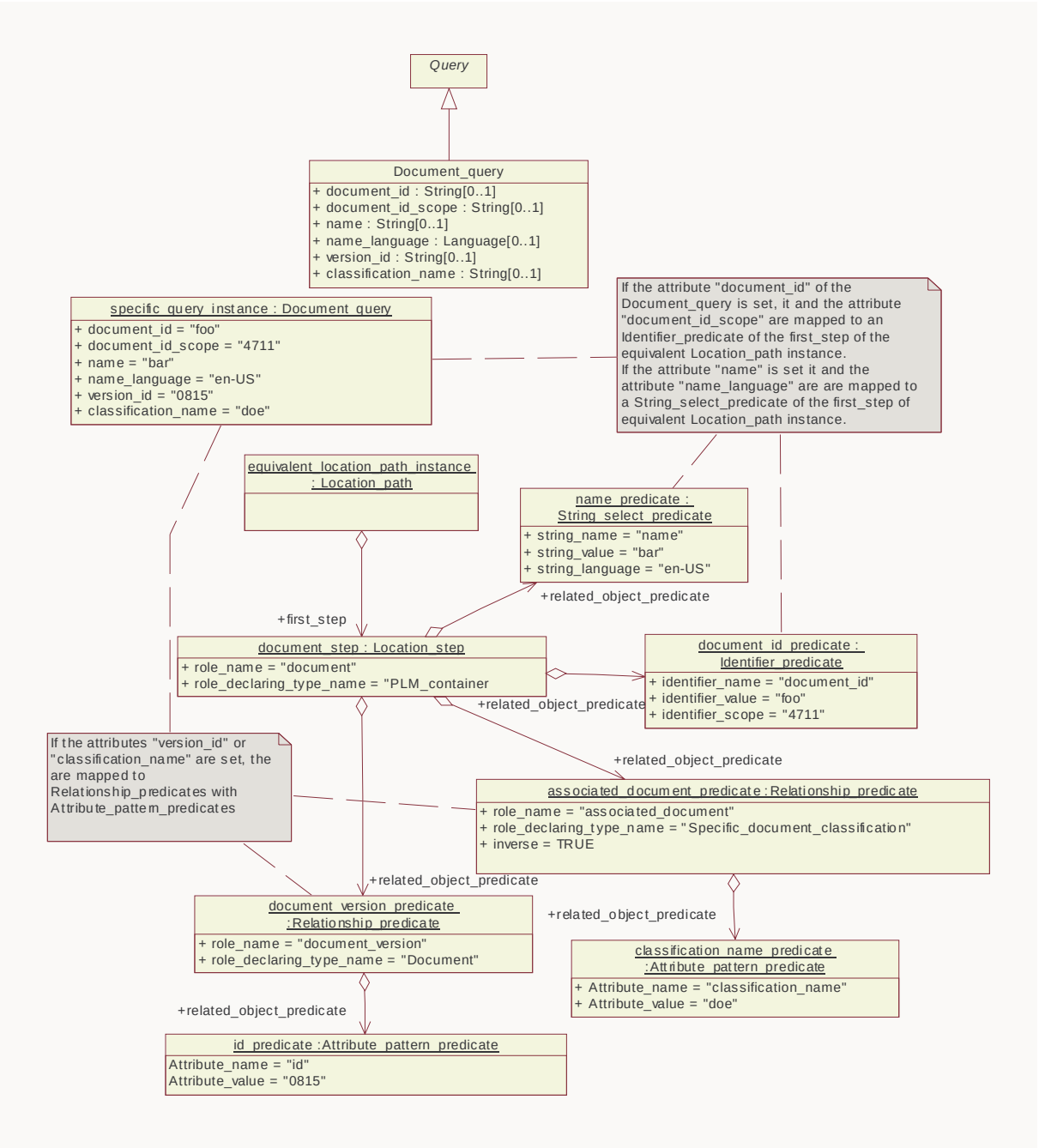


Figure 162 - Definition, sample instance and equivalent Location_path instance of the Document_query

8.14.29 Document_representation_query

The Document_representation_query traverses Document_representation objects from Document_version objects.

Parameters

- id: String [0..1]
- id_scope: String [0..1]

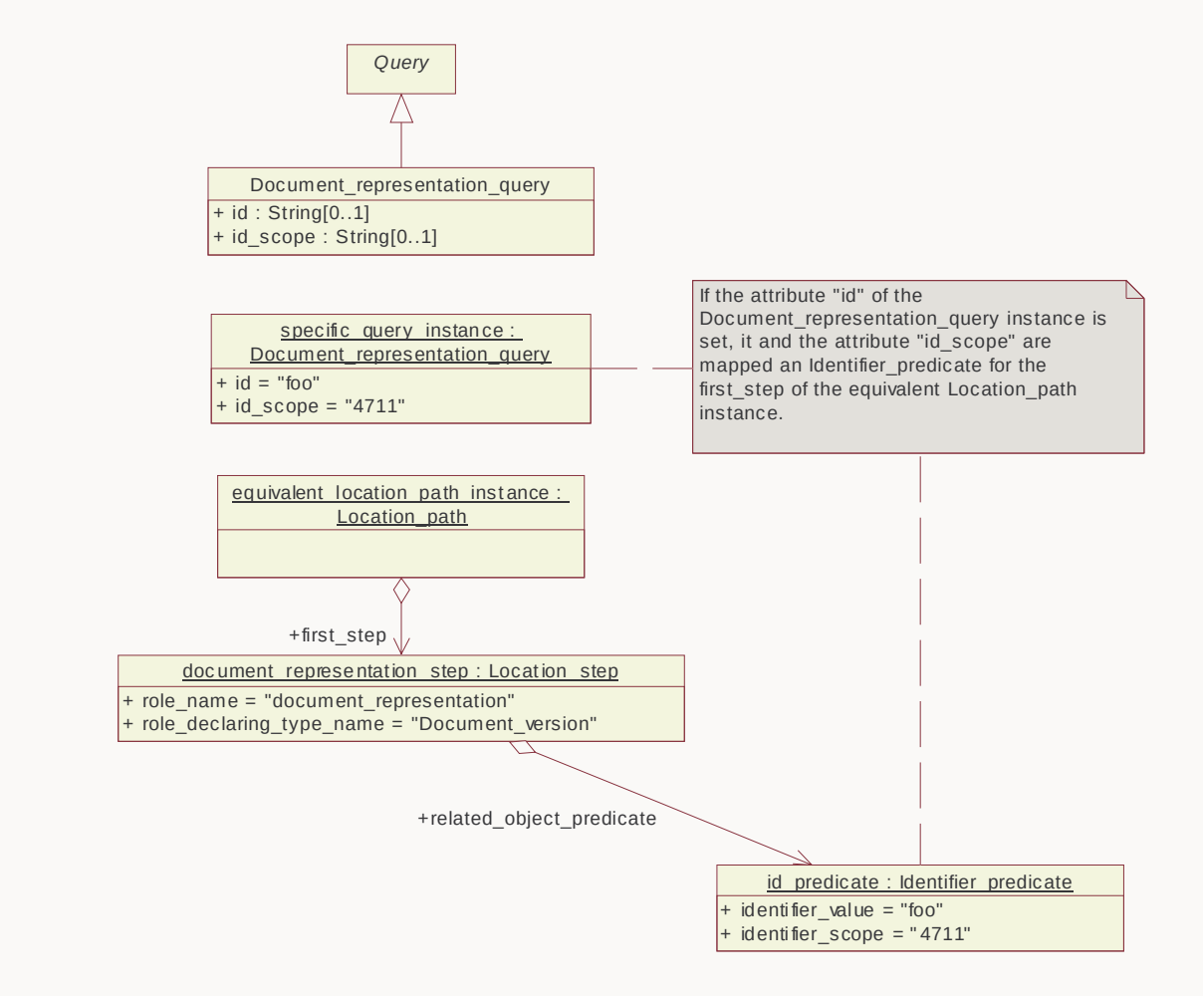


Figure 163 - Definition, sample instance and equivalent Location_path instance of the Document_representation_query

8.14.30 Document_structure_query

The Document_structure_query traverses the subdocuments from documents.

Parameters

- maximum_recursion_number: Integer [0..1]
  limits the recursion level of the query.
- relation_type: String [0..1]
  the specific type of the relations which form the structure

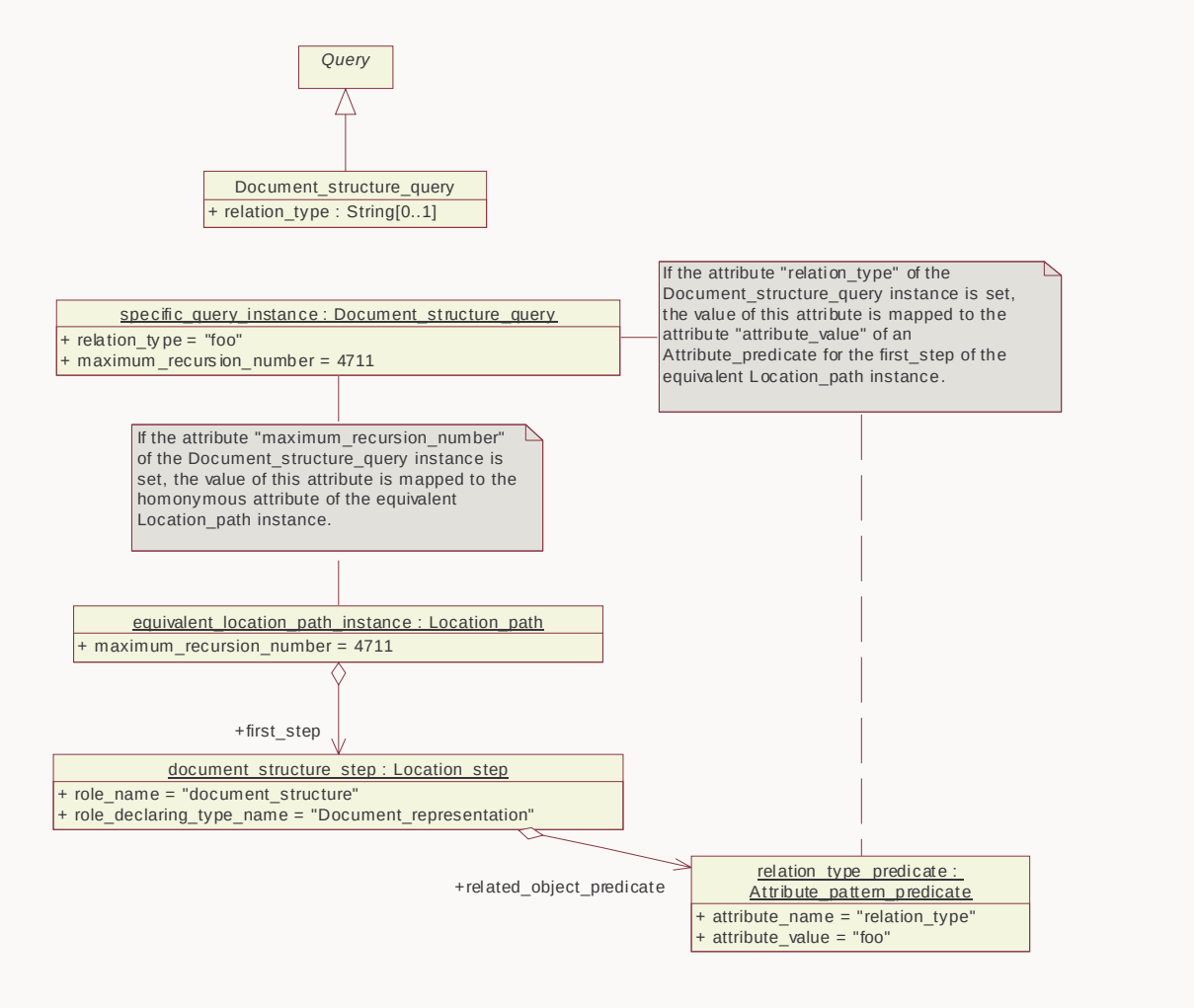


Figure 164 - Definition, sample instance and equivalent Location_path instance of the Document_structure_query

8.14.31 Document_version_query

The Document_version_query traverses Document_version objects of Document objects.

Parameters

- id: String [0..1]

- id_scope: String [0..1]

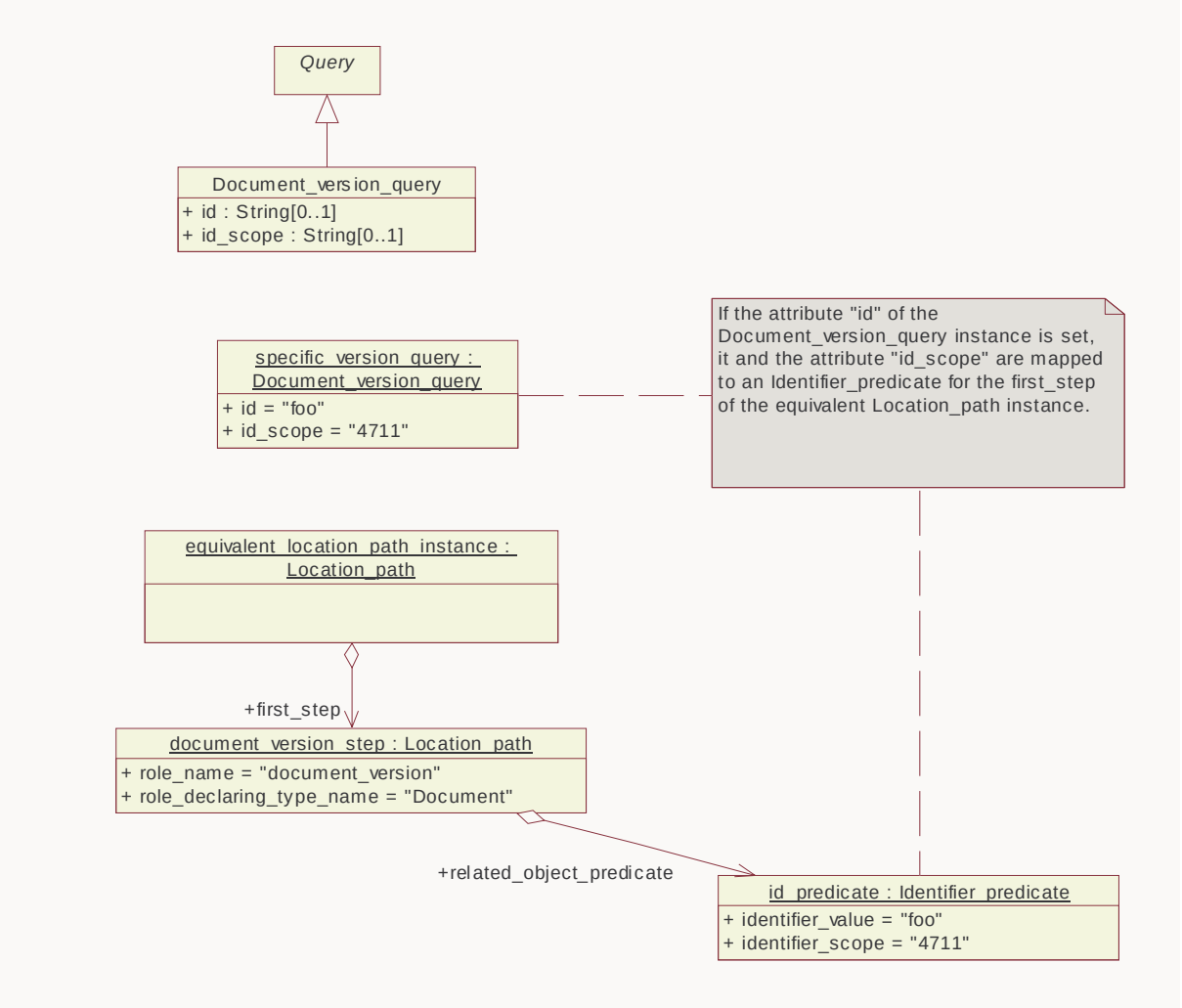


Figure 165 - Definition, sample instance and equivalent Location_path instance of the Document_version_query
Figure 166

8.14.32 Effectivity_query

The Effectivity_query traverses detail information from selected Effectivity objects.

Parameters

- id: String [0..1]
- version_id: String [0..1]
- effectivity_context: String [0..1]

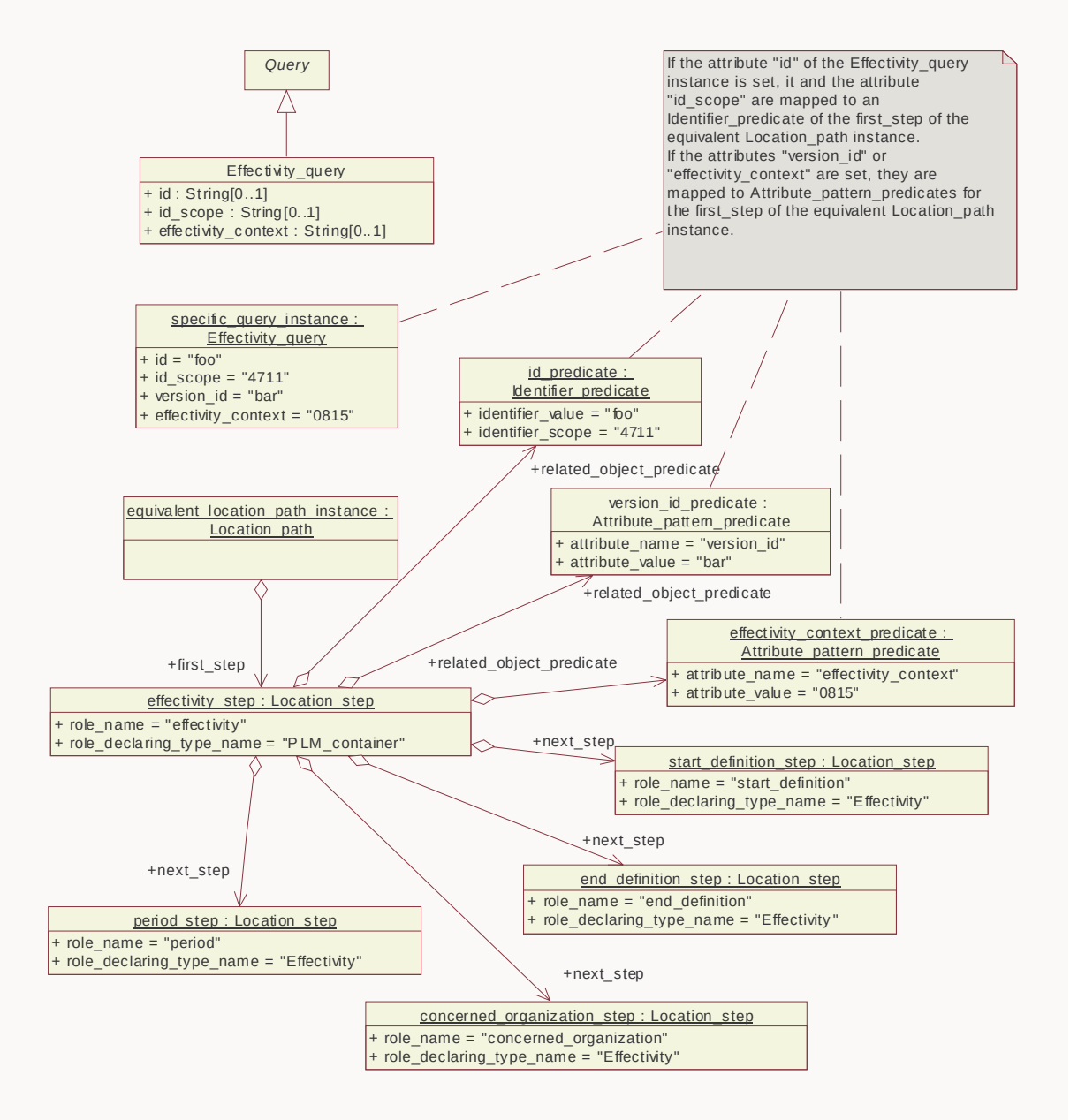


Figure 167 - Definition, sample instance and equivalent Location_path instance of the Effectivity_query

8.14.33 Item_classification_query

The Item_classification_query traverses the Specific_item_classification objects from Item objects.

Parameters

- classification_name: String [0..1]

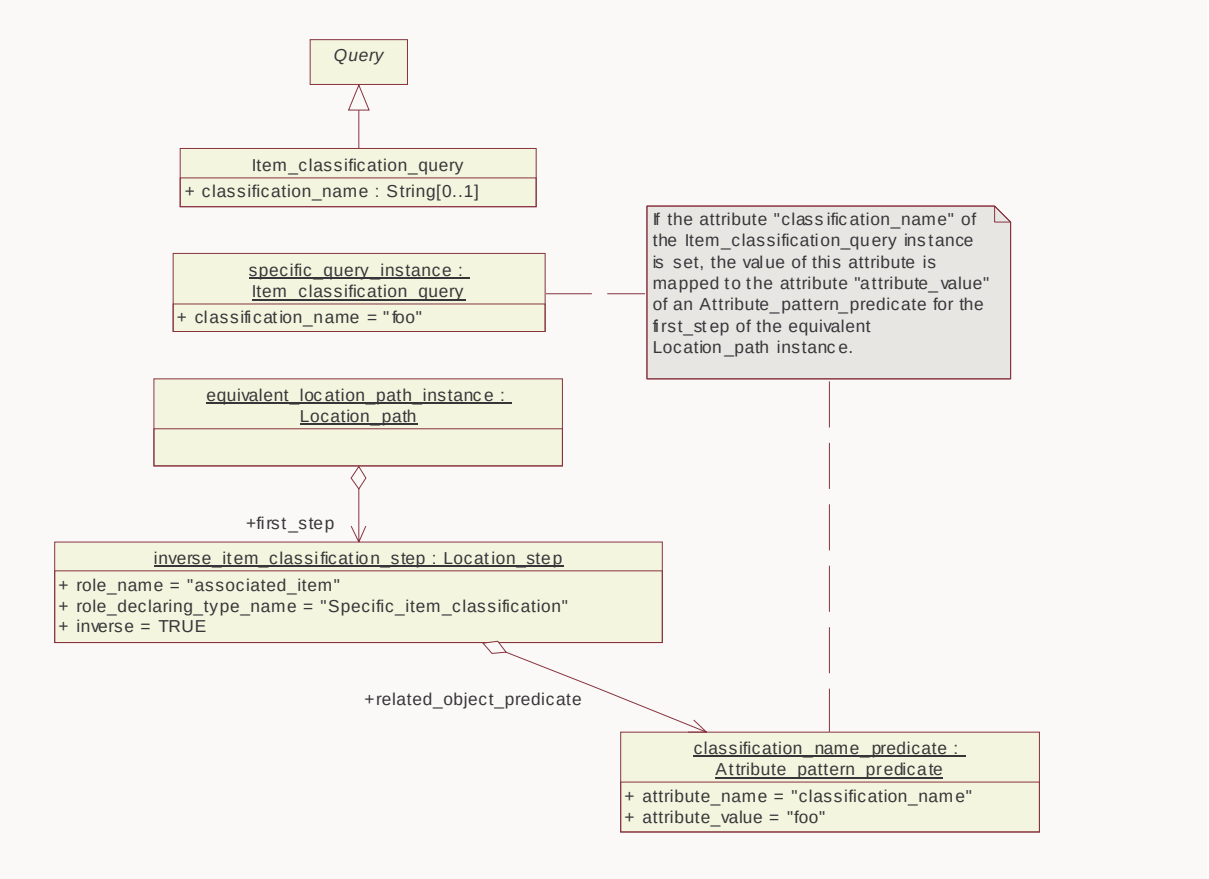


Figure 168 - Definition, sample instance and equivalent Location_path instance of the Item_classification_query

8.14.34 Item_query

The Item_query selects Item objects.

Parameters

- id: String
- id_scope: String [0..1]
- name: String [0..1]
- name_language: Language
- version_id: String [0..1]
- classification_name: String [0..1]

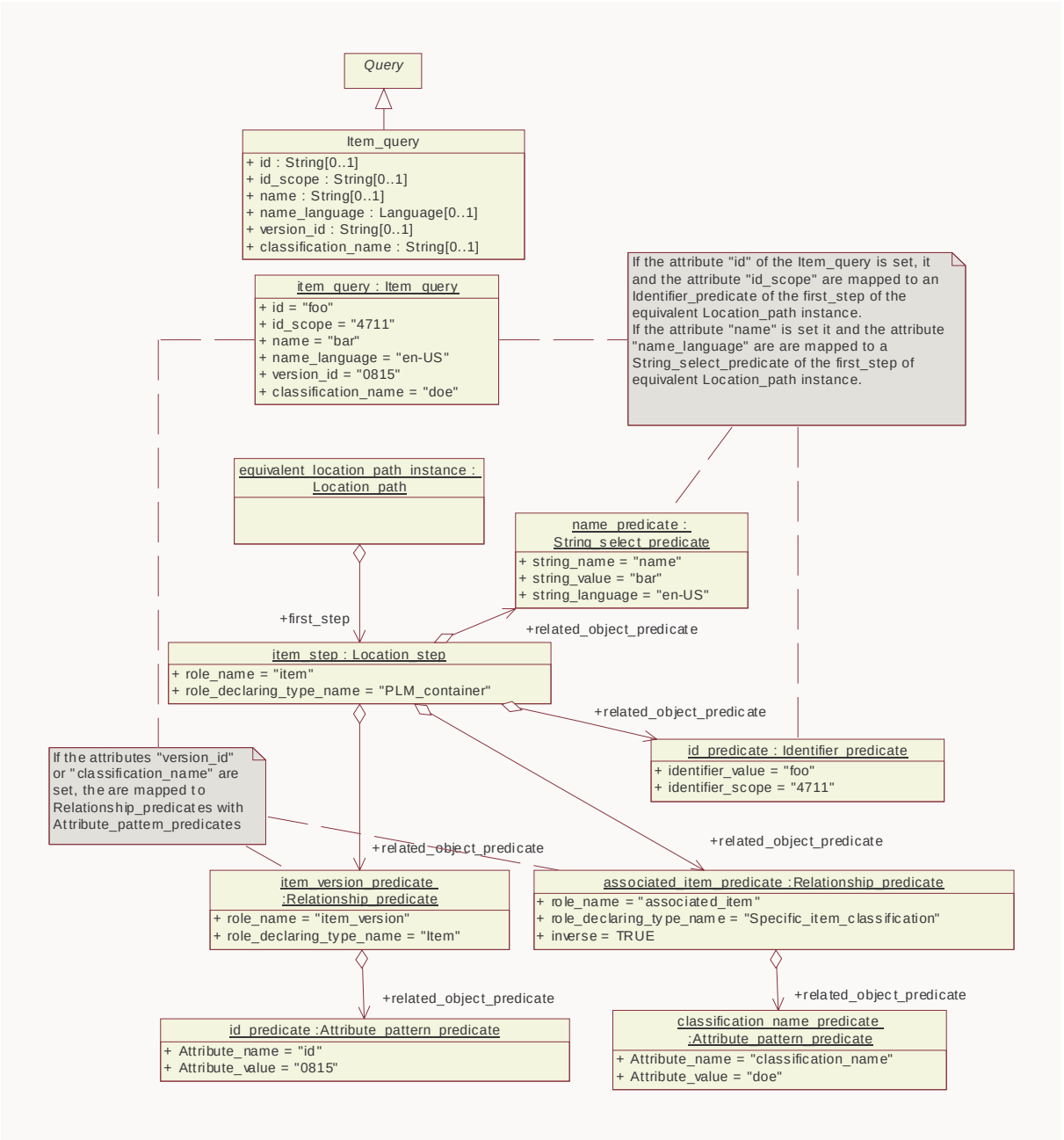


Figure 169 - Definition, sample instance and equivalent Location_path instance of the Item_query

8.14.35 Item_use_query

The Item_use_query traverses those assemblies from Design_discipline_item_definition objects where the Design_discipline_item_definition objects are used as components.

Parameters

- maximum_recursion_number: Integer [0..1]
  limits the recursion level of the query.

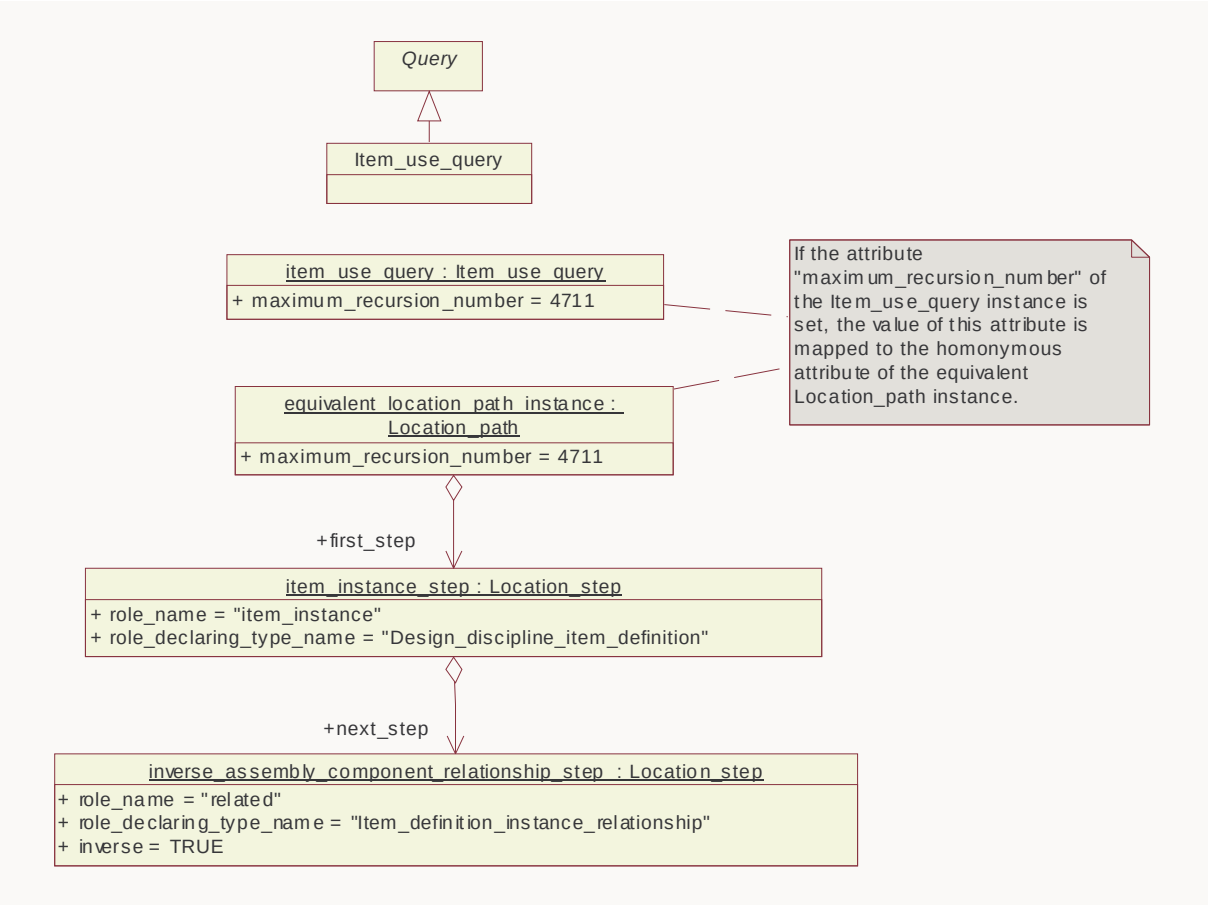


Figure 170 - Definition, sample instance and equivalent Location_path instance of the Item_use_query

8.14.36 Item_version_query

The Item_version_query traverses Item_version objects from Item objects.

Parameters

- id: String [0..1]
- id_scope: String [0..1]

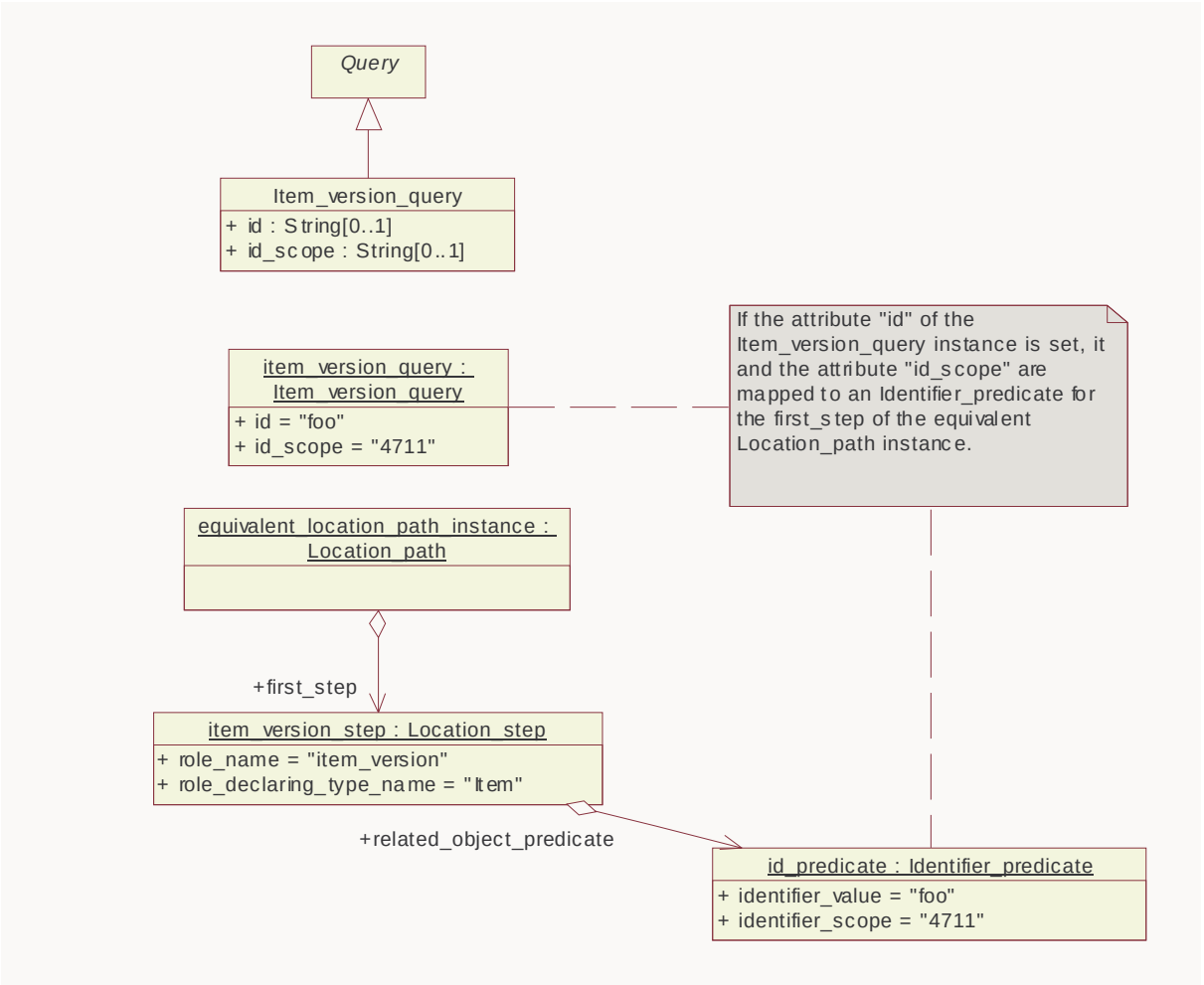


Figure 171 - Definition, sample instance and equivalent Location_path instance of the Item_version_query

8.14.37 Item_version_relationship_query

The Item_version_relationship_query traverses form Item_version objects via Item_version_relationship objects to Item_version objects.

Parameters

- relation_type: String [0..1]the relation_type attribute of the queried relationships
- maximum_recursion_number: Integer [0..1]
- inverse: Boolena[0..1]

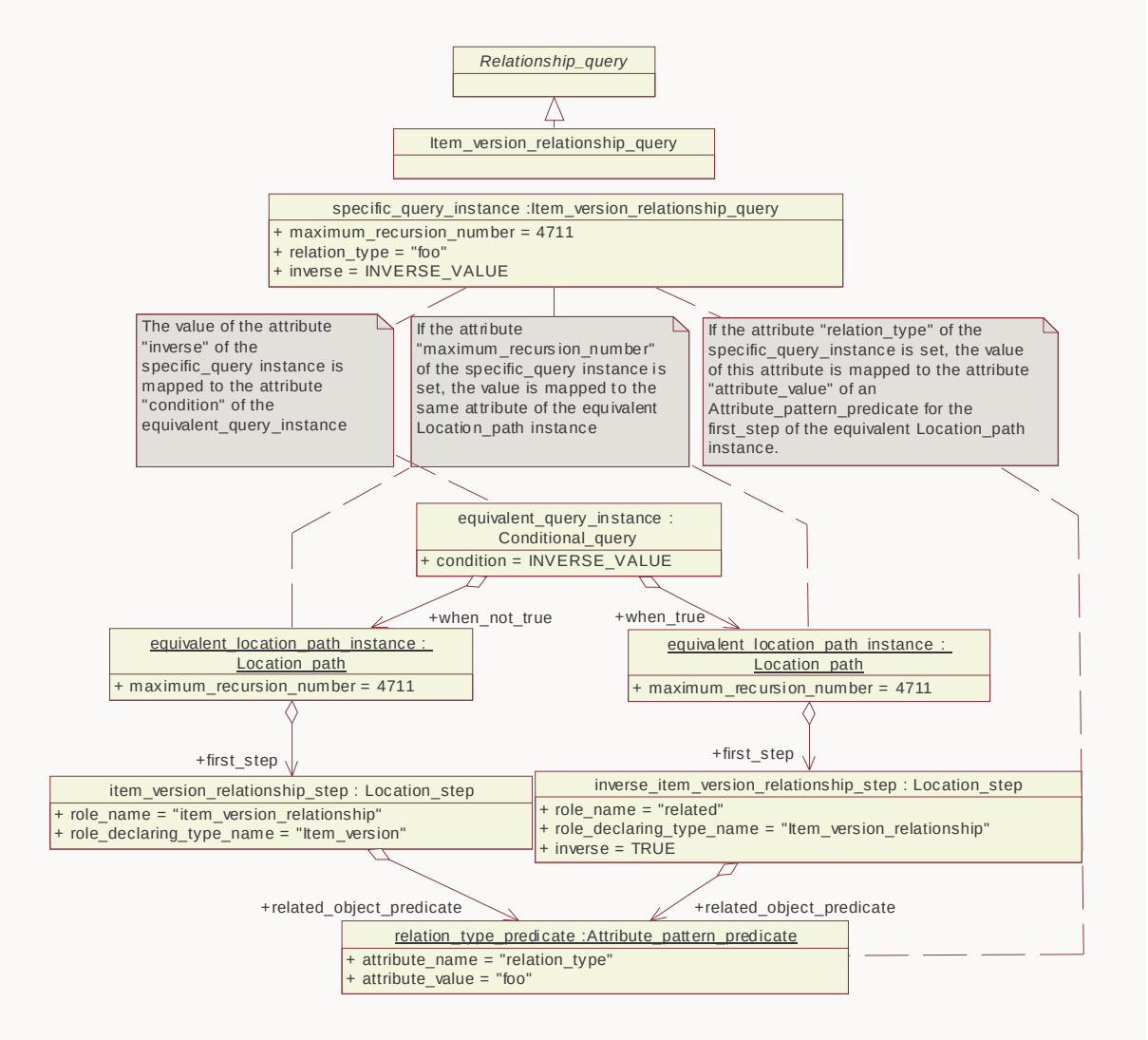


Figure 172 - Definition, sample instance and equivalent Location_path instance of the Item_version_relationship_query

8.14.38 Object_by_uid_query

The Object_by_uid_query selects an object by its uid.

Parameters

- uid: UID

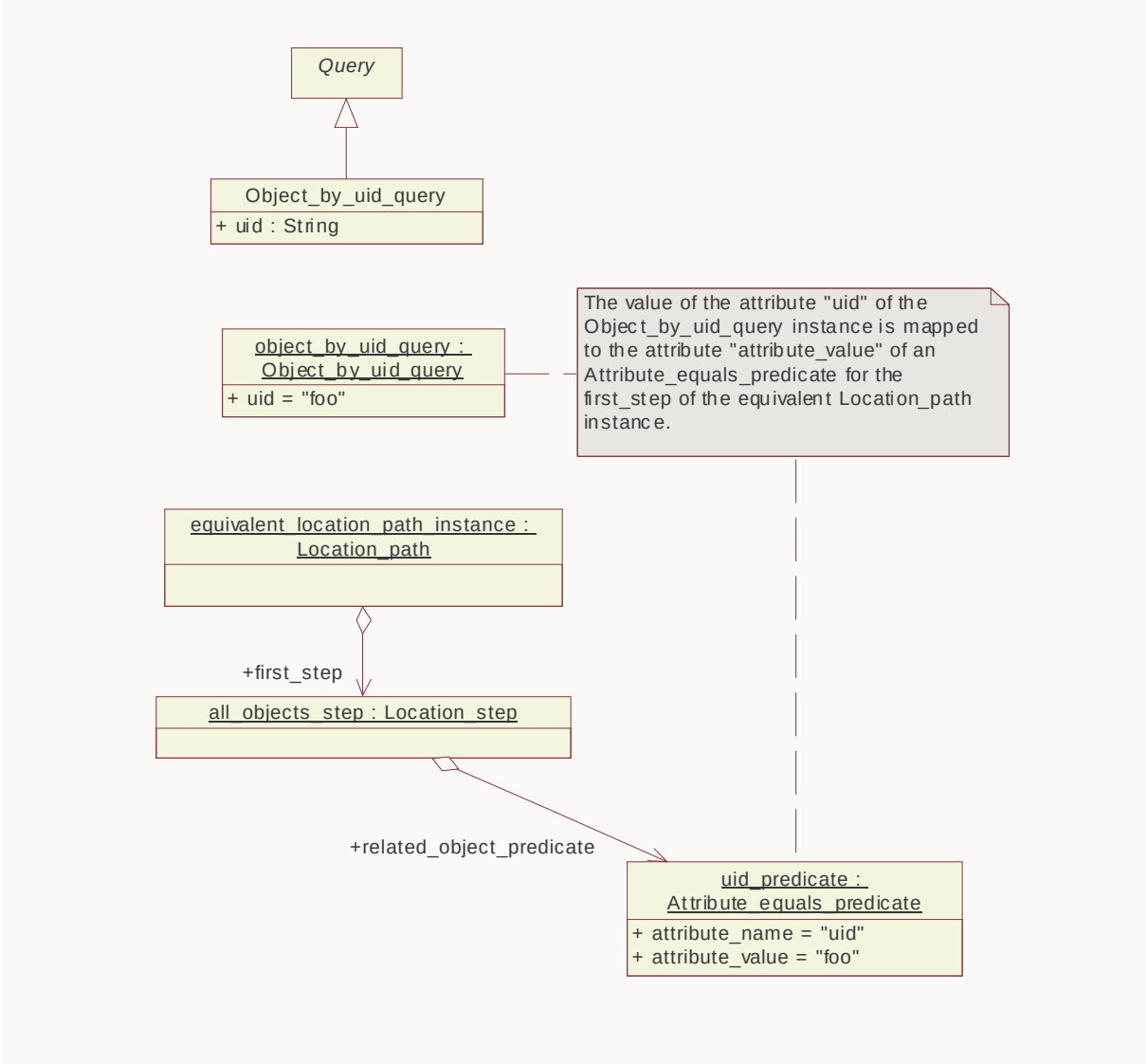


Figure 173 - Definition, sample instance and equivalent Location_path instance of the Object_by_uid_query

8.14.39 Objects_by_uids_query

The Objects_by_uids_query selects a set of objects by its uids.

Parameters

- uids: UID[1..*]

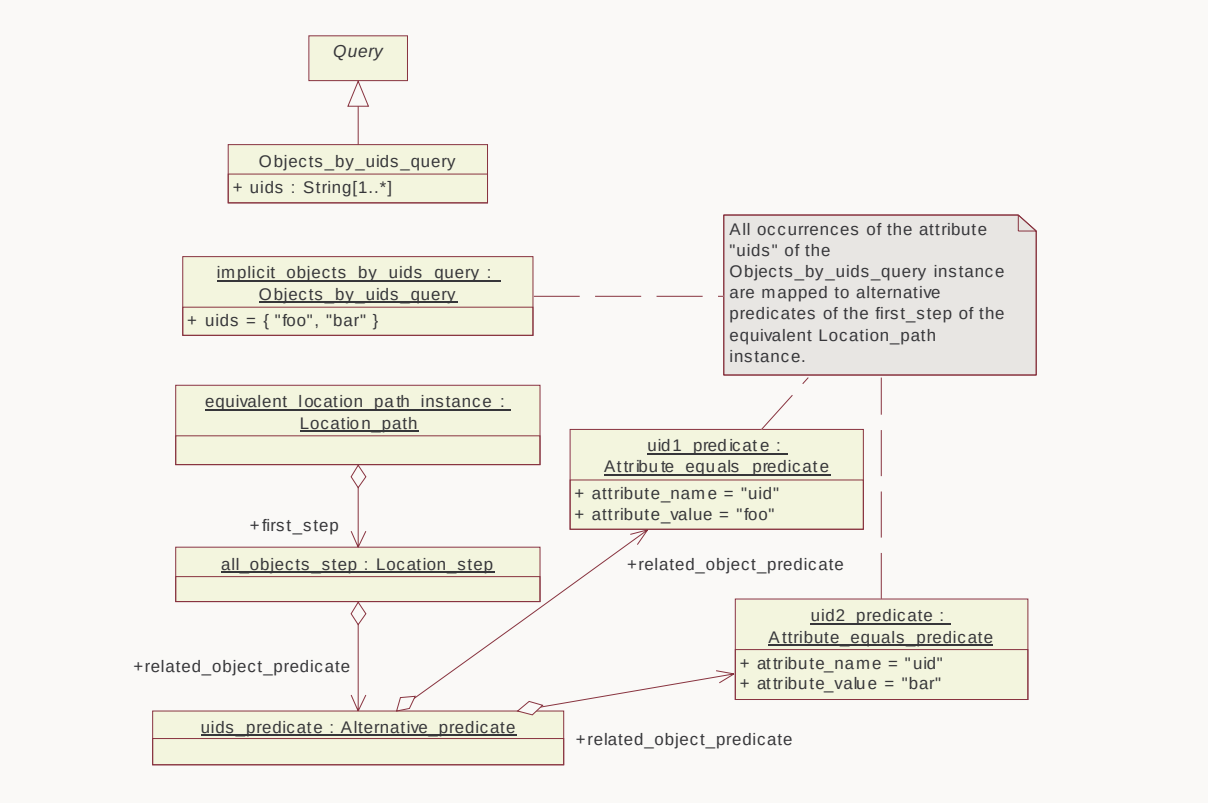


Figure 174 - Definition, instance and equivalent explicit Location_path instance of the Objects_by_uids_query

8.14.40 Organization_query

The Organization_query selects Organization objects.

Parameters

- id: String [0..1]the id of the Organization for which the information is queried
- id_scope: String [0..1]
- organization_name: String [0..1]
- organization_type: String [0..1]

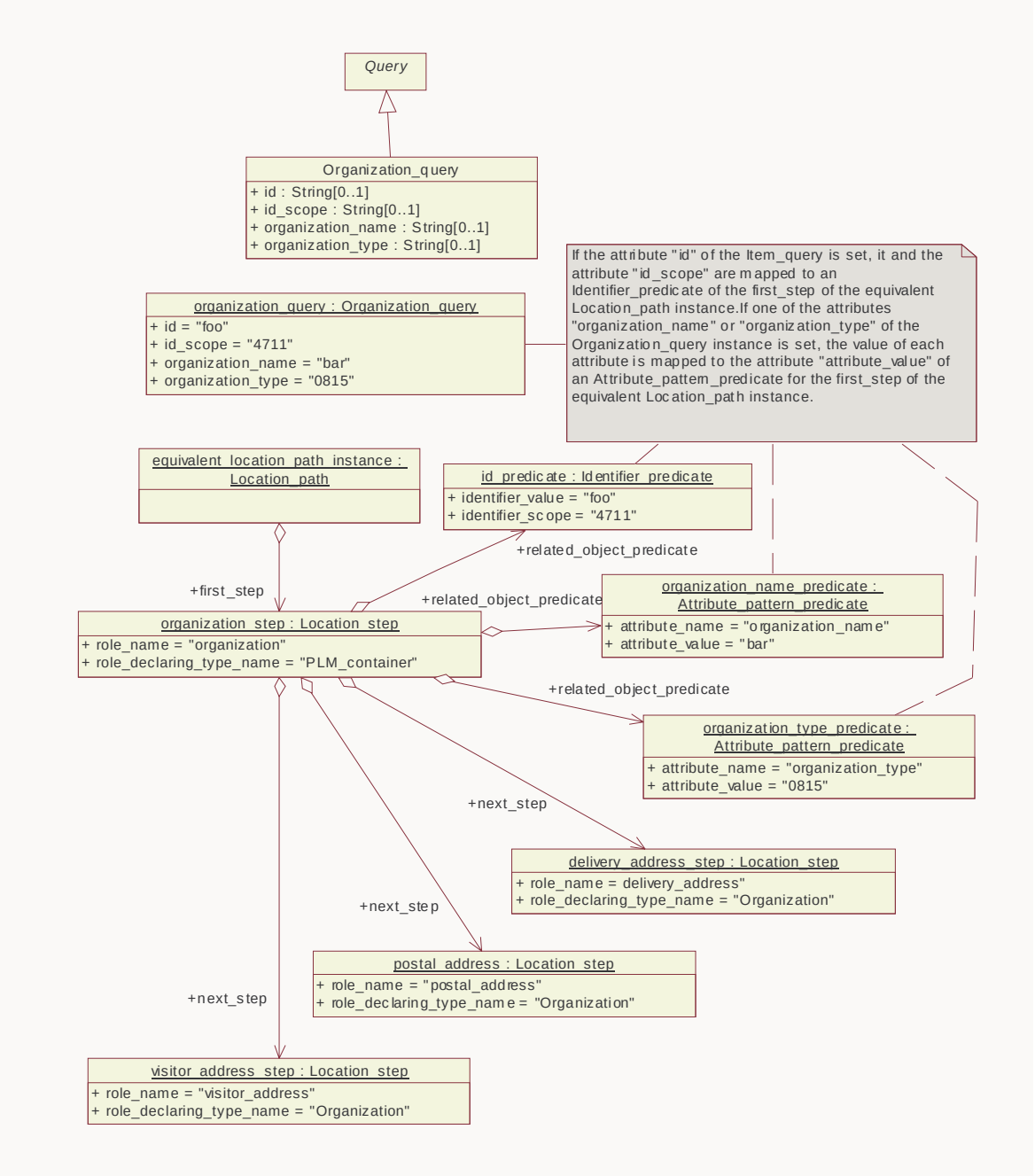


Figure 175 - Definition, sample instance and equivalent Location_path instance of the Organization_query

8.14.41 Organization_relationship_query

The Organization_relationship_query traverses from Organization objects via Organization_relationship objects to Organization objects.

Parameters

- relation_type : String [0..1]
- maximum_recursion_number : Integer [0..1]
- inverse : Boolean [0..1]

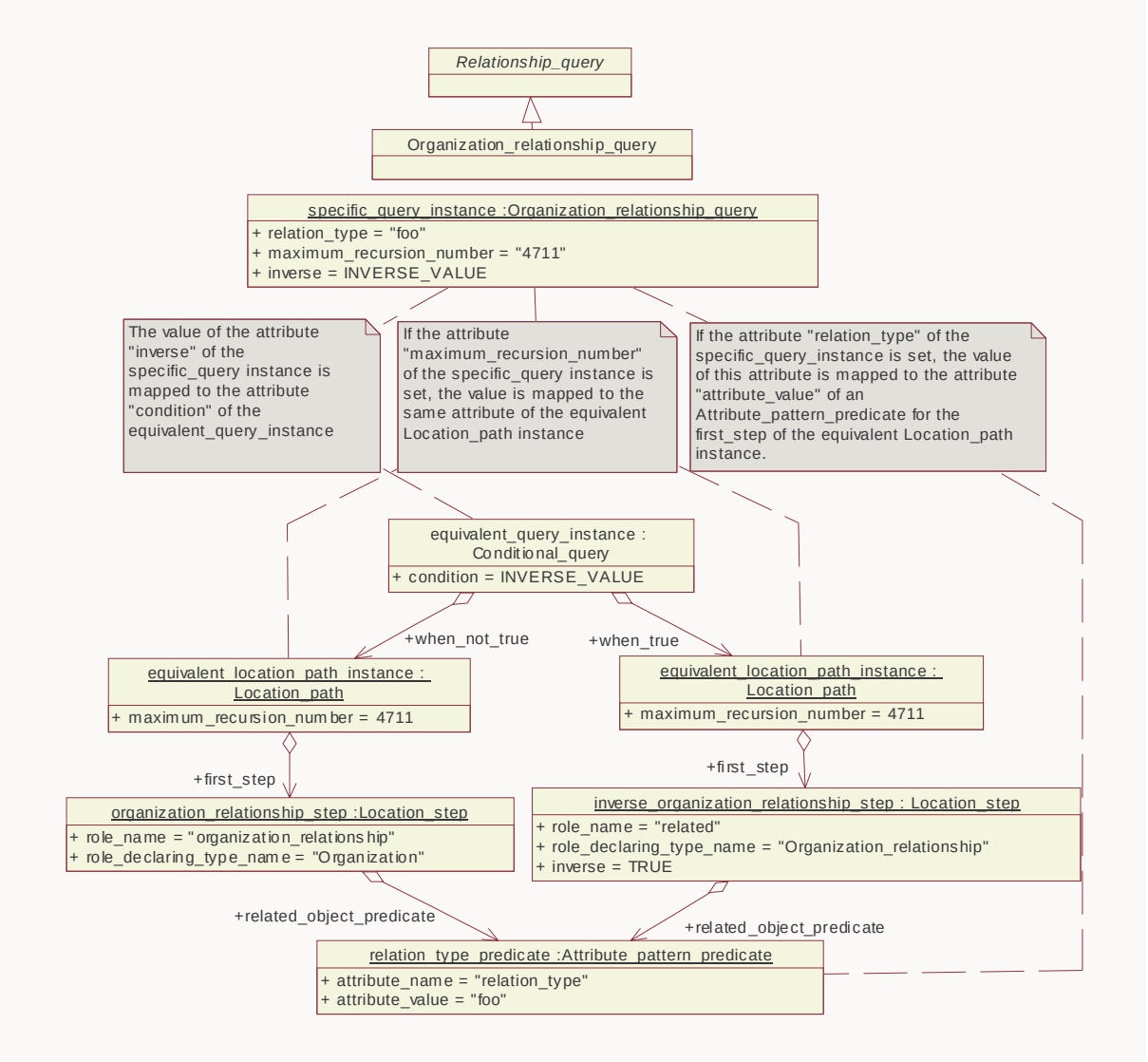


Figure 176 Definition, sample instance and equivalent Location_path instance of the Organization_relationship_query

8.14.42 Person_in_organization_query

The Person_in_organization_ query traverses from Person objects via Person_in_organization objects to Organization objects.

Parameters

- person_name : String [0..1]
- id: String [0..1]
- role: String [0..1]
- organization_id : String [0..1]

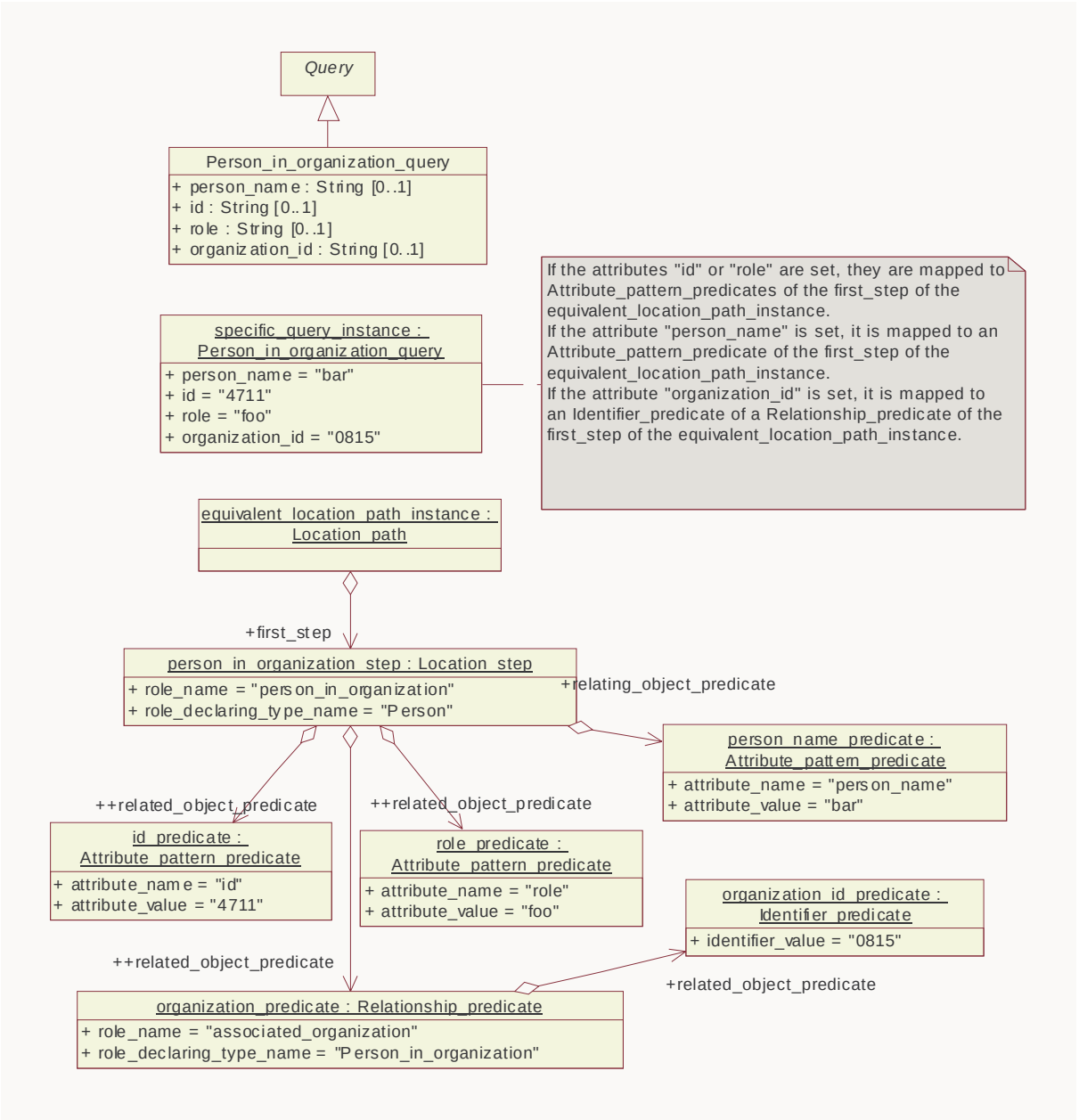


Figure 177 Definition, sample instance and equivalent Location_path instance of the Person_in_organization_query

8.14.43 Person_in_organization_relationship_query

The Person_in_organization_relationship_query traverses from Person_in_organization objects via Person_in_organization_relationship objects to Person_in_organization objects.

Parameters

- relation_type : String [0..1]
- maximum_recursion_number : Integer [0..1]
- inverse : Boolean [0..1]

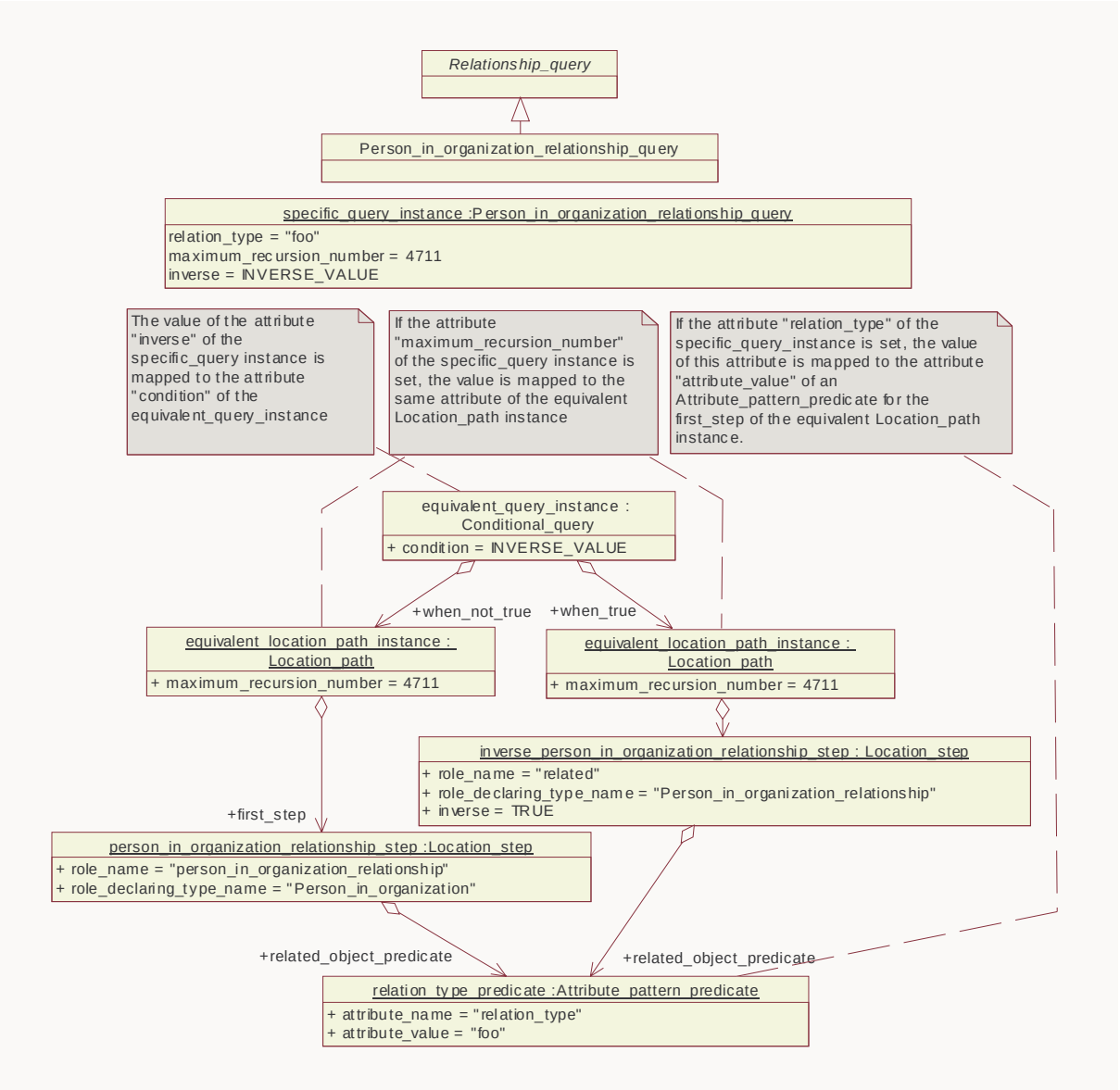


Figure 178 Definition, sample instance and equivalent Location_path instance of the Person_in_organization_relationship_query

8.14.44 Product_class_query

The Product_class_query selects Product_class objects.

Parameters

- id: String
- id_scope: String [0..1]
- name: String
- name_language: Language

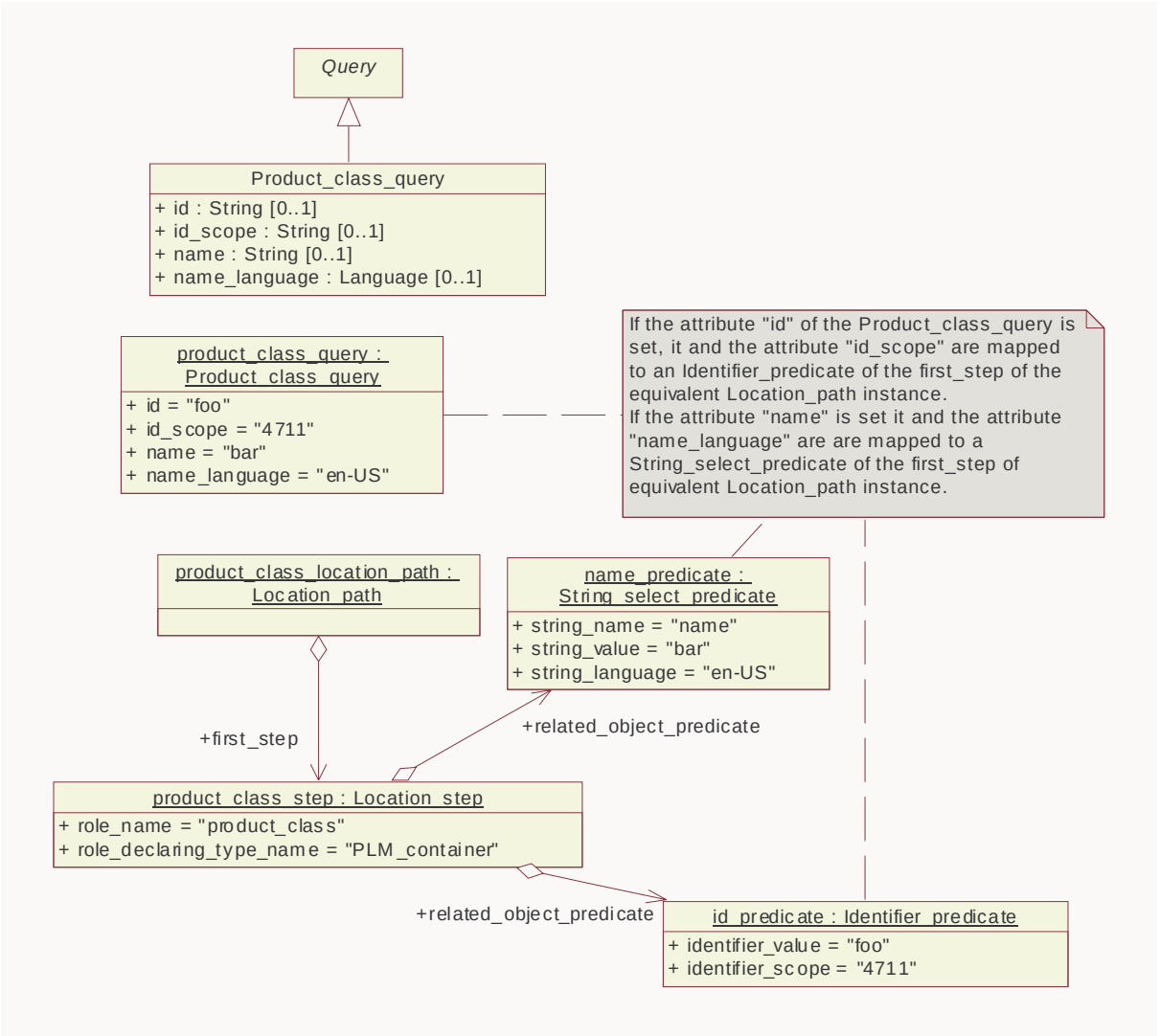


Figure 179 - Definition, sample instance and equivalent Location_path instance of the Product_class_query

8.14.45 Product_structure_query

The Product_structure_query traverses from Complex_product objects via Product_structure_relationship objects to Product_constituent_select objects.

Parameters

- relation_type: String [0..1]

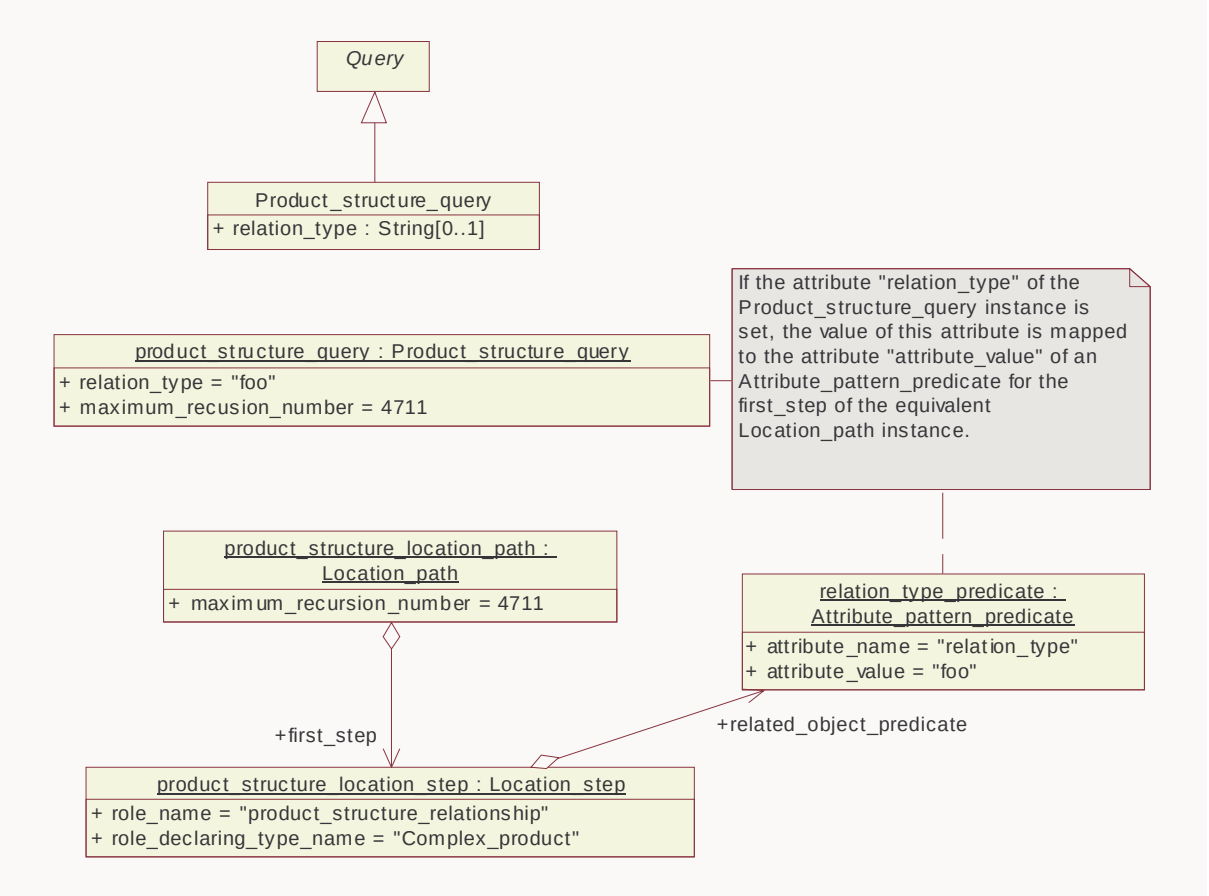


Figure 180 - Definition, sample instance and equivalent Location_path instance of the Product_structure_query

8.14.46 Project_assignment_query

The Project_assignment_query traverses from Project objects via Project_assignment objects to Project_information_select objects.

Parameters

- role : String [0..1]

- related_type_name : String [0..*]

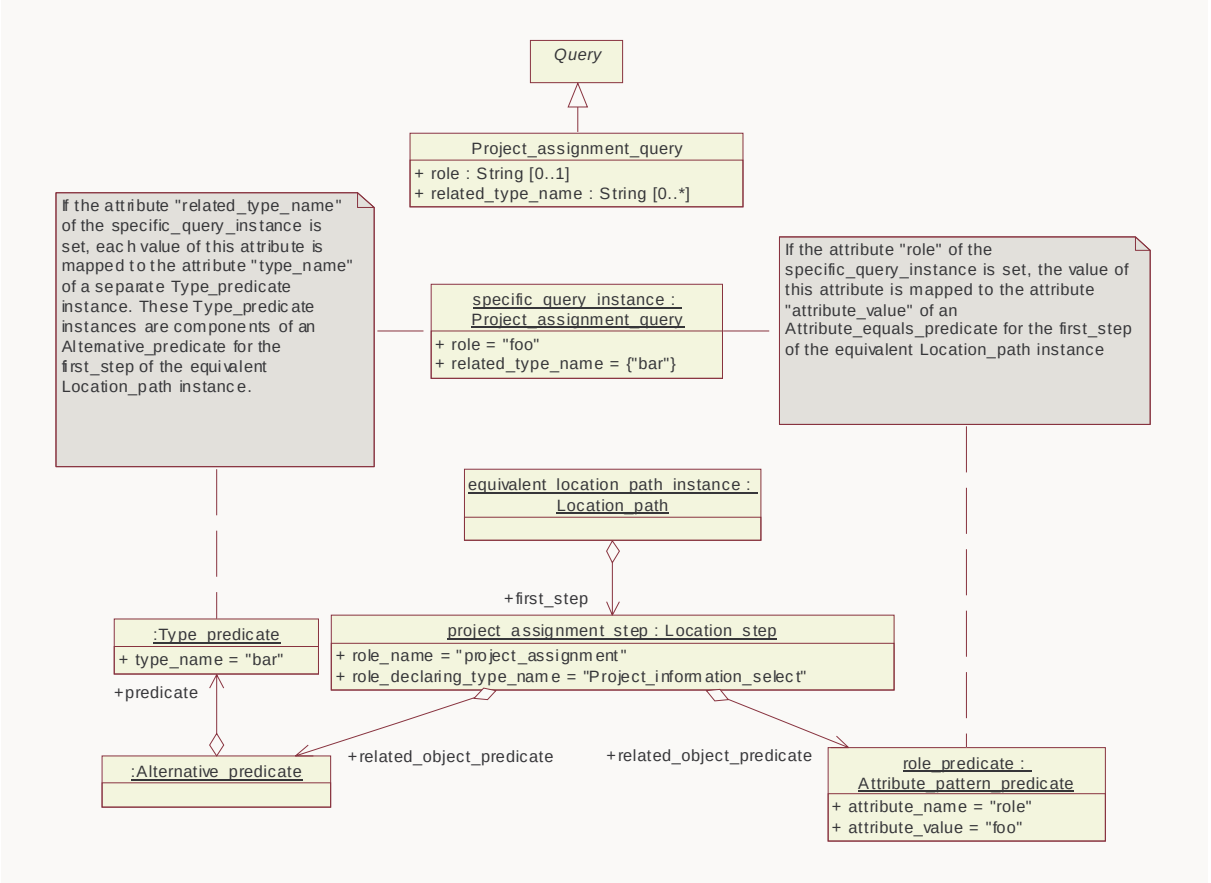


Figure 181 Definition, sample instance and equivalent Location_path instance of the Project_assignment_query

8.14.47 Simple_property_query

The Simple_property_query traverses from Simple_property_select objects via Simple_property_association objects to Simple_property objects.

Parameters

- value_name: String [0..1]
- relating_type_name: String [0..1]

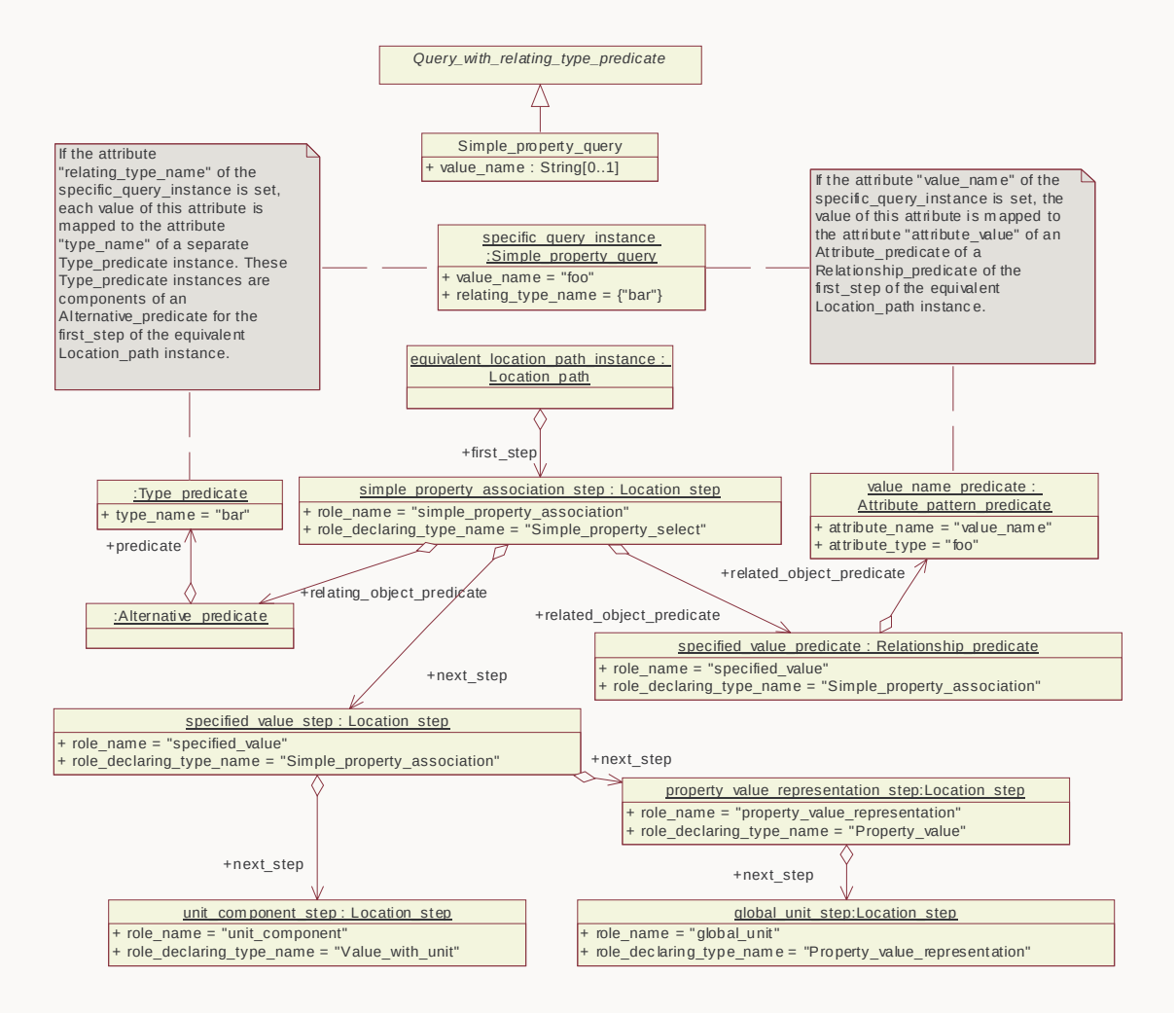


Figure 182 - Definition, sample instance and equivalent Location_path instance of the Simple_property_value_query

8.14.48 Work_request_activity_query

The Work_request_activity_query traverses from Work_request objects to Activity objects.

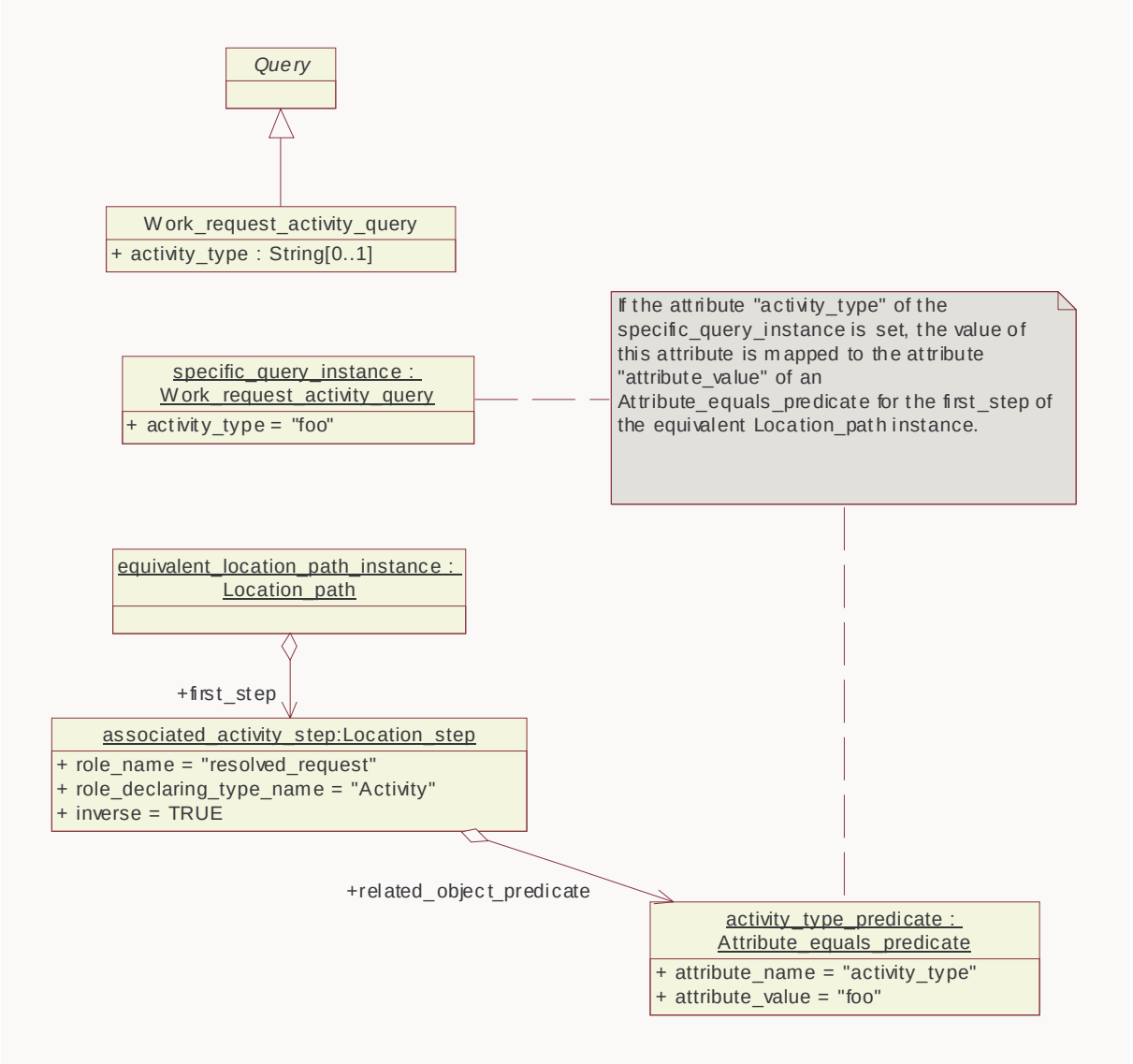


Figure 183 Definition, sample instance and equivalent Location_path instance of theWork_request_activity_query

8.14.49 Work_request_query

The Work_request_query selects Work_request objects.

Parameters

- id : String [0..1]
- request_type : String [0..1]
- status : String [0..1]

- version_id : String [0..1]
- classification_role : String [0..1]
- classification_id : String [0..1]

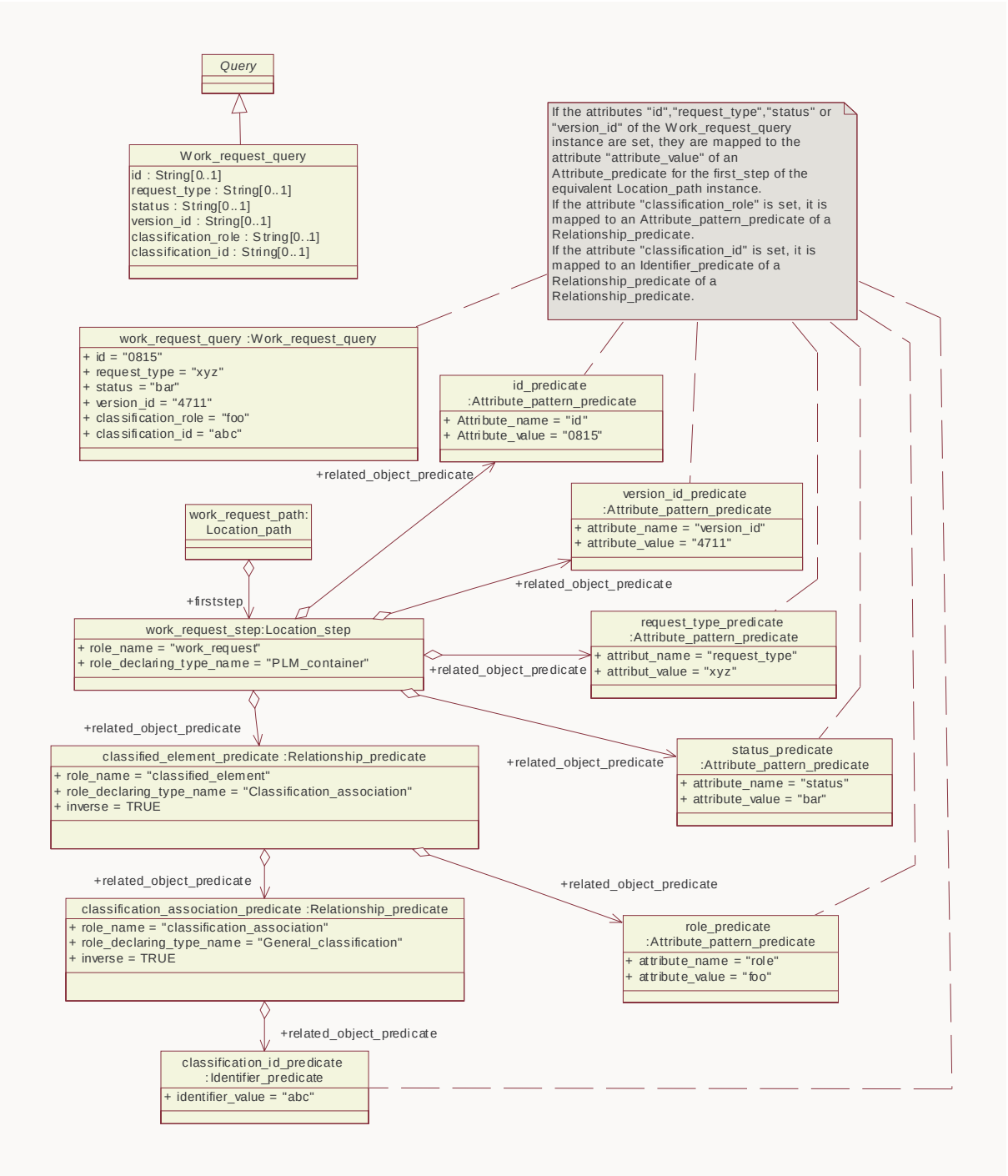


Figure 184 Definition, sample instance and equivalent Location_path instance of the Work_request_query

8.14.50 Work_request_relationship_query

The Work_request_relationship_query traverses from Work_request objects via Work_request_relationship objects to Work_request objects.

Parameters

- relation_type : String [0..1]
- maximum_recursion_number : Integer [0..1]

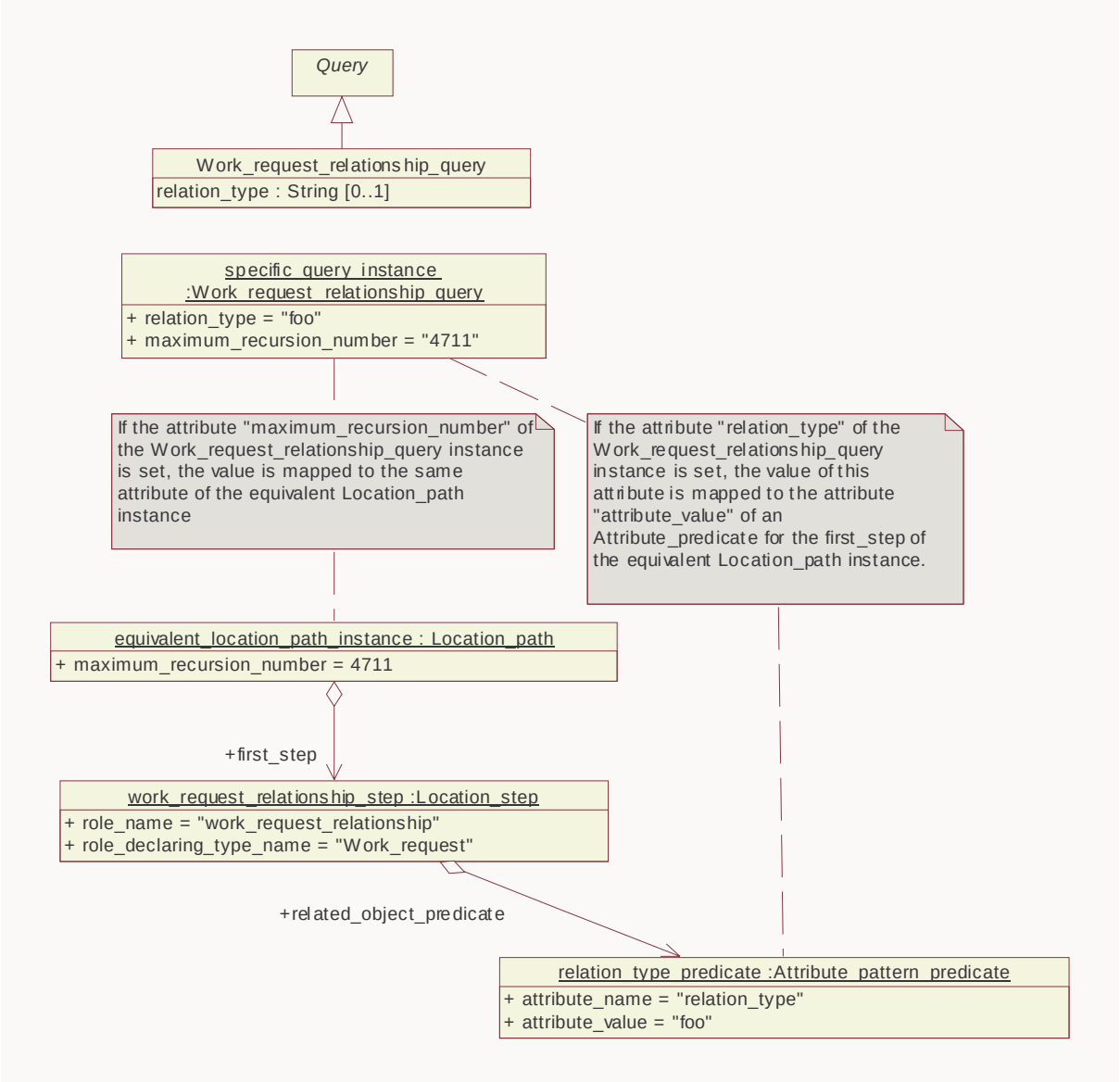


Figure 185 Definition, sample instance and equivalent Location_path instance of the Work_request_relationship_query

8.14.51 Work_request_scope_query

The Work_request_scope_query traverses from Work_request objects to the Activity_element_select objects which are the scope of the Work_request objects.

Parameters

- none

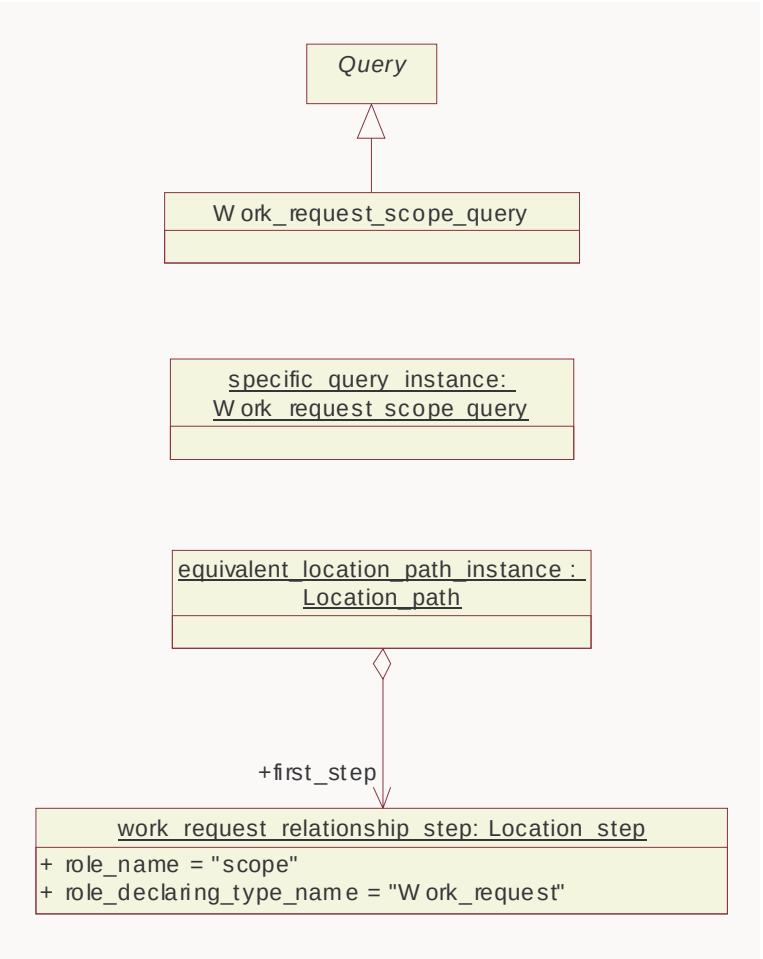


Figure 186 Definition, sample instance and equivalent Location_path instance of the Work_request_scope_query

8.15 PDTnet Queries Conformance Point

The PDTnet Queries Conformance Point defines a set of high level specialized queries that fulfill the requirements of the use cases described in section /CROSSREF 7.2. The semantics of each specialized query of this conformance point is defined by an equivalent Query instance from the Specific Queries Conformance Point.

8.15.1 General_detail_query

The General_detail_query returns general detail information from objects selected by a uid.

Parameters

- uids: UID [1..*]
- relating_type_name: String [0..*]
- add_aliases: Boolean [0..1]
- alias_id: String [0..1]
- add_authorizations: Boolean [0..1]
- authorization_role: String [0..1]
- add_dates: Boolean [0..1]
- date_role: String [0..1]
- add_properties: Boolean [0..1]
- property_name: String [0..1]
- add_classifications: Boolean [0..1]
- classification_role: String [0..1]
- add_approvals: Boolean [0..1]
- approval_level: String [0..1]
- add_activities: Boolean [0..1]
- activity_role: String [0..1]
- add_effectivities: Boolean [0..1]
- effectivity_role: String [0..1]

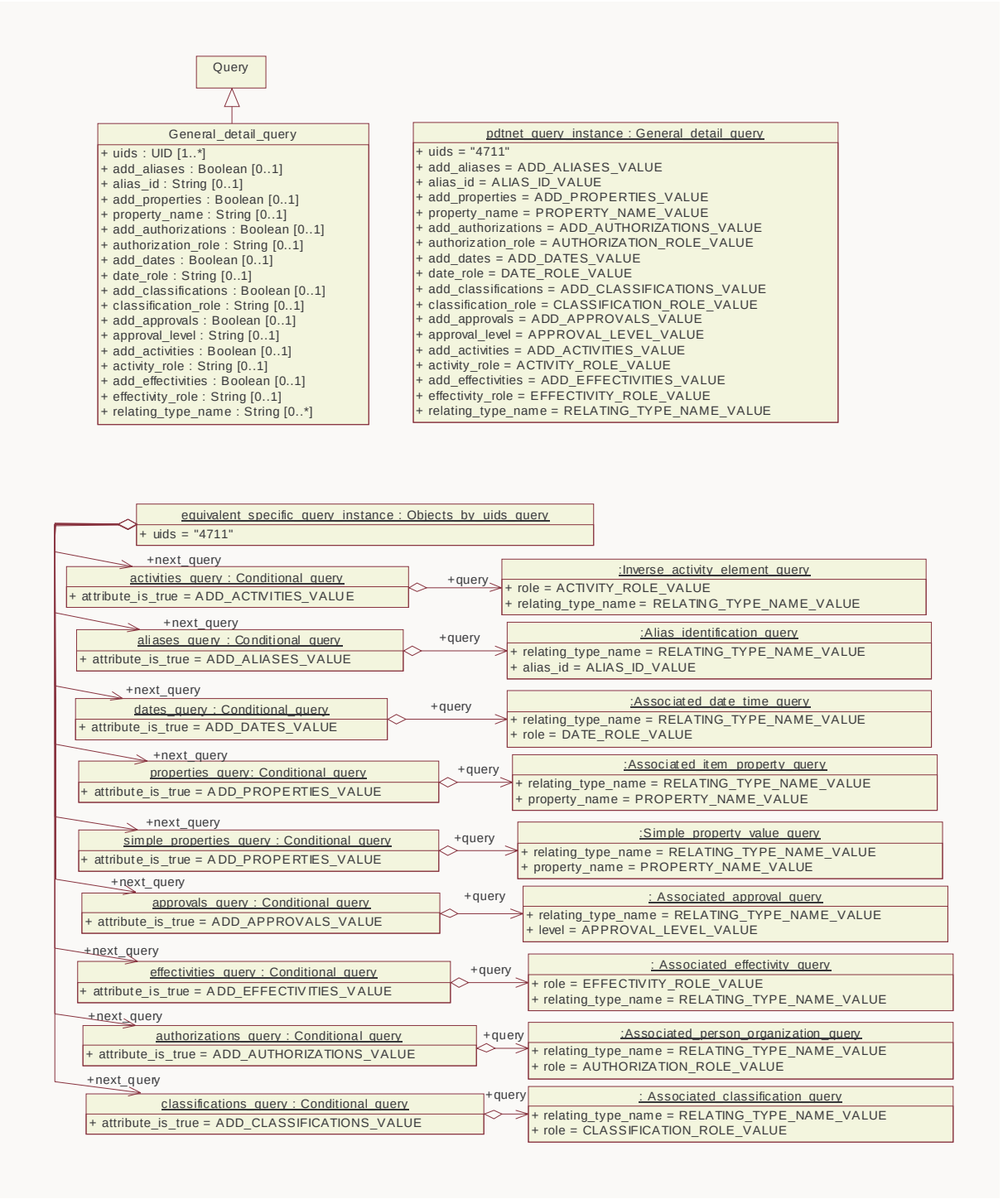


Figure 187 Definition, sample instance and equivalent specific query instance of the General_detail_query

8.15.2 Document_detail_query

The Document_detail_query returns detail information of an Document, Document_version, or Document_representation object selected by a uid.

Parameters (no inherited)

- classification_name: String [0..1]
- add_versions: Boolean [0..1]
- version_id: String [0..1]
- add_representations: Boolean [0..1]

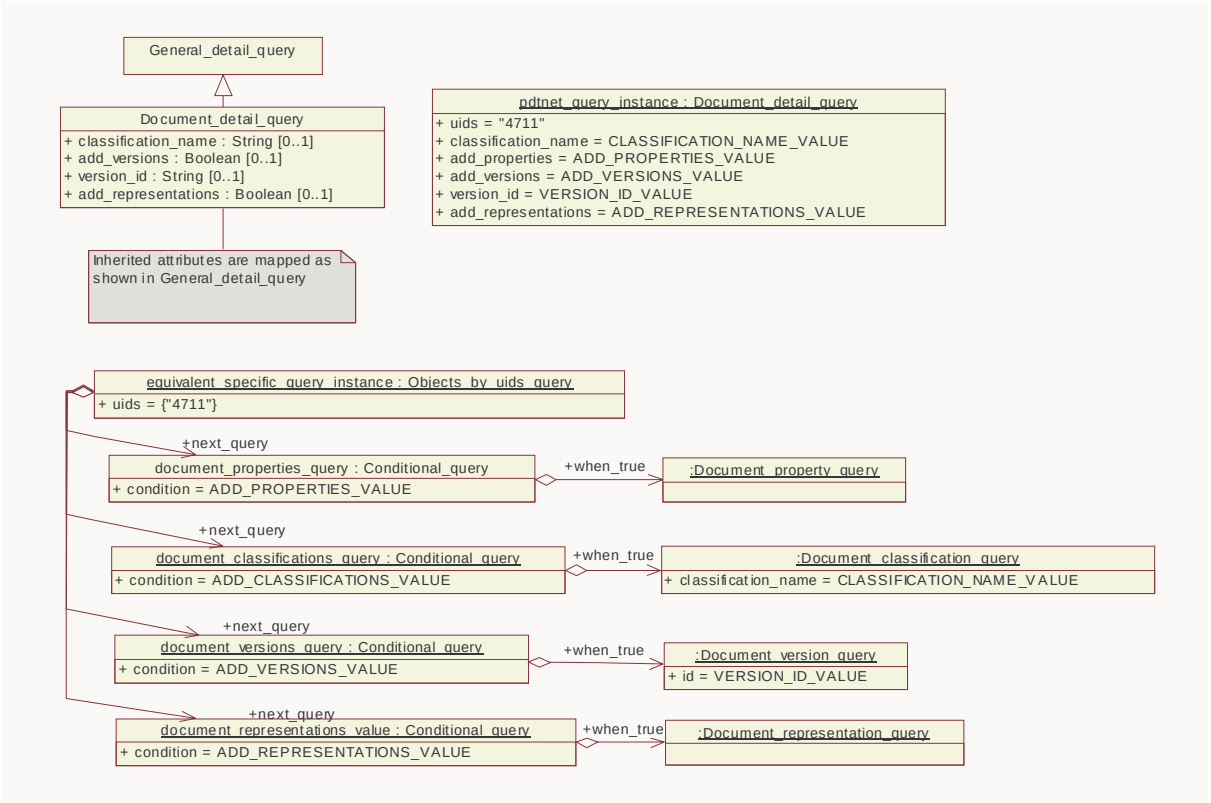


Figure 188 Definition, sample instance and equivalent specific query instance of the Document_detail_query

8.15.3 Document_selection_query

The Document_selection_query selects objects of class Document and includes related Document_version and Document_representation objects.

Parameters

- name: String [0..1]

- id: String [0..1]
- version_id: String [0..1]
- classification_name: String [0..1]

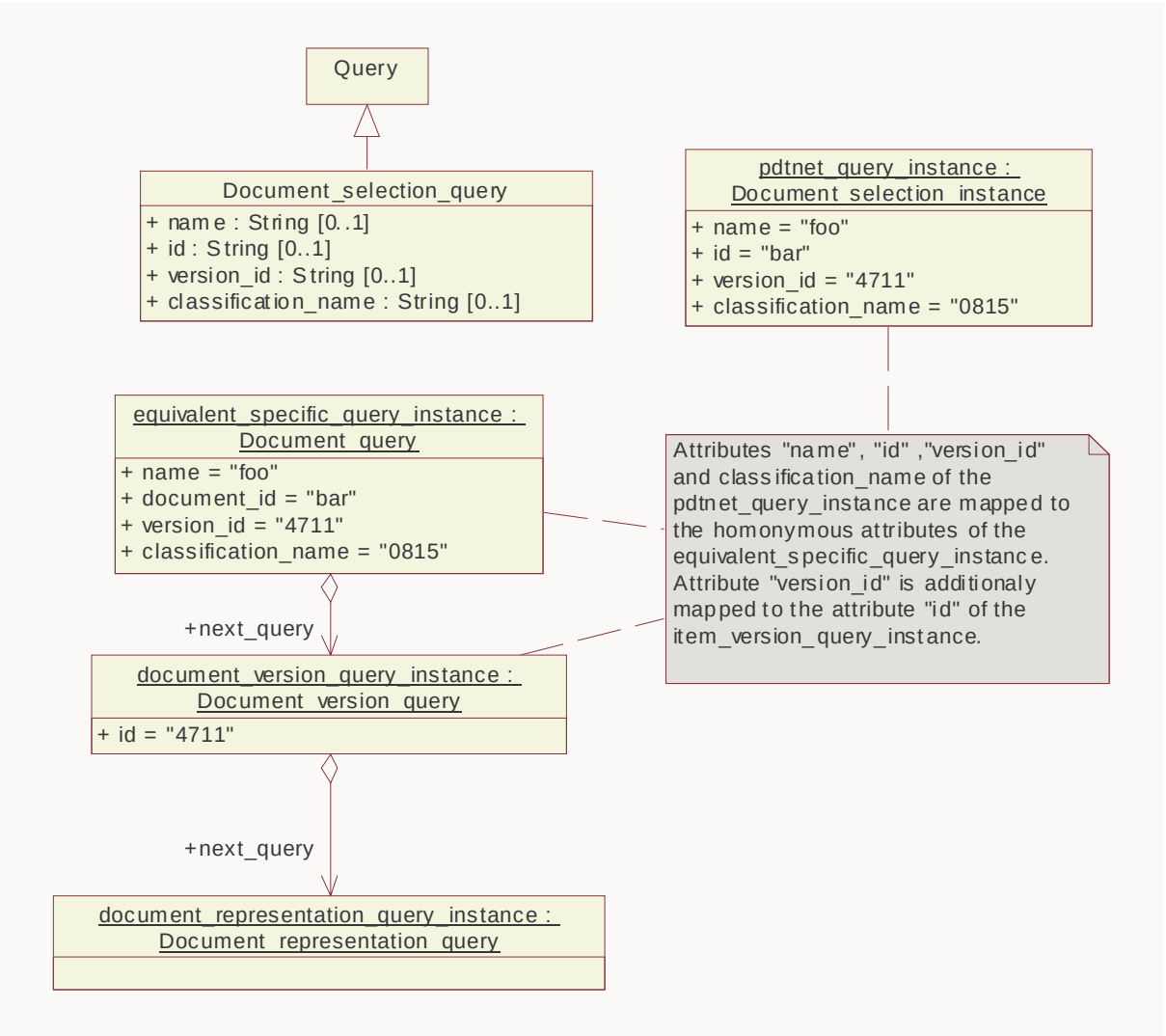


Figure 189 Definition, sample instance and equivalent specific query instance of the Document_selection_query

8.15.4 Document_ traversal_query

The Document_traversal_query traverses from a Document_representation object selected by its uid via Document_structure objects to related Document_representation objects in a document structure.

Parameters

- uid: UID

- maximum_recursion_number: Integer [0..1]
- relation_type: String [0..1]

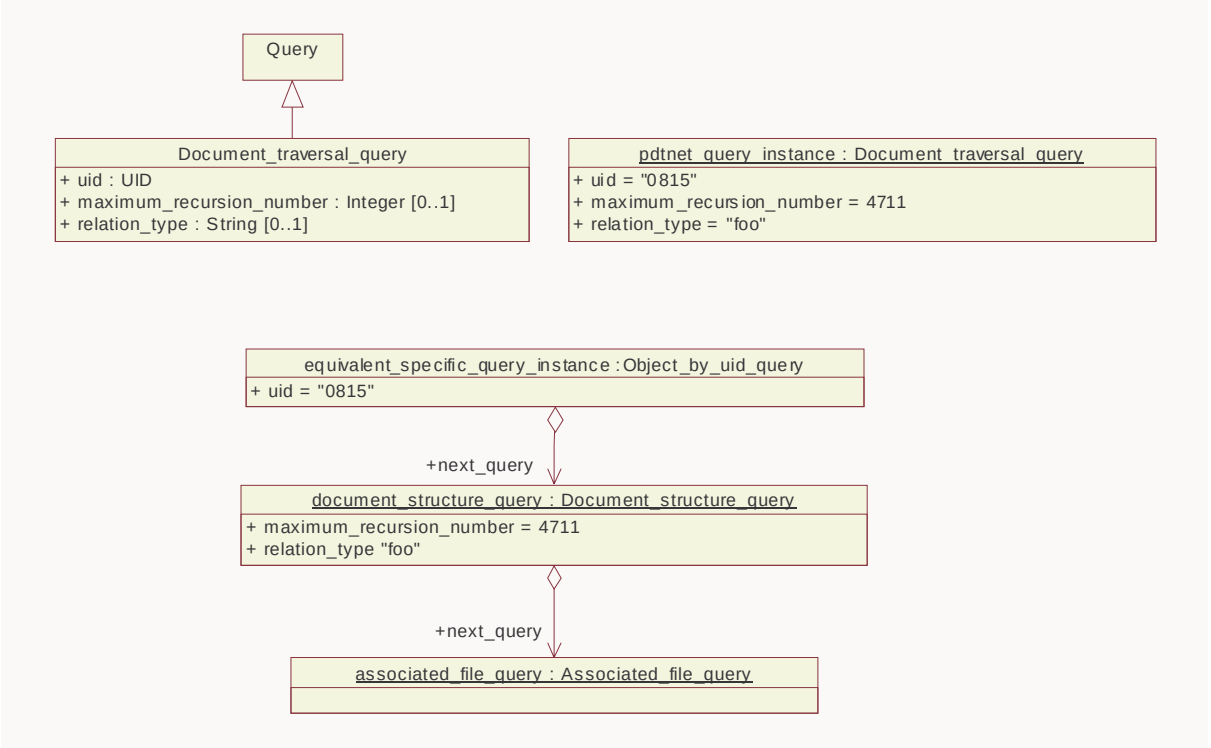


Figure 190 Definition, sample instance and equivalent specific query instance of the Document_traversal_query

8.15.5 Item_detail_query

The Item_detail_query returns detail information of an Item, Item_version, Design_discipline_item_definition, Item_instance or Assembly_component_relationship object selected by a uid.

Parameters (no inherited)

- add_version_relationships: Boolean [0..1]
- version_relationship_type: String [0..1]
- add_documents: Boolean [0..1]
- document_role: String [0..1]
- classification_name: String [0..1]
- add_placement: Boolean [0..1]

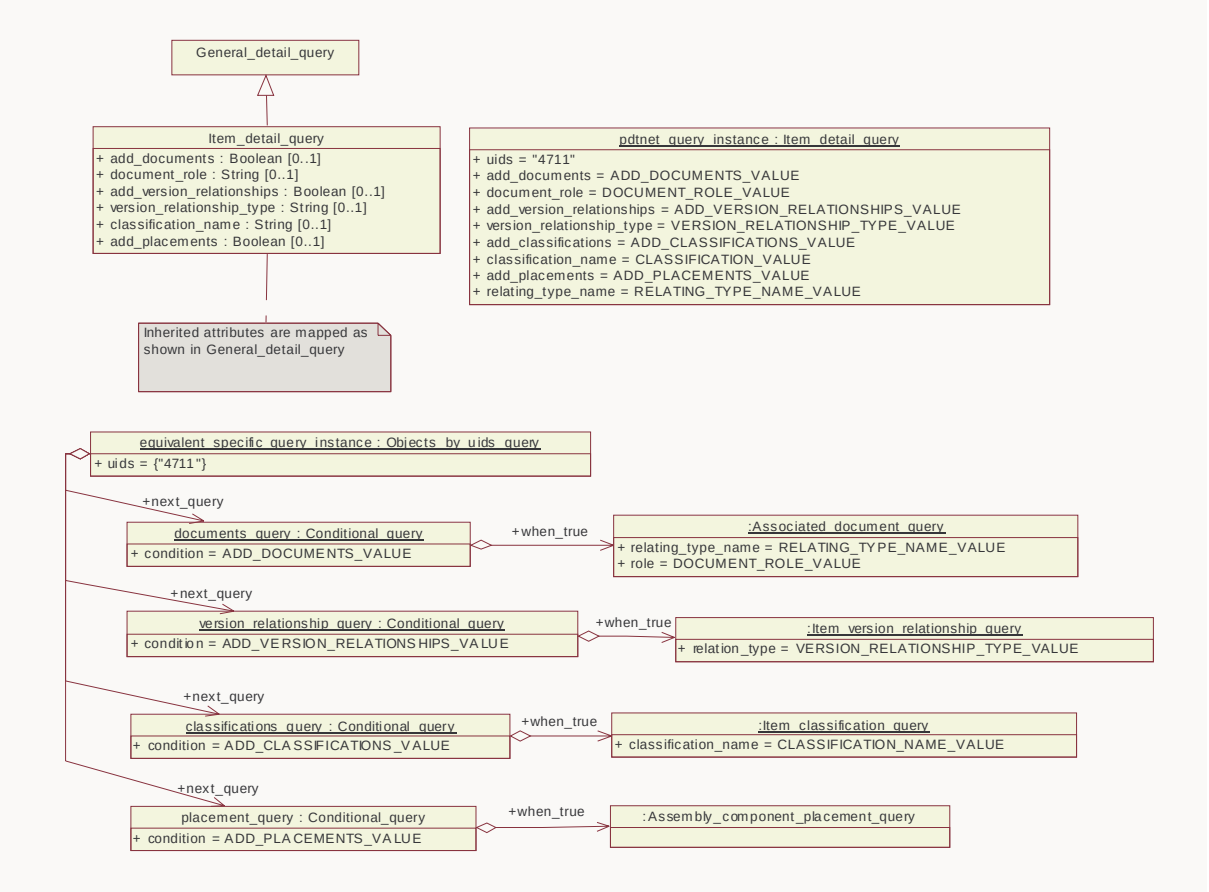


Figure 191 Definition, sample instance and equivalent specific query instance of the Item_detail_query

8.15.6 Item_selection_query

The Item_selection_query selects objects of class Item and includes related Item_version and Design_discipline_item_definition objects.

Parameters

- name: String [0..1]
- id: String [0..1]
- version_id: String [0..1]
- classification_name: String [0..1]
- omit_versions: Boolean[0..1]

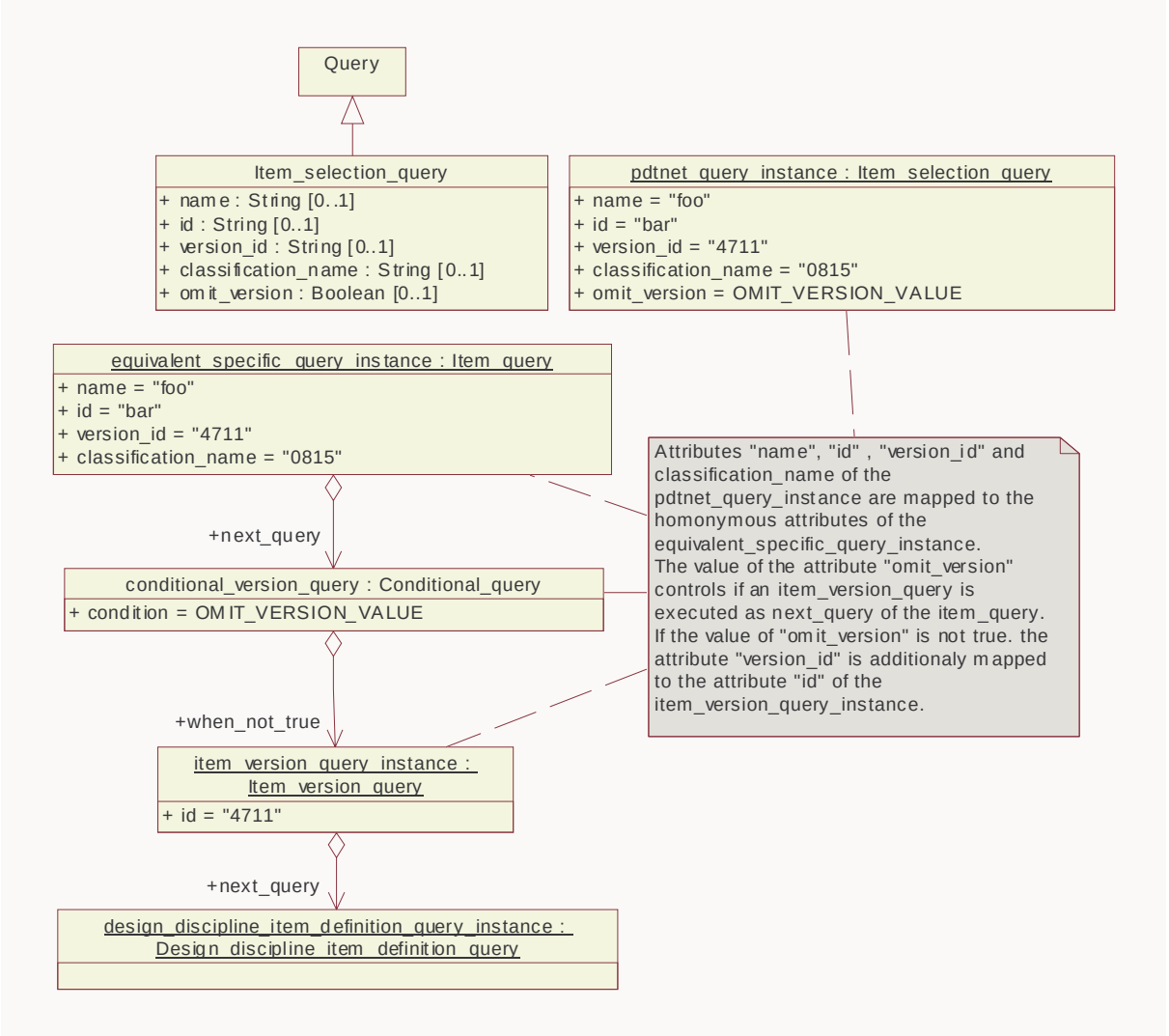


Figure 192 Definition, sample instance and equivalent specific query instance of the Item_selection_query

8.15.7 Item_ traversal_query

The Item_traversal_query traverses from a Design_discipline_item_definition object to the higher or lower Design_discipline_item_definition objects in an assembly structure.

Parameters

- uid: UID
- maximum_recursion_number: Integer [0..1]
- inverse_direction: Boolean [0..1]

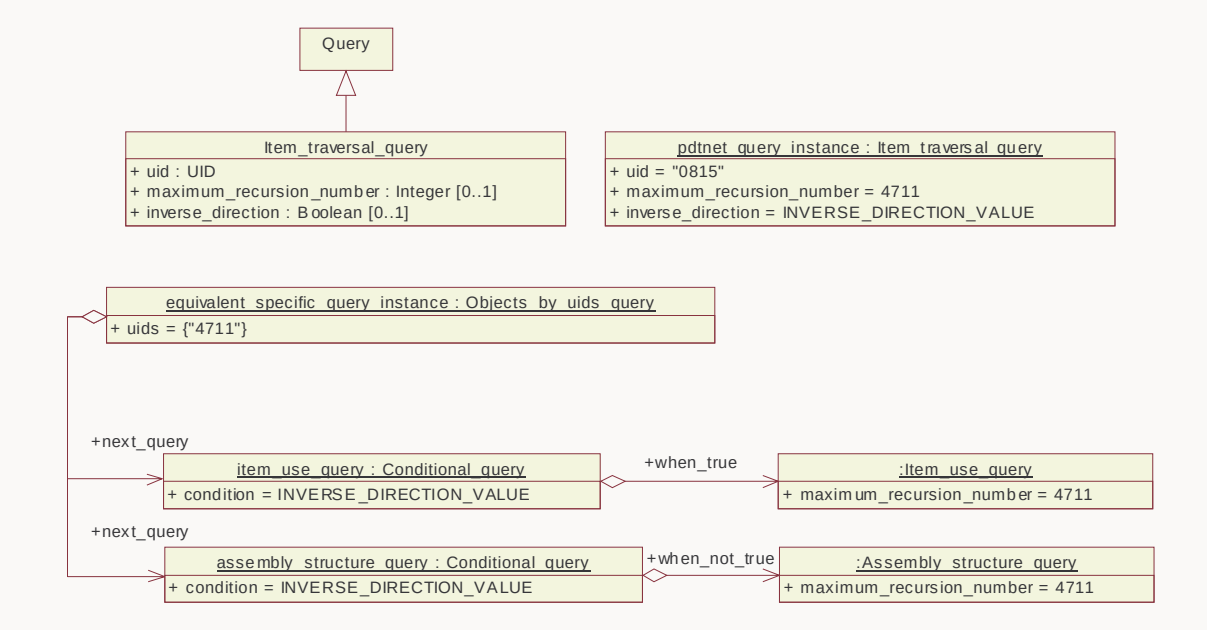


Figure 193 Definition, sample instance and equivalent specific query instance of the Item_traversal_query

8.15.8 Product_ detail_query

The Product_detail_query returns detail information of an Complex_product object selected by a uid.

Parameters (without inherited)

- add_configurations: Boolean [0..1]
- configuration_type: String [0..1]
- add_documents: Boolean [0..1]
- document_role: String [0..1]

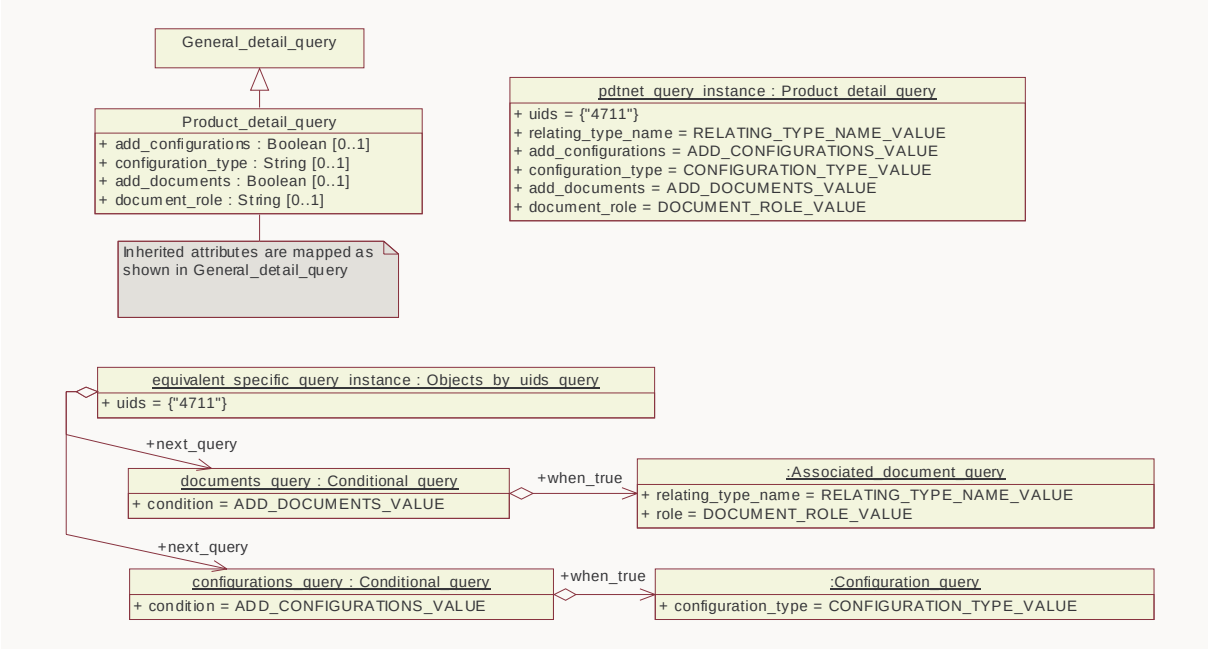


Figure 194 Definition, sample instance and equivalent specific query instance of the Product_detail_query

8.15.9 Product_ selection_query

The Product_selection_query selects objects of class Product_class and includes Product_function_component_select related via Class_structure_relationship objects.

Parameters

- name: String [0..1]
- id: String [0..1]

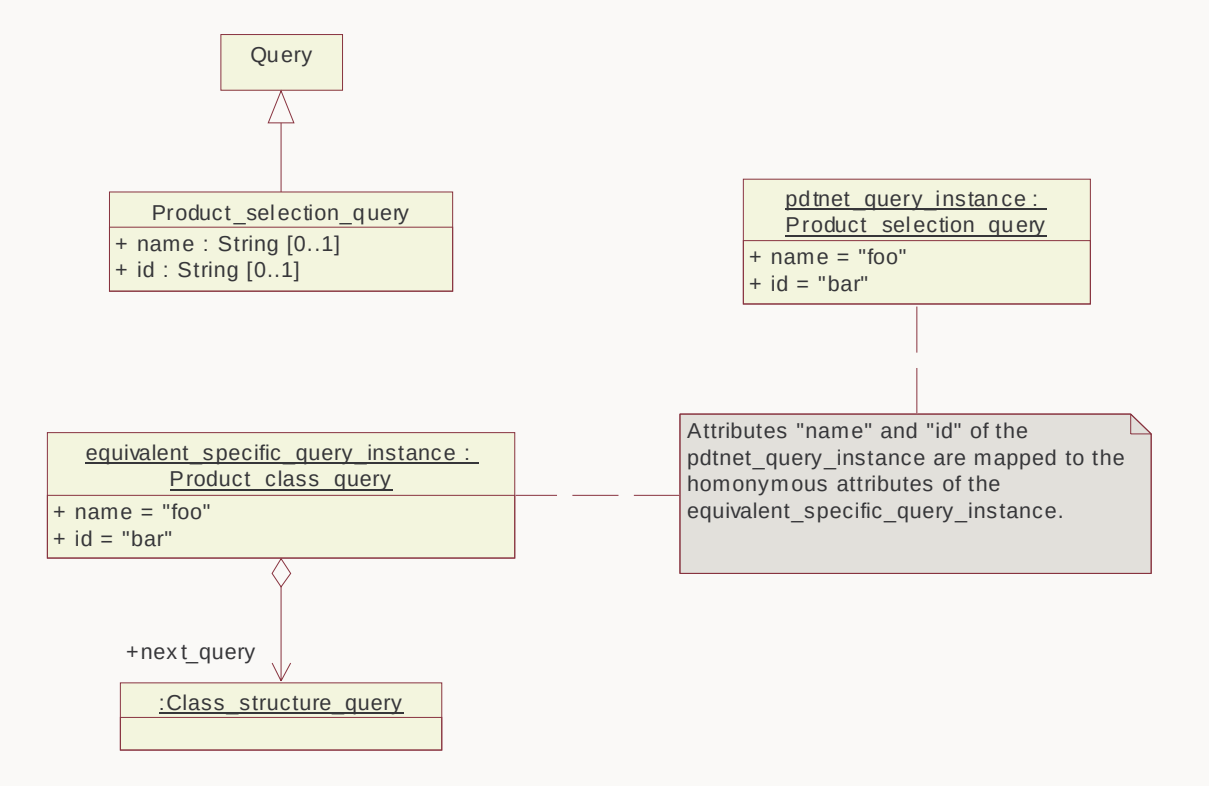


Figure 195 Definition, sample instance and equivalent specific query instance of the Product_selection_query

8.15.10 Product_traversal_query

The Product_traversal_query traverses from a Complex_product object selected by its uid via Product_structure_relationship or Alternative_solution_objects to related Complex_product or Item_instance objects in a product structure.

Parameters

- uid: UID
- maximum_recursion_number: Integer [0..1]
- relation_type: String [0..1]

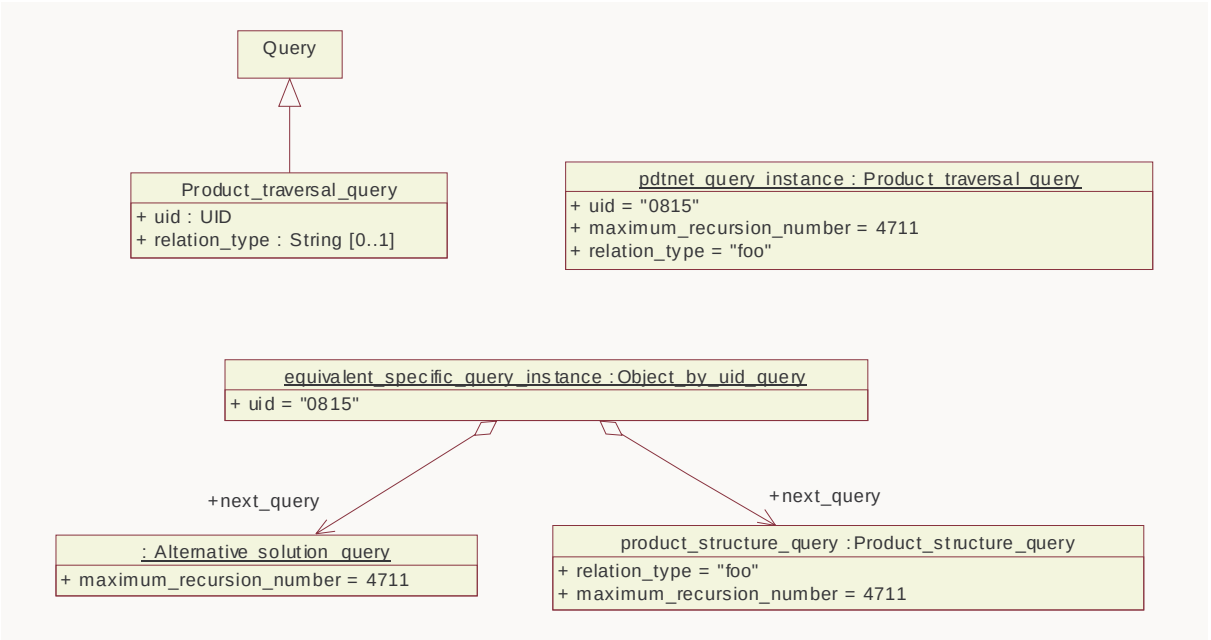


Figure 196 Definition, sample instance and equivalent specific query instance of the Product_traversal_query

9 Web services PSM

9.1 Overview

In the following sections a projection of the PIM into the platform specific model (PSM) with an execution infrastructure given by XML is defined. The projection is done via an enrichment of the model by a customized UML profile for XML Schema. This UML profile is given here for informal purposes.

9.2 UML Profile for XML Schema

To enrich the UML Informational PIM for XML Schema representation an UML profile is used. An UML profile has three key items namely stereotypes, tagged value called properties and constraints.

9.2.1 UML Model

On the entire UML model level the stereotype << XSDschema >> is applied. It can have the following tagged values:

Table 22 - Stereotype <<XSDschema>>

Can be applied to	UML model	
Property	**Value**	**Description**
targetNamespace	namespace URI	The URI which uniquely identifies this schema's namespace.
elementFormDefault	qualified unqualified	Specifies whether elements are quali-fied or unqualified.
attributeFormDefault	qualified unqualified	Specifies whether attributes are quali-fied or unqualified.
version	string value	The version of this schema.
modelGroup	all sequence choice none omitComplexType	Specifies the content model used for generating complexType definitions.
globalElement	true false	Specifies if global element declarations are created for complex types.
attributeMapping	element attribute	Specifies the mapping for UML attrib-utes.
roleMapping	element attribute	Specifies the mapping for roles of UML associations.
anonymousRole	true false	Specifies if role names of UML attrib-utes are mapped to elements.
anonymousType	true false	Specifies if the types of UML attributes are mapped to elements.

Table 22 - Stereotype <<XSDschema>>

typeContainment	true false	Specifies if types are contained instead of referencing them.
elementNamingMapping	firstLetterUpperCase firstLetterLowerCase upperCamelCase lowerCamelCase omitElement	Specifies the naming for elements.
attributeNamingMapping	firstLetterUpperCase firstLetterLowerCase upperCamelCase lowerCamelCase omitAttribute	Specifies the naming for attributes.

The four stereotypes XSDschema, XSDtranslatableString, XSDelement and XSDattribute form an hierarchy in that order. The derived stereotypes need to redefine only those values of the named properties which require new values.

Example

```
Model "PLM_services"
Stereotype << XSDschema >>
targetNamespace = http://www.omg.org/PLMServices1.0/XMLSchema
elementFormDefault      = qualified
attributeFormDefault    = unqualified
version                 = 1.0
modelGroup              = sequence
globalElement           = false
attributeMapping        = element
roleMapping             = element
anonymousRole           = false
anonymousType           = false
typeContainment         = false
elementNamingMapping    = firstLetterUpperCase
attributeNamingMapping  = firstLetterLowerCase
```

```
<xs:schema
    targetNamespace="http://www.omg.org/PLMServices1.0/XMLSchema"
    xmlns="http://www.omg.org/PLMServices1.0/XMLSchema"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified"
    attributeFormDefault="unqualified"
    version="1.0">
```

9.2.2 UML Package

On the UML package “Multi_language_support” the stereotype << XSDtranslatableString >> is applied. It doesn't have any tagged values.

XML provides an own mechanism to specify the language used in the contents and attribute values of any element in a XML document, the predefined attribute xml:lang. Based on this a XML specific concept has been developed to map the multi language support for string values of the PIM model.

If the UML Interface “String_select” is used by any UML composition, the type “Translatable_string” is used in XML instead. Therefore the predefined complex types “Translatable_string”, “Translation” and “Translations” are introduced in the XML schema.

Table 23 - Stereotype << XSDtranslatableString>>

Can be applied to	UML package

Example

```
Package "Multi_language_support"
Stereotype << XSDtranslatableString >>
```

```
<xs:complexType name="Item">
    …
    <xs:element name="Name" type="Translatable_string"/>
    …
</xs:complexType>

<xs:complexType name="Translatable_string">
    <xs:simpleContent>
        <xs:extension base="xs:string">
            <xs:attribute name="translations" type="xs:IDREF" use="optional"/>
                <xs:annotation>
                    <xs:documentation>REFERENCE TO Translations</xs:documentation>
                </xs:annotation>
            </xs:attribute>
            <xs:attribute ref="xml:lang" use="optional"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>

<xs:complexType name="Translation">
    <xs:simpleContent>
        <xs:extension base="xs:string">
            <xs:attribute ref="xml:lang" use="required"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>

<xs:complexType name="Translations">
    <xs:complexContent>
        <xs:extension base="PLM_root_object">
            <xs:sequence>
                <xs:element name="Translation" type="Translation" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
```

The mapping of instance values from UML to XML is as follows:

UML	XML
Default_language_string	Translatable_string
Multi_language_string .primary_language_dependent_string .String_with_language.contents	Translatable_string
Multi_language_string .primary_language_dependent_string .String_with_language .language_specification .Language.language_code	Translatable_string /@xml:lang
Multi_language_string .primary_language_dependent_string .String_with_language .language_specification .Language.country_code	Translatable_string /@xml:lang
Multi_language_string .additional_language_dependent_string .String_with_language.contents	Translation
Multi_language_string .additional_language_dependent_string .String_with_language .language_specification .Language.language_code	Translation /@xml:lang
Multi_language_string .additional_language_dependent_string .String_with_language .language_specification .Language.country_code	Translation /@xml:lang

9.2.3 UML Classes

On each UML class the stereotype << XSDcomplexType >> is applied. It can have the following tagged values:

Table 24 - Stereotype <<XSDcomplexType>>

Can be applied to	UML class	
Property	**Value**	**Description**
modelGroup	all sequence choice multiChoice omitComplexType	Specifies the content model used for generating this complexType definition.

Table 24 - Stereotype <<XSDcomplexType>>

globalElement	true false	Specifies if a global element declaration is created for this complexType.
attributeMapping	element attribute	Specifies the mapping for UML attributes within this complexType.
roleMapping	element attribute	Specifies the mapping for roles of UML associations within this complexType.
anonymousRole	true false	Specifies if the role names of UML attributes are mapped to elements within this complex type.
anonymousType	true false	Specifies if the types of UML attributes are mapped to elements within this complex type.
typeContainment	true false	Specifies if types are contained instead of referencing them within this complex type.
elementNamingMapping	firstLetterUpperCase firstLetterLowerCase upperCamelCase lowerCamelCase omitElement	Specifies the naming for elements within this complex type.
attributeNamingMapping	firstLetterUpperCase firstLetterLowerCase upperCamelCase lowerCamelCase omitAttribute	Specifies the naming for attributes within this complex type.

Each of the above named properties shall apply to all UML classes, attributes, associations and compositions which do not have an own stereotype overwriting these values.

Generalization

Only single inheritance is treated by the UML to XML Schema mapping. This is sufficient since the PIM UML model does not contain any multiple inheritance. Each subclass will be a complexType with complexContent and extension base=”superclass”. Abstract classes are mapped to complex types which are abstract.

Example

```
Class "PLM_container"
Stereotype << XSDcomplexType >>
modelGroup              = multiChoice
globalElement           = true
```

```
<xs:element name="PLM_container" type="PLM_container"/>
<xs:complexType name="PLM_container">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
        …
    </xs:choice>
</xs:complexType>
```

9.2.4 UML Interfaces

UML interfaces are not treated by the UML to XML Schema mapping since the interfaces are only referenced by other classes. These references are mapped to XML schema references of type IDREF and IDREFS, which point to the underlying types of an interface.

9.2.5 UML Attributes, Associations and Compositions

On each UML attribute, association and composition the stereotypes << XSDelement >> or << XSDattribute >> are applied. They can have the following tagged values:

Table 25 - Stereotype <<XSDelement>>

Can be applied to	UML attribute, UML association, UML composition	
Property	**Value**	**Description**
position	integer value	Causes the elements to be ordered within a sequence model group of the containing complexType.
anonymousRole	truefalse	Specifies if the role name of an UML attribute is mapped to an element.
anonymousType	truefalse	Specifies if the type of an UML attribute is mapped to an element.
typeContainment	truefalse	Specifies if the type is contained instead of referencing it.
elementNamingMapping	firstLetterUpperCase firstLetterLowerCase upperCamelCase lowerCamelCase omitElement	Specifies the naming for this element.

Example

```
Attribute "relation_type"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true


Composition "change"
Stereotype << XSDelement >>
```

```
position                = 04
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Association "related"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false
```

```
<xs:complexType name="Item_version_relationship">
    <xs:complexContent>
        <xs:extension base="PLM_object">
            <xs:sequence>
                <xs:element name="Related" type="xs:IDREF">
                    <xs:annotation>
                        <xs:documentation>REFERENCE TO Item_version</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element name="Description" type="Translatable_string" minOccurs="0"/>
                <xs:element name="Relation_type" type="xs:string"/>
                <xs:element name="Change" type="Change" minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
```

Table 26 - Stereotype <<XSDattribute>>

Can be applied to	UML attribute, UML association, UML composition	
Property	**Value**	**Description**
attributeType	qualified name	Specifies the type of the attribute.
use	prohibited optional required fixed	Specifies the use of the attribute.
attributeNamingMapping	firstLetterUpperCase firstLetterLowerCase upperCamelCase lowerCamelCase omitAttribute	Specifies the naming for this attribute.

Example

```
Attribute "uid"
Stereotype << XSDattribute >>
attributeType           = xs:ID
use                     = required
```

```
<xs:complexType name="PLM_object" abstract="true">
    <xs:attribute name="uid" type="xs:ID" use="required"/>
```

```
</xs:complexType>
```

Most of the UML attributes, associations and compositions are mapped to elements in the XML Schema and also a position of these elements are needed if the modelGroup is a sequence. This is done by applying a position value to an UML attribute, association or composition.

If the type of the UML attribute is a data type it is mapped to a corresponding primitive data type of the XML Schema Definition.

Table 27 -:Mapping of UML data types to XSD primitive types

UML datatype	XSD primitive type
String	xs:string
Double	xs:double
Boolean	xs:boolean
Integer	xs:int
UID	xs:ID
Date	xs:date
Time	xs:time

The multiplicity of an UML attribute, association or composition is mapped to the corresponding multiplicity in the XML schema. For elements the values `minOccurs` and `maxOccurs` are used, for attributes the value `use`.

9.3 PLM Services Web services WSDL

The Computational Viewpoint of the Web service PSM is defined in the Web Services Description Language (WSDL) 1.1. The WSDL imports the XML Schema defined by the Informational Viewpoint. The Web service PSM contains definitions of two ports: PLM_connection_factory and PLM_connection. Due to the fact that Web services can not transfer object references as parameters or results of operations, the syntax and semantic of the operation get_connection() has changed in comparison with the PIM. In the Web service PSM get_connection returns a PLM_session instance which contains a Session_context and a Location element. The Session_context identifies a session and has to be added as a soap header element to each operation request to a PLM_connection port for this session. The optional Location element overrides the address element of the PLM_connection port in the WSDL. The PIM object types PLM_resource_adapter and PLM_object_factory have no counterpart in the Web service PSM.

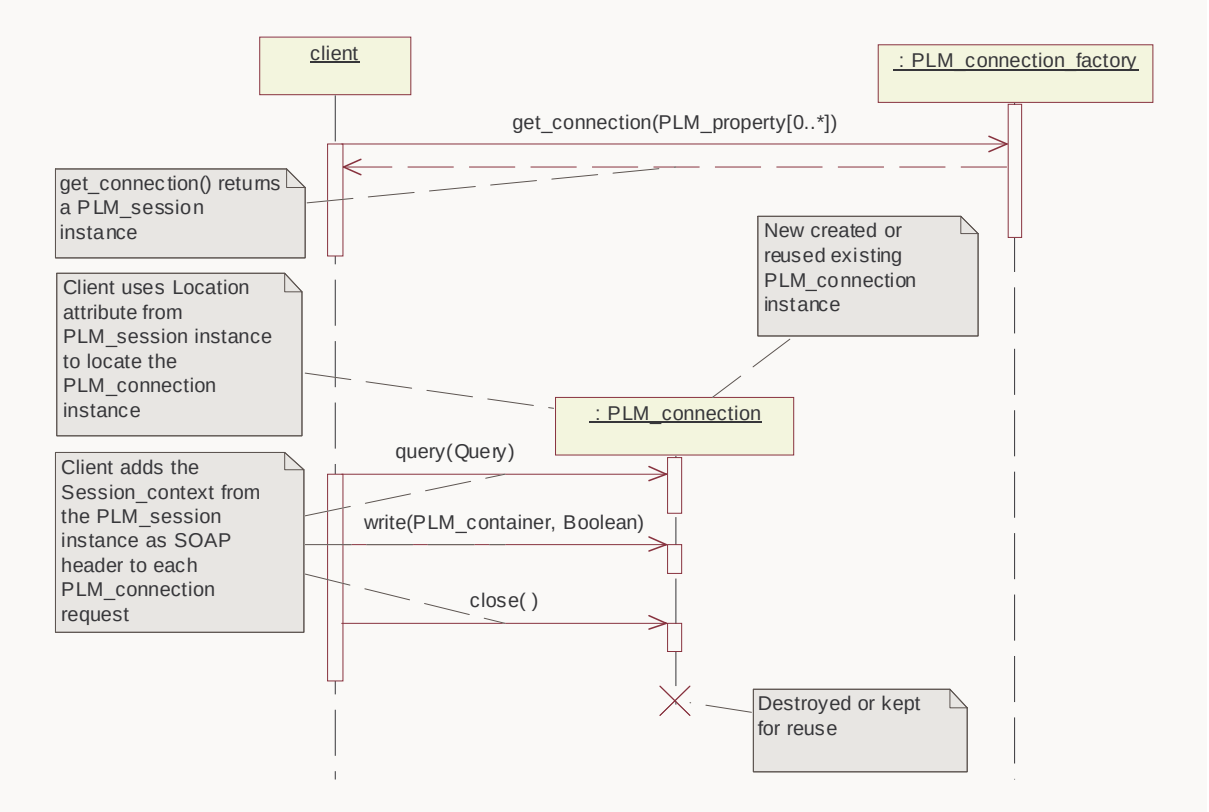


Figure 156 - Sequence diagram of a PLM session

9.3.1 Query Examples

9.3.1.1 Generic Queries Conformance Point Example

Query for all Item_version objects with id='bar' of Item objects with Name='foo'.

```
<Query xsi:type="Location_path">
    <First_step>
        <Role_name>item</Role_name>
        <Predicate xsi:type="Attribute_predicate">
            <Attribute_name>name</Attribute_name>
            <Attribute_value>foo</Attribute_value>
        </Predicate>
        <Next_step>
            <Role_name>item_version</Role_name>
            <Predicate xsi:type="Attribute_predicate">
                <Attribute_name>id</Attribute_name>
                <Attribute_value>bar</Attribute_value>
            </Predicate>
        </Next_step>
    </First_step>
</Query>
```

9.3.1.2 XPath Queries Conformance Point Example

Query for all Item_version objects of Item objects with id='foo'.

```
<Query xsi:type="X_path">
    <Expression>//Item[Id='foo']/Item_version</Expression>
</Query>
```

9.3.1.3 PDTnet Queries Conformance Point Examples

Assembly_structure_query for all Design_discipline_item_definition objects of Item_version objects with id='4711' of Item objects with name='bar' and name language='en-US'

```
<Query xsi:type="Item_query">
    <Name>bar</Name>
    <Name_language>en-US</Name_language>
    <Next_query xsi:type="Item_version_query">
        <Id>4711</Id>
        <Next_query xsi_type="Design_discipline_item_definition">
            <Next_query xsi:type="Assembly_structure_query"/>
        </Next_query>
    </Next_query>
</Query>
```

Assembly_structure_query for Design_discipline_item_definition with an initial_context with application_domain='mechanical design' and life_cycle_stage='design' of all Item_version objects of Item objects with id='foo'. The result is extended by associated Date_time, Organization and Property_value objects.

```
<Query xsi:type="Item_query">
    <Id>foo</Id>
    <Next_query xsi:type="Item_version_query">
        <Next_query xsi_type="Design_discipline_item_definition">
            <Application_domain>mechanical design</Application_domain>
            <Life_cycle_stage>mechanical design</Life_cycle_stage>
            <Next_query xsi:type="Assembly_structure_query">
                <Next_query xsi:type="Associated_date_time_query"/>
                <Next_query xsi:type="Associated_organization_query"/>
                <Next_query xsi:type="Associated_property_query"/>
            </Next_query>
        </Next_query>
    </Next_query>
</Query>
```

Assembly_structure_query for the PLM_object with uid='assembly123' (which should be an Assembly_definition).

```
<Query xsi:type="Object_by_uid_query">
    <uid>assembly123</uid>
    <Next_query xsi:type="Assembly_structure_query"/>
</Query>
```

9.3.2 Realization of Use cases

9.3.2.1 Authentication

Request

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XML-
Schema-instance">
<soapenv:Body>
<get_connection xmlns="http://schema.omg.org/specs/PLM/1.0/
PLM_connection_factory#get_connection">
<properties xmlns="">
<PLM_property>
<Name>user</Name>
<Value>test</Value>
</PLM_property>
<PLM_property>
<Name>password</Name>
<Value>test</Value>
</PLM_property>
</properties>
</get_connection>
</soapenv:Body>
</soapenv:Envelope>
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XML-
Schema-instance">
<soapenv:Body>
<get_connectionResponse xmlns="http://schema.omg.org/specs/PLM/1.0/
PLM_connection_factory#get_connection">
<get_connectionReturn>
<Location xmlns="">http://localhost:8081/axis/services/PLM_connection</Location>
<Id xmlns="">3426814710318558298</Id>
</get_connectionReturn>
</get_connectionResponse>
</soapenv:Body>
</soapenv:Envelope>
```

9.3.2.2 Start node identification

The “start node identification” is realized by the concatenation of the following three queries:

- Item_query
- Item_version_query

- Design_discipline_item_definition_query

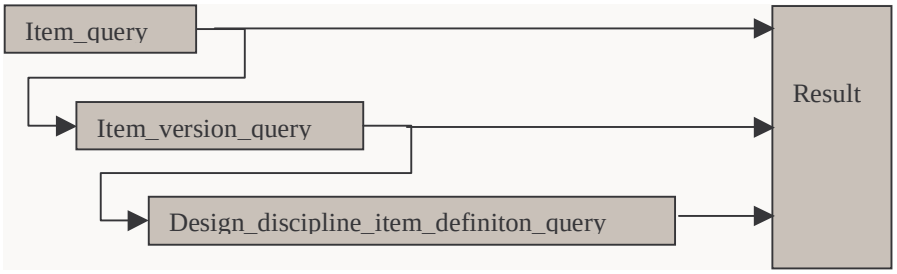


Figure 157 Query concatenation for realizing start node identification

Request

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XML-
Schema-instance">
<soapenv:Header>
<ns1:Id soapenv:mustUnderstand="0" xsi:type="xsd:string" xmlns:ns1="http://www.omg.org/
PLMServices1.0/Services">5119095223646270101</ns1:Id>
</soapenv:Header>
<soapenv:Body>
<query xmlns="http://schema.omg.org/specs/PLM/1.0/PLM_connection#query">
<query xsi:type="ns2:Item_query" xmlns:ns2="http://schema.omg.org/specs/PLM/1.0/Compu-
tationalModel" xmlns="">
<Next_query xsi:type="ns2:Item_version_query">
<Next_query xsi:type="ns2:Design_discipline_item_definition_query"/>
<Id>0001,1</Id>
</Next_query>
<Id>A4000100000</Id>
<Name>Trego</Name>
</query>
</query>
</soapenv:Body>
</soapenv:Envelope>
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XML-
Schema-instance">
<soapenv:Header>
<ns1:Id soapenv:mustUnderstand="0" xsi:type="xsd:long" xmlns:ns1="http://www.omg.org/
PLMServices1.0/Services">5119095223646270101</ns1:Id>
</soapenv:Header>
<soapenv:Body>
<queryResponse xmlns="http://schema.omg.org/specs/PLM/1.0/PLM_connection#query">
<ns2:response xsi:type="ns2:PLM_container" uid="plm_container_0" version_id="1.0"
xmlns:ns2="http://schema.omg.org/specs/PLM/1.0/InformationalModel">
<ns2:Application_context uid="Application_context_40">
<ns2:Application_domain>Application_domain_1</ns2:Application_domain>
<ns2:Life_cycle_stage>design</ns2:Life_cycle_stage>
</ns2:Application_context>
<ns2:Item uid="Item_2920">
<ns2:Id>A4000100000</ns2:Id>
<ns2:Name>Trego</ns2:Name>
<ns2:Item_version uid="Item_version_4070">
<ns2:Id>0001,1</ns2:Id>
<ns2:Design_discipline_item_definition xsi:type="ns2:Assembly_definition"
uid="Assembly_definition_4170">
<ns2:Id>pv0002</ns2:Id>
<ns2:Initial_context>Application_context_40</ns2:Initial_context>
</ns2:Design_discipline_item_definition>
<ns2:Design_discipline_item_definition uid="design_discipline_item_definition_0">
<ns2:Id>design_discipline_item_definition_id_0</ns2:Id>
<ns2:Initial_context>Application_context_40</ns2:Initial_context>
</ns2:Design_discipline_item_definition>
</ns2:Item_version>
</ns2:Item>
</ns2:response>
</queryResponse>
</soapenv:Body>
</soapenv:Envelope>
```

9.3.2.3 Browsing down product structure data

The “browsing down product structure data” is realized by the concatenation of the following five queries:

- Item_query
- Item_version_query
- Design_discipline_item_definition_query
- Assembly_structure_query
- Item_classification_query

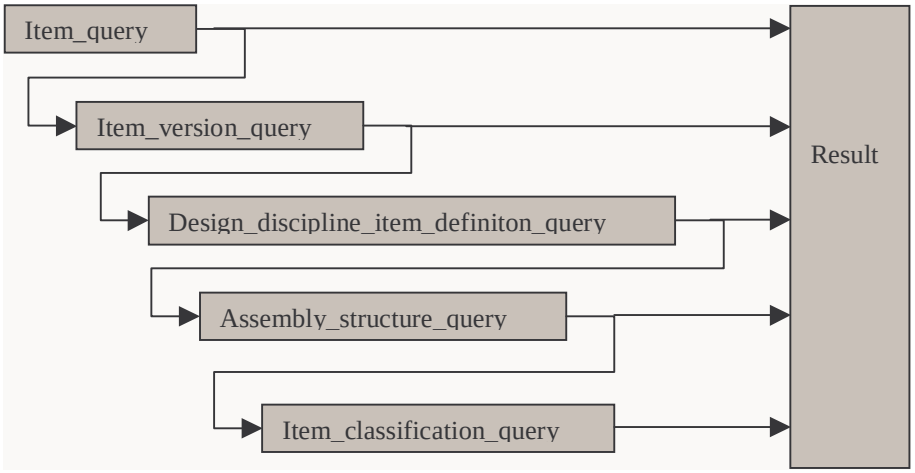


Figure 158 Query concatenation for realizing browsing down product structure data

Request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XML-
Schema-instance">
<soapenv:Header>
<ns1:Id soapenv:mustUnderstand="0" xsi:type="xsd:string" xmlns:ns1="http://
schema.omg.org/specs/PLM/1.0/Services">3624345198239672382</ns1:Id>
</soapenv:Header>
<soapenv:Body>
<query xmlns="http://schema.omg.org/specs/PLM/1.0/PLM_connection#query">
<query xsi:type="ns2:Item_query" xmlns:ns2="http://schema.omg.org/specs/PLM/1.0/Compu-
tationalModel" xmlns="">
<Next_query xsi:type="ns2:Item_version_query">
<Next_query xsi:type="ns2:Design_discipline_item_definition_query">
<Next_query xsi:type="ns2:Assembly_structure_query">
<Maximum_recursion_number>1</Maximum_recursion_number>
<Next_query xsi:type="ns2:Item_classification_query"/>
</Next_query>
</Next_query>
<Id>0001,1</Id>
</Next_query>
<Id>A4000100000</Id>
<Name>Trego</Name>
</query>
</query>
</soapenv:Body>
</soapenv:Envelope>
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XML-
Schema-instance">
<soapenv:Header>
<ns1:Id soapenv:mustUnderstand="0" xsi:type="xsd:long" xmlns:ns1="http://
schema.omg.org/specs/PLM/1.0/Services">3624345198239672382</ns1:Id>
</soapenv:Header>
<soapenv:Body>
<queryResponse xmlns="http://schema.omg.org/specs/PLM/1.0/PLM_connection#query">
<ns2:response xsi:type="ns2:PLM_container" uid="plm_container_0" version_id="1.0"
xmlns:ns2="http://www.omg.org/PLMServices1.0/InformationalModel">
<ns2:Application_context uid="Application_context_40">
<ns2:Application_domain>Application_domain_1</ns2:Application_domain>
<ns2:Life_cycle_stage>design</ns2:Life_cycle_stage>
</ns2:Application_context>
<ns2:Item uid="Item_2920">
<ns2:Id>A4000100000</ns2:Id>
<ns2:Name>Trego</ns2:Name>
<ns2:Item_version uid="Item_version_4070">
<ns2:Id>0001,1</ns2:Id>
<ns2:Design_discipline_item_definition xsi:type="ns2:Assembly_definition"
uid="Assembly_definition_4170">
<ns2:Id>pv0002</ns2:Id>
<ns2:Initial_context>Application_context_40</ns2:Initial_context>
<ns2:Item_definition_instance_relationship xsi:type="ns2:Next_higher_assembly"
uid="Next_higher_assembly_26820">
<ns2:Related>Single_instance_26820</ns2:Related>
</ns2:Item_definition_instance_relationship>
<ns2:Item_definition_instance_relationship xsi:type="ns2:Next_higher_assembly"
uid="Next_higher_assembly_27920">
<ns2:Related>Single_instance_27920</ns2:Related>
</ns2:Item_definition_instance_relationship>
<ns2:Item_definition_instance_relationship xsi:type="ns2:Next_higher_assembly"
uid="Next_higher_assembly_29920">
<ns2:Related>Single_instance_29920</ns2:Related>
</ns2:Item_definition_instance_relationship>
<ns2:Item_definition_instance_relationship xsi:type="ns2:Next_higher_assembly"
uid="Next_higher_assembly_31220">
<ns2:Related>Single_instance_31220</ns2:Related>
</ns2:Item_definition_instance_relationship>
</ns2:Design_discipline_item_definition>
<ns2:Design_discipline_item_definition uid="design_discipline_item_definition_0">
<ns2:Id>design_discipline_item_definition_id_0</ns2:Id>
<ns2:Initial_context>Application_context_40</ns2:Initial_context>
</ns2:Design_discipline_item_definition>
</ns2:Item_version>
</ns2:Item>
<ns2:Item uid="Item_3280">
<ns2:Id>A4000040000</ns2:Id>
<ns2:Name>Mulde</ns2:Name>
<ns2:Item_version uid="Item_version_25300">
<ns2:Id>0001,1</ns2:Id>
<ns2:Design_discipline_item_definition xsi:type="ns2:Assembly_definition"
uid="Assembly_definition_25400">
<ns2:Id>pv0038</ns2:Id>
<ns2:Initial_context>Application_context_40</ns2:Initial_context>
<ns2:Item_instance xsi:type="ns2:Single_instance" uid="Single_instance_26820">
<ns2:Id>Single_instance_26820_ID</ns2:Id>
</ns2:Item_instance>
```

9.3.2.4 Browsing up product structure data

The “browsing up product structure data” is realized by the concatenation of the following five queries:

- Item_query
- Item_version_query
- Design_discipline_item_definition_query
- Item_use_query
- Item_classification_query

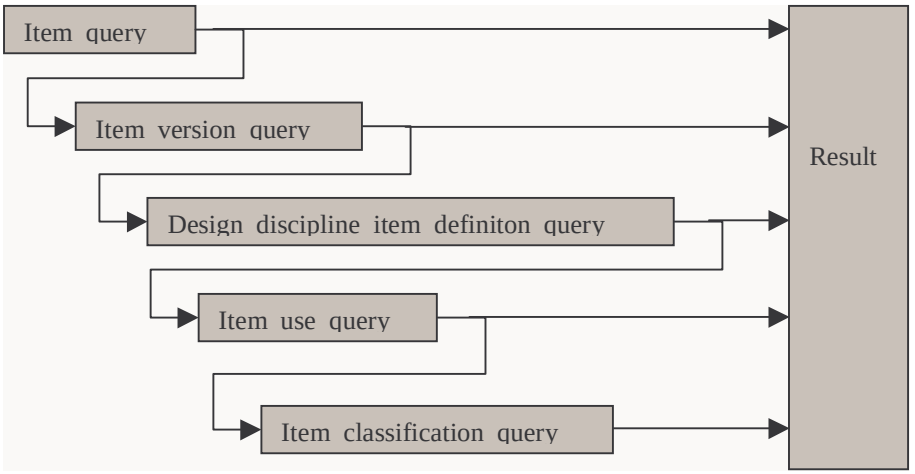


Figure 159 Query concatenation for realizing browsing up product structure data

Request

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XML-
Schema-instance">
<soapenv:Header>
<ns1:Id soapenv:mustUnderstand="0" xsi:type="xsd:string" xmlns:ns1="http://
schema.omg.org/specs/PLM/1.0/Services">-1271763691436743697</ns1:Id>
</soapenv:Header>
<soapenv:Body>
<query xmlns="http://schema.omg.org/specs/PLM/1.0/PLM_connection#query">
<query xsi:type="ns2:Item_query" xmlns:ns2="http://schema.omg.org/specs/PLM/1.0/Compu-
tationalModel" xmlns="">
<Next_query xsi:type="ns2:Item_version_query">
<Next_query xsi:type="ns2:Design_discipline_item_definition_query">
<Next_query xsi:type="ns2:Item_use_query">
<Maximum_recursion_number>1</Maximum_recursion_number>
<Next_query xsi:type="ns2:Item_classification_query"/>
</Next_query>
</Next_query>
<Id>0001,1</Id>
</Next_query>
<Id>A4000040000</Id>
<Name>Mulde</Name>
</query>
</query>
</soapenv:Body>
</soapenv:Envelope>
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XML-
Schema-instance">
<soapenv:Header>
<ns1:Id soapenv:mustUnderstand="0" xsi:type="xsd:long" xmlns:ns1="http://
schema.omg.org/specs/PLM/1.0/Services">-1271763691436743697</ns1:Id>
</soapenv:Header>
<soapenv:Body>
<queryResponse xmlns="http://schema.omg.org/specs/PLM/1.0/PLM_connection#query">
<ns2:response xsi:type="ns2:PLM_container" uid="plm_container_0" version_id="1.0"
xmlns:ns2="http://schema.omg.org/specs/PLM/1.0/InformationalModel">
<ns2:Application_context uid="Application_context_40">
<ns2:Application_domain>Application_domain_1</ns2:Application_domain>
<ns2:Life_cycle_stage>design</ns2:Life_cycle_stage>
</ns2:Application_context>
<ns2:Item uid="Item_3280">
<ns2:Id>A4000040000</ns2:Id>
<ns2:Name>Mulde</ns2:Name>
<ns2:Item_version uid="Item_version_25300">
<ns2:Id>0001,1</ns2:Id>
<ns2:Design_discipline_item_definition xsi:type="ns2:Assembly_definition"
uid="Assembly_definition_25400">
<ns2:Id>pv0038</ns2:Id>
<ns2:Initial_context>Application_context_40</ns2:Initial_context>
<ns2:Item_instance xsi:type="ns2:Single_instance" uid="Single_instance_26820">
<ns2:Id>Single_instance_26820_ID</ns2:Id>
</ns2:Item_instance>
</ns2:Design_discipline_item_definition>
</ns2:Item_version>
</ns2:Item>
<ns2:Item uid="Item_2920">
<ns2:Id>A4000100000</ns2:Id>
<ns2:Name>Trego</ns2:Name>
<ns2:Item_version uid="Item_version_4070">
<ns2:Id>0001,1</ns2:Id>
<ns2:Design_discipline_item_definition xsi:type="ns2:Assembly_definition"
uid="Assembly_definition_4170">
<ns2:Id>pv0002</ns2:Id>
<ns2:Initial_context>Application_context_40</ns2:Initial_context>
<ns2:Item_definition_instance_relationship xsi:type="ns2:Next_higher_assembly"
uid="Next_higher_assembly_26820">
<ns2:Related>Single_instance_26820</ns2:Related>
</ns2:Item_definition_instance_relationship>
</ns2:Design_discipline_item_definition>
</ns2:Item_version>
</ns2:Item>
<ns2:Specific_item_classification uid="Specific_item_classification_31870">
<ns2:Associated_item>Item_2920 Item_3280</ns2:Associated_item>
<ns2:Classification_name>assembly</ns2:Classification_name>
</ns2:Specific_item_classification>
<ns2:Specific_item_classification uid="Specific_item_classification_31890">
<ns2:Associated_item>Item_2920 Item_3280</ns2:Associated_item>
<ns2:Classification_name>part</ns2:Classification_name>
</ns2:Specific_item_classification>
</ns2:response>
</queryResponse>
</soapenv:Body>
</soapenv:Envelope>
```

9.3.2.5 Download of Metadata including structures

The “download of metadata including structures” is realized by the concatenation of the following five queries:

- Item_query
- Item_version_query
- Design_discipline_item_definition_query
- Item_version_relationship_query
- Item_classification_query

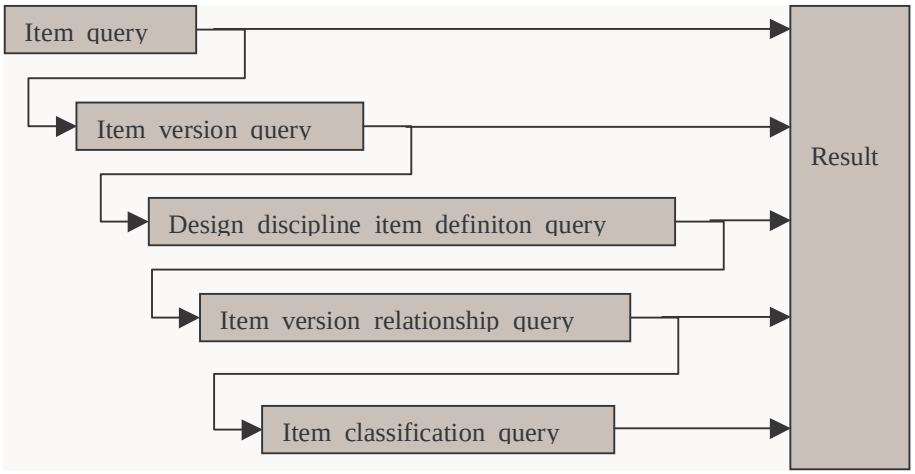


Figure 160 Query concatenation for realizing download of metadata including structures

Request

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XML-
Schema-instance">
<soapenv:Header>
<ns1:Id soapenv:mustUnderstand="0" xsi:type="xsd:string" xmlns:ns1="http://
schema.omg.org/specs/PLM/1.0/Services">-7356016370443115288</ns1:Id>
</soapenv:Header>
<soapenv:Body>
<query xmlns="http://schema.omg.org/specs/PLM/1.0/PLM_connection#query">
<query xsi:type="ns2:Item_query" xmlns:ns2="http://schema.omg.org/specs/PLM/1.0/Compu-
tationalModel" xmlns="">
<Next_query xsi:type="ns2:Item_version_query">
<Next_query xsi:type="ns2:Design_discipline_item_definition_query">
<Next_query xsi:type="ns2:Item_version_relationship_query">
<Next_query xsi:type="ns2:Item_classification_query"/>
</Next_query>
</Next_query>
<Id>0001,1</Id>
</Next_query>
<Id>A4000100000</Id>
<Name>Trego</Name>
</query>
</query>
</soapenv:Body>
</soapenv:Envelope>
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XML-
Schema-instance">
<soapenv:Header>
<ns1:Id soapenv:mustUnderstand="0" xsi:type="xsd:long" xmlns:ns1="http://
schema.omg.org/specs/PLM/1.0/Services">-7356016370443115288</ns1:Id>
</soapenv:Header>
<soapenv:Body>
<queryResponse xmlns="http://schema.omg.org/specs/PLM/1.0/PLM_connection#query">
<ns2:response xsi:type="ns2:PLM_container" uid="plm_container_0" version_id="1.0"
xmlns:ns2="http://schema.omg.org/specs/PLM/1.0/InformationalModel">
<ns2:Application_context uid="Application_context_40">
<ns2:Application_domain>Application_domain_1</ns2:Application_domain>
<ns2:Life_cycle_stage>design</ns2:Life_cycle_stage>
</ns2:Application_context>
<ns2:Item uid="Item_2920">
<ns2:Id>A4000100000</ns2:Id>
<ns2:Name>Trego</ns2:Name>
<ns2:Item_version uid="Item_version_4070">
<ns2:Id>0001,1</ns2:Id>
<ns2:Item_version_relationship uid="item_version_relationship_0">
<ns2:Related>Item_version_12450</ns2:Related>
<ns2:Relation_type>relation_type_0</ns2:Relation_type>
</ns2:Item_version_relationship>
<ns2:Item_version_relationship uid="item_version_relationship_1">
<ns2:Related>Item_version_10310</ns2:Related>
<ns2:Relation_type>relation_type_1</ns2:Relation_type>
</ns2:Item_version_relationship>
<ns2:Design_discipline_item_definition xsi:type="ns2:Assembly_definition"
uid="Assembly_definition_4170">
<ns2:Id>pv0002</ns2:Id>
<ns2:Initial_context>Application_context_40</ns2:Initial_context>
</ns2:Design_discipline_item_definition>
<ns2:Design_discipline_item_definition uid="design_discipline_item_definition_0">
<ns2:Id>design_discipline_item_definition_id_0</ns2:Id>
<ns2:Initial_context>Application_context_40</ns2:Initial_context>
</ns2:Design_discipline_item_definition>
</ns2:Item_version>
</ns2:Item>
<ns2:Item uid="Item_3070">
<ns2:Id>A4000002101</ns2:Id>
<ns2:Name>Rad</ns2:Name>
<ns2:Item_version uid="Item_version_12450">
<ns2:Id>0001,1</ns2:Id>
</ns2:Item_version>
</ns2:Item>
<ns2:Item uid="Item_3030">
<ns2:Id>A4000003902</ns2:Id>
<ns2:Name>Tuer rechts</ns2:Name>
<ns2:Item_version uid="Item_version_10310">
<ns2:Id>0001,1</ns2:Id>
</ns2:Item_version>
</ns2:Item>
<ns2:Specific_item_classification uid="Specific_item_classification_31870">
<ns2:Associated_item>Item_2920</ns2:Associated_item>
<ns2:Classification_name>assembly</ns2:Classification_name>
</ns2:Specific_item_classification>
<ns2:Specific_item_classification uid="Specific_item_classification_31890">
<ns2:Associated_item>Item_2920 Item_3030 Item_3070</ns2:Associated_item>
```

A PIM for Product Lifecycle Management Services

A.1 PIM Equivalence Model

In this section the EXPRESS platform independent equivalence model is defined. This EXPRESS model is produced by the EXPRESS-X mapping specification described in Chapter 7.4 and is equivalent to the ARM of the relevant subset of ISO10303 AP214 [8]. It is listed here with no further explanation. For a documentation see the explanation to the corresponding UML elements in Chapter 7.7.

A.1.1 Part Identification

```
 ENTITY application_context;
  description : OPTIONAL string_select;
  application_domain : STRING;
  life_cycle_stage : STRING;
 END_ENTITY;

 ENTITY item;
  id : STRING;
  name : string_select;
  description : OPTIONAL string_select;
  INVERSE
   associated_version : SET[1:?] OF item_version FOR associated_item;
   item_classification : SET[1:?] OF specific_item_classification FOR asso-
ciated_item;
 END_ENTITY;

 ENTITY item_definition_relationship
  ABSTRACT SUPERTYPE OF (
ONEOF(replaced_definition_relationship,geometrical_relationship,tool_part_relationsh
ip,make_from_relationship,general_item_definition_relationship) );
  relating : design_discipline_item_definition;
  related : design_discipline_item_definition;
 END_ENTITY;

 ENTITY item_version;
  id : STRING;
  associated_item : item;
  description : OPTIONAL string_select;
 INVERSE
  associated_product : SET [0:1] OF product_design FOR design;
 END_ENTITY;

 ENTITY item_version_relationship;
  relating : item_version;
  related : item_version;
```

```
  description : OPTIONAL string_select;
  relation_type : STRING;
 END_ENTITY;
```

A.1.2 Part Structure

```
 ENTITY assembly_component_relationship
  SUPERTYPE OF (next_higher_assembly)
  SUBTYPE OF (item_definition_instance_relationship);
  SELF\item_definition_instance_relationship.relating : assem-bly_definition;
  placement : OPTIONAL transformation_select;
 END_ENTITY;

 ENTITY assembly_definition
  SUBTYPE OF (design_discipline_item_definition);
  assembly_type : OPTIONAL STRING;
 END_ENTITY;

 ENTITY collected_item_association
  SUBTYPE OF (item_definition_instance_relationship);
  SELF\item_definition_instance_relationship.relating : collec-tion_definition;
 END_ENTITY;

 ENTITY collection_definition
  SUBTYPE OF (design_discipline_item_definition);
  purpose : OPTIONAL string_select;
 INVERSE
  collected_items : SET [2:?] OF collected_item_association FOR relating;
 END_ENTITY;

 ENTITY design_discipline_item_definition;
  name : OPTIONAL string_select;
  id : STRING;
  associated_item_version : item_version;
  additional_context : SET[0:?] OF application_context;
  initial_context : application_context;
 END_ENTITY;

ENTITY general_item_definition_instance_relationship
  SUBTYPE OF (item_definition_instance_relationship);
  description : OPTIONAL string_select;
  relation_type : STRING;
 END_ENTITY;

 ENTITY general_item_definition_relationship
  SUBTYPE OF (item_definition_relationship);
  relation_type : STRING;
  description : OPTIONAL string_select;
 END_ENTITY;
```

```
 ENTITY general_item_instance_relationship
  SUBTYPE OF (item_instance_relationship);
  relation_type : STRING;
  description : OPTIONAL string_select;
 END_ENTITY;

 ENTITY item_definition_instance_relationship
  ABSTRACT SUPERTYPE OF (
ONEOF(collected_item_association,assembly_component_relationship,general_item_defini
tion_instance_relationship));
  related : item_instance;
  relating : design_discipline_item_definition;
 END_ENTITY;

 ENTITY item_instance
  ABSTRACT SUPERTYPE OF
(ONEOF(single_instance,quantified_instance,selected_instance,specified_instance));
  description : OPTIONAL string_select;
  definition : instance_definition_select;
  id : STRING;
 END_ENTITY;

 ENTITY item_instance_relationship
  ABSTRACT SUPERTYPE OF (
ONEOF(replaced_usage_relationship,general_item_instance_relationship) );
  relating : item_instance;
  related : item_instance;
 END_ENTITY;

 ENTITY make_from_relationship
  SUBTYPE OF (item_definition_relationship);
  description : OPTIONAL string_select;
 END_ENTITY;

 ENTITY next_higher_assembly
  SUBTYPE OF (assembly_component_relationship);
 END_ENTITY;

 ENTITY physical_assembly_relationship;
  physical_component : physical_instance;
  physical_assembly : physical_instance;
  is_realization_of : item_instance;
 END_ENTITY;

 ENTITY quantified_instance
  SUBTYPE OF (item_instance);
  quantity : numerical_value;
 END_ENTITY;

 ENTITY replaced_definition_relationship
  SUBTYPE OF (item_definition_relationship);
```

```
 description : OPTIONAL string_select;
END_ENTITY;

ENTITY replaced_usage_relationship
 SUBTYPE OF (item_instance_relationship);
 usage_context : instance_usage_context_select;
 description : OPTIONAL string_select;
END_ENTITY;

ENTITY selected_instance
 SUBTYPE OF (item_instance);
 selection_control : STRING;
 selected_quantity : value_with_unit;
END_ENTITY;

ENTITY single_instance
 SUBTYPE OF (item_instance);
END_ENTITY;

ENTITY specified_instance
 SUBTYPE OF (item_instance);
 upper_usage : item_instance;
 related_instance : item_instance;
 assembly_context : assembly_definition;
END_ENTITY;

ENTITY tool_part_relationship
 SUBTYPE OF (item_definition_relationship);
 placement : OPTIONAL transformation_select;
 used_technology_description : OPTIONAL string_select;
END_ENTITY;

TYPE instance_definition_select = SELECT (
  design_discipline_item_definition,
  product_identification
 );
END_TYPE;

TYPE instance_usage_context_select = SELECT (
  product_structure_relationship,
  item_definition_instance_relationship,
  process_operation_input_or_output
 );
END_TYPE;

TYPE item_information_select = SELECT (
  design_discipline_item_definition,
  item_instance,
  physical_instance,
  product_component
 );
```

```
END_TYPE;

TYPE product_constituent_select = SELECT (
  item_instance,
  product_component,
  product_function
 );
END_TYPE;
```

A.1.3 Document and File Management

```
ENTITY digital_document
 SUBTYPE OF (document_representation);
 file : SET[0:?] OF digital_file;
END_ENTITY;

ENTITY digital_file
 SUBTYPE OF (document_file);
INVERSE
 associated_model_space : SET [0:1] OF external_model FOR is_defined_as;
END_ENTITY;

ENTITY document;
 description : OPTIONAL string_select;
 name : string_select;
 document_id : STRING;
 INVERSE
  associated_version : SET[1:?] OF document_version FOR associ-ated_document;
END_ENTITY;

ENTITY document_assignment;
 assigned_document : assigned_document_select;
 is_assigned_to : documented_element_select;
 role : STRING;
END_ENTITY;

ENTITY document_content_property;
 detail_level : OPTIONAL STRING;
 geometry_type : OPTIONAL STRING;
 real_world_scale : OPTIONAL numerical_value;
 languages : SET[0:?] OF language;
END_ENTITY;

ENTITY document_creation_property;
 creating_system : STRING;
 operating_system : OPTIONAL STRING;
 creating_interface : OPTIONAL STRING;
END_ENTITY;

ENTITY document_file
```

```
 ABSTRACT SUPERTYPE OF (ONEOF(digital_file,hardcopy));
 file_id : STRING;
 version_id : OPTIONAL STRING;
 document_file_type : OPTIONAL document_type_property;
 external_id_and_location : SET[0:?] OF external_file_id_and_location;
 size : OPTIONAL document_size_property;
 file_format : OPTIONAL document_format_property;
 content : OPTIONAL document_content_property;
 creation : OPTIONAL document_creation_property;
END_ENTITY;

ENTITY document_format_property;
 data_format : OPTIONAL STRING;
 character_code : OPTIONAL STRING;
 size_format : OPTIONAL rectangular_size;
END_ENTITY;

ENTITY document_location_property;
 location_name : STRING;
END_ENTITY;

ENTITY document_representation
 ABSTRACT SUPERTYPE OF (ONEOF(physical_representation,digital_document));
 description : OPTIONAL string_select;
 id : STRING;
 associated_document_version : document_version;
 creation : OPTIONAL document_creation_property;
 common_location : SET[0:?] OF document_location_property;
 representation_format : OPTIONAL document_format_property;
 size : OPTIONAL document_size_property;
 content : OPTIONAL document_content_property;
END_ENTITY;

ENTITY document_size_property;
 file_size : OPTIONAL value_with_unit;
 page_count : OPTIONAL value_with_unit;
END_ENTITY;

ENTITY document_structure;
 relating : document_representation;
 related : document_representation;
 description : OPTIONAL string_select;
 relation_type : STRING;
END_ENTITY;

ENTITY document_type_property;
 document_type_name : STRING;
 used_classification_system : OPTIONAL classification_system;
END_ENTITY;

ENTITY document_version;
```

```
 associated_document : document;
 id : STRING;
 description : OPTIONAL string_select;
END_ENTITY;

ENTITY document_version_relationship;
 description : OPTIONAL string_select;
 relating : document_version;
 related : document_version;
 relation_type : STRING;
END_ENTITY;

ENTITY external_file_id_and_location;
 location : document_location_property;
 external_id : OPTIONAL STRING;
END_ENTITY;

ENTITY hardcopy
 SUBTYPE OF (document_file);
END_ENTITY;

ENTITY physical_document
 SUBTYPE OF (physical_representation);
 component : SET [0:?] OF hardcopy;
END_ENTITY;

ENTITY physical_representation
 ABSTRACT SUPERTYPE OF (physical_document)
 SUBTYPE OF (document_representation);
END_ENTITY;

ENTITY named_size
 SUBTYPE OF (rectangular_size);
 referenced_standard : OPTIONAL classification_system;
 size : STRING;
END_ENTITY;

ENTITY rectangular_size;
 density : OPTIONAL value_with_unit;
 width : value_with_unit;
 height : value_with_unit;
END_ENTITY;

TYPE assigned_document_select = SELECT (
  document,
  document_version,
  document_file,
  document_representation
 );
END_TYPE;
```

```
TYPE documented_element_select = SELECT (
  material,
  activity,
  approval,
  change,
  item_instance,
  design_constraint,
  design_discipline_item_definition,
  descriptive_specification,
  general_classification,
  classification_attribute,
  classification_system,
  activity_method,
  item_shape,
  item_definition_instance_relationship,
  item_instance_relationship,
  item_definition_relationship,
  complex_product,
  physical_assembly_relationship,
  physical_instance,
  physical_instance_test_result,
  process_plan,
  process_operation_occurrence,
  product_identification,
  product_class,
  product_structure_relationship,
  project,
  property,
  class_structure_relationship,
  item,
  activity_element,
  item_version,
  person,
  documented_element_sub_select,
  organization
 );
END_TYPE;

TYPE documented_element_sub_select = SELECT (
  specification_category,
  work_request,
  work_order,
  shape_element,
  shape_element_relationship,
  specific_item_classification,
  specification
 );
END_TYPE;
```

A.1.4 Shape Definition and Transformation

```
 ENTITY accuracy;
  accuracy_value : REAL;
  accuracy_type : STRING;
  is_defined_for : SET [1:?] OF accuracy_select;
  description : OPTIONAL string_select;
 END_ENTITY;

 ENTITY axis2_placement_3d;
  location : cartesian_point;
  axis : OPTIONAL direction;
  ref_direction : OPTIONAL direction;
 END_ENTITY;

 ENTITY cartesian_coordinate_space
  ABSTRACT SUPERTYPE OF (
ONEOF(cartesian_coordinate_space_2d,cartesian_coordinate_space_3d) );
  unit_of_values : OPTIONAL SET[2:?] OF unit;
 END_ENTITY;

 ENTITY cartesian_coordinate_space_2d
  SUBTYPE OF (cartesian_coordinate_space);
 END_ENTITY;

 ENTITY cartesian_coordinate_space_3d
  SUBTYPE OF (cartesian_coordinate_space);
 END_ENTITY;

 ENTITY cartesian_point;
  coordinates : LIST[3:3] OF REAL;
 END_ENTITY;

 ENTITY direction;
  direction_ratios : LIST[3:3] OF REAL;
 END_ENTITY;

 ENTITY explicit_transformation_3d
  SUBTYPE OF (transformation_3d);
  local_origin : cartesian_point;
  axis1 : OPTIONAL direction;
  axis2 : OPTIONAL direction;
  axis3 : OPTIONAL direction;
 END_ENTITY;

 ENTITY external_geometric_model
  SUBTYPE OF (external_model);
  model_extent : OPTIONAL STRING;
 END_ENTITY;

 ENTITY external_model
```

```
 ABSTRACT SUPERTYPE OF ( ONEOF(external_picture,external_geometric_model) );
 is_defined_as : digital_file;
 is_defined_in : cartesian_coordinate_space;
 description : OPTIONAL string_select;
 model_id : STRING;
END_ENTITY;

ENTITY external_picture
 SUBTYPE OF (external_model);
 SELF\external_model.is_defined_in : cartesian_coordinate_space_2d;
END_ENTITY;

ENTITY geometric_model;
 is_defined_in : cartesian_coordinate_space;
 model_id : STRING;
 description : OPTIONAL string_select;
 model_extent : OPTIONAL NUMBER;
END_ENTITY;

ENTITY geometric_model_relationship;
 relating : geometric_or_external_model_select;
 related : geometric_or_external_model_select;
 description : OPTIONAL string_select;
 relation_type : STRING;
END_ENTITY;

ENTITY geometric_model_relationship_with_transformation
 SUBTYPE OF (geometric_model_relationship);
 model_placement : transformation;
END_ENTITY;

ENTITY geometrical_relationship
 SUBTYPE OF (item_definition_relationship);
 description : OPTIONAL string_select;
 definition_placement : transformation_select;
END_ENTITY;

ENTITY implicit_transformation_3d
 SUBTYPE OF (transformation_3d);
 transformation_origin : axis2_placement_3d;
 transformation_target : transformation_target_select;
END_ENTITY;

ENTITY item_shape;
 described_object : item_information_select;
 description : OPTIONAL string_select;
END_ENTITY;

ENTITY material;
 material_name : STRING;
 described_element : SET[1:?] OF item_property_select;
```

```
END_ENTITY;

ENTITY shape_description_association;
 is_defining_shape_for : shape_information_select;
 role : STRING;
 defining_geometry : shape_definition_select;
END_ENTITY;

ENTITY shape_element;
 description : OPTIONAL string_select;
 composition : item_shape;
 element_name : OPTIONAL STRING;
END_ENTITY;

ENTITY shape_element_relationship;
 relating : shape_element;
 related : shape_element;
 description : OPTIONAL string_select;
 relation_type : STRING;
END_ENTITY;

ENTITY transformation
 ABSTRACT SUPERTYPE OF (transformation_3d);
END_ENTITY;

ENTITY transformation_3d
 SUBTYPE OF (transformation);
END_ENTITY;

TYPE accuracy_select = SELECT (
  geometric_model,
  external_geometric_model
 );
END_TYPE;

TYPE geometric_or_external_model_select = SELECT (
  external_model,
  geometric_model
 );
END_TYPE;

TYPE shape_definition_select = SELECT (
  external_geometric_model,
  geometric_model
 );
END_TYPE;

TYPE shaped_element_select = SELECT (
  shape_element,
  item_shape
 );
```

```
 END_TYPE;

 TYPE shape_information_select = SELECT (
   shape_element_relationship,
   shaped_element_select
  );
 END_TYPE;

 TYPE transformation_select = SELECT (
   geometric_model_relationship_with_transformation
  );
 END_TYPE;

 TYPE transformation_target_select = SELECT (
   axis2_placement_3d,
   explicit_transformation_3d
  );
 END_TYPE;
```

A.1.5 Classification

```
 ENTITY classification_association;
  associated_classification : general_classification;
  role : OPTIONAL STRING;
  definitional : OPTIONAL BOOLEAN;
  classified_element : classified_element_select;
 END_ENTITY;

 ENTITY classification_attribute;
  id : STRING;
  name : OPTIONAL string_select;
  description : OPTIONAL string_select;
  allowed_value : SET [0:?] OF property_value_representation;
  attribute_definition : property;
  associated_classification : general_classification;
 END_ENTITY;

 ENTITY classification_system;
  description : OPTIONAL string_select;
  id : STRING;
 INVERSE
  allowed_classification : SET [0:?] OF general_classification FOR
used_classification_system;
 END_ENTITY;

 ENTITY external_library_reference;
  external_id : STRING;
  library_type : STRING;
  description : OPTIONAL string_select;
 END_ENTITY;
```

```
ENTITY general_classification;
 classification_source : OPTIONAL class_source_select;
 used_classification_system : OPTIONAL classification_system;
 description : OPTIONAL string_select;
 id : STRING;
 version_id : OPTIONAL STRING;
END_ENTITY;

ENTITY general_classification_hierarchy;
 super_classification : general_classification;
 sub_classification : general_classification;
END_ENTITY;

ENTITY specific_document_classification;
 associated_document : SET [1:?] OF document;
 description : OPTIONAL string_select;
 classification_name : STRING;
END_ENTITY;

ENTITY specific_document_classification_hierarchy;
 super_classification : specific_document_classification;
 sub_classification : specific_document_classification;
END_ENTITY;

ENTITY specific_item_classification;
 associated_item : SET[1:?] OF item;
 classification_name : STRING;
 description : OPTIONAL string_select;
END_ENTITY;

ENTITY specific_item_classification_hierarchy;
 super_classification : specific_item_classification;
 sub_classification : specific_item_classification;
END_ENTITY;

TYPE class_source_select = SELECT (
  external_library_reference
 );
END_TYPE;

TYPE classified_element_select = SELECT (
  design_constraint,
  item,
  approval_status,
  product_class,
  document,
  document_representation,
  project,
  activity_method,
  property,
```

```
   material,
   product_identification,
   complex_product,
   activity,
   item_version,
   property_value_association,
   item_instance,
   design_discipline_item_definition,
   document_version,
   shape_element,
   specification_category,
   work_order,
   work_request,
   process_plan,
   process_operation_definition,
   process_operation_occurrence,
   document_file
  );
END_TYPE;
```

A.1.6 Properties

```
ENTITY cost_property
 SUBTYPE OF (property);
END_ENTITY;

ENTITY data_environment;
 environment_name : STRING;
 description : OPTIONAL string_select;
END_ENTITY;

ENTITY duration_property
 SUBTYPE OF (property);
END_ENTITY;

ENTITY general_property
 SUBTYPE OF (property);
 property_type : STRING;
END_ENTITY;

ENTITY item_property_association
 SUBTYPE OF (property_value_association);
 definitional : OPTIONAL BOOLEAN;
 described_element : item_property_select;
END_ENTITY;

ENTITY mass_property
 SUBTYPE OF (property);
END_ENTITY;
```

```
 ENTITY material_property
  SUBTYPE OF (property);
  property_name : STRING;
 END_ENTITY;

 ENTITY material_property_association;
  described_material : material;
  associated_property_value : material_property_value_representation;
  definitional : OPTIONAL BOOLEAN;
 END_ENTITY;

 ENTITY material_property_value_representation
  SUBTYPE OF (property_value_representation);
  environment_condition : data_environment;
  SELF\property_value_representation.definition : material_property;
 END_ENTITY;

 ENTITY numerical_value
  SUBTYPE OF (value_with_unit);
  value_component : NUMBER;
 END_ENTITY;

 ENTITY property
  ABSTRACT SUPERTYPE OF
(ONEOF(cost_property,quality_property,duration_property,material_property,general_pr
operty,recyclability_property,mass_property));
  allowed_unit : SET[0:?] OF unit;
  property_source : OPTIONAL property_source_select;
  description : OPTIONAL string_select;
  id : STRING;
  version_id : OPTIONAL STRING;
 END_ENTITY;

 ENTITY property_value
  ABSTRACT SUPERTYPE OF (ONEOF(value_list,value_with_unit,string_value));
  value_name : STRING;
 END_ENTITY;

 ENTITY property_value_association
  ABSTRACT SUPERTYPE OF (
ONEOF(item_property_association,process_property_association) );
  validity_context : OPTIONAL validity_context_select;
  description : OPTIONAL string_select;
  describing_property_value : property_value_representation;
 END_ENTITY;

 ENTITY property_value_representation;
  specified_value : property_value;
  value_determination : OPTIONAL STRING;
  global_unit : OPTIONAL unit;
  qualifier : OPTIONAL STRING;
```

```
 definition : property;
END_ENTITY;

ENTITY quality_property
 SUBTYPE OF (property);
END_ENTITY;

ENTITY recyclability_property
 SUBTYPE OF (property);
END_ENTITY;

ENTITY simple_property_association
 described_element : simple_property_select;
 specified_value : string_value
 value_type : STRING;
END_ENTITY;

ENTITY simple_string_value
 SUBTYPE OF (simple_property_value);
 value_specification : string_select;
END_ENTITY;

ENTITY string_value
 SUBTYPE OF (property_value);
 value_specification : string_select;
END_ENTITY;

ENTITY unit;
 unit_name : STRING;
END_ENTITY;

ENTITY value_limit
 SUBTYPE OF (value_with_unit);
 limit_qualifier : STRING;
 limit : NUMBER;
END_ENTITY;

ENTITY value_list
 SUBTYPE OF (property_value);
 values : LIST[1:?] OF property_value;
END_ENTITY;

ENTITY value_range
 SUBTYPE OF (value_with_unit);
 upper_limit : NUMBER;
 lower_limit : NUMBER;
END_ENTITY;

ENTITY value_with_unit
 ABSTRACT SUPERTYPE OF (  ONEOF(numerical_value,value_range,value_limit) )
 SUBTYPE OF (property_value);
```

```
 unit_component : OPTIONAL unit;
 significant_digits : OPTIONAL INTEGER;
END_ENTITY;

TYPE item_property_select = SELECT (
  product_class,
  design_constraint,
  item_instance,
  design_discipline_item_definition,
  product_structure_relationship,
  item_definition_relationship,
  item_definition_instance_relationship,
  item_instance_relationship,
  item_shape,
  shape_element,
  shape_element_relationship,
  complex_product,
  document_file,
  document_representation,
  product_identification,
  physical_instance
 );
END_TYPE;

TYPE property_source_select = SELECT (
  external_library_reference
 );
END_TYPE;

TYPE simple_property_select = SELECT (
  item_property_select,
  process_property_select
 );
END_TYPE;

TYPE validity_context_select = SELECT (
  organization,
  product_identification,
  product_class
 );
END_TYPE;
```

A.1.7 Alias Identification

```
ENTITY alias_identification;
 alias_id : STRING;
 alias_version_id : OPTIONAL STRING;
 is_applied_to : alias_select;
 alias_scope : OPTIONAL organization;
 description : OPTIONAL string_select;
```

```
END_ENTITY;

TYPE alias_select = SELECT (
  organization,
  product_class,
  approval_status,
  item,
  document,
  document_version,
  specification,
  item_version,
  item_instance,
  specification_category,
  document_representation,
  document_type_property,
  physical_instance,
  geometric_model,
  general_classification,
  complex_product,
  classification_system,
  property,
  classification_attribute,
  design_discipline_item_definition
 );
END_TYPE;
```

A.1.8 Authorization

```
ENTITY address;
 internal_location : OPTIONAL STRING;
 street_number : OPTIONAL STRING;
 street : OPTIONAL STRING;
 postal_box : OPTIONAL STRING;
 town : OPTIONAL STRING;
 region : OPTIONAL STRING;
 postal_code : OPTIONAL STRING;
 country : OPTIONAL STRING;
 facsimile_number : OPTIONAL STRING;
 telephone_number : OPTIONAL STRING;
 electronic_mail_address : OPTIONAL STRING;
 telex_number : OPTIONAL STRING;
END_ENTITY;

ENTITY approval;
 status : approval_status;
 is_applied_to : SET[1:?] OF approval_element_select;
 is_approved_by : SET[0:?] OF date_and_person_organization;
 planned_date : OPTIONAL date_time;
 actual_date : OPTIONAL date_time;
 scope : SET[0:?] OF organization;
```

```
 level : OPTIONAL STRING;
END_ENTITY;

ENTITY approval_relationship;
 relating : approval;
 related : approval;
 relation_type : STRING;
 description : OPTIONAL string_select;
END_ENTITY;

ENTITY approval_status;
 status_name : STRING;
 used_classification_system : OPTIONAL classification_system;
END_ENTITY;

ENTITY date_and_person_assignment;
 is_applied_to : SET [1:?] OF date_time_person_organization_element_select;
 assigned_date_and_person : date_and_person_organization;
 role : STRING;
 description : OPTIONAL string_select;
END_ENTITY;

ENTITY date_and_person_organization;
 person_or_organization : person_organization_select;
 actual_date : date_time;
END_ENTITY;

ENTITY date_time;
 time : OPTIONAL STRING;
 date : STRING;
END_ENTITY;

ENTITY date_time_assignment;
 assigned_date_time : date_time;
 role : STRING;
 is_applied_to : SET[1:?] OF date_time_person_organization_element_select;
 description : OPTIONAL string_select;
END_ENTITY;

ENTITY duration;
 time : STRING;
 time_unit : STRING;
END_ENTITY;

ENTITY event_reference;
 offset : OPTIONAL duration;
 event_context : OPTIONAL general_organizational_data_select;
 event_type : STRING;
 description : OPTIONAL string_select;
END_ENTITY;
```

```
ENTITY organization;
 organization_name : STRING;
 visitor_address : OPTIONAL address;
 organization_type : OPTIONAL STRING;
 id : STRING;
 delivery_address : OPTIONAL address;
 postal_address : OPTIONAL address;
END_ENTITY;

ENTITY organization_relationship;
 description : STRING;
 related : organization;
 relating : organization;
 relation_type : STRING;
END_ENTITY;

ENTITY person;
 person_name : STRING;
 preferred_business_address : OPTIONAL address;
 INVERSE
  associated_organization : SET[1:?] OF person_in_organization FOR associ-
ated_person;
END_ENTITY;

ENTITY person_in_organization;
 associated_person : person;
 associated_organization : organization;
 role : STRING;
 location : OPTIONAL address;
 id : OPTIONAL STRING;
END_ENTITY;

ENTITY person_in_organization_relationship;
 description : STRING;
 related : person_in_organization;
 relating : person_in_organization;
 relation_type : STRING;
END_ENTITY;

ENTITY person_organization_assignment;
 is_applied_to : SET[1:?] OF date_time_person_organization_element_select;
 assigned_person_organization : person_organization_select;
 role : STRING;
 description : OPTIONAL string_select;
END_ENTITY;

TYPE approval_element_select = SELECT (
  document,
  document_version,
  document_representation,
  geometric_model,
```

```
  activity_method_assignment,
  design_constraint,
  specification_category,
  class_category_association,
  class_specification_association,
  class_condition_association,
  specification_expression,
  specification_inclusion,
  product_class,
  physical_instance_test_result,
  document_file,
  class_inclusion_association,
  specification,
  configuration,
  material,
  activity,
  activity_element,
  process_plan,
  work_order,
  project,
  work_request,
  physical_assembly_relationship,
  design_discipline_item_definition,
  physical_instance,
  product_structure_relationship,
  manufacturing_configuration,
  complex_product,
  property_value_association,
  item_version,
  property,
  class_structure_relationship,
  item_definition_instance_relationship,
  item_definition_relationship,
  item_instance,
  item_instance_relationship,
  general_classification,
  classification_association,
  classification_system
 );
END_TYPE;

TYPE date_time_person_organization_element_select = SELECT (
  event_reference,
  general_organizational_data_select
 );
END_TYPE;

TYPE event_or_date_select = SELECT (
  event_reference,
  date_time
 );
```

```
END_TYPE;

TYPE general_organizational_data_select = SELECT (
  product_identification,
  design_discipline_item_definition,
  class_category_association,
  class_specification_association,
  class_condition_association,
  class_inclusion_association,
  design_constraint,
  product_class,
  activity,
  activity_element,
  document,
  document_version,
  configuration,
  process_plan,
  classification_system,
  classification_association,
  document_representation,
  document_file,
  process_operation_occurrence,
  material,
  physical_instance,
  physical_assembly_relationship,
  physical_instance_test_result,
  product_structure_relationship,
  manufacturing_configuration,
  complex_product,
  activity_method_assignment,
  approval_status,
  class_structure_relationship,
  complex_product_relationship,
  general_classification,
  geometric_model,
  item,
  item_definition_instance_relationship,
  item_definition_relationship,
  item_instance,
  item_instance_relationship,
  item_version,
  item_version_relationship,
  process_operation_definition,
  process_operation_resource_assignment,
  person_in_organization,
  general_organizational_data_sub_select
 );
END_TYPE;

TYPE general_organizational_data_sub_select = SELECT (
  project,
```

```
  property,
  property_value_association,
  specification_category,
  specification,
  specification_expression,
  specification_inclusion,
  work_order,
  work_request
 );
END_TYPE;

TYPE period_or_date_select = SELECT (
  duration,
  event_reference,
  date_time
 );
END_TYPE;

TYPE person_organization_select = SELECT (
  person_in_organization,
  organization
 );
END_TYPE;
```

A.1.9 Configuration Management

```
ENTITY alternative_solution
 SUBTYPE OF (complex_product);
 base_element : complex_product_select;
END_ENTITY;

ENTITY class_category_association;
 associated_product_class : product_class;
 mandatory : BOOLEAN;
 associated_category : specification_category;
END_ENTITY;

ENTITY class_condition_association;
 condition_type : STRING;
 associated_product_class : product_class;
 description : OPTIONAL string_select;
 associated_condition : specification_expression;
END_ENTITY;

ENTITY class_inclusion_association;
 associated_product_class : product_class;
 description : OPTIONAL string_select;
 associated_inclusion : specification_inclusion;
END_ENTITY;
```

```
 ENTITY class_specification_association;
  associated_product_class : product_class;
  association_type : STRING;
  associated_specification : specification;
 END_ENTITY;

 ENTITY class_structure_relationship;
  related : product_function_component_select;
  relating : product_class;
  description : OPTIONAL string_select;
  relation_type : STRING;
 END_ENTITY;

 ENTITY complex_product
  ABSTRACT SUPERTYPE OF
(ONEOF(product_component,product_function,alternative_solution));
  id : STRING;
  version_id : OPTIONAL STRING;
 END_ENTITY;

 ENTITY complex_product_relationship;
  relating : complex_product;
  related : complex_product;
  description : OPTIONAL string_select;
  relation_type : STRING;
 END_ENTITY;

 ENTITY component_placement;
  placed_component : product_component;
  placement : transformation_select;
  reference_product_component : product_component;
 END_ENTITY;

 ENTITY configuration;
  configured_element : configured_item_select;
  is_solution_for : configured_specification_select;
  configuration_type : STRING;
  inheritance_type : STRING;
 END_ENTITY;

 ENTITY dated_configuration
  SUBTYPE OF (manufacturing_configuration);
  start_date : STRING;
  end_date : OPTIONAL STRING;
 END_ENTITY;

 ENTITY descriptive_specification;
  description : string_select;
  id : OPTIONAL STRING;
 END_ENTITY;
```

```
ENTITY design_constraint;
 constraint_id : STRING;
 name : OPTIONAL string_select;
 description : OPTIONAL string_select;
 is_valid_for : SET [0:?] OF product_class;
END_ENTITY;

ENTITY design_constraint_association;
 is_based_on : design_constraint;
 name : OPTIONAL string_select;
 is_constraining : complex_product;
END_ENTITY;

ENTITY design_constraint_relationship;
 related : design_constraint;
 relating : design_constraint;
 relation_type : STRING;
 description : OPTIONAL string_select;
END_ENTITY;

ENTITY design_constraint_version
 SUBTYPE OF (design_constraint);
 version_id : STRING;
END_ENTITY;

ENTITY effectivity;
 concerned_organization : SET[0:?] OF organization;
 description : OPTIONAL string_select;
 id : OPTIONAL STRING;
 version_id : OPTIONAL STRING;
 effectivity_context : OPTIONAL STRING;
 period : OPTIONAL duration;
 start_definition : OPTIONAL event_or_date_select;
 end_definition : OPTIONAL event_or_date_select;
END_ENTITY;

ENTITY effectivity_assignment;
 assigned_effectivity : effectivity;
 effective_element : effective_element_select;
 role : STRING;
 effectivity_indication : BOOLEAN;
END_ENTITY;

ENTITY final_solution
 SUBTYPE OF (alternative_solution);
 final_specification : SET [1:?] OF final_definition_select;
 final_status : STRING;
END_ENTITY;

ENTITY instance_placement;
 reference_product_component : product_component;
```

```
  placed_instance : single_instance;
  placement : transformation_select;
 END_ENTITY;

 ENTITY item_function_association;
  associated_function : product_function;
  associated_item : design_discipline_item_definition;
  description : OPTIONAL string_select;
  association_type : STRING;
 END_ENTITY;

 ENTITY lot_configuration
  SUBTYPE OF (manufacturing_configuration);
  lot_id : STRING;
  lot_size : STRING;
 END_ENTITY;

 ENTITY manufacturing_configuration
  ABSTRACT SUPERTYPE OF (
ONEOF(serial_configuration,dated_configuration,lot_configuration) );
  is_solution_for : product_design;
  configured_element : item_instance;
  concerned_organization : SET [0:?] OF organization;
 END_ENTITY;

 ENTITY physical_instance;
  is_realization_of : OPTIONAL physical_instance_definition_select;
  serial_number : OPTIONAL STRING;
  lot_id : OPTIONAL STRING;
  description : OPTIONAL string_select;
  inventory_number : OPTIONAL STRING;
 END_ENTITY;

 ENTITY physical_instance_test_result;
  test_activity : OPTIONAL test_activity_select;
  test_result : SET [0:?] OF property_value_representation;
  tested_instance : physical_instance;
  description : OPTIONAL string_select;
  id : STRING;
 END_ENTITY;

 ENTITY product_class;
  name : OPTIONAL string_select;
  id : STRING;
  description : OPTIONAL string_select;
  level_type : OPTIONAL STRING;
  version_id : OPTIONAL STRING;
 END_ENTITY;

 ENTITY product_component
  SUBTYPE OF (complex_product);
```

```
 is_influenced_by : SET[0:?] OF class_category_association;
 name : OPTIONAL string_select;
 description : OPTIONAL string_select;
 is_relevant_for : SET[0:?] OF application_context;
 instance_required : BOOLEAN;
END_ENTITY;

ENTITY product_design;
 design : item_version;
 product : product_identification;
END_ENTITY;

ENTITY product_function
 SUBTYPE OF (complex_product);
 name : OPTIONAL string_select;
 description : OPTIONAL string_select;
 is_relevant_for : SET[0:?] OF application_context;
END_ENTITY;

ENTITY product_identification;
 associated_product_class : product_class;
 name : OPTIONAL string_select;
 version_id : OPTIONAL STRING;
 id : STRING;
 description : OPTIONAL string_select;
 INVERSE
  associated_design : SET[0:1] OF product_design FOR product;
END_ENTITY;

ENTITY product_specification
 SUBTYPE OF (product_identification);
 defining_specification : SET [1:?] OF specification;
END_ENTITY;

ENTITY product_structure_relationship;
 relating : complex_product;
 related : product_constituent_select;
 relation_type : STRING;
 description : OPTIONAL string_select;
END_ENTITY;

ENTITY serial_configuration
 SUBTYPE OF (manufacturing_configuration);
 serial_start_number : STRING;
 serial_end_number : OPTIONAL STRING;
END_ENTITY;

ENTITY specification;
 id : STRING;
 name : OPTIONAL string_select;
 description : OPTIONAL string_select;
```

```
 category : specification_category;
 version_id : OPTIONAL STRING;
 package : BOOLEAN;
END_ENTITY;

ENTITY specification_category;
 implicit_exclusive_condition : BOOLEAN;
 id : STRING;
 description : string_select;
END_ENTITY;

ENTITY specification_category_hierarchy;
 sub_category : specification_category;
 super_category : specification_category;
END_ENTITY;

ENTITY specification_expression;
 description : OPTIONAL string_select;
 operation : STRING;
 operand : SET [1:?] OF specification_operand_select;
 id : OPTIONAL STRING;
END_ENTITY;

ENTITY specification_inclusion;
 if_condition : specification_operand_select;
 included_specification : specification_operand_select;
 description : OPTIONAL string_select;
 id : OPTIONAL STRING;
END_ENTITY;

ENTITY supplier_solution
 SUBTYPE OF (alternative_solution);
 supplier : organization;
 probability_rate : OPTIONAL STRING;
END_ENTITY;

ENTITY technical_solution
 SUBTYPE OF (alternative_solution);
 description : string_select;
END_ENTITY;

TYPE complex_product_select = SELECT (
  alternative_solution,
  product_component,
  product_function
 );
END_TYPE;

TYPE configured_item_select = SELECT (
  process_operation_occurrence,
  item_instance,
```

```
  complex_product_select,
  process_plan
 );
END_TYPE;

TYPE configured_specification_select = SELECT (
  class_specification_association,
  class_condition_association
 );
END_TYPE;

TYPE effective_element_select = SELECT (
  item,
  item_version,
  product_identification,
  item_instance,
  material,
  specification,
  specification_category,
  specification_inclusion,
  specification_expression,
  product_class,
  design_constraint,
  class_inclusion_association,
  class_category_association,
  class_specification_association,
  class_condition_association,
  geometric_model,
  document_file,
  document,
  classification_system,
  product_structure_relationship,
  document_version,
  configuration,
  item_definition_instance_relationship,
  item_definition_relationship,
  item_instance_relationship,
  complex_product,
  property_value_association,
  property,
  class_structure_relationship,
  complex_product_relationship,
  document_representation,
  process_operation_definition,
  process_operation_definition_relationship,
  process_plan,
  process_operation_occurrence,
  process_operation_resource_assignment,
  process_operation_occurrence_relationship
 );
END_TYPE;
```

```
TYPE final_definition_select = SELECT (
  physical_instance,
  design_discipline_item_definition,
  descriptive_specification
 );
END_TYPE;

TYPE physical_instance_definition_select = SELECT (
  product_identification,
  design_discipline_item_definition
 );
END_TYPE;

TYPE product_function_component_select = SELECT (
  product_component,
  product_function
 );
END_TYPE;

TYPE specification_operand_select = SELECT (
  specification_expression,
  specification
 );
END_TYPE;

TYPE test_activity_select = SELECT (
  activity,
  process_operation_occurrence
 );
END_TYPE;
```

A.1.10 Change and Work Management

```
ENTITY activity;
 activity_type : STRING;
 id : STRING;
 status : OPTIONAL STRING;
 description : OPTIONAL string_select;
 resolved_request : SET[0:?] OF work_request;
 concerned_organization : SET[0:?] OF organization;
 supplying_organization : SET[0:?] OF organization;
 requestor : OPTIONAL date_and_person_organization;
 actual_end_date : OPTIONAL date_time;
 planned_end_date : OPTIONAL period_or_date_select;
 planned_start_date : OPTIONAL event_or_date_select;
 actual_start_date : OPTIONAL date_time;
 internal : OPTIONAL BOOLEAN;
 chosen_method : OPTIONAL activity_method;
 INVERSE
```

```
  authorization : SET[0:1] OF work_order FOR is_controlling;
  associated_project : SET[0:1] OF project FOR work_program;
END_ENTITY;

ENTITY activity_element;
 element : activity_element_select;
 associated_activity : activity;
 role : STRING;
END_ENTITY;

ENTITY activity_method;
 description : string_select;
 consequence : OPTIONAL STRING;
 name : string_select;
END_ENTITY;

ENTITY activity_method_assignment;
 assigned_method : activity_method;
 associated_request : work_request;
 relation_type : STRING;
END_ENTITY;

ENTITY activity_relationship;
 related : activity;
 relating : activity;
 description : OPTIONAL string_select;
 relation_type : STRING;
END_ENTITY;

ENTITY element_delivery;
 destination : organization;
 quantity : value_with_unit;
 deliverable_element : activity_element;
END_ENTITY;

ENTITY project;
 id : STRING;
 name : string_select;
 description : OPTIONAL string_select;
 actual_start_date : OPTIONAL date_time;
 actual_end_date : OPTIONAL date_time;
 planned_start_date : OPTIONAL event_or_date_select;
 work_program : SET[0:?] OF activity;
 planned_end_date : OPTIONAL period_or_date_select;
END_ENTITY;

ENTITY project_assignment;
 assigned_project: project;
 role : STRING;
 is_applied_to : SET[1:?] OF project_information_select;
END_ENTITY;
```

```
ENTITY project_relationship;
 related : project;
 relating : project;
 relation_type : STRING;
 description : OPTIONAL string_select;
END_ENTITY;

ENTITY work_order;
 is_controlling : SET [1:?] OF activity;
 id : STRING;
 version_id : OPTIONAL STRING;
 description : OPTIONAL string_select;
 work_order_type : STRING;
END_ENTITY;

ENTITY work_request;
 id : STRING;
 request_type : STRING;
 status : STRING;
 notified_person : SET[1:?] OF date_and_person_organization;
 version_id : OPTIONAL STRING;
 requestor : date_and_person_organization;
 scope : SET[0:?] OF activity_element_select;
 description : OPTIONAL string_select;
END_ENTITY;

ENTITY work_request_relationship;
 related: work_request;
 relating: work_request;
 relation_type : STRING;
 description : OPTIONAL string_select;
END_ENTITY;

TYPE activity_element_select = SELECT (
  property,
  specification,
  specification_category,
  specification_expression,
  specification_inclusion,
  class_category_association,
  class_inclusion_association,
  class_specification_association,
  product_class,
  design_constraint,
  activity_method,
  configuration,
  item_instance,
  product_identification,
  document_representation,
  geometric_model,
```

```
  document_file,
  document,
  document_version,
  product_structure_relationship,
  item_definition_instance_relationship,
  item_definition_relationship,
  item_instance_relationship,
  complex_product,
  process_plan,
  property_value_association,
  process_operation_definition,
  design_discipline_item_definition,
  class_condition_association,
  class_structure_relationship,
  item,
  item_version,
  manufacturing_configuration,
  material,
  process_operation_occurrence,
  physical_instance,
  physical_assembly_relationship
 );
END_TYPE;

TYPE change_relationship_select = SELECT (
  item_version_relationship,
  process_plan_relationship,
  design_constraint_relationship,
  shape_element_relationship,
  replaced_definition_relationship,
  replaced_usage_relationship,
  complex_product_relationship,
  process_operation_occurrence_relationship
 );
END_TYPE;

TYPE project_information_select = SELECT (
  item,
  product_identification,
  document_version,
  product_class,
  document,
  physical_instance,
  complex_product,
  item_version
 );
END_TYPE;
```

A.1.11 Process Planning

```
ENTITY process_operation_definition;
 id : STRING;
 name : OPTIONAL string_select;
 process_type : STRING;
 description : OPTIONAL string_select;
 version_id : OPTIONAL STRING;
END_ENTITY;

ENTITY process_operation_definition_relationship;
 relating : process_operation_definition;
 related : process_operation_definition;
 relation_type : STRING;
END_ENTITY;

ENTITY process_operation_input_or_output;
 role : STRING;
 description : OPTIONAL string_select;
 operation : process_operation_occurrence;
 element : process_operation_input_or_output_select;
 placement : OPTIONAL transformation;
 concerned_shape : SET [0:?] OF shape_element;
END_ENTITY;

ENTITY process_operation_occurrence;
 plan : process_plan;
 is_defined_in : OPTIONAL cartesian_coordinate_space;
 operation_definition : process_operation_definition;
 id : STRING;
END_ENTITY;

ENTITY process_operation_occurrence_relationship;
 related : process_operation_occurrence;
 relation_type : STRING;
 relating : process_operation_occurrence;
 waiting_time : OPTIONAL property_value;
 description : OPTIONAL string_select;
 cycle_time : OPTIONAL duration;
END_ENTITY;

ENTITY process_operation_resource_assignment;
 reference_tool : BOOLEAN;
 operation : process_operation_occurrence;
 reason : OPTIONAL string_select;
 resource_definition : resource_definition_select;
 placement : OPTIONAL transformation;
END_ENTITY;

ENTITY process_plan;
 plan_id : STRING;
```

```
 name : OPTIONAL string_select;
 description : OPTIONAL string_select;
 produced_output : OPTIONAL SET[1:?] OF item_version;
END_ENTITY;

ENTITY process_plan_relationship;
 relating : process_plan;
 related : process_plan;
 description : OPTIONAL string_select;
 relation_type : STRING;
END_ENTITY;

ENTITY process_plan_version
 SUBTYPE OF (process_plan);
 version_id : STRING;
END_ENTITY;

ENTITY process_property_association
 SUBTYPE OF (property_value_association);
 described_element : process_property_select;
END_ENTITY;

ENTITY process_state
 SUBTYPE OF (design_discipline_item_definition);
 related_item_definition : design_discipline_item_definition;
END_ENTITY;

TYPE process_operation_input_or_output_select = SELECT (
  design_discipline_item_definition,
  assembly_component_relationship,
  item_instance
 );
END_TYPE;

TYPE process_property_select = SELECT (
  process_plan,
  process_operation_occurrence,
  process_operation_resource_assignment,
  activity,
  activity_method_assignment,
  process_operation_definition
 );
END_TYPE;

TYPE resource_definition_select = SELECT (
  descriptive_specification,
  design_discipline_item_definition,
  item_instance,
  physical_instance,
  product_component
 );
```

```
END_TYPE;
```

A.1.12 Multi-Language Support

```
ENTITY language;
 language_code : STRING;
 country_code : OPTIONAL STRING;
END_ENTITY;

ENTITY multi_language_string;
 additional_language_dependent_string : SET [0:?] OF string_with_language;
 primary_language_dependent_string : string_with_language;
END_ENTITY;

ENTITY string_with_language;
 contents : STRING;
 language_specification : language;
INVERSE
 used_by : SET [1:?] OF multi_language_string FOR pri-mary_language_dependent_string;
END_ENTITY;

TYPE default_language_string = STRING;
END_TYPE;

TYPE string_select = SELECT (
  multi_language_string,
  default_language_string
 );
END_TYPE;
```

A.2 PIM

The PIM for Product Lifecycle Management Services is defined in XMI and provided in an extra OMG document.

B Webservices PSM for Product LifeCycle Management Services

B.1 UML Profile for XML

B.1.1 Model PLM_services

```
Model "PLM_services"
Stereotype << XSDschema >>
targetNamespace         = http://www.omg.org/PLMServices1.0/XMLSchema
elementFormDefault      = qualified
attributeFormDefault    = unqualified
version                 = 1.0
modelGroup              = sequence
globalElement           = false
attributeMapping        = element
roleMapping             = element
anonymousRole           = false
anonymousType           = false
typeContainment         = false
elementNamingMapping    = firstLetterUpperCase
attributeNamingMapping  = firstLetterLowerCase
```

B.1.2 PLM Base

```
Class PLM_container
Class "PLM_container"
Stereotype << XSDcomplexType >>
modelGroup              = multiChoice
globalElement           = true

Attribute "uid"
Stereotype << XSDattribute >>
attributeType           = xs:ID
use                     = required

Attribute "version_id"
Stereotype << XSDattribute >>
attributeType           = xs:string
use                     = required

Composition "activity"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = true
anonymousType           = false
typeContainment         = true
```

```
Composition "classification_system"
Stereotype << XSDelement >>
position                = 11
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "classification_attribute"
Stereotype << XSDelement >>
position                = 10
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "complex_product"
Stereotype << XSDelement >>
position                = 12
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "address"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "application_context"
Stereotype << XSDelement >>
position                = 05
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "data_environment"
Stereotype << XSDelement >>
position                = 13
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "activity_method"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "approval_status"
```

```
Stereotype << XSDelement >>
position                = 06
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "axis2_placement_3d"
Stereotype << XSDelement >>
position                = 07
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "cartesian_coordinate_space"
Stereotype << XSDelement >>
position                = 08
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "cartesian_point"
Stereotype << XSDelement >>
position                = 09
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "accuracy"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "design_constraint"
Stereotype << XSDelement >>
position                = 16
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "direction"
Stereotype << XSDelement >>
position                = 17
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "date_time"
Stereotype << XSDelement >>
position                = 14
```

```
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "descriptive_specification"
Stereotype << XSDelement >>
position                = 15
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "document_content_property"
Stereotype << XSDelement >>
position                = 19
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "document"
Stereotype << XSDelement >>
position                = 18
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "document_file"
Stereotype << XSDelement >>
position                = 21
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "document_format_property"
Stereotype << XSDelement >>
position                = 22
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "document_location_property"
Stereotype << XSDelement >>
position                = 23
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "document_creation_property"
Stereotype << XSDelement >>
position                = 20
anonymousRole           = true
anonymousType           = false
```

```
typeContainment         = true

Composition "document_type_property"
Stereotype << XSDelement >>
position                = 25
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "duration"
Stereotype << XSDelement >>
position                = 26
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "document_size_property"
Stereotype << XSDelement >>
position                = 24
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "item"
Stereotype << XSDelement >>
position                = 32
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "item_shape"
Stereotype << XSDelement >>
position                = 33
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "language"
Stereotype << XSDelement >>
position                = 34
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "effectivity"
Stereotype << XSDelement >>
position                = 27
anonymousRole           = true
anonymousType           = false
typeContainment         = true
```

```
Composition "event_reference"
Stereotype << XSDelement >>
position                = 28
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "external_library_reference"
Stereotype << XSDelement >>
position                = 29
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "material"
Stereotype << XSDelement >>
position                = 35
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "organization"
Stereotype << XSDelement >>
position                = 36
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "person"
Stereotype << XSDelement >>
position                = 37
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "physical_instance"
Stereotype << XSDelement >>
position                = 38
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "general_classification"
Stereotype << XSDelement >>
position                = 30
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "geometric_model"
Stereotype << XSDelement >>
```

```
position                = 31
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "rectangular_size"
Stereotype << XSDelement >>
position                = 46
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "specific_document_classification"
Stereotype << XSDelement >>
position                = 47
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "specific_item_classification"
Stereotype << XSDelement >>
position                = 48
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "specification"
Stereotype << XSDelement >>
position                = 49
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "process_operation_definition"
Stereotype << XSDelement >>
position                = 39
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "process_operation_occurrence"
Stereotype << XSDelement >>
position                = 40
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "process_plan"
Stereotype << XSDelement >>
position                = 41
anonymousRole           = true
```

```
anonymousType           = false
typeContainment         = true

Composition "product_class"
Stereotype << XSDelement >>
position                = 42
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "project"
Stereotype << XSDelement >>
position                = 43
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "specification_expression"
Stereotype << XSDelement >>
position                = 51
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "unit"
Stereotype << XSDelement >>
position                = 54
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "work_request"
Stereotype << XSDelement >>
position                = 56
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "work_order"
Stereotype << XSDelement >>
position                = 55
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "property_value"
Stereotype << XSDelement >>
position                = 45
anonymousRole           = true
anonymousType           = false
typeContainment         = true
```

```
Composition "property"
Stereotype << XSDelement >>
position                = 44
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "specification_category"
Stereotype << XSDelement >>
position                = 50
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "transformation"
Stereotype << XSDelement >>
position                = 52
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Class "PLM_object"
Stereotype << XSDcomplexType >>

Attribute "uid"
Stereotype << XSDattribute >>
attributeType           = xs:ID
use                     = required

Class PLM_root_object
Class "PLM_root_object"
Stereotype << XSDcomplexType >>
```

B.1.3 Part Identification

```
Class Application_context
Class "Application_context"
Stereotype << XSDcomplexType >>

Attribute "application_domain"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "life_cycle_stage"
Stereotype << XSDelement >>
```

```
position                = 03
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Class Design_discipline_item_definition
Class "Design_discipline_item_definition"
Stereotype << XSDcomplexType >>

Attribute "id"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "item_instance"
Stereotype << XSDelement >>
position                = 05
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "item_definition_relationship"
Stereotype << XSDelement >>
position                = 09
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "name"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "document_assignment"
Stereotype << XSDelement >>
position                = 06
anonymousRole           = true
anonymousType           = false
typeContainment         = true
```

```
Composition "item_function_association"
Stereotype << XSDelement >>
position                = 08
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "alias_identification"
Stereotype << XSDelement >>
position                = 07
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "simple_property_value"
Stereotype << XSDelement >>
position                = 10
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "Item_definition_instance_relationship"
Stereotype << XSDelement >>
position                = 11
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Association "initial_context"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "additional_context"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Item
Class "Item"
Stereotype << XSDcomplexType >>

Attribute "id"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
```

```
typeContainment         = true

Composition "item_version"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "name"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "alias_identification"
Stereotype << XSDelement >>
position                = 06
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "document_assignment"
Stereotype << XSDelement >>
position                = 05
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Class Item_definition_relationship
Class "Item_definition_relationship"
Stereotype << XSDcomplexType >>

Composition "document_assignment"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "simple_property_value"
Stereotype << XSDelement >>
position                = 03
```

```
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Association "related"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Item_version
Class "Item_version"
Stereotype << XSDcomplexType >>

Attribute "id"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "item_version_relationship"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "product_design"
Stereotype << XSDelement >>
position                = 07
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "design_discipline_item_definition"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "alias_identification"
```

```
Stereotype << XSDelement >>
position                = 06
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "document_assignment"
Stereotype << XSDelement >>
position                = 05
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Class Item_version_relationship
Class "Item_version_relationship"
Stereotype << XSDcomplexType >>

Attribute "relation_type"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "change"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Association "related"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false
```

B.1.4 Part Structure

```
Class Assembly_component_relationship
```

```
Class "Assembly_component_relationship"
Stereotype << XSDcomplexType >>

Association "placement"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Assembly_definition
Class "Assembly_definition"
Stereotype << XSDcomplexType >>

Attribute "assembly_type"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Class Collected_item_association
Class "Collected_item_association"
Stereotype << XSDcomplexType >>

Class Collection_definition
Class "Collection_definition"
Stereotype << XSDcomplexType >>

Composition "purpose"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Class General_item_definition_instance_relationship
Class "General_item_definition_instance_relationship"
Stereotype << XSDcomplexType >>

Attribute "relation_type"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
```

```
anonymousType           = true
typeContainment         = true

Class General_item_definition_relationship
Class "General_item_definition_relationship"
Stereotype << XSDcomplexType >>

Attribute "relation_type"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Class General_item_instance_relationship
Class "General_item_instance_relationship"
Stereotype << XSDcomplexType >>

Attribute "relation_type"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Class Item_definition_instance_relationship
Class "Item_definition_instance_relationship"
Stereotype << XSDcomplexType >>

Composition "document_assignment"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "simple_property_value"
```

```
Stereotype << XSDelement >>
position                = 03
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Association "related"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Item_instance
Class "Item_instance"
Stereotype << XSDcomplexType >>

Attribute "id"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "item_instance_relationship"
Stereotype << XSDelement >>
position                = 05
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "manufacturing_configuration"
Stereotype << XSDelement >>
position                = 07
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "configuration"
Stereotype << XSDelement >>
position                = 06
anonymousRole           = true
anonymousType           = false
typeContainment         = true
```

```
Composition "alias_identification"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "document_assignment"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "simple_property_value"
Stereotype << XSDelement >>
position                = 08
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Class Item_instance_relationship
Class "Item_instance_relationship"
Stereotype << XSDcomplexType >>

Composition "document_assignment"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "simple_property_value"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Association "related"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Make_from_relationship
Class "Make_from_relationship"
Stereotype << XSDcomplexType >>
```

```
Composition "description"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Class Next_higher_assembly
Class "Next_higher_assembly"
Stereotype << XSDcomplexType >>

Class Physical_assembly_relationship
Class "Physical_assembly_relationship"
Stereotype << XSDcomplexType >>

Composition "document_assignment"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Association "physical_component"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "is_realization_of"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Quantified_instance
Class "Quantified_instance"
Stereotype << XSDcomplexType >>

Association "quantity"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Replaced_definition_relationship
Class "Replaced_definition_relationship"
Stereotype << XSDcomplexType >>
```

```
Composition "change"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Class Replaced_usage_relationship
Class "Replaced_usage_relationship"
Stereotype << XSDcomplexType >>

Composition "description"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Association "usage_context"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Selected_instance
Class "Selected_instance"
Stereotype << XSDcomplexType >>

Attribute "selection_control"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Association "selected_quantity"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Single_instance
```

```
Class "Single_instance"
Stereotype << XSDcomplexType >>

Composition "instance_placement"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Class Specified_instance
Class "Specified_instance"
Stereotype << XSDcomplexType >>

Association "assembly_context"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "related_instance"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "upper_usage"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Tool_part_relationship
Class "Tool_part_relationship"
Stereotype << XSDcomplexType >>

Composition "used_technology_description"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Association "placement"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
```

```
typeContainment         = false
```

B.1.5 Document and File Management

```
Class Digital_document
Class "Digital_document"
Stereotype << XSDcomplexType >>

Association "file"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Digital_file
Class "Digital_file"
Stereotype << XSDcomplexType >>

Composition "external_model"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Class Document
Class "Document"
Stereotype << XSDcomplexType >>

Attribute "document_id"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "document_version"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "name"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
```

```
anonymousType           = true
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "alias_identification"
Stereotype << XSDelement >>
position                = 05
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Class Document_assignment
Class "Document_assignment"
Stereotype << XSDcomplexType >>

Attribute "role"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Association "assigned_document"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Document_content_property
Class "Document_content_property"
Stereotype << XSDcomplexType >>

Attribute "detail_level"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "geometry_type"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
```

```
typeContainment         = true

Association "languages"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "real_world_scale"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Document_creation_property
Class "Document_creation_property"
Stereotype << XSDcomplexType >>

Attribute "creating_system"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "operating_system"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "creating_interface"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Class Document_file
Class "Document_file"
Stereotype << XSDcomplexType >>

Attribute "file_id"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true
```

```
Attribute "version_id"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "simple_property_value"
Stereotype << XSDelement >>
position                = 09
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Association "creation"
Stereotype << XSDelement >>
position                = 08
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "content"
Stereotype << XSDelement >>
position                = 07
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "file_format"
Stereotype << XSDelement >>
position                = 06
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "size"
Stereotype << XSDelement >>
position                = 05
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "external_id_and_location"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "document_file_type"
```

```
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Document_format_property
Class "Document_format_property"
Stereotype << XSDcomplexType >>

Attribute "data_format"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "character_code"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Association "size_format"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Document_location_property
Class "Document_location_property"
Stereotype << XSDcomplexType >>

Attribute "location_name"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "external_file_id_and_location"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Class Document_representation
Class "Document_representation"
```

```
Stereotype << XSDcomplexType >>

Attribute "id"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "document_structure"
Stereotype << XSDelement >>
position                = 08
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "alias_identification"
Stereotype << XSDelement >>
position                = 09
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "simple_property_value"
Stereotype << XSDelement >>
position                = 10
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Association "content"
Stereotype << XSDelement >>
position                = 07
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "size"
Stereotype << XSDelement >>
position                = 06
anonymousRole           = false
anonymousType           = false
typeContainment         = false
```

```
Association "representation_format"
Stereotype << XSDelement >>
position                = 05
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "common_location"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "creation"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Document_size_property
Class "Document_size_property"
Stereotype << XSDcomplexType >>

Association "page_count"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "file_size"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Document_structure
Class "Document_structure"
Stereotype << XSDcomplexType >>

Attribute "relation_type"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "description"
```

```
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Association "related"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Document_type_property
Class "Document_type_property"
Stereotype << XSDcomplexType >>

Attribute "document_type_name"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "alias_identification"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Association "used_classification_system"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Document_version
Class "Document_version"
Stereotype << XSDcomplexType >>

Attribute "id"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "document_version_relationship"
Stereotype << XSDelement >>
```

```
position                = 05
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "document_representation"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "alias_identification"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Class Document_version_relationship
Class "Document_version_relationship"
Stereotype << XSDcomplexType >>

Attribute "relation_type"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Association "related"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = false
typeContainment         = false
```

```
Class External_file_id_and_location
Class "External_file_id_and_location"
Stereotype << XSDcomplexType >>

Attribute "external_id"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Class Hardcopy
Class "Hardcopy"
Stereotype << XSDcomplexType >>

Class Named_size
Class "Named_size"
Stereotype << XSDcomplexType >>

Attribute "size"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Association "referenced_standard"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Physical_document
Class "Physical_document"
Stereotype << XSDcomplexType >>

Association "component"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Physical_representation
Class "Physical_representation"
Stereotype << XSDcomplexType >>

Class Rectangular_size
Class "Rectangular_size"
Stereotype << XSDcomplexType >>
```

```
Association "density"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "height"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "width"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = false
typeContainment         = false
```

B.1.6 Shape Definition and Transformation

```
Class Accuracy
Class "Accuracy"
Stereotype << XSDcomplexType >>

Attribute "accuracy_value"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "accuracy_type"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Association "is_defined_for"
```

```
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Axis2_placement_3d
Class "Axis2_placement_3d"
Stereotype << XSDcomplexType >>

Association "ref_direction"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "axis"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "location"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Cartesian_coordinate_space
Class "Cartesian_coordinate_space"
Stereotype << XSDcomplexType >>

Association "unit_of_values"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Cartesian_coordinate_space_2d
Class "Cartesian_coordinate_space_2d"
Stereotype << XSDcomplexType >>

Class Cartesian_coordinate_space_3d
Class "Cartesian_coordinate_space_3d"
Stereotype << XSDcomplexType >>

Class Cartesian_point
```

```
Class "Cartesian_point"
Stereotype << XSDcomplexType >>

Attribute "coordinates"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Class Direction
Class "Direction"
Stereotype << XSDcomplexType >>

Attribute "direction_ratios"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Class Explicit_transformation_3d
Class "Explicit_transformation_3d"
Stereotype << XSDcomplexType >>

Association "axis3"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "axis2"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "axis1"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "local_origin"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
```

```
typeContainment         = false

Class External_geometric_model
Class "External_geometric_model"
Stereotype << XSDcomplexType >>

Attribute "model_extent"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Class External_model
Class "External_model"
Stereotype << XSDcomplexType >>

Attribute "model_id"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "geometric_model_relationship"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Association "is_defined_in"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class External_picture
Class "External_picture"
Stereotype << XSDcomplexType >>

Class Geometric_model
Class "Geometric_model"
```

```
Stereotype << XSDcomplexType >>

Attribute "model_id"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "model_extent"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "geometric_model_relationship"
Stereotype << XSDelement >>
position                = 05
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Association "is_defined_in"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Geometric_model_relationship
Class "Geometric_model_relationship"
Stereotype << XSDcomplexType >>

Attribute "relation_type"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 02
```

```
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Association "related"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Geometric_model_relationship_with_transformation
Class "Geometric_model_relationship_with_transformation"
Stereotype << XSDcomplexType >>

Association "model_placement"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Geometrical_relationship
Class "Geometrical_relationship"
Stereotype << XSDcomplexType >>

Composition "description"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Association "definition_placement"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Implicit_transformation_3d
Class "Implicit_transformation_3d"
Stereotype << XSDcomplexType >>

Association "transformation_origin"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false
```

```
Association "transformation_target"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Item_shape
Class "Item_shape"
Stereotype << XSDcomplexType >>

Composition "shape_element"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "shape_description_association"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "document_assignment"
Stereotype << XSDelement >>
position                = 05
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "simple_property_value"
Stereotype << XSDelement >>
position                = 06
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Association "described_object"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
```

```
typeContainment         = false

Class Material
Class "Material"
Stereotype << XSDcomplexType >>

Attribute "material_name"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "document_assignment"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "material_property_association"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Association "described_element"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Shape_description_association
Class "Shape_description_association"
Stereotype << XSDcomplexType >>

Attribute "role"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Association "defining_geometry"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = false
typeContainment         = false
```

```
Class Shape_element
Class "Shape_element"
Stereotype << XSDcomplexType >>

Attribute "element_name"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "change"
Stereotype << XSDelement >>
position                = 06
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "document_assignment"
Stereotype << XSDelement >>
position                = 05
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "shape_description_association"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "shape_element_relationship"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "simple_property_value"
Stereotype << XSDelement >>
position                = 07
anonymousRole           = true
```

```
anonymousType           = false
typeContainment         = true

Class Shape_element_relationship
Class "Shape_element_relationship"
Stereotype << XSDcomplexType >>

Attribute "relation_type"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "document_assignment"
Stereotype << XSDelement >>
position                = 05
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "shape_description_association"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "simple_property_value"
Stereotype << XSDelement >>
position                = 06
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Association "related"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Transformation
Class "Transformation"
```

```
Stereotype << XSDcomplexType >>

Class Transformation_3d
Class "Transformation_3d"
Stereotype << XSDcomplexType >>
```

B.1.7 Classification

```
Class Classification_association
Class "Classification_association"
Stereotype << XSDcomplexType >>

Attribute "role"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "definitional"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Association "classified_element"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Classification_attribute
Class "Classification_attribute"
Stereotype << XSDcomplexType >>

Attribute "id"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "alias_identification"
Stereotype << XSDelement >>
position                = 07
anonymousRole           = true
anonymousType           = false
typeContainment         = true
```

```
Composition "document_assignment"
Stereotype << XSDelement >>
position                = 08
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "name"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Association "attribute_definition"
Stereotype << XSDelement >>
position                = 05
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "allowed_value"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "associated_classification"
Stereotype << XSDelement >>
position                = 06
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Classification_system
Class "Classification_system"
Stereotype << XSDcomplexType >>

Attribute "id"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
```

```
anonymousType           = true
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "document_assignment"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "alias_identification"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Class External_library_reference
Class "External_library_reference"
Stereotype << XSDcomplexType >>

Attribute "external_id"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "library_type"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = true
typeContainment         = true


Class General_classification
```

```
Class "General_classification"
Stereotype << XSDcomplexType >>

Attribute "id"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "version_id"
Stereotype << XSDelement >>
position                = 05
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "general_classification_hierarchy"
Stereotype << XSDelement >>
position                = 08
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "classification_association"
Stereotype << XSDelement >>
position                = 09
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "alias_identification"
Stereotype << XSDelement >>
position                = 07
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "document_assignment"
Stereotype << XSDelement >>
position                = 06
anonymousRole           = true
anonymousType           = false
typeContainment         = true
```

```
Association "classification_source"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "used_classification_system"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class General_classification_hierarchy
Class "General_classification_hierarchy"
Stereotype << XSDcomplexType >>

Association "sub_classification"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Specific_document_classification
Class "Specific_document_classification"
Stereotype << XSDcomplexType >>

Attribute "classification_name"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "specific_document_classification_hierarchy"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true
```

```
Association "associated_document"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Specific_document_classification_hierarchy
Class "Specific_document_classification_hierarchy"
Stereotype << XSDcomplexType >>

Association "sub_classification"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Specific_item_classification
Class "Specific_item_classification"
Stereotype << XSDcomplexType >>

Attribute "classification_name"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "specific_item_classification_hierarchy"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "document_assignment"
Stereotype << XSDelement >>
position                = 05
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Association "associated_item"
```

```
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Specific_item_classification_hierarchy
Class "Specific_item_classification_hierarchy"
Stereotype << XSDcomplexType >>

Association "sub_classification"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false
```

B.1.8 Properties

```
Class Cost_property
Class "Cost_property"
Stereotype << XSDcomplexType >>

Class Data_environment
Class "Data_environment"
Stereotype << XSDcomplexType >>

Attribute "environment_name"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Class Duration_property
Class "Duration_property"
Stereotype << XSDcomplexType >>

Class General_property
Class "General_property"
Stereotype << XSDcomplexType >>

Attribute "property_type"
```

```
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Class Item_property_association
Class "Item_property_association"
Stereotype << XSDcomplexType >>

Attribute "definitional"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Association "described_element"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Mass_property
Class "Mass_property"
Stereotype << XSDcomplexType >>

Class Material_property
Class "Material_property"
Stereotype << XSDcomplexType >>

Attribute "property_name"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Class Material_property_association
Class "Material_property_association"
Stereotype << XSDcomplexType >>

Attribute "definitional"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Association "associated_property_value"
```

```
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Material_property_value_representation
Class "Material_property_value_representation"
Stereotype << XSDcomplexType >>

Association "environment_condition"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Numerical_value
Class "Numerical_value"
Stereotype << XSDcomplexType >>

Attribute "value_component"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Class Property
Class "Property"
Stereotype << XSDcomplexType >>

Attribute "id"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "version_id"
Stereotype << XSDelement >>
position                = 05
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = true
```

```
typeContainment         = true

Composition "alias_identification"
Stereotype << XSDelement >>
position                = 06
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "document_assignment"
Stereotype << XSDelement >>
position                = 07
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Association "property_source"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "allowed_unit"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Property_value
Class "Property_value"
Stereotype << XSDcomplexType >>

Attribute "value_name"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "property_value_representation"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Class Property_value_association
Class "Property_value_association"
Stereotype << XSDcomplexType >>
```

```
Composition "description"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Association "validity_context"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Property_value_representation
Class "Property_value_representation"
Stereotype << XSDcomplexType >>

Attribute "value_determination"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "qualifier"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "property_value_association"
Stereotype << XSDelement >>
position                = 05
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Association "definition"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "global_unit"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
```

```
anonymousType           = false
typeContainment         = false

Class Quality_property
Class "Quality_property"
Stereotype << XSDcomplexType >>

Class Recyclability_property
Class "Recyclability_property"
Stereotype << XSDcomplexType >>

Class Simple_property_value
Class "Simple_property_value"
Stereotype << XSDcomplexType >>

Attribute "value_name"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "value_type"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Class Simple_string_value
Class "Simple_string_value"
Stereotype << XSDcomplexType >>

Composition "value_specification"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Class String_value
Class "String_value"
Stereotype << XSDcomplexType >>

Composition "value_specification"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true
```

```
Class Unit
Class "Unit"
Stereotype << XSDcomplexType >>

Attribute "unit_name"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Class Value_limit
Class "Value_limit"
Stereotype << XSDcomplexType >>

Attribute "limit_qualifier"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "limit"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Class Value_list
Class "Value_list"
Stereotype << XSDcomplexType >>

Association "values"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Value_range
Class "Value_range"
Stereotype << XSDcomplexType >>

Attribute "upper_limit"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true
```

```
Attribute "lower_limit"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Class Value_with_unit
Class "Value_with_unit"
Stereotype << XSDcomplexType >>

Attribute "significant_digits"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Association "unit_component"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false
```

B.1.9 Alias Identification

```
Class Alias_identification
Class "Alias_identification"
Stereotype << XSDcomplexType >>

Attribute "alias_id"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "alias_version_id"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = false
anonymousType           = true
```

```
typeContainment         = true

Association "alias_scope"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = false
typeContainment         = false
```

B.1.10 Authorization

```
Class Address
Class "Address"
Stereotype << XSDcomplexType >>

Attribute "internal_location"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "street_number"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "street"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "postal_box"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "town"
Stereotype << XSDelement >>
position                = 05
anonymousRole           = false
anonymousType           = true
typeContainment         = true
```

```
Attribute "region"
Stereotype << XSDelement >>
position                = 06
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "postal_code"
Stereotype << XSDelement >>
position                = 07
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "country"
Stereotype << XSDelement >>
position                = 08
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "facsimile_number"
Stereotype << XSDelement >>
position                = 09
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "telephone_number"
Stereotype << XSDelement >>
position                = 10
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "electronic_mail_address"
Stereotype << XSDelement >>
position                = 11
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "telex_number"
Stereotype << XSDelement >>
position                = 12
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Class Approval
Class "Approval"
```

```
Stereotype << XSDcomplexType >>

Attribute "level"
Stereotype << XSDelement >>
position                = 06
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "approval_relationship"
Stereotype << XSDelement >>
position                = 07
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "document_assignment"
Stereotype << XSDelement >>
position                = 08
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Association "scope"
Stereotype << XSDelement >>
position                = 05
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "actual_date"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "planned_date"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "is_approved_by"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = false
typeContainment         = false
```

```
Association "is_applied_to"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Approval_relationship
Class "Approval_relationship"
Stereotype << XSDcomplexType >>

Attribute "relation_type"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Association "related"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Approval_status
Class "Approval_status"
Stereotype << XSDcomplexType >>

Attribute "status_name"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "approval"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "alias_identification"
```

```
Stereotype << XSDelement >>
position                = 04
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Association "used_classification_system"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Date_and_person_assignment
Class "Date_and_person_assignment"
Stereotype << XSDcomplexType >>

Attribute "role"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Association "is_applied_to"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Date_and_person_organization
Class "Date_and_person_organization"
Stereotype << XSDcomplexType >>

Composition "date_and_person_assignment"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Association "actual_date"
Stereotype << XSDelement >>
```

```
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Date_time
Class "Date_time"
Stereotype << XSDcomplexType >>

Attribute "time"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "date"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "date_time_assignment"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Class Date_time_assignment
Class "Date_time_assignment"
Stereotype << XSDcomplexType >>

Attribute "role"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Association "is_applied_to"
Stereotype << XSDelement >>
position                = 02
```

```
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Duration
Class "Duration"
Stereotype << XSDcomplexType >>

Attribute "time"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "time_unit"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Class Event_reference
Class "Event_reference"
Stereotype << XSDcomplexType >>

Attribute "event_type"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Association "event_context"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "offset"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
```

```
anonymousType           = false
typeContainment         = false

Class Organization
Class "Organization"
Stereotype << XSDcomplexType >>

Attribute "organization_name"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "organization_type"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "id"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "document_assignment"
Stereotype << XSDelement >>
position                = 10
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "person_organization_assignment"
Stereotype << XSDelement >>
position                = 09
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "date_and_person_organization"
Stereotype << XSDelement >>
position                = 08
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "alias_identification"
Stereotype << XSDelement >>
```

```
position                = 07
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Association "postal_address"
Stereotype << XSDelement >>
position                = 06
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "delivery_address"
Stereotype << XSDelement >>
position                = 05
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "visitor_address"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Person
Class "Person"
Stereotype << XSDcomplexType >>

Attribute "person_name"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "person_in_organization"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "document_assignment"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = true
anonymousType           = false
typeContainment         = true
```

```
Association "preferred_business_address"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Person_in_organization
Class "Person_in_organization"
Stereotype << XSDcomplexType >>

Attribute "role"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "id"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "person_organization_assignment"
Stereotype << XSDelement >>
position                = 06
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "date_and_person_organization"
Stereotype << XSDelement >>
position                = 05
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Association "location"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "associated_organization"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
```

```
typeContainment         = false

Class Person_organization_assignment
Class "Person_organization_assignment"
Stereotype << XSDcomplexType >>

Attribute "role"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Association "is_applied_to"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false
```

B.1.11 Configuration Management

```
Class Alternative_solution
Class "Alternative_solution"
Stereotype << XSDcomplexType >>

Composition "configuration"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Association "base_element"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Class_category_association
Class "Class_category_association"
Stereotype << XSDcomplexType >>
```

Attribute "mandatory"
Stereotype << XSDelement >>
position = 01
anonymousRole = false
anonymousType = true
typeContainment = true

Association "associated_category"
Stereotype << XSDelement >>
position = 02
anonymousRole = false
anonymousType = false
typeContainment = false

Class Class_condition_association
Class "Class_condition_association"
Stereotype << XSDcomplexType >>

Attribute "condition_type"
Stereotype << XSDelement >>
position = 01
anonymousRole = false
anonymousType = true
typeContainment = true

Composition "description"
Stereotype << XSDelement >>
position = 02
anonymousRole = false
anonymousType = true
typeContainment = true

Association "associated_condition"
Stereotype << XSDelement >>
position = 03
anonymousRole = false
anonymousType = false
typeContainment = false

Class Class_inclusion_association
Class "Class_inclusion_association"
Stereotype << XSDcomplexType >>

Composition "description"
Stereotype << XSDelement >>
position = 01
anonymousRole = false
anonymousType = true
typeContainment = true

```
Association "associated_inclusion"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Class_specification_association
Class "Class_specification_association"
Stereotype << XSDcomplexType >>

Attribute "association_type"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Association "associated_specification"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Class_structure_relationship
Class "Class_structure_relationship"
Stereotype << XSDcomplexType >>

Attribute "relation_type"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Association "related"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Complex_product
```

```
Class "Complex_product"
Stereotype << XSDcomplexType >>

Attribute "id"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "version_id"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "product_structure_relationship"
Stereotype << XSDelement >>
position                = 05
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "design_constraint_association"
Stereotype << XSDelement >>
position                = 07
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "complex_product_relationship"
Stereotype << XSDelement >>
position                = 06
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "alias_identification"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "document_assignment"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = true
anonymousType           = false
typeContainment         = true
```

```
Composition "simple_property_value"
Stereotype << XSDelement >>
position                = 08
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Class Complex_product_relationship
Class "Complex_product_relationship"
Stereotype << XSDcomplexType >>

Attribute "relation_type"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Association "related"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Component_placement
Class "Component_placement"
Stereotype << XSDcomplexType >>

Association "reference_product_component"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "placement"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false
```

```
Class Configuration
Class "Configuration"
Stereotype << XSDcomplexType >>

Attribute "configuration_type"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "inheritance_type"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Association "is_solution_for"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Dated_configuration
Class "Dated_configuration"
Stereotype << XSDcomplexType >>

Attribute "start_date"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "end_date"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Class Descriptive_specification
Class "Descriptive_specification"
Stereotype << XSDcomplexType >>

Attribute "id"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
```

```
anonymousType           = true
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "document_assignment"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Class Design_constraint
Class "Design_constraint"
Stereotype << XSDcomplexType >>

Attribute "constraint_id"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "design_constraint_relationship"
Stereotype << XSDelement >>
position                = 05
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "name"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "document_assignment"
Stereotype << XSDelement >>
```

```
position                = 06
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "simple_property_value"
Stereotype << XSDelement >>
position                = 07
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Association "is_valid_for"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Design_constraint_association
Class "Design_constraint_association"
Stereotype << XSDcomplexType >>

Composition "name"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Association "is_based_on"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Design_constraint_relationship
Class "Design_constraint_relationship"
Stereotype << XSDcomplexType >>

Attribute "relation_type"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 03
```

```
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Association "related"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Design_constraint_version
Class "Design_constraint_version"
Stereotype << XSDcomplexType >>

Attribute "version_id"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Class Effectivity
Class "Effectivity"
Stereotype << XSDcomplexType >>

Attribute "id"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "version_id"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "effectivity_context"
Stereotype << XSDelement >>
position                = 05
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "effectivity_assignment"
Stereotype << XSDelement >>
position                = 09
anonymousRole           = true
```

```
anonymousType           = false
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Association "end_definition"
Stereotype << XSDelement >>
position                = 08
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "start_definition"
Stereotype << XSDelement >>
position                = 07
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "period"
Stereotype << XSDelement >>
position                = 06
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "concerned_organization"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Effectivity_assignment
Class "Effectivity_assignment"
Stereotype << XSDcomplexType >>

Attribute "role"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "effectivity_indication"
Stereotype << XSDelement >>
```

```
position                = 03
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Association "effective_element"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Final_solution
Class "Final_solution"
Stereotype << XSDcomplexType >>

Attribute "final_status"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Association "final_specification"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Instance_placement
Class "Instance_placement"
Stereotype << XSDcomplexType >>

Association "reference_product_component"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "placement"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Item_function_association
Class "Item_function_association"
Stereotype << XSDcomplexType >>
```

```
Attribute "association_type"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Association "associated_function"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Lot_configuration
Class "Lot_configuration"
Stereotype << XSDcomplexType >>

Attribute "lot_id"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "lot_size"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Class Manufacturing_configuration
Class "Manufacturing_configuration"
Stereotype << XSDcomplexType >>

Association "concerned_organization"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = false
typeContainment         = false
```

```
Association "is_solution_for"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Physical_instance
Class "Physical_instance"
Stereotype << XSDcomplexType >>

Attribute "serial_number"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "lot_id"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "inventory_number"
Stereotype << XSDelement >>
position                = 05
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "physical_instance_test_result"
Stereotype << XSDelement >>
position                = 09
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "physical_assembly_relationship"
Stereotype << XSDelement >>
position                = 06
anonymousRole           = true
anonymousType           = false
```

```
typeContainment         = true

Composition "document_assignment"
Stereotype << XSDelement >>
position                = 07
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "alias_identification"
Stereotype << XSDelement >>
position                = 08
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "simple_property_value"
Stereotype << XSDelement >>
position                = 10
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Association "is_realization_of"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Physical_instance_test_result
Class "Physical_instance_test_result"
Stereotype << XSDcomplexType >>

Attribute "id"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "Document_assignment"
Stereotype << XSDelement >>
position                = 05
```

```
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Association "test_result"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "test_activity"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Product_class
Class "Product_class"
Stereotype << XSDcomplexType >>

Attribute "id"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "level_type"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "version_id"
Stereotype << XSDelement >>
position                = 05
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "product_identification"
Stereotype << XSDelement >>
position                = 13
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "description"
```

```
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "name"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "class_structure_relationship"
Stereotype << XSDelement >>
position                = 12
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "class_specification_association"
Stereotype << XSDelement >>
position                = 11
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "class_inclusion_association"
Stereotype << XSDelement >>
position                = 10
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "class_condition_association"
Stereotype << XSDelement >>
position                = 09
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "class_category_association"
Stereotype << XSDelement >>
position                = 06
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "document_assignment"
Stereotype << XSDelement >>
position                = 07
```

```
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "alias_identification"
Stereotype << XSDelement >>
position                = 08
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "simple_property_value"
Stereotype << XSDelement >>
position                = 14
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Class Product_component
Class "Product_component"
Stereotype << XSDcomplexType >>

Attribute "instance_required"
Stereotype << XSDelement >>
position                = 05
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "name"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "configuration"
Stereotype << XSDelement >>
position                = 07
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "component_placement"
```

```
Stereotype << XSDelement >>
position                = 06
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Association "is_relevant_for"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "is_influenced_by"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Product_design
Class "Product_design"
Stereotype << XSDcomplexType >>

Association "product"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Product_function
Class "Product_function"
Stereotype << XSDcomplexType >>

Composition "description"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "name"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "configuration"
Stereotype << XSDelement >>
```

```
position                = 04
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Association "is_relevant_for"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Product_identification
Class "Product_identification"
Stereotype << XSDcomplexType >>

Attribute "version_id"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "id"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "name"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "item_instance"
Stereotype << XSDelement >>
position                = 06
anonymousRole           = true
anonymousType           = false
typeContainment         = true
```

```
Composition "document_assignment"
Stereotype << XSDelement >>
position                = 05
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "simple_property_value"
Stereotype << XSDelement >>
position                = 07
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Class Product_specification
Class "Product_specification"
Stereotype << XSDcomplexType >>

Association "defining_specification"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Product_structure_relationship
Class "Product_structure_relationship"
Stereotype << XSDcomplexType >>

Attribute "relation_type"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "document_assignment"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "simple_property_value"
```

```
Stereotype << XSDelement >>
position                = 05
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Association "related"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Serial_configuration
Class "Serial_configuration"
Stereotype << XSDcomplexType >>

Attribute "serial_start_number"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "serial_end_number"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Class Specification
Class "Specification"
Stereotype << XSDcomplexType >>

Attribute "id"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "version_id"
Stereotype << XSDelement >>
position                = 05
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "package"
Stereotype << XSDelement >>
```

```
position                = 06
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "specification_inclusion"
Stereotype << XSDelement >>
position                = 09
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "name"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "alias_identification"
Stereotype << XSDelement >>
position                = 08
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "document_assignment"
Stereotype << XSDelement >>
position                = 07
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Association "category"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Specification_category
Class "Specification_category"
Stereotype << XSDcomplexType >>
```

```
Attribute "implicit_exclusive_condition"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "id"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "specification_category_hierarchy"
Stereotype << XSDelement >>
position                = 05
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "document_assignment"
Stereotype << XSDelement >>
position                = 06
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "alias_identification"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Class Specification_category_hierarchy
Class "Specification_category_hierarchy"
Stereotype << XSDcomplexType >>

Association "sub_category"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
```

```
typeContainment         = false

Class Specification_expression
Class "Specification_expression"
Stereotype << XSDcomplexType >>

Attribute "operation"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "id"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "specification_inclusion"
Stereotype << XSDelement >>
position                = 05
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Association "operand"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Specification_inclusion
Class "Specification_inclusion"
Stereotype << XSDcomplexType >>

Attribute "id"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = true
typeContainment         = true
```

```
Composition "description"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Association "included_specification"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Supplier_solution
Class "Supplier_solution"
Stereotype << XSDcomplexType >>

Attribute "probability_rate"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Association "supplier"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Technical_solution
Class "Technical_solution"
Stereotype << XSDcomplexType >>

Composition "description"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true
```

B.1.12 Change and Work Management

```
Class Activity
Class "Activity"
Stereotype << XSDcomplexType >>
```

```
Attribute "activity_type"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "id"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "status"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "internal"
Stereotype << XSDelement >>
position                = 13
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "activity_relationship"
Stereotype << XSDelement >>
position                = 15
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "activity_element"
Stereotype << XSDelement >>
position                = 16
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "document_assignment"
Stereotype << XSDelement >>
```

```
position                = 17
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "simple_property_value"
Stereotype << XSDelement >>
position                = 18
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Association "chosen_method"
Stereotype << XSDelement >>
position                = 14
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "actual_start_date"
Stereotype << XSDelement >>
position                = 12
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "planned_start_date"
Stereotype << XSDelement >>
position                = 11
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "planned_end_date"
Stereotype << XSDelement >>
position                = 10
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "actual_end_date"
Stereotype << XSDelement >>
position                = 09
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "requestor"
Stereotype << XSDelement >>
position                = 08
anonymousRole           = false
```

```
anonymousType           = false
typeContainment         = false

Association "supplying_organization"
Stereotype << XSDelement >>
position                = 07
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "concerned_organization"
Stereotype << XSDelement >>
position                = 06
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "resolved_request"
Stereotype << XSDelement >>
position                = 05
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Activity_element
Class "Activity_element"
Stereotype << XSDcomplexType >>

Attribute "role"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "element_delivery"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "document_assignment"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Association "element"
Stereotype << XSDelement >>
```

```
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Activity_method
Class "Activity_method"
Stereotype << XSDcomplexType >>

Attribute "consequence"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "activity_method_assignment"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "name"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "document_assignment"
Stereotype << XSDelement >>
position                = 05
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Association "description"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Activity_method_assignment
Class "Activity_method_assignment"
Stereotype << XSDcomplexType >>

Attribute "relation_type"
Stereotype << XSDelement >>
position                = 02
```

```
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "simple_property_value"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Association "associated_request"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Activity_relationship
Class "Activity_relationship"
Stereotype << XSDcomplexType >>

Attribute "relation_type"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Association "related"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Change
Class "Change"
Stereotype << XSDcomplexType >>

Composition "description"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
```

```
anonymousType           = true
typeContainment         = true

Composition "document_assignment"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Class Element_delivery
Class "Element_delivery"
Stereotype << XSDcomplexType >>

Association "quantity"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "destination"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Project
Class "Project"
Stereotype << XSDcomplexType >>

Attribute "id"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "Project_relationship"
Stereotype << XSDelement >>
position                = 10
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = true
```

```
typeContainment         = true

Composition "name"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "document_assignment"
Stereotype << XSDelement >>
position                = 11
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Association "planned_end_date"
Stereotype << XSDelement >>
position                = 09
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "work_program"
Stereotype << XSDelement >>
position                = 08
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "planned_start_date"
Stereotype << XSDelement >>
position                = 06
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "actual_end_date"
Stereotype << XSDelement >>
position                = 05
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "actual_start_date"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = false
anonymousType           = false
typeContainment         = false
```

```
Association "is_applied_to"
Stereotype << XSDelement >>
position                = 07
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Project_relationship
Class "Project_relationship"
Stereotype << XSDcomplexType >>

Attribute "relation_type"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Association "related"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Work_order
Class "Work_order"
Stereotype << XSDcomplexType >>

Attribute "id"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "version_id"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "work_order_type"
```

```
Stereotype << XSDelement >>
position                = 05
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "document_assignment"
Stereotype << XSDelement >>
position                = 06
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Association "is_controlling"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Work_request
Class "Work_request"
Stereotype << XSDcomplexType >>

Attribute "id"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "request_type"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "status"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = true
typeContainment         = true
```

```
Attribute "version_id"
Stereotype << XSDelement >>
position                = 05
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 08
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "document_assignment"
Stereotype << XSDelement >>
position                = 09
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Association "notified_person"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "scope"
Stereotype << XSDelement >>
position                = 07
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "requestor"
Stereotype << XSDelement >>
position                = 06
anonymousRole           = false
anonymousType           = false
typeContainment         = false
```

B.1.13 Process Planning

```
Class Process_operation_definition
Class "Process_operation_definition"
Stereotype << XSDcomplexType >>

Attribute "id"
```

```
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "process_type"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "version_id"
Stereotype << XSDelement >>
position                = 05
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "process_operation_definition_relationship"
Stereotype << XSDelement >>
position                = 06
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "name"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "simple_property_value"
Stereotype << XSDelement >>
position                = 07
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Class Process_operation_definition_relationship
Class "Process_operation_definition_relationship"
Stereotype << XSDcomplexType >>
```

```
Attribute "relation_type"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Association "related"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Process_operation_input_or_output
Class "Process_operation_input_or_output"
Stereotype << XSDcomplexType >>

Attribute "role"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Association "concerned_shape"
Stereotype << XSDelement >>
position                = 05
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "placement"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "element"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
```

```
anonymousType           = false
typeContainment         = false

Class Process_operation_occurrence
Class "Process_operation_occurrence"
Stereotype << XSDcomplexType >>

Attribute "id"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "process_operation_resource_assignment"
Stereotype << XSDelement >>
position                = 06
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "process_operation_occurence_relationship"
Stereotype << XSDelement >>
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "process_operation_input_or_output"
Stereotype << XSDelement >>
position                = 07
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "configuration"
Stereotype << XSDelement >>
position                = 09
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "document_assignment"
Stereotype << XSDelement >>
position                = 08
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "simple_property_value"
Stereotype << XSDelement >>
position                = 10
```

```
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Association "operation_definition"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "is_defined_in"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "plan"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Process_operation_occurrence_relationship
Class "Process_operation_occurrence_relationship"
Stereotype << XSDcomplexType >>

Attribute "relation_type"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "change"
Stereotype << XSDelement >>
position                = 06
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Association "cycle_time"
```

```
Stereotype << XSDelement >>
position                = 05
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "waiting_time"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Association "related"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Process_operation_resource_assignment
Class "Process_operation_resource_assignment"
Stereotype << XSDcomplexType >>

Attribute "reference_tool"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "reason"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "simple_property_value"
Stereotype << XSDelement >>
position                = 05
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Association "placement"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = false
anonymousType           = false
typeContainment         = false
```

```
Association "resource_definition"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Process_plan
Class "Process_plan"
Stereotype << XSDcomplexType >>

Attribute "plan_id"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "process_plan_relationship"
Stereotype << XSDelement >>
position                = 05
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "name"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "configuration"
Stereotype << XSDelement >>
position                = 07
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Composition "document_assignment"
Stereotype << XSDelement >>
position                = 06
anonymousRole           = true
```

```
anonymousType           = false
typeContainment         = true

Composition "simple_property_value"
Stereotype << XSDelement >>
position                = 08
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Association "produced_output"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Process_plan_relationship
Class "Process_plan_relationship"
Stereotype << XSDcomplexType >>

Attribute "relation_type"
Stereotype << XSDelement >>
position                = 03
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "description"
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Composition "change"
Stereotype << XSDelement >>
position                = 04
anonymousRole           = true
anonymousType           = false
typeContainment         = true

Association "related"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Process_plan_version
Class "Process_plan_version"
```

```
Stereotype << XSDcomplexType >>

Attribute "version_id"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Class Process_property_association
Class "Process_property_association"
Stereotype << XSDcomplexType >>

Association "described_element"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false

Class Process_state
Class "Process_state"
Stereotype << XSDcomplexType >>

Association "related_item_definition"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = false
typeContainment         = false
```

B.1.14 Multi Language Support

```
Package "Multi_language_support"
Stereotype << XSDtranslatableString >>

Class Language
Class "Language"
Stereotype << XSDcomplexType >>

Attribute "language_code"
Stereotype << XSDelement >>
position                = 01
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Attribute "country_code"
```

```
Stereotype << XSDelement >>
position                = 02
anonymousRole           = false
anonymousType           = true
typeContainment         = true

Class Multi_language_string
Class "Multi_language_string"
Stereotype << XSDcomplexType >>
modelGroup              = omitComplexType

Class String_with_language
Class "String_with_language"
Stereotype << XSDcomplexType >>
modelGroup              = omitComplexType
```

B.2 XML Schema for PLM Services

The XML Schema for Product Lifecycle Management Services is defined in two separate parts:

- PLMInformationalModel.xsd
- PLMComputationalModel.xsd

Both documents are provided by separate OMG documents bound to http://schema.omg.org/specs/PLM/1.1.