INTERNATIONAL STANDARD

ISO/IEC 19763-10

First edition
2014-12-15

Information technology — Metamodel framework for interoperability (MFI) —

Part 10: Core model and basic mapping

Technologies de l'information — Cadre du métamodèle pour l'interopérabilité (MFI) —

Partie 10: Modèle de base et de cartographie de base

Reference number
ISO/IEC 19763-10:2014(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 19763-10 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 32, *Data management and interchange*.

ISO/IEC 19763 consists of the following parts, under the general title *Information technology — Metamodel framework for interoperability (MFI)*:

— *Part 1: Framework*

— *Part 3: Metamodel for ontology registration*

— *Part 5: Metamodel for process model registration*

— *Part 6: Registry Summary*

— *Part 10: MFI Core model and basic mapping*

— *Part 12: Metamodel for information model registration*

The following parts are under preparation:

— *Part 7: Metamodel for service registration*

— *Part 8: Metamodel for role and goal registration*

— *Part 9: On demand model selection*

— *Part 13: Metamodel for forms registration*

Introduction

Industrial consortia have engaged in the standardization of domain-specific business objects including business process models and software components using common modelling facilities and interchange facilities such as UML and XML. They are very active in standardizing domain-specific business process models and standard modelling constructs such as data elements, entity profiles, and value domains.

However, to promote interoperability across business domains, a generic framework for registering a variety of models and the mapping between them is required. This part of ISO/IEC 19763 provides a core metamodel as the basis for the other parts of ISO/IEC 19763 and a metamodel for registering the mappings between models registered in those other parts of ISO/IEC 19763.

NOTE UML is a trademark of the Object Management Group.

Information technology — Metamodel framework for interoperability (MFI) —

Part 10:
Core model and basic mapping

1 Scope

The primary purpose of this International Standard is to specify a metamodel framework for interoperability. This part of ISO/IEC 19763 specifies the metamodel that provides a facility to register administrative information and common semantics of models and mapping between two models.

This part of ISO/IEC 19763 does not specify the metamodel of models in a specific language, but provides a common core metamodel for the other parts of ISO/IEC 19763, each of which specifies a metamodel for a registry that can register models of a specific type, such as ontologies, process models or information models, in a number of different languages.

This part of ISO/IEC 19763 also provides a metamodel for registering the mappings between two models registered in those other parts of ISO/IEC 19763.

This part of ISO/IEC 19763 utilises the common facilities specified in ISO/IEC 11179-3. ISO/IEC 11179 (all parts) specifies a metadata registry (MDR). These common facilities provide the ability to identify and register models and their associated model elements and modelling languages within a metadata registry used to register models.

Figure 1 shows the relationship between this part of ISO/IEC 19763 and other parts of ISO/IEC 19763.

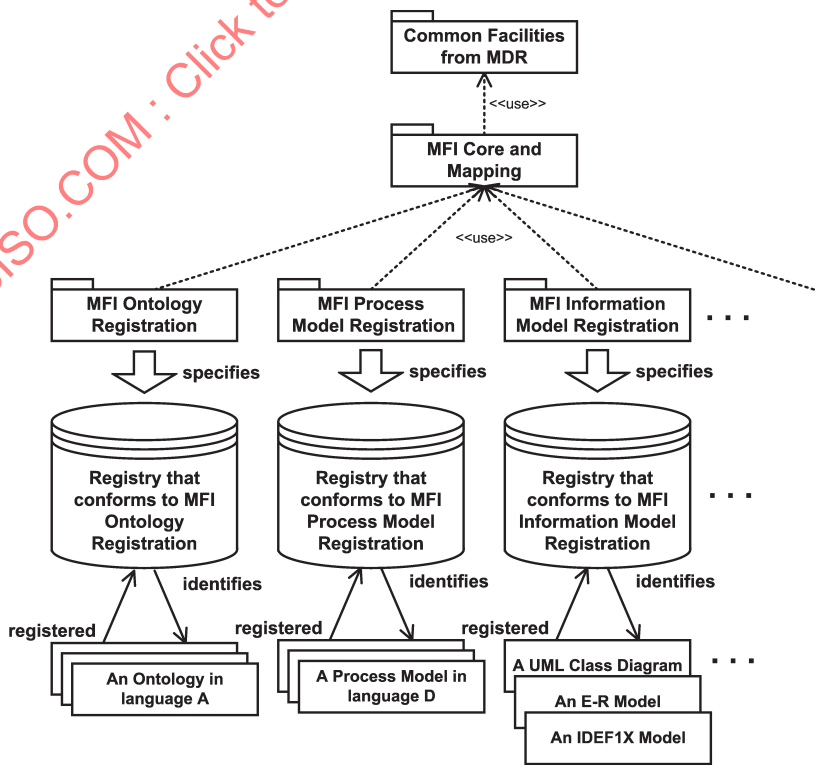


Figure 1 — Relationships between MFI Core and mapping and other parts

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 11179-3:2013, *Information technology — Metadata registries (MDR) — Part 3: Registry metamodel and basic attributes*

ISO/IEC 11179-6, *Information technology — Metadata registries (MDR) — Part 6: Registration*

ISO/IEC 19763-1[1)], *Information technology — Metamodel framework for interoperability (MFI) — Part 1: Reference model*

ISO/IEC 19763-3:2010, *Information technology — Metamodel framework for interoperability (MFI) — Part 3: Metamodel for ontology registration*

3 Terms, definitions and abbreviated terms

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 11179-3, ISO/IEC 11179-6, ISO/IEC 19763-1, and the following apply.

3.1.1
modelling language
language or notation that is used to model some aspect of a domain of interest

3.1.2
model
representation of some aspect of a domain of interest using a normative modelling facility and modelling constructs

Note 1 to entry: Models can be used to express a set of information requirements, processes, services, roles, goals, or some other aspect of a domain of interest.

[SOURCE: ISO/IEC 19763-1:2007, 4.1.7, modified]

3.1.3
model element
element or component in a model

Note 1 to entry: Examples of model elements are a representation of an entity type in an information model, a representation of an event in a process model, a representation of a service operation in a service model, or a representation of an actor in a role and goal model.

3.1.4
model element set
means by which a set of model elements are grouped together for the purposes of mapping, with each set forming one half of a model element set mapping

3.1.5
model mapping
collection of model element set mappings

Note 1 to entry: A model mapping is a statement that the domains represented by two models intersect and there is a need to register details of the correspondence between the structures in the models that semantically represent this overlap to achieve interoperability.

1) This standard is under revision.

3.1.6
model element set mapping
part of a model mapping, comprising two sets of model elements, one from each of the models being mapped, that model similar real-world requirements.

3.1.7
model element set mapping type
classification for a set of model element set mappings

3.2 Abbreviated terms

MFI

metamodel framework for interoperability

MFI Core and mapping

ISO/IEC 19763-10, *Information technology — Metamodel framework for interoperability (MFI) — Part 10: Core model and basic mapping*

MFI Ontology registration

ISO/IEC 19763-3, *Information technology — Metamodel framework for interoperability (MFI) — Part 3: Metamodel for ontology registration*

MDR

metadata registry

4 Conformance

4.1 General

An implementation claiming conformance to this part of ISO/IEC 19763 shall conform as further described in this Clause.

4.2 Conformance Levels

4.2.1 Conformance Level 1

The metamodel specified in Clause 7 (The Core_Model package) is supported.

4.2.2 Conformance Level 2

The metamodels specified in Clause 7 (The Core_Model package) and Clause 8 (The Basic_Mapping_Model package) are supported.

4.3 Implementation Conformance Statement (ICS)

An implementation claiming conformance to this part of ISO/IEC 19763 shall include an Implementation Conformance Statement stating:

a) which conformance level it claims (4.2);

b) for each of the ISO/IEC 19763 metaclasses supported in the implementation, what types specified in ISO/IEC 11179-3:2013, 5.5 are used to extend that metaclass.

NOTE 1 The use of the types specified in ISO/IEC 11179-3:2013, 5.5 is explained in Clause 9.

NOTE 2 Other parts of ISO/IEC 19763 will require a conformance to this part of ISO/IEC 19763 as a part of its conformance claim. It will follow 4.3.

5 Overview of MFI Core and mapping

5.1 Basic Structure

MFI Core and mapping consists of two packages: **Core_Model** package and **Basic_Mapping_Model** package. Figure 2 shows the package structure of MFI Core and mapping.

The **Core_Model** package specifies a core model that is inherited by other parts of ISO/IEC 19763 to register administrative information and common semantics of models. The **Basic_Mapping_Model** package specifies a basic mapping model that allows the registration of information about mappings between models.

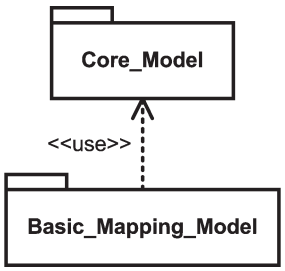
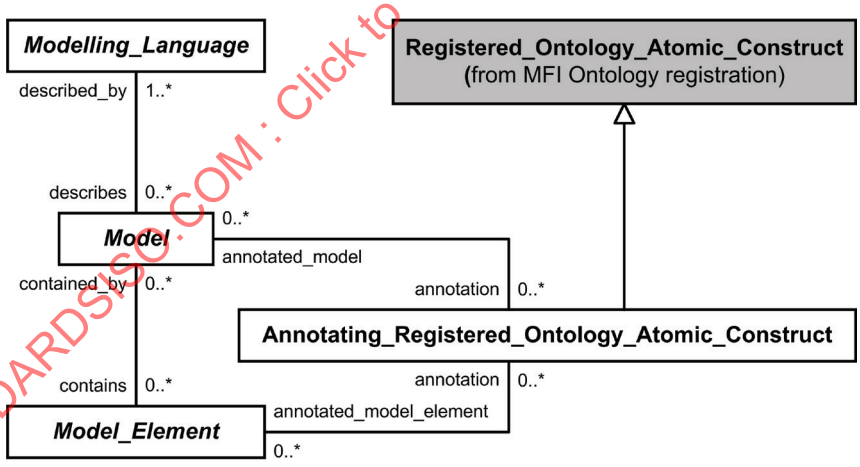


Figure 2 — Package structure of MFI Core and Mapping

5.2 Overview of the Core_Model package

Figure 3 shows the metamodel in the **Core_Model** package.



NOTE 1 Metaclasses whose names are italicized are abstract metaclasses.

NOTE 2 Metaclasses grey shaded are metaclasses that are defined in the other parts of ISO/IEC 19763.

Figure 3 — The Metamodel in the Core_Model package

It comprises the following metaclasses:

— **Modelling_Language**;

— **Model**;

— **Model_Element**;

— **Annotating_Registered_Ontology_Atomic_Construct**.

The metaclasses that represent models in other parts of ISO/IEC 19763 will be subclasses of **Model** and the metaclasses representing the content of these models will be subclasses of **Model_Element**. Some of the metaclasses in other parts of ISO/IEC 19763 inherited from **Model** or **Model_Element** may override some attributes and/or references of the metaclasses defined in this part. The metaclasses that represent the language in which these models are expressed will be subclasses of **Modelling_Language**.

Each modelling language is a language or notation that is used to model concepts found in the other parts of the ISO/IEC 19763 family of standards.

NOTE 1 Examples are languages used to express ontologies and/or to model information requirements, processes, services, forms, roles, goals, or some other set of concepts that can be modelled.

Each model is a representation of concepts found in the other parts of the ISO/IEC 19763 family of standards.

NOTE 2 Examples are an ontology or a model used to represent a set of information requirements, processes, services, forms roles, goals, or some other set of concepts that can be represented in a model.

Each model element is an element or component in a model, such as a non-logical symbol in an ontology, an entity type in an information model, an event in a process model, a service operation in a service model, or an actor in a role and goal model.

Each model element is an element or component in a model, such as those found in the other parts of the ISO/IEC 19763 family of standards.

NOTE 3 Examples are non-logical symbol in an ontology, an entity type in an information model, an event in a process model, a service operation in a service model, a section or question on a form, or an actor in a role and goal model or other types found in ISO/IEC 19763 family of standards.

Each model shall be expressed in one or more modelling language. Each modelling language is used to describe zero, one, or more models.

Each model contains zero, one, or more model elements. Each model element is part of zero, one, or more models.

Each annotating registered ontology atomic construct annotates zero, one, or more models. Each model is annotated by zero, one, or more annotating registered ontology atomic constructs.

Each annotating registered ontology atomic construct annotates zero, one, or more model elements. Each model element is annotated by zero, one, or more annotating registered ontology atomic constructs.

The exact specification of each metaclass in the **Core_Model** package is given in Clause 7.

5.3 Overview of the Basic_Mapping_Model package

Figure 4 shows the metamodel in the **Basic_Mapping_Model** package.

It comprises the following metaclasses:

— **Model_Mapping**;

— **Model_Element_Set_Mapping**;

— **Model_Element_Set_Mapping_Type**;

— **Model_Element_Set**.

Each model mapping is a mapping from one model to another model, one of which is the subject model and the other is the object model.

Each model element set is the means by which a set of model elements are grouped together for the purposes of mapping, with each set forming one half of a model element set mapping.

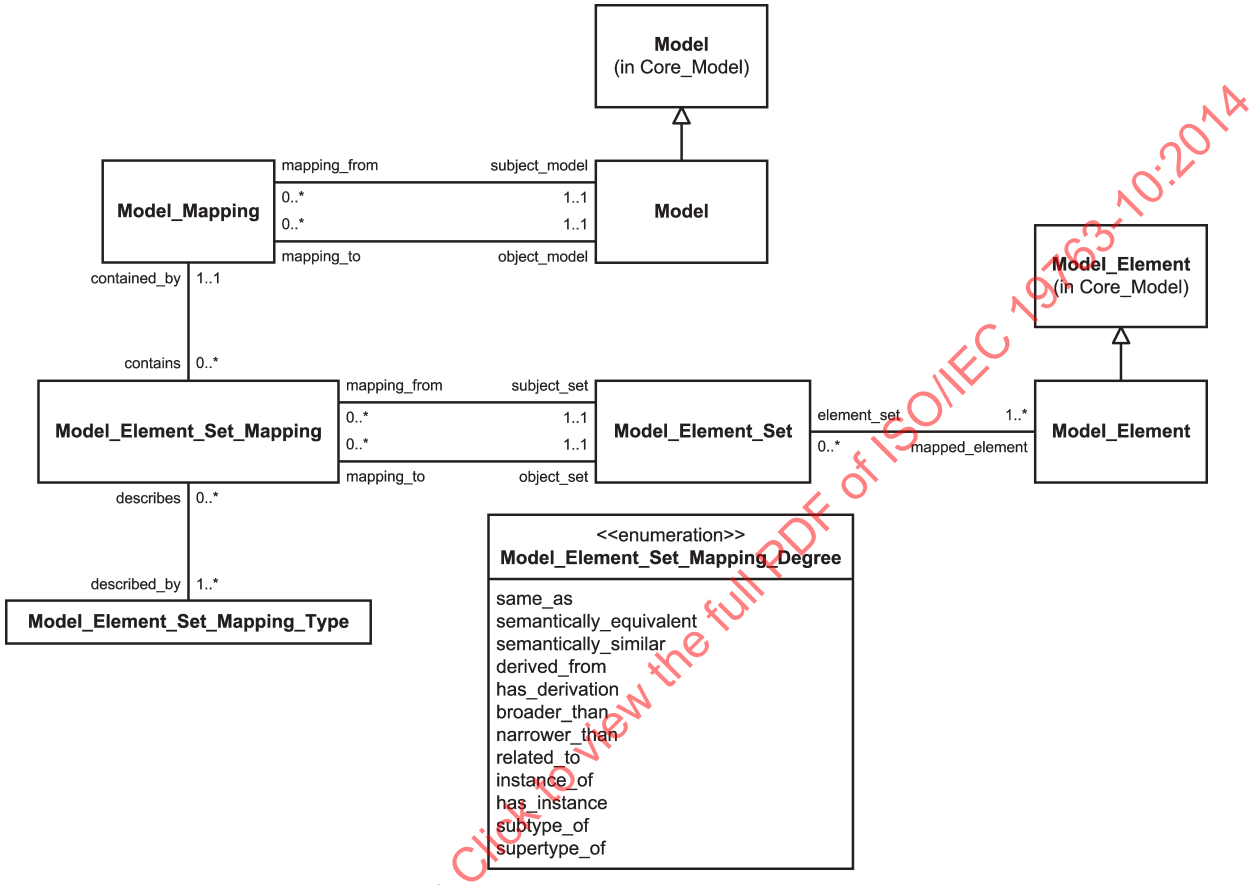


Figure 4 — The Metamodel in the Basic_Mapping_Model package

Each model element set mapping is part of a model mapping. Each model element set mapping comprises a set of model elements from the subject model and a set of elements from the object model that model similar real-world requirements. Each model element set mapping is subject to a model element set mapping degree.

Each model element set mapping type is a classification for a set of model element mappings.

Each model mapping shall be from one and only one model, which is the subject model. Each model is the subject of zero, one, or more model mappings.

Each model mapping shall be to one and only one model, which is the object model. Each model is the object of zero, one, or more model mappings.

Each model element set mapping shall be described by one and only one model element set mapping type. Each model element set mapping type is the description of zero, one, or more model element set mappings.

Each model element set mapping shall be from one and only one model element set, which is the subject model element set. Each model element set is the subject of zero, one, or more model element set mappings.

Each model element set mapping shall be to one and only one model element set, which is the object model element set. Each model element set is the object of zero, one, or more model element set mappings.

Each model mapping shall be comprised of one or more model element set mappings. Each model element set mapping shall be part of one or more model mappings.

Each model element set contains zero, one, or more model elements. Each model element is mapped though zero, one, or more model element sets.

The exact specification of each metaclass in the **Basic_Mapping_Model** package is given in Clause 8.

6 Detail provided in each metaclass definition

In this part of ISO/IEC 19763, each metaclass is defined in the following form. Other parts of ISO/IEC 19763 also use this notation and format unless otherwise specified.

Each metaclass is defined at its relevant subclause.

Title of a subclause is the metaclass name and the first paragraph is a brief description of the metaclass.

Its direct superclass is specified under the heading of **Superclass**.

Its attributes are listed and defined under the headings **Attribute**, **DataType**, **Multiplicity**, and **Description**, under each of which, the name of the attribute, its datatype, its multiplicity, and a description are given, respectively.

Its references are listed and defined under the headings **Reference**, **Class**, **Multiplicity**, **Description**, **Inverse**, and **Precedence**, under each of which, the name of the reference, the name of the associated metaclass, its multiplicity, a description, its inverse reference name, and whether it has precedence over its inverse reference or not, are given, respectively. The inverse reference name shall be the name of the reference in the associated metaclass which is complementary to this reference.

If a class has any other constraints, they are described in a natural language under the heading of **Constraints**.

In UML, an association is equivalent to a reference and its inverse reference. In this part of ISO/IEC 19763, to formally define a metaclass, associations are not used, but each association is replaced by two references, a reference and its inverse reference. If a reference has precedence over its inverse reference, it means that instances of the relevant metaclass have responsibility for maintenance of the links of the association defined by it and its inverse reference.

For better understandability, diagrams shows associations, rather than references. A role name of an association is a reference name of the associated metaclass.

For simplicity, a diagram does not show attributes.

Multiplicity constraints of attributes and references and other constraints of a metaclass are to be enforced when registration status of an instance of the metaclass is “recorded” or higher.

NOTE Registration statuses are defined in ISO/IEC 11179-6.

7 The Core_Model package

7.1 Modelling_Language

Modelling_Language is an abstract metaclass each instance of which represents a language or notation that is used to model concepts found in the other parts of the ISO/IEC 19763 family of standards. The subclasses of Modelling_Language are specified in the subordinate parts of ISO/IEC 19763.

NOTE Examples are languages used to express ontologies and/or to model information requirements, processes, services, forms, roles, goals, or some other set of concepts that can be modelled.

Superclass

[None]

Attribute

[None]

Reference	Class	Multiplicity	Description	Inverse	Precedence
describes	Model	0..*	The set of models that this language describes	described_by	No

Constraints

[None]

7.2 Model

Model is an abstract metaclass each instance of which represents a representation of concepts found in the other parts of the ISO/IEC 19763 family of standards. The subclasses of Model are specified in the subordinate parts of ISO/IEC 19763.

NOTE Examples are an ontology or is a model used to represent a set of information requirements, processes, services, form designs, roles, goals, or some other set of concepts that can be represented in a model.

Superclass

[None]

Attribute

[None]

Reference	Class	Multiplicity	Description	Inverse	Precedence
described_by	Modelling_Language	1..*	The set of languages used to express this model	describes	Yes
contains	Model_Element	0..*	The set of model elements that are part of this model	contained_by	Yes
annotation	Annotating_Registered_Ontology_Atomic_Construct	0..*	The set of annotating registered ontology atomic constructs that annotate this model	annotated_model	No

Constraints

[None]

7.3 Model_Element

Model_Element is an abstract metaclass each instance of which represents an element or component in a model, such as those found in the other parts of the ISO/IEC 19763 family of standards. The subclasses of Model_Element are specified in the subordinate parts of ISO/IEC 19763.

NOTE Examples are non-logical symbol in an ontology, an entity type in an information model, an event in a process model, a service operation in a service model, a section or question on a form, or an actor in a role and goal model or other types found in ISO/IEC 19763 family of standards.

Superclass

[None]

Attribute

[None]

Reference	Class	Multiplicity	Description	Inverse	Precedence
contained_by	Model	0..*	The set of models that contain this model element	contains	No
annotation	Annotating_Registered_Ontology_Atomic_Construct	0..*	The set of annotating registered ontology atomic constructs that annotate this model element	annotated_model_element	No

Constraints

[None]

7.4 Annotating_Registered_Ontology_Atomic_Construct

Annotating_Registered_Ontology_Atomic_Construct is a metaclass each instance of which is a particular instance of Registered_Ontology_Atomic_Construct that annotates one or more instances of Model or Model_Element.

Superclass

Registered_Ontology_Atomic_Construct (from MFI Ontology registration)

Attribute

[None]

Reference	Class	Multiplicity	Description	Inverse	Precedence
annotated_model	Model	0..*	The set of models that are annotated by this annotating registered ontology atomic construct	annotation	Yes
annotated_model_element	Model_Element	0..*	The set of model elements that are annotated by this annotating registered ontology atomic construct	annotation	Yes

Constraints

At least, one of the cardinalities of annotated_model or of annotated_model_element shall be one or more.

8 The Basic_Mapping_Model package

8.1 Model (as specialized)

Model is a subclass of Model in the Core_Model package, each instance of which represents a particular model that is mapped to another model through zero, one, or more model mappings.

Superclass

Model (in the Core_Model package)

Attribute	Datatype	Multiplicity	Description
[None]			

Reference	Class	Multiplicity	Description	Inverse	Precedence
mapping_from	Model_Mapping	0..*	The set of model mappings that map from this model	subject_model	No
mapping_to	Model_Mapping	0..*	The set of model mappings that map to this model	object_model	No

Constraints

[None]

8.2 Model_Element (as specialized)

Model_Element is a subclass of Model_Element in the Core_Model package, each instance of which represents a model element that is an element of zero, one, or more model element sets.

Superclass

Model_Element (in the Core_Model package)

Attribute

[None]

Reference	Class	Multiplicity	Description	Inverse	Precedence
element_set	Model_Element_Set	0..*	The set of model elements of which this model element is an element for the purposes of mapping	mapped_element	Yes

Constraints

[None]

8.3 Model_Mapping

Model_Mapping is a metaclass each instance of which represents a mapping from one model to another model, one of which is known as the subject model and the other is known as the object model.

Superclass

[None]

Attribute

[None]

Reference	Class	Multiplicity	Description	Inverse	Precedence
subject_model	Model	1..1	The model that is the subject of this model mapping.	mapping_from	Yes
object_model	Model	1..1	The model that is the object of this model mapping.	mapping_to	Yes
contains	Model_Element_Set_Mapping	1..*	The set of model element set mappings that are contained by this model mapping.	contained_by	Yes

Constraints

[None]

8.4 Model_Element_Set_Mapping

Model_Element_Set_Mapping is a metaclass each instance of which represents some part of a model mapping; each model element set mapping comprises a set of model elements from the subject model and a set of model elements from the object model that model similar real-world requirements.

Superclass

[None]

Attribute	Datatype	Multiplicity	Description
degree	Model_Element_Set_Mapping_Degree	1..1	The degree of equivalence exhibited by this model element set mapping.

Reference	Class	Multiplicity	Description	Inverse	Precedence
subject_set	Model_Element_Set	1..1	The model element set that forms the subject of this model element set mapping	mapping_from	Yes
object_set	Model_Element_Set	1..1	The model element set that forms the object of this model element set mapping	mapping_to	Yes
described_by	Model_Element_Mapping_Type	1..1	The model element mapping type that describes or classifies this model element set mapping	describes	Yes

contained_by	Model_Mapping	0..*	The set of model mappings this model element set mapping is contained by	contains	No

Constraints

[None]

8.5 Model_Element_Set_Mapping_Type

Model_Element_Set_Mapping_Type is a metaclass each instance of which represents a classification for a set of model element set mappings.

Superclass

[None]

Attribute	Datatype	Multiplicity	Description
label	String	1..1	The label that defines the classification described by this model element set mapping type. Examples are “Object Class to Entity Type mapping” when mapping two information models and “Sequence dependency replacement” when mapping two process models.

Reference	Class	Multiplicity	Description	Inverse	Precedence
describes	Model_Element_Set_Mapping	0..*	The set of model element set mappings that are described or classified by this model element set mapping type	described_by	Yes

Constraints

[None]

8.6 Model_Element_Set

Model_Element_Set is a metaclass each instance of which represents the means by which a set of model elements are grouped together for the purposes of mapping; each set forms one half of a model element set mapping.

Superclass

[None]

Attribute

[None]

Reference	Class	Multiplicity	Description	Inverse	Precedence
mapped_element	Model_Element	1..*	The set of model elements that comprise this model mapping component set	element_set	Yes

mapping_from	Model_Element_Set_Mapping	0..*	The set of model element set mappings that include this model element set as the subject set	subject_set	No
mapping_to	Model_Element_Set_Mapping	0..*	The set of model element set mappings that include this model element set as the object set	object_set	No

Constraints

[None]

8.7 Model_Element_Set_Mapping_Degree

Model_Element_Set_Mapping_Degree is an enumerated datatype with the following values:

Value	Description
same_as	Indicates that the creator of the mapping believes the subject set and the object set are syntactically and semantically identical.
semantically_equivalent	Indicates that the creator of the mapping believes the subject set and the object set are semantically identical.
semantically_similar	Indicates that the creator of the mapping believes the subject set and the object set are semantically similar although not identical.
derived_from	Indicates that the creator of the mapping has asserted that the subject set has been used in the creation of the object set.
has_derivation	Inverse of derived_from.
broader_than	Indicates that the creator of the mapping believes the subject set and the object set are linked, but the object set has a broader meaning than the subject set.
narrower_than	Inverse of broader_than.
related_to	Indicates that the subject_set and the object_set share some form of relationship.
instance_of	Indicates that the object set is an implementation of the subject set.
has_instance	Inverse of instance_of.
subtype_of	Indicates that the object set is an implementation of the subject set and additionally that the subject set satisfies requirements for participating in computing operation(s) in some framework which will also complete on the object set.
supertype_of	Inverse of subtype_of.

9 Use of the common facilities types specified in ISO/IEC 11179 within ISO/IEC 19763

9.1 General principles

Any model registry based on the specifications of the ISO/IEC 19763 family of standards shall use the common facilities specified in ISO/IEC 11179-3.

Instances of each of the metaclasses specified in the ISO/IEC 19763 family of standards shall be extended by one or more of the types specified in ISO/IEC 11179-3:2013, 5.5.

The principles are:

— Any metadata item that is to be identified, shall be an instance of **Identified_Item**.

— Any metadata item that is to be registered, shall be an instance of **Registered_Item**, either an **Administered_Item** with its own administrative information, or an **Attached_Item**, associated with another **Administered_Item**.

— Any metadata item that is to be named and/or defined, shall be an instance of **Designatable_Item**.

— Any metadata item that is to be classified, shall be an instance of **Classifiable_Item**.

9.2 specifies the possible types for instances of the metaclasses specified in this part of ISO/IEC 19763.

Each subordinate part of ISO/IEC 19763 shall specify the types for the instances of each metaclass that are specified in that part.

NOTE For clarity and completeness, subordinate parts can include "name" (or similar) attributes in metaclasses whose instances are to be typed as **Designatable_Items.** It is assumed that these attributes will not be implemented in any model registry and designations, names, labels, etc. will be applied through the typing as **Designatable_Items.**

9.2 Application to this part

Listed below are the possible types for instances of the metaclasses specified in this part of ISO/IEC 19763.

— Instances of **Modelling_Language** should be instances of **Identified_Item**.

— Most instances of **Model** should be instances of **Registered_Item**, but there are exceptions. For example, in ISO/IEC 19763-3, instances of **Registered_Ontology_Whole** will be instances of **Registered_Item** while instances of **Unregistered_Ontology_Whole** will not be registered, by definition. Those instances of **Model** which are registered will more likely be **Administered_Items**, rather than **Attached_Items.**

— Most instances of **Model_Element** should be instances of **Registered_Item**, but there are exceptions. For example, in ISO/IEC 19763-3, instances of **Registered_Ontology_Atomic_Construct** will be instances of **Registered_Item** while instances of **Unregistered_Ontology_Atomic_Construct** will not be registered, by definition. Those instances of **Model_Element** which are registered might be **Administered_Items** in their own right, but are more likely to be **Attached_Items**, attached to the **Administered_Item** that represents the containing **Model**.Instances of **Model_Mapping**, **Model_Element_Set_Mapping** and **Model_Element_Set** should be instances of **Administered_Item**.

— Instances of **Model_Element_Set_Mapping_Type** are examples of instances which may not be instances of **Identified_Item**.

— Instances of any of the metaclasses might be instances of **Designatable_Item**, allowing them to be defined and or designated (named).

— Instances of any of the metaclasses might be instances of **Classifiable_Item**, allowing them to be classified.

Annex A
(informative)

Examples

A.1 Introduction

This Annex illustrates the registration using the core model of models described using the metamodels specified in the subordinate parts of ISO/IEC 19763 and the registration of the relevant mappings between those models.

The examples are not exhaustive and only the objects relevant to the mappings are described. Not every possible mapping between the models is illustrated.

Note that the registration of mappings is only possible in registries that are at Conformance Level 2. Conformance Level 1 only allows the registration of models without mappings.

The examples all follow the same format. First the example models are presented, two for each example. This is, then, followed by three sets of illustrative object instances.

Each object instance is identified with the name of the metaclass in angle brackets. Where appropriate the name from this part is followed by the name of the metaclass from the subordinate part in parenthesis. For simplicity, some attributes and references are omitted.

Object identifiers, of the form "ObjectXXXX" are introduced to help with the description of the examples. The detailed specification these identifiers are beyond the scope of this part of the standard.

A.2 Example 1 — Registration and mapping of two information models

This example illustrates two examples of mapping between the information model used by System A, which is expressed as a UML Class Diagram (see Figure A.1), and an equivalent information model that is used by System B, which is expressed using the IDEF1X notation (see Figure A.2).

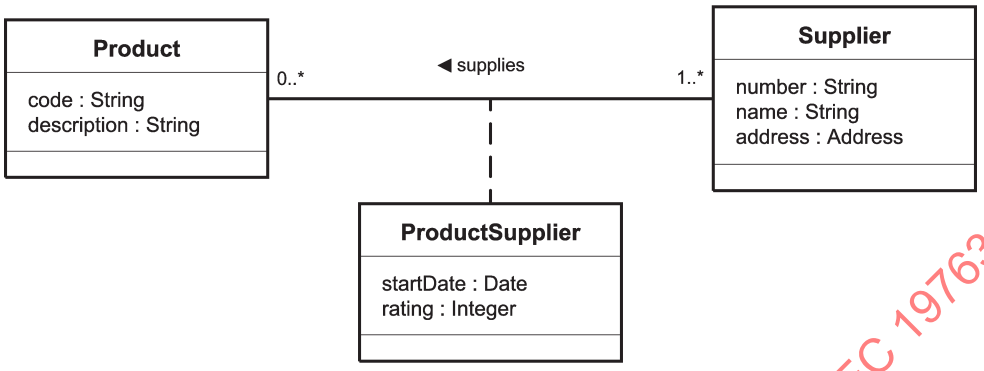


Figure A.1 — Example UML Class Diagram for the Product Supplier concept (as used by System A)

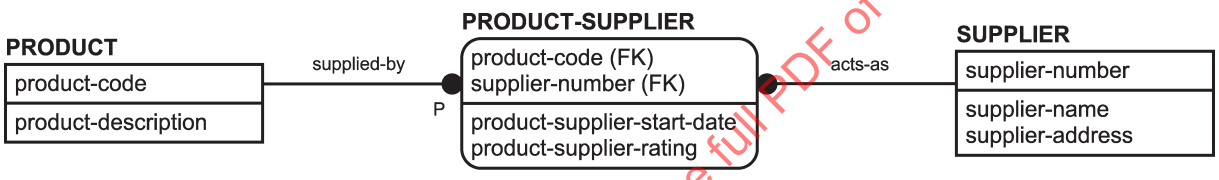


Figure A.2 — Example IDEF1X Model for the Product Supplier concept (as used by System B)

The UML Class Diagram used by System A is registered using Object1101 to Object1115 as shown in Figure A.3 below. The IDEF1X Model used by System B is registered using Object1201 to Object1218 as shown in Figure A.4 below. The mappings are registered using Object1301 to Object1309 as shown in Figure A.5 below. The mapping of the UML **Product** class to the IDEF1X **PRODUCT** entity type is illustrated using Object1302 to Object1305 and the mapping of the UML **supplies** association and the associative **ProductSupplier** class to the IDEF1X **supplied-by** and **acts-as** relationships with the **PRODUCT-SUPPLIER** associative entity type is illustrated using Object1306 to Object1309.

<Modelling_Language (Information_Modelling_Language)>
Object1101

Attribute/Reference	Literal/Instance
name	"UML Class Diagram"
expressed_model	Object1102

<Model (Information_Model)>
Object1102

Attribute/Reference	Literal/Instance
name	"Product Supplier Model A"
describing_language	Object1101
diagram_model_element	Object1103
mapping_from	Object1301

<Model_Element (Diagram)>
Object1103

Attribute/Reference	Literal/Instance
name	"Product Supplier Model A – Dia 1"
containing_model	Object1102
entity_type_model_element	Object1104, Object1107, Object1108
relationship_model_element	Object1109

<Model_Element (Entity_Type)>
Object1104

Attribute/Reference	Literal/Instance
name	"Product"
containing_diagram	Object1103
characteristic_partial_description	Object1105, Object1106
relationship_end_group_ partial_description	Object1110
element_set	Object1302

<Model_Element (Non_Key_Attribute)>
Object1105

Attribute/Reference	Literal/Instance
name	"code"
described_entity_type	Object1104
element_set	Object1302

<Model_Element (Non_Key_Attribute)>
Object1106

Attribute/Reference	Literal/Instance
name	"description"
described_entity_type	Object1104
element_set	Object1302

<Model_Element (Entity_Type)>
Object1107

Attribute/Reference	Literal/Instance
name	"Supplier"
containing_diagram	Object1103
relationship_end_group_ partial_description	Object1112

<Model_Element (Entity_Type)>
Object1108

Attribute/Reference	Literal/Instance
name	"ProductSupplier"
containing_diagram	Object1103
relationship_end_group_ partial_description	Object1114
element_set	Object1306

<Model_Element (Relationship)>
Object1109

Attribute/Reference	Literal/Instance
name	"supplies"
containing_diagram	Object1103
contained_relationship_end	Object1111, Object1113, Object1115
element_set	Object1306

<Model_Element (Relationship_End_Group)>
Object1110

Attribute/Reference	Literal/Instance
described_entity_type	Object1104
mutually_excluive_component	Object1111
element_set	Object1306

<Model_Element (Relationship_End)>
Object1111

Attribute/Reference	Literal/Instance
minimum_cardinality	"0"
maximum_cardinality	"*"
associative_indicator	"No"
containing_relationship	Object1109
containing_group	Object1110
element_set	Object1306

<Model_Element (Relationship_End_Group)>
Object1112

Attribute/Reference	Literal/Instance
described_entity_type	Object1107
mutually_excluive_component	Object1113
element_set	Object1306

<Model_Element (Relationship_End)>
Object1113

Attribute/Reference	Literal/Instance
minimum_cardinality	"1"
maximum_cardinality	"*"
associative_indicator	"No"
containing_relationship	Object1109
containing_group	Object1112
element_set	Object1306

<Model_Element (Relationship_End_Group)>
Object1114

Attribute/Reference	Literal/Instance
described_entity_type	Object1108
mutually_excluive_component	Object1115
element_set	Object1306

<Model_Element (Relationship_End)>
Object1115

Attribute/Reference	Literal/Instance
associative_indicator	"Yes"
containing_relationship	Object1109
containing_group	Object1114
element_set	Object1306

Figure A.3 — Registration of the System A UML Class Diagram for the Product Supplier concept

<Modelling_Language (Information_Modelling_Language)>
Object1201

Attribute/Reference	Literal/Instance
name	"IDEF1X"
expressed_model	Object1202

<Model (Information_Model)>
Object1202

Attribute/Reference	Literal/Instance
name	"Product Supplier Model B"
describing_language	Object1201
diagram_model_element	Object1203
mapping_to	Object1301

<Model_Element (Diagram)>
Object1203

Attribute/Reference	Literal/Instance
name	"Product Supplier Model B – Dia 1"
containing_model	Object1202
entity_type_model_element	Object1204, Object1207, Object1208
relationship_model_element	Object1209, Object1214
mapping_to	Object1301

<Model_Element (Entity_Type)>
Object1204

Attribute/Reference	Literal/Instance
name	"PRODUCT"
containing_diagram	Object1203
characteristic_partial_description	Object1205, Object1206
relationship_end_group_partial_description	Object1210
element_set	Object1303

<Model_Element (Native_Key_Attribute)>
Object1205

Attribute/Reference	Literal/Instance
name	"product-code"
described_entity_type	Object1204
element_set	Object1303

<Model_Element (Non_Key_Attribute)>
Object1206

Attribute/Reference	Literal/Instance
name	"product-description"
described_entity_type	Object1204
element_set	Object1303

<Model_Element (Entity_Type)>
Object1207

Attribute/Reference	Literal/Instance
name	"SUPPLIER"
containing_diagram	Object1203
relationship_end_group_partial_description	Object1215

<Model_Element (Entity_Type)>
Object1208

Attribute/Reference	Literal/Instance
name	"PRODUCT-SUPPLIER"
containing_diagram	Object1203
relationship_end_group_partial_description	Object1212, Object1217
element_set	Object1307

<Model_Element (Relationship)>
Object1209

Attribute/Reference	Literal/Instance
name	"supplied-by"
identifying_indicator	"Yes"
containing_diagram	Object1203
contained_relationship_end	Object1211, Object1213
element_set	Object1307

<Model_Element (Relationship_End_Group)>
Object1210

Attribute/Reference	Literal/Instance
described_entity_type	Object1204
mutually_excluive_component	Object1211
element_set	Object1307

<Model_Element (Relationship_End)>
Object1211

Attribute/Reference	Literal/Instance
minimum_cardinality	"1"
maximum_cardinality	"1"
containing_relationship	Object1209
containing_group	Object1210
element_set	Object1307

<Model_Element (Relationship_End_Group)>
Object1212

Attribute/Reference	Literal/Instance
described_entity_type	Object1208
mutually_excluive_component	Object1213
element_set	Object1307

<Model_Element (Relationship_End)>
Object1213

Attribute/Reference	Literal/Instance
minimum_cardinality	"1"
maximum_cardinality	"*"
containing_relationship	Object1209
containing_group	Object1212
element_set	Object1307

<Model_Element (Relationship)>
Object1214

Attribute/Reference	Literal/Instance
name	"acts-as"
identifying_indicator	"Yes"
containing_diagram	Object1203
contained_relationship_end	Object1216, Object1218
element_set	Object1307

<Model_Element (Relationship_End_Group)>
Object1215

Attribute/Reference	Literal/Instance
described_entity_type	Object1207
mutually_excluive_component	Object1216
element_set	Object1307

<Model_Element (Relationship_End)>
Object1216

Attribute/Reference	Literal/Instance
minimum_cardinality	"1"
maximum_cardinality	"1"
containing_relationship	Object1214
containing_group	Object1215
element_set	Object1307

<Model_Element (Relationship_End_Group)>
Object1217

Attribute/Reference	Literal/Instance
described_entity_type	Object1208
mutually_excluive_component	Object1218
element_set	Object1307

<Model_Element (Relationship_End)>
Object1218

Attribute/Reference	Literal/Instance
minimum_cardinality	"0"
maximum_cardinality	"*"
containing_relationship	Object1214
containing_group	Object1217
element_set	Object1307

Figure A.4 — Registration of the System B IDEF1X Model for the Product Supplier concept

<Model_Mapping>
Object1301

Attribute/Reference	Literal/Instance
subject_model	Object1102
object_model	Object1202
contains	Object1305, Object1309

<Model_Element_Set>
Object1302

Attribute/Reference	Literal/Instance
mapped_element	Object1104, Object1105, Object1106
mapping_from	Object1305

<Model_Element_Set>
Object1303

Attribute/Reference	Literal/Instance
mapped_element	Object1204, Object1205, Object1206
mapping_to	Object1305

<Model_Element_Set_Mapping_Type>
Object1304

Attribute/Reference	Literal/Instance
label	"ObjectClass to Entity Type mapping"
describes	Object1305

<Model_Element_Set_Mapping>
Object1305

Attribute/Reference	Literal/Instance
degree	"semantically_equivalent"
contained_by	Object1301
subject_set	Object1302
object_set	Object1303
described_by	Object1304

<Model_Element_Set>
Object1306

Attribute/Reference	Literal/Instance
mapped_element	Object1108, Object1109, Object1110, Object1111, Object1112, Object1113, Object1114, Object1115
mapping_from	Object1309

<Model_Element_Set>
Object1307

Attribute/Reference	Literal/Instance
mapped_element	Object1208, Object1209, Object1210, Object1211, Object1212, Object1213, Object1214, Object1215, Object1216, Object1217, Object1218
mapping_to	Object1309

<Model_Element_Set_Mapping_Type>
Object1308

Attribute/Reference	Literal/Instance
label	"UML many-to-many association with associative class to entity-relationship associative entity type with relationships mapping"
describes	Object1309

<Model_Element_Set_Mapping>
Object1309

Attribute/Reference	Literal/Instance
degree	"semantically_equivalent"
contained_by	Object1301
subject_set	Object1306
object_set	Object1307
described_by	Object1308

Figure A.5 — Registration of the mappings between the System A UML Class Diagram and the System B IDEF1X Model

A.3 Example 2 — Registration and mapping of two further information models

This example illustrates mapping between the information model used by System C, which is expressed using the Ellis-Barker entity-relationship notation (see Figure A.6), and an equivalent information model that is used by System D, which is expressed as a UML Class Diagram (see Figure A.7).

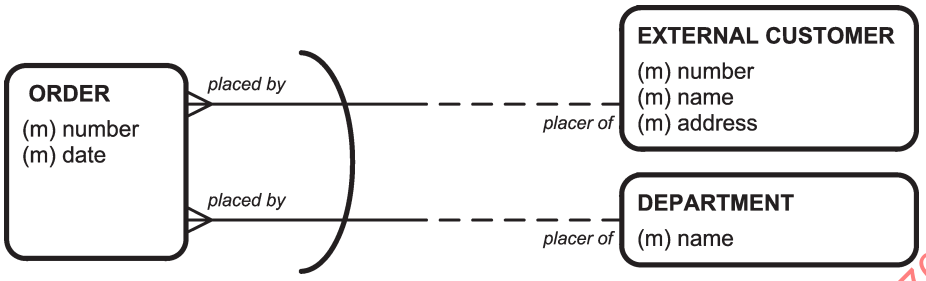


Figure A.6 — Example Ellis-Barker model for the Order Placement concept (as used by System C)

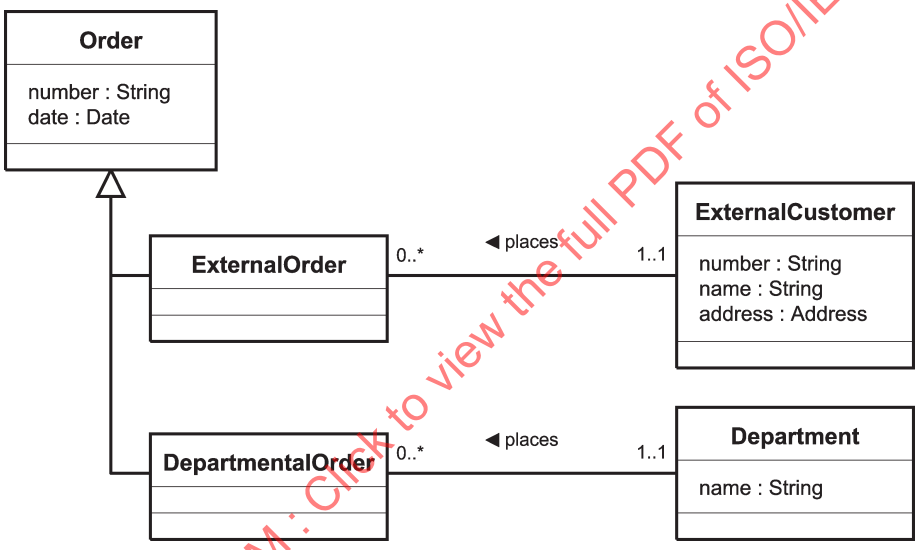


Figure A.7 — Example UML Class Diagram for the Order Placement concept (as used by System D)

The Ellis Barker model used by System C is registered using Object2101 to Object2115 as shown in Figure A.8 below. The UML Class Diagram used by System D is registered using Object2201 to Object2219 as shown in Figure A.9 below. The mapping of the Ellis Barker exclusive arc concept, and the associated relationships, to the UML Class Diagram equivalent which is the use of the **ExternalOrder** and **DepartmentalOrder** sub-classes of **Order** with associated associations is registered using Object2301 to Object2305 as shown in Figure A.10 below.

<Modelling_Language (Information_Modelling_Language)>
Object2101

Attribute/Reference	Literal/Instance
name	"Ellis Barker"
expressed_model	Object2102

<Model (Information_Model)>
Object2102

Attribute/Reference	Literal/Instance
name	"Order Placement Model C"
describing_language	Object2101
diagram_model_element	Object2103
mapping_from	Object2301

<Model_Element (Diagram)>
Object2103

Attribute/Reference	Literal/Instance
name	"Order Placement Model C - #1"
containing_model	Object2102
entity_type_model_element	Object2104, Object2105, Object2106
relationship_model_element	Object2107, Object2108

<Model_Element (Entity_Type)>
Object2104

Attribute/Reference	Literal/Instance
name	"ORDER"
containing_diagram	Object2103
relationship_end_group_ partial_description	Object2109

<Model_Element (Entity_Type)>
Object2105

Attribute/Reference	Literal/Instance
name	"EXTERNALCUSTOMER"
containing_diagram	Object2103
relationship_end_group_ partial_description	Object2110

<Model_Element (Entity_Type)>
Object2106

Attribute/Reference	Literal/Instance
name	"DEPARTMENT"
containing_diagram	Object2103
relationship_end_group_ partial_description	Object2111

<Model_Element (Relationship)>
Object2107

Attribute/Reference	Literal/Instance
containing_diagram	Object2103
contained_relationship_end	Object2112, Object2113
element_set	Object2302

<Model_Element (Relationship)>
Object2108

Attribute/Reference	Literal/Instance
containing_diagram	Object2103
contained_relationship_end	Object2114, Object2115
element_set	Object2302

<Model_Element (Relationship_End_Group)>
Object2109

Attribute/Reference	Literal/Instance
described_entity_type	Object2104
mutually_exclusive_component	Object2112, Object2114
element_set	Object2302

<Model_Element (Relationship_End_Group)>
Object2110

Attribute/Reference	Literal/Instance
described_entity_type	Object2105
mutually_exclusive_component	Object2113
element_set	Object2302

<Model_Element (Relationship_End_Group)>
Object2111

Attribute/Reference	Literal/Instance
described_entity_type	Object2106
mutually_exclusive_component	Object2115
element_set	Object2302

<Model_Element (Relationship_End)>
Object2112

Attribute/Reference	Literal/Instance
minimum_cardinality	"0"
maximum_cardinality	"*"
link_phrase	"placed by"
containing_relationship	Object2107
containing_group	Object2109
element_set	Object2302

<Model_Element (Relationship_End)>
Object2113

Attribute/Reference	Literal/Instance
minimum_cardinality	"1"
maximum_cardinality	"1"
link_phrase	"placer of"
containing_relationship	Object2107
containing_group	Object2110
element_set	Object2302

<Model_Element (Relationship_End)>
Object2114

Attribute/Reference	Literal/Instance
minimum_cardinality	"0"
maximum_cardinality	"*"
link_phrase	"placed by"
containing_relationship	Object2108
containing_group	Object2109
element_set	Object2302

<Model_Element (Relationship_End)>
Object2115

Attribute/Reference	Literal/Instance
minimum_cardinality	"1"
maximum_cardinality	"1"
link_phrase	"placer of"
containing_relationship	Object2108
containing_group	Object2111
element_set	Object2302

Figure A.8 — Registration of the System C Ellis-Barker Model for the Order concept

<Modelling_Language (Information_Modelling_Language)>
Object2201

Attribute/Reference	Literal/Instance
name	"UML Class Diagram"
expressed_model	Object2202

<Model (Information_Model)>
Object2202

Attribute/Reference	Literal/Instance
name	"Order Placement Model D"
describing_language	Object2201
diagram_model_element	Object2203
mapping_to	Object2301

<Model_Element (Diagram)>
Object2202

Attribute/Reference	Literal/Instance
name	"Order Placement Model D - #1"
containing_model	Object2202
entity_type_model_element	Object2204, Object2205, Object2206, Object2207, Object2208
relationship_model_element	Object2210, Object2211

<Model_Element (Entity_Type)>
Object2204

Attribute/Reference	Literal/Instance
name	"Order"
containing_diagram	Object2203
classification_scheme_ partial_description	Object2209

<Model_Element (Entity_Type)>
Object2205

Attribute/Reference	Literal/Instance
name	"ExternalOrder"
containing_diagram	Object2203
relationship_end_group_ partial_description	Object2212
containing_hierarchy	Object2209
element_set	Object2303

<Model_Element (Entity_Type)>
Object2206

Attribute/Reference	Literal/Instance
name	"DepartmentalOrder"
containing_diagram	Object2203
relationship_end_group_ partial_description	Object2214
containing_hierarchy	Object2209
element_set	Object2303

<Model_Element (Entity_Type)>
Object2207

Attribute/Reference	Literal/Instance
name	"ExternalCustomer"
containing_diagram	Object2203
relationship_end_group_ partial_description	Object2213

<Model_Element (Entity_Type)>
Object2208

Attribute/Reference	Literal/Instance
name	"Department"
containing_diagram	Object2203
relationship_end_group_ partial_description	Object2215

<Model_Element (Entity_Specialisation_Hierarchy)>
Object2209

Attribute/Reference	Literal/Instance
completeness_indicator	"Yes"
exclusivity_indicator	"Yes"
described_entity_type	Object2204
subtype_entity_type	Object2205, Object2206
element_set	Object2303

<Model_Element (Relationship)>
Object2210

Attribute/Reference	Literal/Instance
name	"places"
containing_diagram	Object2203
contained_relationship_end	Object2216, Object2217
element_set	Object2303

<Model_Element (Relationship)>
Object2211

Attribute/Reference	Literal/Instance
name	"places"
containing_diagram	Object2203
contained_relationship_end	Object2218, Object2219
element_set	Object2303

<Model_Element (Relationship_End_Group)>
Object2212

Attribute/Reference	Literal/Instance
described_entity_type	Object2205
mutually_exclusive_component	Object2216
element_set	Object2303

<Model_Element (Relationship_End_Group)>
Object2213

Attribute/Reference	Literal/Instance
described_entity_type	Object2207
mutually_exclusive_component	Object2217
element_set	Object2303

<Model_Element (Relationship_End_Group)>
Object2214

Attribute/Reference	Literal/Instance
described_entity_type	Object2206
mutually_exclusive_component	Object2218
element_set	Object2303

<Model_Element (Relationship_End_Group)>
Object2215

Attribute/Reference	Literal/Instance
described_entity_type	Object2208
mutually_exclusive_component	Object2219
element_set	Object2303

<Model_Element (Relationship_End)>
Object2216

Attribute/Reference	Literal/Instance
minimum_cardinality	"0"
maximum_cardinality	"*"
containing_relationship	Object2210
containing_group	Object2212
element_set	Object2303

<Model_Element (Relationship_End)>
Object2217

Attribute/Reference	Literal/Instance
minimum_cardinality	"1"
maximum_cardinality	"1"
containing_relationship	Object2210
containing_group	Object2213
element_set	Object2303

<Model_Element (Relationship_End)>
Object2218

Attribute/Reference	Literal/Instance
minimum_cardinality	"0"
maximum_cardinality	"*"
containing_relationship	Object2211
containing_group	Object2214
element_set	Object2303

<Model_Element (Relationship_End)>
Object2219

Attribute/Reference	Literal/Instance
minimum_cardinality	"1"
maximum_cardinality	"1"
containing_relationship	Object2211
containing_group	Object2215
element_set	Object2303

Figure A.9 — Registration of the System D UML Class Diagram for the Order concept

<Model_Mapping>
Object2301

Attribute/Reference	Literal/Instance
subject_model	Object2102
object_model	Object2202
contains	Object2305

<Model_Element_Set>
Object2302

Attribute/Reference	Literal/Instance
mapped_element	Object2107, Object2108, Object2109, Object2110, Object2111, Object2112, Object2113, Object2114, Object2115
mapping_from	Object2305

<Model_Element_Set>
Object2303

Attribute/Reference	Literal/Instance
mapped_element	Object2205, Object2206, Object2209, Object2210, Object2211, Object2212, Object2213, Object2214, Object2215, Object2216, Object2217, Object2218, Object2219
mapping_to	Object2305

<Model_Element_Set_Mapping_Type>
Object2304

Attribute/Reference	Literal/Instance
label	"Exclusive arc replaced by sub-classes"
describes	Object2305

<Model_Element_Set_Mapping>
Object2305

Attribute/Reference	Literal/Instance
degree	"semantically_equivalent"
contained_by	Object2301
subject_set	Object2302
object_set	Object2303
described_by	Object2304

Figure A.10 — Registration of the mappings between the System C Ellis-Barker Model and the System D UML Class Diagram

A.4 Example 3 — Registration and mapping of two process models

This example illustrates two examples of mapping between the process model used to explain the functionality of System E, which is expressed as a UML Activity Diagram (see Figure A.11), and an equivalent set of process models used to explain the functionality of System F, which is expressed using BPMN (see Figure A.12). In this case, the overall process is decomposed into two separate processes.

Figure A.11 — Example UML Activity Diagram for the Handle Order concept (as used by System E)

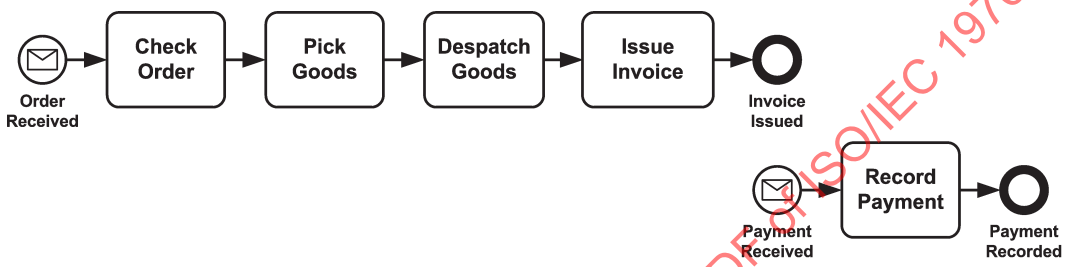


Figure A.12 — Example workflow model for the Handle Order concept (as used by System F)

The UML Activity Diagram used to explain the functionality of System E is registered using Object3101 to Object3109 as shown in Figure A.13 below. The set of BPMN models used to explain the functionality of System F is registered using Object3201 to Object3214 as shown in Figure A.14 below. The mappings are registered using Object3301 to Object3309 as shown in Figure A.15 below. The mapping of the UML **FULFILL ORDER** activity to the BPMN **CHECK ORDER**, **PICK GOODS** and **DESPATCH GOODS** activities is illustrated using Objects3302 to Object3305 and the mapping of the UML sequence dependency between the **ISSUE INVOICE** and **RECORD PAYMENT** activities in the System E model and the Invoice Issued and the Payment Received events in the System F model is registered using Object3306 to Object3309.

<Modelling_Language (Process_Modelling_Language)>
Object3101

Attribute/Reference	Literal/Instance
name	"UML Activity Diagram"
expressed_model	Object3102

<Model (Process_Model)>
Object3102

Attribute/Reference	Literal/Instance
name	"Handle Order Model E"
describing_language	Object3101
contained_process_model_element	Object3103, Object3104, Object3105, Object3108, Object3109
mapping_from	Object3301

<Model_Element (Process)>
Object3103

Attribute/Reference	Literal/Instance
name	"FULFILL ORDER"
containing_model	Object3102
trigger	Object3106
successor	Object3108
element_set	Object3302

<Model_Element (Process)>
Object3104

Attribute/Reference	Literal/Instance
name	"ISSUE INVOICE"
containing_model	Object3102
precedent	Object3108
successor	Object3109

<Model_Element (Process)>
Object3105

Attribute/Reference	Literal/Instance
name	"RECORD PAYMENT"
containing_model	Object3102
precedent	Object3109
produced_event	Object3107

<Model_Element (Event)>
Object3106

Attribute/Reference	Literal/Instance
name	"Order Received"
triggered_process	Object3103

<Model_Element (Event)>
Object3107

Attribute/Reference	Literal/Instance
name	"Payment Recorded"
producer	Object3105

<Model_Element (Sequence_Dependency)>
Object3108

Attribute/Reference	Literal/Instance
containing_model	Object3102
preceding_process	Object3103
following_process	Object3104

<Model_Element (Sequence_Dependency)>
Object3109

Attribute/Reference	Literal/Instance
containing_model	Object3102
preceding_process	Object3104
following_process	Object3105
element_set	Object3306

Figure A.13 — Registration of the System E UML Activity Diagram for the Handle Order concept

<Modelling_Language (Process_Modelling_Language)>
Object3201

Attribute/Reference	Literal/Instance
name	"BPMN 2.0"
expressed_model	Object3202

<Model (Process_Model)>
Object3202

Attribute/Reference	Literal/Instance
name	"Handle Order Model F"
describing_language	Object3201
contained_process_model_element	Object3203, Object3204, Object3205, Object3206, Object3209, Object3210, Object3211, Object3212
mapping_to	Object3301

<Model_Element (Process)>
Object3203

Attribute/Reference	Literal/Instance
name	"Check Order"
containing_model	Object3202
trigger	Object3207
successor	Object3209
element_set	Object3303

<Model_Element (Process)>
Object3204

Attribute/Reference	Literal/Instance
name	"Pick Goods"
containing_model	Object3202
precedent	Object3209
successor	Object3210
element_set	Object3303

<Model_Element (Process)>
Object3205

Attribute/Reference	Literal/Instance
name	"Despatch Goods"
containing_model	Object3202
precedent	Object3210
successor	Object3211
element_set	Object3303

<Model_Element (Process)>
Object3206

Attribute/Reference	Literal/Instance
name	"Issue Invoice"
containing_model	Object3202
precedent	Object3211
produced_event	Object3208

<Model_Element (Event)>
Object3207

Attribute/Reference	Literal/Instance
name	"Order Received"
triggered_process	Object3203

<Model_Element (Event)>
Object3208

Attribute/Reference	Literal/Instance
name	"Invoice Issued"
producer	Object3206
element_set	Object3307

<Model_Element (Sequence_Dependency)>
Object3209

Attribute/Reference	Literal/Instance
containing_model	Object3202
preceding_process	Object3203
following_process	Object3204
element_set	Object3303

<Model_Element (Sequence_Dependency)>
Object3210

Attribute/Reference	Literal/Instance
containing_model	Object3202
preceding_process	Object3204
following_process	Object3205
element_set	Object3303

<Model_Element (Sequence_Dependency)>
Object3211

Attribute/Reference	Literal/Instance
containing_model	Object3202
preceding_process	Object3205
following_process	Object3206

<Model_Element (Process)>
Object3212

Attribute/Reference	Literal/Instance
name	"Record Payment"
containing_model	Object3202
trigger	Object3213
produced_event	Object3214

<Model_Element (Event)>
Object3213

Attribute/Reference	Literal/Instance
name	"Payment Received"
triggered_process	Object3212
element_set	Object3307

<Model_Element (Event)>
Object3214

Attribute/Reference	Literal/Instance
name	"Payment Recorded"
producer	Object3212

Figure A.14 — Registration of the System F BPMN model for the Handle Order concept