

ISO/IEC JTC 1/SC 32 N

Date: 2012-04-18

ISO/IEC WD 19763-xx.02

ISO/IEC JTC 1/SC 32/WG 2N1640

Secretariat:

**Information technology — Metamodel Framework for Interoperability
(MFI) — Part xx: Metamodel for Fact Based Information Model
Registration**

Warning

This document is not an ISO International Standard. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an International Standard.

Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Copyright notice

This ISO document is a working draft or committee draft and is copyright-protected by ISO. While the reproduction of working drafts or committee drafts in any form for use by participants in the ISO standards development process is permitted without prior permission from ISO, neither this document nor any extract from it may be reproduced, stored or transmitted in any form for any other purpose without prior written permission from ISO.

Requests for permission to reproduce this document for the purpose of selling it should be addressed as shown below or to ISO's member body in the country of the requester:

[Indicate the full address, telephone number, fax number, telex number, and electronic mail address, as appropriate, of the Copyright Manger of the ISO member body responsible for the secretariat of the TC or SC within the framework of which the working document has been prepared.]

Reproduction for sales purposes may be subject to royalty payments or a licensing agreement.

Violators may be prosecuted.

Contents

Page

| | |
|--|----|
| Foreword | iv |
| Introduction..... | v |
| 1 Scope | 1 |
| 2 Conformance | 1 |
| 3 Normative references..... | 1 |
| 4 Terms and definitions | 2 |
| 5 Symbols (and abbreviated terms)..... | 7 |
| 6 Structure of MFI fact based information model registration..... | 8 |
| 6.1 Convention | 8 |
| 6.2 Overview of MFI fact based information model registration..... | 9 |
| 6.3 Relationship of metaclasses to the MDR Metamodel | 9 |
| 6.4 Metaclasses in MFI Fact Based Information Model Registration | 10 |
| 6.4.1 Fact_Based_Information_Modelling_Language | 10 |
| 6.4.2 Fact_Based_Information_Model | 10 |
| 6.4.3 Fact and fact type | 11 |
| 6.4.4 Object type and fact type's role | 12 |
| 6.4.5 Fact type derivation..... | 17 |
| 6.4.6 Predicate..... | 18 |
| 6.4.7 Fact reading | 22 |
| 6.4.8 Constraint..... | 23 |
| Bibliography..... | 36 |

Figures

| | |
|---|----|
| Figure 1 — Metamodel of MFI fact based information model registration | 9 |
| Figure 2 — Fact is instance of Fact Type..... | 11 |
| Figure 3 — Object Type and fact type's role..... | 12 |
| Figure 4 — Object Type | 12 |
| Figure 5 — Role | 14 |
| Figure 6 — Fact type derivation | 17 |
| Figure 7 — Predicate..... | 18 |
| Figure 8 — Fact reading..... | 22 |
| Figure 9 — Constraint | 23 |
| Figure 10 — Mandatory constraint | 24 |
| Figure 11 — Uniqueness constraint | 24 |
| Figure 12 — Frequency constraint | 25 |
| Figure 13 — Ring constraint..... | 26 |
| Figure 14 — Compared_Role_List_Occurrence..... | 27 |
| Figure 15 — Equality constraint | 29 |
| Figure 16 — Exclusion constraint | 29 |
| Figure 17 — Subset constraint..... | 30 |
| Figure 18 — Value comparison constraint | 31 |
| Figure 19 — Cardinality constraint | 32 |
| Figure 20 — Value constraint | 33 |
| Figure 21 — Ad hoc constraint | 35 |

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-xx was prepared by Joint Technical Committee ISO/IEC JTC 1, , Subcommittee SC 32, .

This second/third/... edition cancels and replaces the first/second/... edition (), [clause(s) / subclause(s) / table(s) / figure(s) / annex(es)] of which [has / have] been technically revised.

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

- *Part 1: Reference model*
- *Part 2: Core model and model mapping*
- *Part 3: Metamodel for ontology registration*
- *Part 4: Metamodel for model mapping*
- *Part 5: Metamodel for process model registration*
- *Part 6: Registry Summary*
- *Part 7: Metamodel for service registration*
- *Part 8: Metamodel for role and goal registration*
- *Part 9: On Demand Model Selection (ODMS) [Technical Report]*
- *Part 10: Core model and basic mapping*
- *Part 11: Advanced Mapping [Technical Report]*
- *Part 12: Metamodel for information model registration*
- *Part xx: Metamodel for Fact Based Information Model*

Introduction

There is an increasing demand for systems to interoperate by exchanging data. For these data exchanges to be meaningful it is essential that the business information requirements that are met by the data stored in these systems are understood so that suitable data exchange mechanisms can be developed.

Business information requirements, including the semantics of the information, are often represented by information models before the databases that are an integral part of the systems.

This part of ISO/IEC 19763 intends to provide a generic framework for registering these information models using the Fact Based Modelling approaches so that the mappings between models can be recognised and registered using the facilities specified in ISO/IEC 19763 Part 2.

In order to be within the spirit of interoperability of ISO/IEC WD 19763-xx, the conceptual schema (as defined in ISO TR9007) modeller needs to be given the option to select

- either a role based view or a variable based view,
- elementary or compound fact types, and
- nominalisation or avoid nominalisation.

This standard treats ISO TR9007 (1987) as its base. The combination of the ISO TR9007 100% principle and the ISO TR9007 Conceptualisation principle make a new dimension possible in conceptual modelling, namely the automatic transformation of a model expressed in one conceptual notation into another conceptual notation. Models that do not satisfy the ISO TR9007 principle cannot be automatically transformed as part of the model is implemented in the code, and will require other techniques to affect the transformation. ISO TR9007 opens the way to a productive approach: that is able to trap all errors as early as possible in the conceptual specification and have the subject matter experts involved in the validation as early as possible, long before program code development and testing. The ISO TR9007 based architecture permits true 100% conceptual modelling and transformation.

A short description of some concepts:

A variable based view is the view adopted by first order logic as well as ISO Common Logic (ISO/IEC 24707). This is the view that is closest to the subject matter expert and as such permits the best basis for communication with the subject matter expert.

The role based view is the view adopted by the modeller. The role based view prefers shorter names where the variable based view applies ready to use sentence patterns. The role view has a close affinity with the UML view, whereas the variable view has close affinity with the SQL view.

By giving an organisation or project the option to select between role based or variable based, conceptual modelling becomes a service to be selected, not a dictate.

Elementary fact types are preferred by some conceptual modellers while others prefer well-formed compound fact types. Both options can be seen as being equivalent with respect to semantics if properly modelled. Hence in this dimension the user can be given another option.

This option is the use of nominalisation or not. For example, the OWL (Web Ontology Language) does not use nominalisation. Other communities prefer to use nominalisation in order to minimize the distance between the model of the subject matter experts and the conceptual model.

Information technology — Metamodel Framework for Interoperability (MFI) — Part xx: Metamodel for Fact Based Information Model Registration

1 Scope

The primary purpose of the multipart standard ISO/IEC 19763 is to specify a metamodel framework for interoperability. This part of ISO/IEC 19763 specifies a metamodel for registering information models – Fact Based Models. This metamodel was developed taking into account information models developed using a number of different Fact Based Modelling diagramming techniques and notations.

These notations are:

- NIAM - Natural language Information Analysis Method
- CogNIAM - Cognition enhanced NIAM
- DOGMA - Developing Ontology-Grounded Methods and Applications
- FCO-IM - Fully Communication Oriented Information Modelling
- ORM – Object Role Modelling

An information model can be used to represent the information requirements that are met by a system. Where there is an overlap of the universe of discourse of two systems the information models using Fact Based Modelling for these two systems can be registered using the facilities specified by this part of ISO/IEC 19763. The mappings between these two models can then be registered using the facilities specified by Part 2 of ISO/IEC 19763. An interface between the two systems can then be designed, enabling the two systems to interoperate.

2 Conformance

To be developed

3 Normative references

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

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

ISO/IEC 19763-2, Information technology – Metamodel framework for interoperability (MFI) – Part 2: Core model and model mapping

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

ISO TR9007:1997 Concepts and Terminology for a Conceptual Schema

4 Terms and definitions

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

4.4

asserted fact

fact that is declared to be the case rather than being derived from other facts

SYNONYMS primitive fact, base fact, extensional fact

4.5

asserted fact type

fact type, each of whose instances is an asserted fact

SYNONYMS primitive fact type, base fact type, extensional fact type

4.7

atomic fact

fact that cannot be decomposed, without loss of information, into multiple facts involving exactly the same objects

NOTE An atomic fact is either an elementary fact or an existential fact. .

SYNONYM irreducible fact

4.8

atomic fact type

fact type, each of whose instances is an atomic fact

4.9

cardinality constraint

constraint on an object type or role that determines the possible number of instances that the object type or the role may contain for any given state of the fact base

NOTE 1 The full terms “object cardinality constraint” and “role cardinality constraint” should be used to distinguish them from cardinality constraints in other approaches (e.g. ER and OWL).

NOTE 2 The possible number of instances can be expressed by means of one or more ranges specified by their minimum and/or maximum values.

4.10

compatible object types

object types whose populations may have some instances in common

NOTE 1 Two object types are compatible if and only if one of the following applies: (1) the object types are the same, (2) one is a subtype of the other; (3) the object types share a common supertype and are not declared mutually exclusive.

NOTE 2 Value types based on the same data type are considered to be compatible value types.

4.11

compatible roles

roles whose associated object types are compatible

4.12**compound fact**

fact that can be decomposed without loss of information into multiple facts involving exactly the same objects

NOTE A compound fact is not atomic.

4.13**compound fact type**

fact type, each of whose instances is a compound fact

4.14**conceptual schema**

schema that specifies the possible and permitted populations and the possible and permitted transitions of the fact base, as well as an explanation of every term that could be misunderstood by the intended audience

NOTE A fact based conceptual schema declares the fact types, constraints, derivation rules, and concept definitions relevant to the universe of discourse.

4.15**conceptualization principle**

principle governing how the conceptual data model maps the universe of discourse

NOTE Conceptual data models include only conceptually relevant aspects, both static and dynamic, of the universe of discourse, thus excluding all aspects of (external or internal) data representation, physical data organization and access as well as all aspects of a particular external user representation such as message format, data structures, refer to ISO TC97/SC5/WG3 TR9007 (1987).

4.16**constraint**

restriction on what populations or transitions of the fact base are possible for or permitted by the conceptual schema

4.17**constraint arity**

number of roles in a sequence of roles

4.20**derivation rule**

rule that specifies how to derive instances of a derived fact type or semiderived fact type from other facts

4.22**derived fact**

fact that is deduced from other facts by means of a derivation rule

NOTE A fact that is not derived is an asserted fact.

4.23**derived fact type**

fact type, each of whose instances is a derived fact

4.29**equality constraint**

set-comparison constraint that specifies that, for each state of the fact base, the populations of the constrained sequences of role occurrences must be equal

4.30**exclusion constraint**

set-comparison constraint that specifies that, for each state of the fact base, the populations of the constrained sequences of role occurrences must be mutually exclusive i.e. do not overlap

4.31

exclusive-or constraint

combination of an inclusive-or constraint and an exclusion constraint over the same set of roles

SYNONYM XOR constraint

4.34

fact

proposition that is taken to be true by the relevant community whereby the proposition is represented by a predicate with an associated ordered set of objects

NOTE A fact is neither a constraint nor a derivation rule.

4.35

fact base

set of facts of interest to the relevant community and that satisfies the rules of the conceptual schema

NOTE The set of asserted facts is called the extensional database. The set of derived facts is called the intentional database.

4.36

fact reading

sentence expressing a fact

4.37

fact type

type, each of whose instances are facts that express the same kind of information

NOTE 1 A fact type is a non-empty set of typed predicates, i.e. predicates where each placeholder is applied to a specific object type, with the same semantics.

NOTE 2 Each instance of a fact type includes one object for each role.

4.40

frequency constraint

constraint that restricts, for each state of the fact base, the number of times any given sequence of objects that instantiate the constrained role sequence appears in the population of that role sequence

NOTE 1 The allowed number of instances can be expressed by means of one or more ranges, i.e. minimum and/or maximum values.

NOTE 2 A uniqueness constraint should always be used instead of a frequency constraint having a maximum frequency of 1.

4.43

inclusive-or constraint

constraint over two or more roles played by the same object type or by compatible object types, which specifies that all instances of the object type(s) must play at least one of the constrained roles

EXAMPLE disjunctive mandatory role constraint (mandatory constraint over a logical disjunction of two or more roles)

4.44

instance

member of a specified type

4.45

mandatory role

role that must be played by all instances in the population of the role's associated object type

SYNONYM total role

4.46

mandatory role constraint

constraint requiring that each instance in the population of a given object type must play the constrained role or at least one of the constrained roles

NOTE 1 A mandatory role constraint applying to one role only is also known as simple mandatory role constraint.

NOTE 2 A mandatory role constraint applying to more than one role is also known as disjunctive mandatory role constraint or inclusive-or constraint.

4.47

non-lexical object

object that is referenced by a definite description that relates it to other objects

NOTE An non-lexical object can typically change its state over time (e.g. by participating in new facts). Not a lexical constant, such as a name or numeral.

EXAMPLE 1 The Country named "Australia".

EXAMPLE 2 The President named "Barack Obama".

4.48

non-lexical object type

object type, each of whose instances are non-lexical objects

4.49

object

ground level single thing, individual or value, about which the community wants to exchange facts

NOTE An object is either an non-lexical object or a lexical object. It cannot be both.

EXAMPLES the person named "Barack Obama", the country code "AU"

4.50

object type

concept used to classify the objects in the universe of discourse into different kinds

NOTE Object type is either an non-lexical object type or a lexical object type.

4.51

objectification

process of making an object from a fact in order to talk about the state of affairs that corresponds to, but is not identical to, the fact

NOTE 1 Propositional nominalization is not supported. Propositional nominalization is the use of a noun phrase to refer to the proposition expressed by a declarative sentence. For example, in the sentence "I know that it's snowing outside", the noun phrase "that it's snowing outside" is used to nominalize the proposition expressed by the sentence "It's snowing outside 1".

NOTE 2 Situational nominalization is the use of a noun phrase to refer to the state of affairs denoted by another linguistic expression. For example, in the sentence "That snowing is beautiful", the noun phrase "that snowing" is used to nominalize the state of affairs referenced in the sentence "It's snowing outside".

4.52

objectified fact type

object type resulting from the objectification of a fact type

4.53

population

set of instances in a given state of the fact base, of an object type, of a fact type, or a role under discussion

NOTE The population of a type may change over time.

4.54

predicate

result of replacing the objects of interest in a proposition by placeholders for those objects

NOTE 1 A predicate is denoted by a declarative sentence with the object terms of interest replaced by placeholders.

NOTE 2 Predicate is a semantic, not a syntactic construct.

SYNONYM logical predicate

4.55

predicate reading

sentence with placeholders for object(s) that expresses a logical predicate

NOTE A predicate must have at least one predicate reading.

EXAMPLES "... employs ...", "... was born in ...", "... played ... for ..."

4.58

proposition

that which is asserted by a declarative sentence

NOTE A proposition is true or false, but not both.

4.62

ring constraint

logical constraint between two type-compatible role occurrences that specifies how the populations of these role occurrences may be related

EXAMPLES irreflexive, asymmetric, intransitive, antisymmetric, acyclic ring constraints.

4.63

role

constituting part of a fact type that indicates the function played by the instances of the associated object type

4.64

role occurrence

instance of navigating through the role of a fact type

EXAMPLE The derived fact type definition "Person1 is a grandparent of Person2 if and only if Person1 is a parent of some Person3 who is a parent of Person2" involves a defining role path that includes two occurrences of each of the roles in the fact type "Person is a parent of Person".

4.65

semiderived fact type

fact type that may have a population that includes some asserted facts and some derived facts

4.67

set-comparison constraint

constraint that specifies a condition to be satisfied when comparing the populations of compatible role-sequences

NOTE 1 The set-comparison constraints are: "subset constraint", "equality constraint" and "exclusion constraint".

NOTE 2 Compatible role-sequences imply that the role occurrences across matching positions in the sequences must be compatible.

4.68**subset constraint**

set-comparison constraint that specifies that, for each state of the fact base, the population of a sequence of one or more role occurrences must be a subset of the population of another compatible sequence of role occurrences

4.69**subtype**

object type, each of whose instances belong to an encompassing type

EXAMPLE Woman is a (proper) subtype of Person.

4.70**supertype**

object type that has at least one subtype and may have some instances not in that subtype

4.71**uniqueness constraint**

constraint over a sequence of one or more roles that requires that in each state of the fact base, each instantiation of that role sequence occurs only once

4.72**value**

object that identifies itself and is unchangeable

EXAMPLES 12, "Amsterdam"

SYNONYMS lexical object, label

4.73**value comparison constraint**

constraint that specifies how the values of instances of two roles with co-roles played by the same object are related by one of the following comparison operators: $<$, \leq , $>$, \geq

4.74**value constraint**

constraint that specifies the possible values for the instances of a role or a value type

4.76**value type**

object type, each of whose instances must be values

5 Symbols (and abbreviated terms)

CogNIAM cognition enhanced natural language information analysis method

FBM fact based modelling

FCO-IM fully communication oriented information modelling

IT information technology

NIAM natural language information analysis method

| | |
|------|---|
| ORM | object role modelling |
| OWL | web ontology language |
| SBVR | semantics of business vocabulary and business rules |
| UoD | universe of discourse |
| UUID | universal unique identifier |

6 Structure of MFI fact based information model registration

6.1 Convention

The metamodel for the registration of fact based information models is developed using the industry standard diagramming techniques and notations listed in Clause 1 above.

For each metaclass the following detail is shown:

- A definition that describes the role or significance of instances of the metaclass.
- The name of its immediate supertype.
- Any alternative names (synonyms or aliases) for the metaclass.
- A list of attributes.
- A list of references.

For each attribute the following detail is shown:

- The name of the attribute; where the attribute is one that is provided by the type defined in the MDR metamodel by which the instances of the metaclass are extended the name is italicised.
- The datatype for values of the attribute.
- The multiplicity of the attribute.
- A description that describes the role or significance of values of the attribute.

For each reference the following detail is shown:

- The name of the reference; this is the role name that describes the role played by the referenced metaclass with respect to the association identified by this reference.
- The name of the referenced metaclass.
- The multiplicity of the reference.
- A description that describes the role or significance of the instance, or instances, of the referenced metaclass with respect to an instance of this metaclass.
- The name of the reference in the referenced metaclass that provides the inverse definition for the association.

An indication as to whether this metaclass is responsible for the maintenance of the association, i.e. the precedence of the metaclass with respect to the association.

6.2 Overview of MFI fact based information model registration

The following figure shows an overview of the metamodel for the registration of fact based information models.

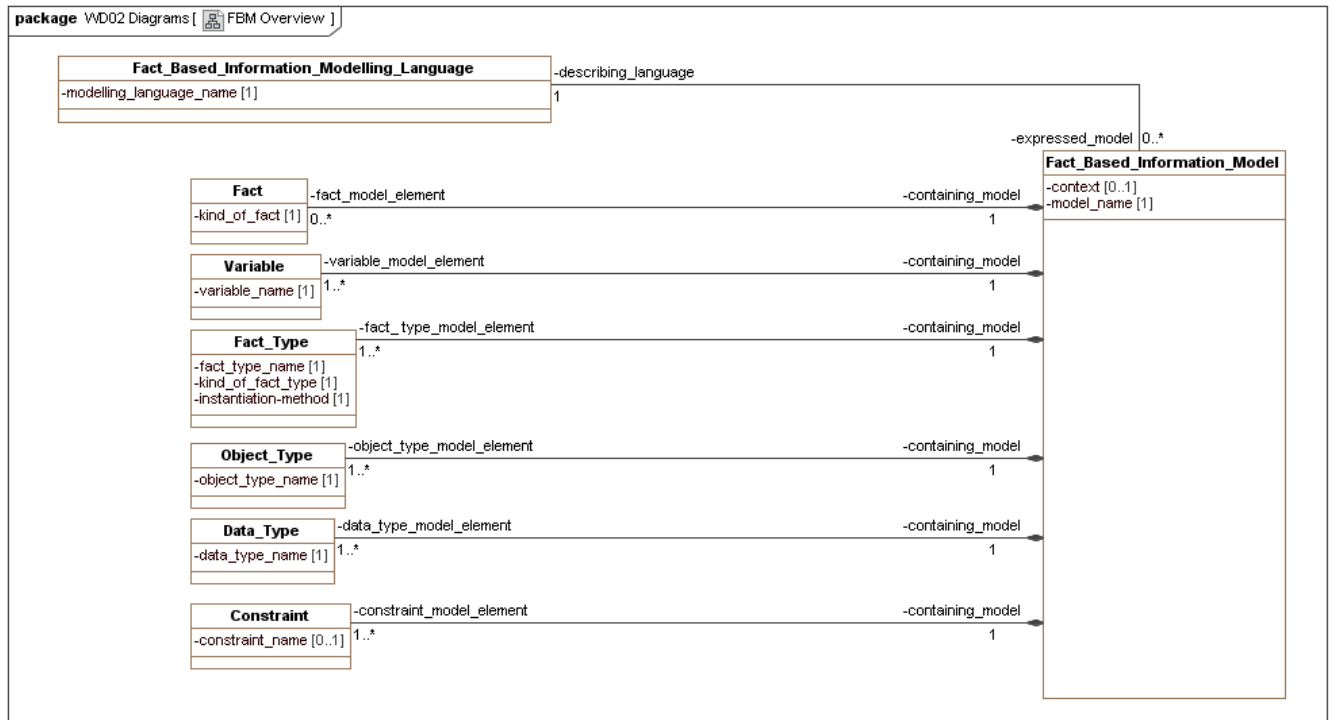


Figure 1 — Metamodel of MFI fact based information model registration

The metamodel for fact based information model registration comprises the following main metaclasses:

- Fact_Based_Information_Modelling_Language
- Fact_Based_Information_Model
- Fact
- Variable
- Fact_Type
- Object_Type
- Data_Type
- Constraint

The detailed specification of each main metaclass and their relationships are provided in 6.4 below.

6.3 Relationship of metaclasses to the MDR Metamodel

As explained in ISO/IEC 19763 Part 2, instances of the metaclasses defined in this part of ISO/IEC 19763 may be extended by the types defined in the MDR Metamodel as follows:

- Fact_Based_Information_Modelling_Language may be extended as an Identified_Item and as a Designatable_Item.
- Fact_Based_Information_Model may be extended as an Administered_Item and as a Designatable_Item.
- Fact may be extended as an Attached_Item and as a Designatable_Item.
- Variable may be extended as an Attached_Item and as a Designatable_Item.
- Fact_Type may be extended as an Attached_Item and as a Designatable_Item.
- Object_Type may be extended as an Attached_Item and as a Designatable_Item.
- Data_Type may be extended as an Attached_Item and as a Designatable_Item.
- Constraint may be extended as an Attached_Item and as a Designatable_Item.

6.4 Metaclasses in MFI Fact Based Information Model Registration

6.4.1 Fact_Based_Information_Modelling_Language

Fact_Based_Information_Modelling_Language is a metaclass each instance of which represents a language or notation that is used to model information requirements in line with a fact based modelling method.

Superclass

Modelling_Language (defined in MFI Core and Model Mapping)

| Attribute | Data Type | Multiplicity | Description |
|-----------|-----------|--------------|--|
| Name | String | 1..1 | A unique name by which this information modelling language is known. |

| Reference | Class | Multiplicity | Description | Inverse | Precedence |
|-----------------|-------------------|--------------|--|---------------------|------------|
| expressed_model | Information_Model | 0..* | The set of information models that are expressed in this language. | describing_language | No |

6.4.2 Fact_Based_Information_Model

Fact_Based_Information_Model is a metaclass each instance of which represents a set of definitions for things of significance to an organisation, about which information needs to be held, and the relationships between them.

Superclass

Model (defined in MFI Core and Model Mapping)

Aliases

Data_Model, Conceptual_Data_Model, Fact_Based_Data_Model

| Attribute | Data Type | Multiplicity | Description |
|-----------|-----------|--------------|---|
| Name | String | 1..1 | A unique name by which this information model is known. |
| Context | String | 0..1 | A description of the universe of discourse covered by this information model. |

| Reference | Class | Multiplicity | Description | Inverse | Precedence |
|---------------------|--------------------------------|--------------|--|-----------------|------------|
| describing_language | Information_Modelling_Language | 1..1 | The information modelling language in which this model is expressed. | expressed_model | Yes |

6.4.3 Fact and fact type

6.4.3.1 Overview

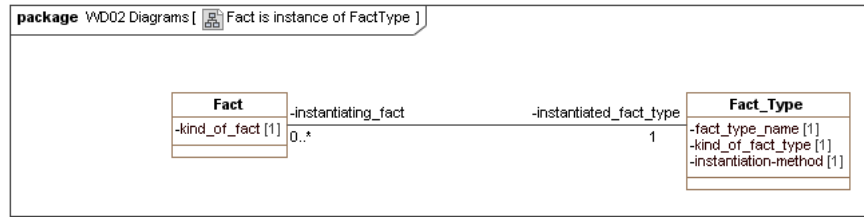


Figure 2 — Fact is instance of Fact Type

6.4.3.1.1 Fact

Fact is a metaclass each instance of which represents a proposition taken to be true.

Superclass

Model_Element (defined in MFI Core and Model Mapping)

| Attribute | Data Type | Multiplicity | Description |
|--------------|-----------|--------------|---|
| kind_of_fact | string | 1..1 | specification whether or not the fact is an atomic fact or a compound fact. |

| Reference | Class | Multiplicity | Description | Inverse | Precedence |
|-------------------------|------------------|--------------|---|--------------------|------------|
| containing_model | Fact_Based_Model | 1..1 | The fact based model which includes the fact | fact_model_element | |
| instantiated_fact_type | Fact_Type | 1..1 | The fact type for which the fact is an instance | instantiating_fact | |
| expressing_fact_reading | Fact_Reading | 1..* | The fact reading used to express the fact | expressed_fact | |

6.4.3.2 Fact_Type

Fact_Type is a metaclass each instance of which represents a fact type to be used to express facts of the same kind.

Superclass

Model_Element (defined in MFI Core and Model Mapping)

| Attribute | Data Type | Multiplicity | Description |
|-------------------|-----------|--------------|--|
| fact_type_name | String | 0..1 | A name by which the fact type is alternatively known |
| kind_of_fact_type | String | 1..1 | Specification whether or not the fact type is an atomic fact type or a compound fact type. |

| Reference | Class | Multiplicity | Description | Inverse | Precedence |
|-----------------------|--------------------|--------------|---|-----------------------------|------------|
| containing_model | Fact_Based_Model | 1..1 | The fact based model which includes the fact type | Fact_type_ Model_element | |
| compounding_fact_type | Compound_Fact_Type | 0..* | The compound fact type of which the fact type is | compounded_ fact_type | |

| | | | | |
|---------------------------|-------------------|------|--|------------------------|
| objectifying_object_type | Object_Type | 0..1 | part Object type that is the result of objectification of the fact type | objectified_fact_type |
| instantiating_Fact | Fact | 0..* | facts that are instances of the fact type | instantiated_fact_type |
| owned_predicate | Predicate | 1..* | predicate corresponding to one role ordering in the fact type | owning_fact_type |
| contained_role | Role | 1..* | role that is part of the fact type | containing_fact_type |
| applied_ad_hoc_constraint | Ad_Hoc_Constraint | 0..* | Ad hoc constraint restricting the fact type | owning_fact_type |

6.4.4 Object type and fact type's role

6.4.4.1 Overview

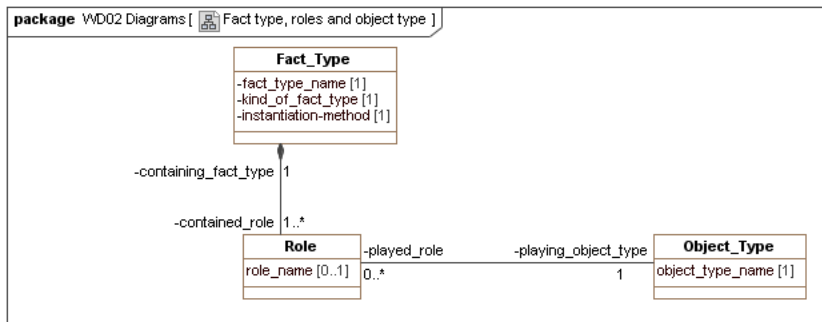


Figure 3 — Object Type and fact type's role

6.4.4.2 Object_Type

6.4.4.2.1 Overview

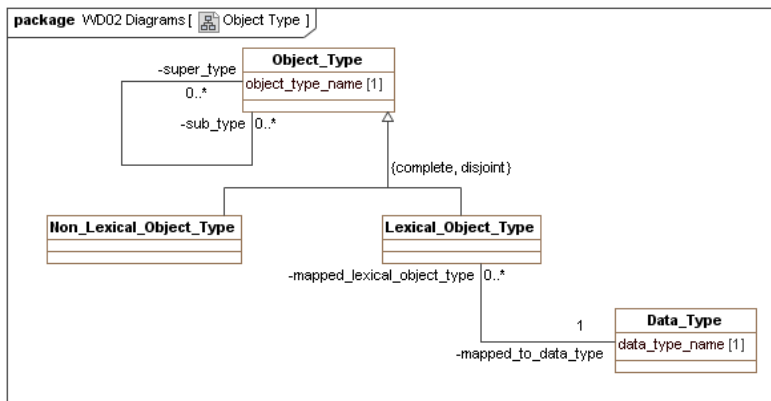


Figure 4 — Object Type

6.4.4.2.2 General

Object_Type is a metaclass each instance of which represents a concept that is used to classify the object in the UoD into different kinds.

Superclass

Model_Element (defined in MFI Core and Model Mapping)

Subclasses

Non_Lexical_ObjectType, Lexical_Object_Type

| Attribute | DataType | Multiplicity | Description | | |
|--|--------------------------------------|--------------|---|-------------------------------|------------|
| object_type_ name | String | 1..1 | Name of an object type, used to reference the object type. | | |
| Reference | Class | Multiplicity | Description | Inverse | Precedence |
| containing_model | Fact_Based_ Information_ Model | 1..1 | Model to which the object type belongs | object_type_ model_element | |
| played_role | Role | 0..* | Role played by the object type | Playing_ object_type | |
| objectified_fact_ type | Fact_Type | 0..1 | Fact type that is objectified by the object type | objectifying_ object_type | |
| super_type | Object_Type | 0..* | object type of which the object type under consideration is a subtype | sub_type | |
| sub_type | Object_Type | 0..* | object type that is a specialization of the object type under consideration | super_type | |
| restricting_value_ constraint | Value_Constraint | 0..1 | value constraint restricting the object type | restricted_object_ type | |
| restricting_cardinality_ constraint | Cardinality_ Constraint | 0..1 | Cardinality constraint restricting the object type | restricted_object_ type | |
| applied_ad-hoc_ constraint | Ad-Hoc_ Constraint | 0..* | Ad hoc constraint restricting the object type | owning_ object_type | |

6.4.4.2.3 Non_Lexical_Object_Type

Non_Lexical_Object_Type is a metaclass each instance of which represents a concept that is used to represent entities.

Superclass

Object_Type

| Attribute | DataType | Multiplicity | Description | | |
|-----------|----------|--------------|-------------|---------|------------|
| [none] | | | | | |
| Reference | Class | Multiplicity | Description | Inverse | Precedence |
| [none] | | | | | |

6.4.4.2.4 Lexical_Object_Type

Lexical_Object_Type is a metaclass each instance of which represents a concept that is used to represent values.

Superclass

Object_Type

| Attribute | Data Type | Multiplicity | Description |
|-----------|-----------|--------------|-------------|
|-----------|-----------|--------------|-------------|

[none]

| Reference | Class | Multiplicity | Description | Inverse | Precedence |
|---------------------|-----------|--------------|-------------------------------------|----------------------------|------------|
| mapped_to_data_type | Data_Type | 1..1 | Data type associated with the value | mapped_lexical_object_type | |

6.4.4.2.5 Data_Type

Data_Type is a metaclass each instance of which represents a format specification.

Superclass

Model_Element (defined in MFI Core and Model Mapping)

| Attribute | Data Type | Multiplicity | Description |
|----------------|-----------|--------------|-----------------------|
| data_type_name | string | 1..1 | name of the data type |

| Reference | Class | Multiplicity | Description | Inverse | Precedence |
|----------------------------|------------------------------|--------------|---|-------------------------|------------|
| containing_model | Fact_Based_Information_Model | 1..1 | model in which the data type is defined | data_type_model_element | |
| mapped_lexical_object_type | Lexical_Object_Type | 0..* | Lexical object type that uses the data type | mapped_to_data_type | |

6.4.4.3 Role

6.4.4.3.1 Overview

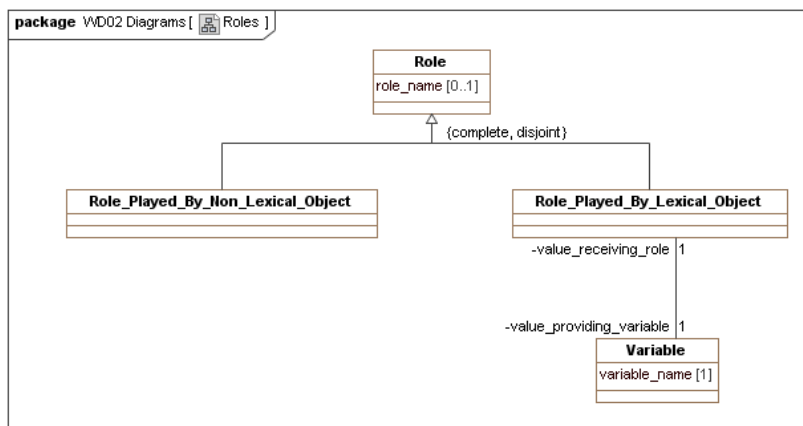


Figure 5 — Role

6.4.4.3.2 General

Role is a metaclass each instance of which represents a constituting part of a fact type which indicates the function played by the instances of the associated object type.

Superclass

Model_Element (defined in MFI Core and Model Mapping)

Subclasses

Role_Played_By_Lexical_Object_Type; Role_Played_By_Non_Lexical_Object_Type

| Attribute | DataType | Multiplicity | Description | | |
|--|--|---------------------|--|------------------------|-------------------|
| role_name | string | 0..1 | name of the role | | |
| Reference | Class | Multiplicity | Description | Inverse | Precedence |
| containing_fact_type | Fact_Type | 1..1 | fact type to which the role belongs | contained_role | |
| playing_object_type | Object_Type | 1..1 | object type that plays the role | played_role | |
| positioning_role_position | Role_in_Predicate | 1..* | position of a role in a predicate | positioned_role | |
| positioning_role_position2 | Role_in_Predicate_Reading | 1..* | position of a role in a predicate | positioned_role | |
| restricting_mandatory_constraint | Mandatory_Constraint | 0..* | mandatory constraint restricting the role | restricted_role | |
| restricting_role_position | Role_Position_In_Uniqueness_Constraint | 0..* | position, taken by the role in a uniqueness constraint | restricted_role | |
| restricting_frequency_constraint | Frequency_Constraint | 0..* | frequency constraints restricting the role | restricted_role | |
| restricting_ring_constraint1 | Ring_Constraint | 0..* | ring constraint restricting the role | first_restricted_role | |
| restricting_ring_constraint2 | Ring_Constraint | 0..* | ring constraint restricting the role | second_restricted_role | |
| restricting_value_constraint | Value_Constraint | 0..1 | value constraint restricting the role | restricted_role | |
| including_included_role_position | Included_role_Position | 0..* | xx | included_role | |
| restricting_cardinality_constraint | Cardinality_Constraint | 0..1 | Cardinality constraint restricting the role | restricted_role | |
| comparing_value_comparison_constraint1 | Value_Comparison_Constraint | 0..* | value comparison constraint restricting the role | first_compared_role | |

| | | | | |
|--|---------------------------------|------|---|--------------------------|
| comparing_value_ comparison_ constraint2 | Value_Comparison_ Constraint | 0..* | value comparison constraint restricting the role | second_compared _role |
| applied_ad_hoc_ constraint | Ad_Hoc_Constraint | 0..* | Ad constraint restricting the role | hoc owning_role |

6.4.4.3.3 Role_Played_By_Non_Lexical_Object_Type

Role_Played_By_Lexical_Object_Type is a metaclass each instance of which represents a constituting part of a fact type which indicates the function played by the instances of a non lexical object type

Superclass

Role

| Attribute | DataType | Multiplicity | Description |
|-----------|----------|--------------|-------------|
|-----------|----------|--------------|-------------|

[none]

| Reference | Class | Multiplicity | Description | Inverse | Precedence |
|-----------|-------|--------------|-------------|---------|------------|
|-----------|-------|--------------|-------------|---------|------------|

[none]

6.4.4.3.4 Role_Played_By_Lexical_Object_Type

Role_Played_By_Lexical_Object_Type is a metaclass each instance of which represents a constituting part of a fact type which indicates the function played by the instances of a lexical object type

Superclass

Role

| Attribute | DataType | Multiplicity | Description |
|-----------|----------|--------------|-------------|
|-----------|----------|--------------|-------------|

[none]

| Reference | Class | Multiplicity | Description | Inverse | Precedence |
|-----------|-------|--------------|-------------|---------|------------|
|-----------|-------|--------------|-------------|---------|------------|

| | | | | | |
|------------------------------|----------|------|--|----------------------|--|
| value_providing_ variable | Variable | 1..1 | Variable that is used to populate the role | value_receiving_Role | |
|------------------------------|----------|------|--|----------------------|--|

6.4.4.3.5 Variable

Variable is a metaclass each instance of which represents a position in a fact type that can only be instantiated with a lexical object.

Superclass

Model_Element (defined in MFI Core and Model Mapping)

| Attribute | DataType | Multiplicity | Description |
|-----------|----------|--------------|-------------|
|-----------|----------|--------------|-------------|

| | | | |
|---------------|--------|------|----------------------|
| variable_name | string | 1..1 | name of the variable |
|---------------|--------|------|----------------------|

| Reference | Class | Multiplicity | Description | Inverse | Precedence |
|-----------|-------|--------------|-------------|---------|------------|
|-----------|-------|--------------|-------------|---------|------------|

| | | | | | |
|------------------------------|----------------------------------|------|--|------------------------|--|
| containing_model | Fact_Based_ information_Model | 1..1 | Model containing the variable | variable_model_element | |
| value_receiving_ variable | Role | 1..1 | Role which gets its values from the variable | value_providing_Role | |

6.4.5 Fact type derivation

6.4.5.1 Overview

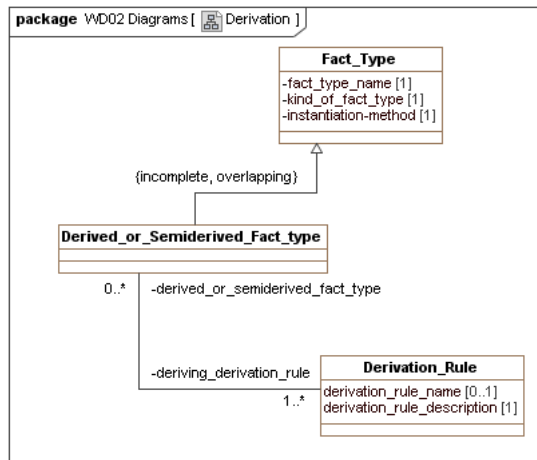


Figure 6 — Fact type derivation

6.4.5.2 Derived_or_Semiderived_Fact_type

Derived_or_Semiderived_Fact_type is a metaclass each instance of which represents a fact type whose instances are either derived or asserted

Superclass

Fact_Type

| Attribute | Data Type | Multiplicity | Description |
|----------------------|-----------|--------------|---|
| instantiation_method | string | 1..1 | specification of the manner in which the fact type is populated. Possible values are derived, semi-derived or asserted. |

| Reference | Class | Multiplicity | Description | Inverse | Precedence |
|--------------------------|-----------------|--------------|--|----------------------------------|------------|
| deriving_derivation_rule | Derivation_Rule | 1..* | Derivation rules that are used to derive the instances of the derived or semiderived fact type | derived_or_semiderived_fact_type | |

6.4.5.3 Derivation_Rule

Derivation_Rule is a metaclass each instance of which represents a rule that is used to derive the instances of a derived or semiderived fact type

Superclass

Model_Element (defined in MFI Core and Model Mapping)

| Attribute | Data Type | Multiplicity | Description |
|-----------------------------|-----------|--------------|---|
| derivation_rule_name | string | 0..1 | name of the derivation rule |
| derivation_rule_description | string | 1..1 | description (informal) of the derivation rule |

| Reference | Class | Multiplicity | Description | Inverse | Precedence |
|-----------|-------|--------------|-------------|---------|------------|
|-----------|-------|--------------|-------------|---------|------------|

| | | | | |
|----------------------------------|---------------------------------|------|---|-------------------------------|
| containing_model | Fact_Based_Information_Model | 1..1 | information model to which the derivation rule belongs | derivation_rule_model_element |
| derived_or_semiderived_fact_type | Derived_or_Semiderived_FactType | 0..* | fact type whose instances are derived by means of the derivation rule | deriving_derivation_rule |

6.4.6 Predicate

6.4.6.1 overview

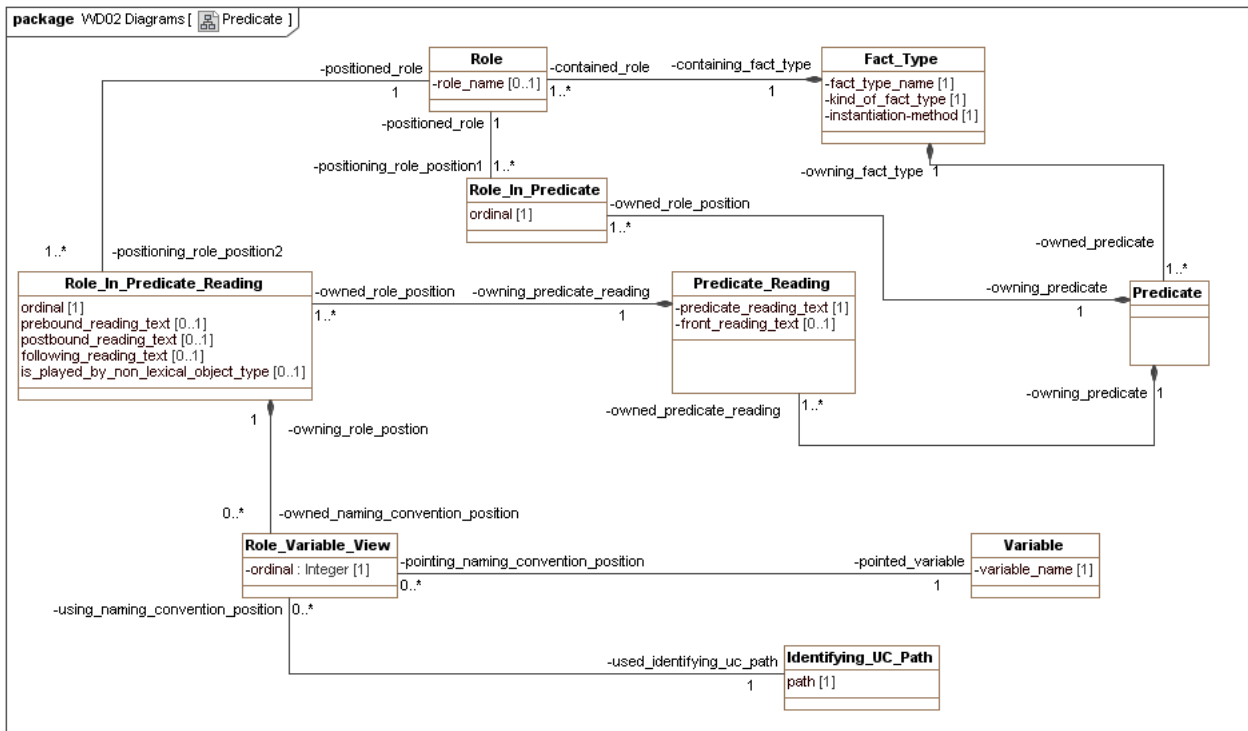


Figure 7 — Predicate

6.4.6.2 Predicate

Predicate is a metaclass each instance of which represents an ordering of roles of a fact type

Superclass

Model_Element (defined in MFI Core and Model Mapping)

| Attribute | Data Type | Multiplicity | Description | Inverse | Precedence |
|-------------------------|-------------------|--------------|---|------------------|------------|
| [None] | | | | | |
| owning_fact_type | Fact_Type | 1..1 | Fact type to which the predicate belongs | owned_predicate | |
| owned_predicate_reading | Predicate_Reading | 1..* | Predicate reading used to express the predicate | owning_predicate | |

following_reading_text string 0..1 reading text behind the last role in the predicate reading.

is_played_by_non_lexical_object_type Boolean 1..1 specifies whether or not the role in the predicate reading is played by a non-lexical object type.

| Reference | Class | Multiplicity | Description | Inverse | Precedence |
|----------------------------------|--------------------|--------------|--|----------------------------|------------|
| positioned_role | Role | 1..1 | role that is positioned within the predicate reading | positioning_role_position2 | |
| owned_naming_convention_position | Role_Variable_View | 0..* | position of a variable within a role within a predicate reading | owning_role_position | |
| owning_predicate_reading | Predicate_Reading | 1..1 | predicate reading to which the role in predicate reading belongs | owned_role_position | |

6.4.6.6 Role_Variable_View

Role_Variable_View is a metaclass each instance of which represents the order of the variables which correspond to the role.

Superclass

Model_Element (defined in MFI Core and Model Mapping)

| Attribute | Data Type | Multiplicity | Description |
|-----------|-----------|--------------|--------------------------------------|
| ordinal | integer | 1..1 | position of the variable in the role |

| Reference | Class | Multiplicity | Description | Inverse | Precedence |
|--------------------------|---------------------------|--------------|---|-------------------------------------|------------|
| owning_role_position | Role_In_Predicate_Reading | 1..1 | role that owns the position of a variable | owned_naming_convention_position | |
| pointed_variable | Variable | 1..1 | variable that is referenced at that position within a role | pointing_naming_convention_position | |
| used_identifying_uc_path | Identifying_UC_Path | 1..1 | identification path used for identifying the lexical object type playing the role. This identification path determines which variables are required for population the role | using_naming_convention_position | |

6.4.6.7 Variable

Variable is a metaclass each instance of which represents a position in a fact type that can only be instantiated with a lexical object.

Superclass

Model_Element (defined in MFI Core and Model Mapping)

| Attribute | DataType | Multiplicity | Description | | |
|-------------------------------------|------------------------------|--------------|--|------------------------|------------|
| variable_name | string | 1..1 | name of the variable | | |
| Reference | Class | Multiplicity | Description | Inverse | Precedence |
| pointing_naming_convention_position | Role_Variable_View | 0..* | position within a predicate reading that references the variable | pointed_variable | |
| containing_model | Fact_Based_Information_Model | 1..1 | information model that contains the variable | variable_model_element | |

6.4.6.8 Identifying_UC_Path

Identifying_UC_Path is a metaclass each instance of which represents an identification path that is used to specify which variables are used to populate the roles played by a non-lexical object type.

Superclass

Model_Element (defined in MFI Core and Model Mapping)

| Attribute | DataType | Multiplicity | Description | | |
|----------------------------------|------------------------------|--------------|---|------------------------------------|------------|
| Path | string | 1..1 | Description of the path, in terms of roles | | |
| Reference | Class | Multiplicity | Description | Inverse | Precedence |
| using_naming_convention_position | Role_Variable_View | 0..* | position within a predicate reading that uses the identifying uc path | used_identifying_uc_path | |
| containing_model | Fact_Based_Information_Model | 1..1 | the fact base modelling model that contains the identifying uc path | identifying_ucC_path_model_element | |

6.4.7 Fact reading

6.4.7.1 Overview

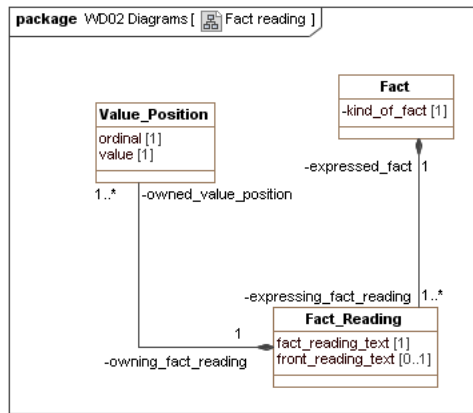


Figure 8 — Fact reading

6.4.7.2 Fact_Reading

Fact_Reading is a metaclass each instance of which represents a reading for a fact

Superclass

Model_Element (defined in MFI Core and Model Mapping)

| Attribute | DataType | Multiplicity | Description | | |
|----------------------|----------------|--------------|---|-------------------------|------------|
| fact_reading_text | string | 1..1 | text used to express the fact | | |
| front_reading_text | string | 0..1 | text before the variable in the fact reading | | |
| Reference | Class | Multiplicity | Description | Inverse | Precedence |
| expressed_fact | Fact | 1..1 | fact for which the fact reading is used to express the fact | expressing_fact_reading | |
| owned_value_position | Value_Position | 1..* | value, on a position, given to | owning_fact_reading | |

6.4.7.3 Value_Position

Value_Position is a metaclass each instance of which represents the position of a value in a fact reading

Superclass

Model_Element (defined in MFI Core and Model Mapping)

| Attribute | DataType | Multiplicity | Description | | |
|---------------------|--------------|--------------|---|-------------------------|------------|
| ordinal | integer | 1..1 | position of the value | | |
| value | string | 1..1 | the value itself on a specific position | | |
| Reference | Class | Multiplicity | Description | Inverse | Precedence |
| owning_fact_reading | Fact_Reading | 1..1 | Fact reading to which the | expressing_fact_reading | |

6.4.8 Constraint

6.4.8.1 General

6.4.8.1.1 Overview

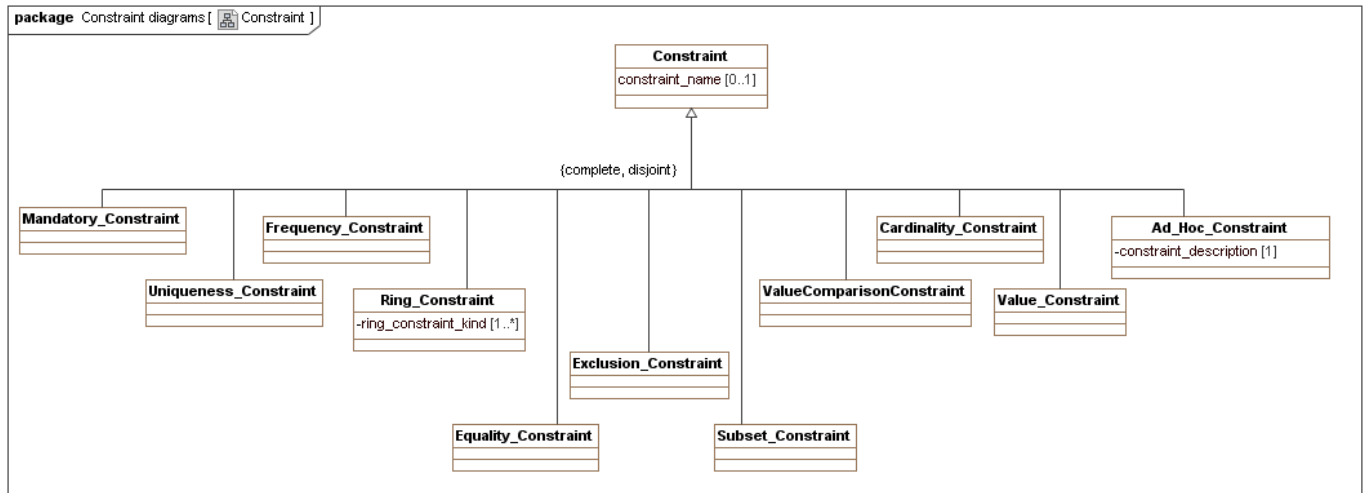


Figure 9 — Constraint

6.4.8.1.2 Constraint

Constraint is a metaclass each instance of which represents a means to restrict the values of a role of a fact type

Superclass

Model_Element (defined in MFI Core and Model Mapping)

Subclasses

Mandatory_Constraint, Uniqueness_Constraint, Frequency_Constraint, Ring_Constraint, Equality_Constraint, Exclusion_Constraint, Subset_Constraint, Value_Comparison_Constraint, Cardinality_Constraint, Value_Constraint, Ad_Hoc_Constraint

| Attribute | Data Type | Multiplicity | Description |
|-----------------|-----------|--------------|------------------------|
| Constraint_name | String | 0..1 | Name of the constraint |

| Reference | Class | Multiplicity | Description | Inverse | Precedence |
|------------------|------------------------------|--------------|---------------------------------------|--------------------------|------------|
| containing_model | Fact_Based_Information_Model | 1..1 | model to which the constraint belongs | constraint_model_element | |

6.4.8.2 Mandatory_Constraint

6.4.8.2.1 Overview

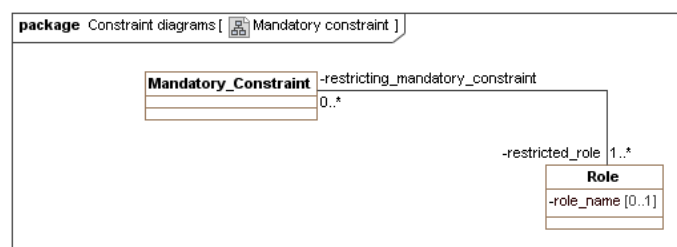


Figure 10 — Mandatory constraint

6.4.8.2.2 Definition

Mandatory Constraint is a metaclass each instance of which represents a means to specify that each instance of the object type playing the role restricted by the mandatory constraint, must play that role

Superclass

Constraint

| Attribute | Data Type | Multiplicity | Description | Inverse | Precedence |
|-----------------------------------|----------------------|---------------------|---|-----------------------------------|-------------------|
| [none] | | | | | |
| Reference | Class | Multiplicity | Description | Inverse | Precedence |
| restricted_role | Role | 1..* | Role restricted by the mandatory constraint | restricting_mandatory_constraint | |
| exclusive-or_exclusion_constraint | Exclusion_Constraint | 0..1 | exclusive-or constraint | exclusive-or_mandatory_constraint | |

6.4.8.3 Uniqueness_Constraint

6.4.8.3.1 Overview

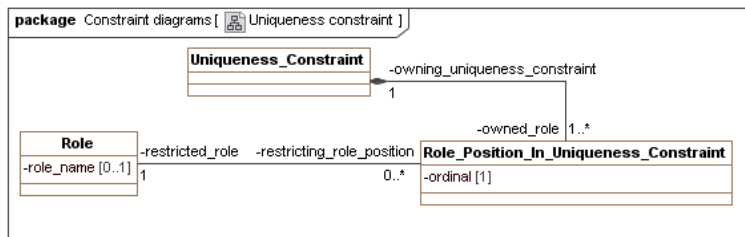


Figure 11 — Uniqueness constraint

6.4.8.3.2 General

Uniqueness Constraint is a metaclass each instance of which represents a means to specify a constraint over one or more roles that requires that in each state of the fact based, each instantiation of that role occurs only once.

Superclass

Constraint

| Attribute | Data Type | Multiplicity | Description | Inverse | Precedence |
|------------------|--|---------------------|-------------------------------------|------------------------------|-------------------|
| [none] | | | | | |
| Reference | Class | Multiplicity | Description | Inverse | Precedence |
| owned_role | Role_Position_In_Uniqueness_Constraint | 1..* | reference to the role in a position | owning_uniqueness_constraint | |

6.4.8.3.3 Role_Position_In_Uniqueness_Constraint

Role_Position_In_Uniqueness_Constraint is a metaclass each instance of which represents a role in a certain position in the sequence of roles

Superclass

Model_Element (defined in MFI Core and Model Mapping)

| Attribute | DataType | Multiplicity | Description |
|-----------|----------|--------------|-------------|
| Ordinal | integer | 1..1 | position |

| Reference | Class | Multiplicity | Description | Inverse | Precedence |
|------------------------------|-----------------------|--------------|---|---------------------------|------------|
| owning_uniqueness_constraint | Uniqueness_Constraint | 1..1 | uniqueness constraint to which the role sequence belongs | owned_role | |
| restricted_role | Role | 1..1 | Role which is restricted by the uniqueness constraint in a certain position | restricting_role_position | |

6.4.8.4 Frequency_Constraint

6.4.8.4.1 Overview

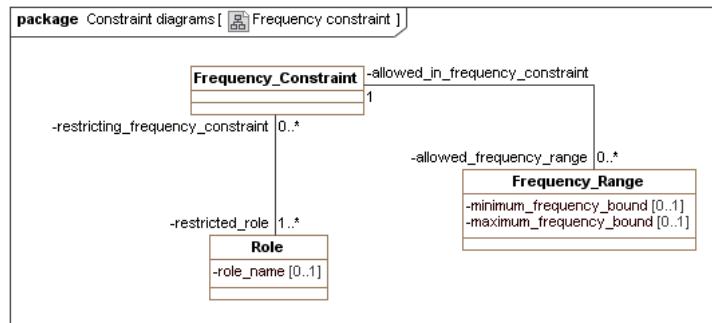


Figure 12 — Frequency constraint

6.4.8.4.2 General

Frequency_Constraint is a metaclass each instance of which represents a constraint that restricts, for each state of the fact base, the number of times any given sequence of objects that instantiate the constrained role(s) appears in the population of these roles.

Superclass

Constraint

| Attribute | DataType | Multiplicity | Description |
|--------------------------|----------|--------------|--|
| discrete_frequency_bound | integer | 0..* | discrete value that specifies of allowed number of times the sequence of objects is allowed. |

| Reference | Class | Multiplicity | Description | Inverse | Precedence |
|-----------------|-------|--------------|---|----------------------------------|------------|
| restricted_role | Role | 1..* | role restricted by the frequency constraint | restricting_frequency_constraint | |

| | | | | |
|-----------------------------|-----------------|------|---|--------------------------------------|
| allowed_frequency_ range | Frequency_Range | 0..* | statement of the interval that specifies the allowed number of times the sequence of objects is allowed. Any value provided in the frequency range (including minimum and maximum) is the allowed number of times | allowed_ frequency_ constraint |
|-----------------------------|-----------------|------|---|--------------------------------------|

6.4.8.4.3 Frequency_Range

Frequency_Range is a metaclass each instance of which represents an interval specifying the minimum and/or maximum number of times, including all in-between number of times, any given sequence of objects that instantiate the constrained roles appear in the population of these roles

Superclass

Model_Element (defined in MFI Core and Model Mapping)

| Attribute | Data Type | Multiplicity | Description |
|-----------------------------|-----------|--------------|--------------------------------------|
| minimum_frequency_ bound | integer | 0..1 | minimum value of the frequency range |
| maximum_frequency_ bound | integer | 0..1 | maximum value of the frequency range |

| Reference | Class | Multiplicity | Description | Inverse | Precedence |
|-------------------------------------|--------------------------------------|--------------|---|---------------------------------------|------------|
| containing_model | Fact_Based_ Information_ Model | 1..1 | model to which the frequency range is used | frequency_ range_ model_element | |
| allowed_in_frequency_ constraint | Frequency_ Constraint | 1..1 | frequency constraint to which the frequency range belongs | frequency_range_ model_element | |

6.4.8.5 Ring_Constraint

6.4.8.5.1 Overview

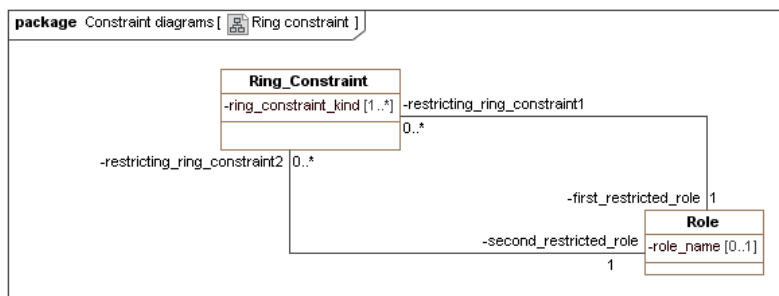


Figure 13 — Ring constraint

6.4.8.5.2 General

Ring_Constraint is a metaclass each instance of which represents a logical constraint between two type-compatible role occurrences that specifies how the populations of these role occurrences may be related

Superclass

Constraint

| Attribute | Data Type | Multiplicity | Description |
|----------------------|-----------|--------------|--|
| ring_constraint_kind | string | 1..* | type of the ring constraint. Possible values are: irreflexive, anti-symmetric, a-symmetric, transitive, acyclic, strongly-intransitive, reflexive, purely-reflexive, symmetric, intransitive |

| Reference | Class | Multiplicity | Description | Inverse | Precedence |
|------------------------|-------|--------------|---|------------------------------|------------|
| first_restricted_role | Role | 1..* | first role restricted by the ring constraint | restricting_ring_constraint1 | |
| second_restricted_role | Role | 1..* | second role restricted by the ring constraint | restricting_ring_constraint2 | |

6.4.8.6 Set comparison constraints

6.4.8.6.1 General

6.4.8.6.1.1 Overview

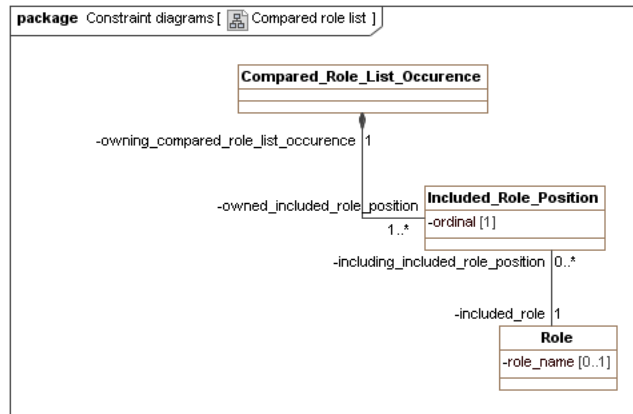


Figure 14 — Compared_Role_List_Occurrence

6.4.8.6.1.2 Compared_Role_List_Occurrence

Compared_Role_List_Occurrence is a metaclass each instance of which represents a sequence of roles

Superclass

Model_Element (defined in MFI Core and Model Mapping)

| Attribute | Data Type | Multiplicity | Description |
|-----------|-----------|--------------|-------------|
| [none] | | | |

| Reference | Class | Multiplicity | Description | Inverse | Precedence |
|--------------------------|------------------------------|--------------|--|---|------------|
| containing_model | Fact_Based_Information_Model | 1..1 | model in which the compared role list occurrence is defined | contained_compared_role_list_occurrence | |
| subset_subset_constraint | Subset_Constraint | 0..1 | subset constraint which uses the compared role list occurrence as a subset sequence of roles | subset_compared_role_list_occurrence | |

| | | | | |
|------------------------------------|------------------------|------|--|--|
| superset_subset_ constraint | Subset_Constraint | 0..1 | subset constraint which uses the compared role list occurrence as superset sequence of roles | superset_compared_ role_list_occurrence |
| comparing_equality_ constraint | Equality_Constraint | 0..1 | Equality constraint which uses the compared role list occurrence as sequences of roles that have to be equal | compared_compared_ role_list_occurrence |
| comparing_exclusion_ constraint | Exclusion_Constraint | 0..1 | Exclusion constraint which uses the compared role list occurrences as sequences of roles that have to exclude each other | compared_compared_ role_list_occurrence |
| owned_included_ role_position | Included_Role_Position | 1..* | reference to the role that is included in a specific position in the role list occurrence | owning_compared_ role_list_occurrence |

6.4.8.6.1.3 Included_Role_Position

Included_Role_Position is a metaclass each instance of which represents a role in a certain position, included in the compared role list occurrence

Superclass

Model_Element (defined in MFI Core and Model Mapping)

| Attribute | DataType | Multiplicity | Description | | |
|--|-----------------------------------|--------------|---|--------------------------------------|------------|
| Ordinal | integer | 1..1 | Position of the role | | |
| Reference | Class | Multiplicity | Description | Inverse | Precedence |
| included_role | Role | 1..1 | role included in the position in a role list occurrence | including_ included_role_position | |
| owning_compared_ role_list_ occurrence | Compared_Role_ List_Occurrence | 1..1 | Compared role list occurrence that includes the role position | owned_included_ role_position | |

6.4.8.6.2 Equality_Constraint

6.4.8.6.2.1 Overview

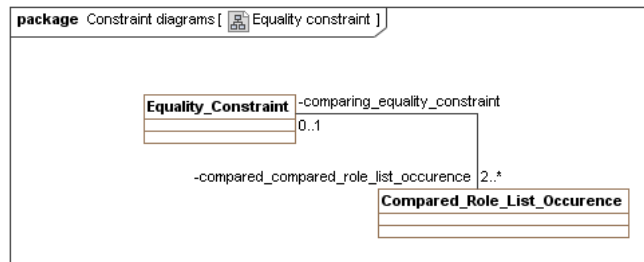


Figure 15 — Equality constraint

6.4.8.6.2.2 General

Equality_Constraint is a metaclass each instance of which represents a constraint that specifies that for each state of the fact base, the population of the constrained sequences of compared role occurrences must be equal

Superclass

Constraint

| Attribute | Data Type | Multiplicity | Description | Inverse | Precedence |
|--|-------------------------------|--------------|---|-------------------------------|------------|
| [none] | | | | | |
| compared_compared_role_list_occurrence | Compared_Role_List_Occurrence | 2..* | role lists for which the values of the roles contained in the role list are compared in the equality constraint | comparing_equality_constraint | |

6.4.8.6.3 Exclusion_Constraint

6.4.8.6.3.1 Overview

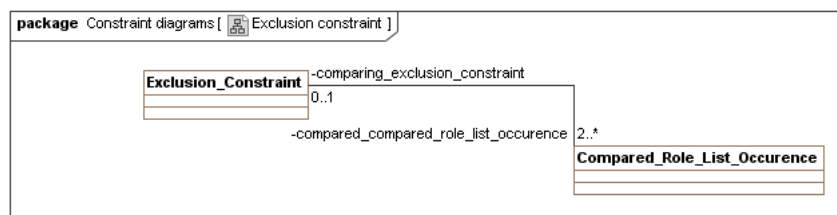


Figure 16 — Exclusion constraint

6.4.8.6.3.2 General

Exclusion_Constraint is a metaclass each instance of which represents a constraint that specifies that for each state of the fact base, the populations of the constrained compared role list occurrences have to be mutually exclusive, i.e., do not overlap

Superclass

Constraint

| Attribute | Data Type | Multiplicity | Description | Inverse | Precedence |
|-----------|-----------|--------------|-------------|---------|------------|
|-----------|-----------|--------------|-------------|---------|------------|

[none]

| Reference | Class | Multiplicity | Description | Inverse | Precedence |
|--|-----------------------------------|--------------|---|--|------------|
| compared_compared_ role_list_occurrence | Compared_Role_ List_Occurrence | 2..* | role lists for which the values of the roles contained in the role list are compared by means of the exclusion constraint | comparing_ exclusion_ constraint | |

6.4.8.6.4 Subset_Constraint

6.4.8.6.4.1 Overview

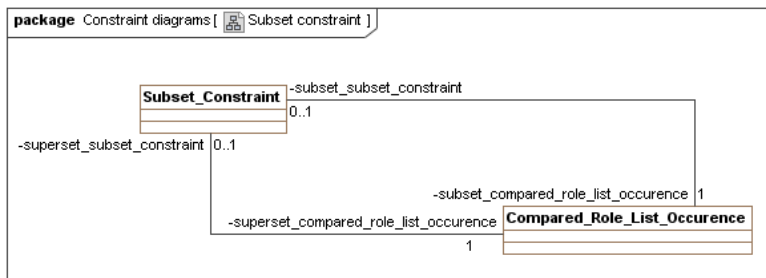


Figure 17 — Subset constraint

6.4.8.6.4.2 General

Subset_Constraint is a metaclass each instance of which represents a constraint that specifies that, for each state of the fact base, the population of a sequence of one or more compared role list occurrences must be a subset of the population of another compatible sequence of compared role list occurrences.

Superclass

Constraint

| Attribute | DataType | Multiplicity | Description |
|-----------|----------|--------------|-------------|
|-----------|----------|--------------|-------------|

[none]

| Reference | Class | Multiplicity | Description | Inverse | Precedence |
|--|-----------------------------------|--------------|--|--------------------------------|------------|
| subset_compared_ role_list_occurrence | Compared_Role_ List_Occurrence | 1..1 | Role list containing the roles that form the subset of the subset constraint | subset_subset_ constraint | |
| superset_compared_ role_list_occurrence | Compared_Role_ List_Occurrence | 1..1 | Role list containing the roles that form the superset of the subset constraint | superset_ subset_constraint | |

6.4.8.7 Value_Comparison_Constraint

6.4.8.7.1 Overview

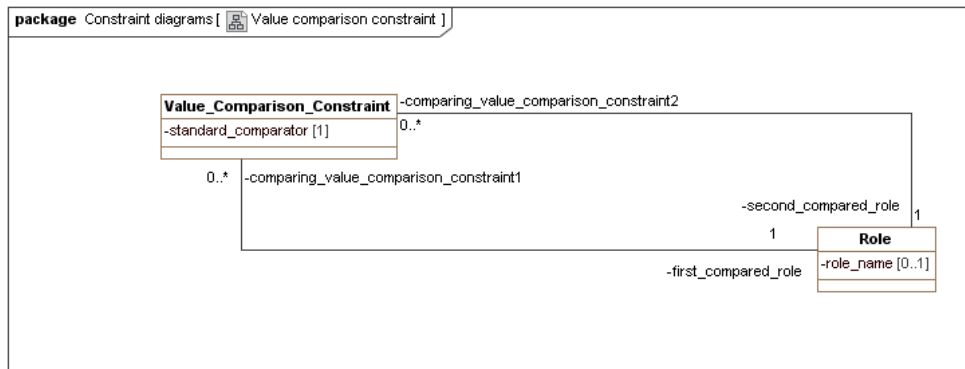


Figure 18 — Value comparison constraint

6.4.8.7.2 General

Value_Comparison_Constraint is a metaclass each instance of which represents a constraint that specifies how the values of instances of two roles, with co-roles played by the same object, are related by one of the standard comparison operators.

Superclass

Constraint

| Attribute | DataType | Multiplicity | Description |
|---------------------|----------|--------------|--|
| standard_comparator | string | 1..1 | comparison operator that is by standard, provided for comparing the instances of two roles. Possible values are: "<", "<=", ">", ">=". |

| Reference | Class | Multiplicity | Description | Inverse | Precedence |
|----------------------|-------|--------------|---|---|------------|
| first_compared_role | Role | 1..1 | one of the two roles restricted by the value comparison constraint | comparing_value_ comparison_constraint1 | |
| second_compared_role | Role | 1..1 | second of the two roles restricted by the value comparison constraint | comparing_value_ comparison_constraint2 | |

6.4.8.8 Cardinality_Constraint

6.4.8.8.1 Overview

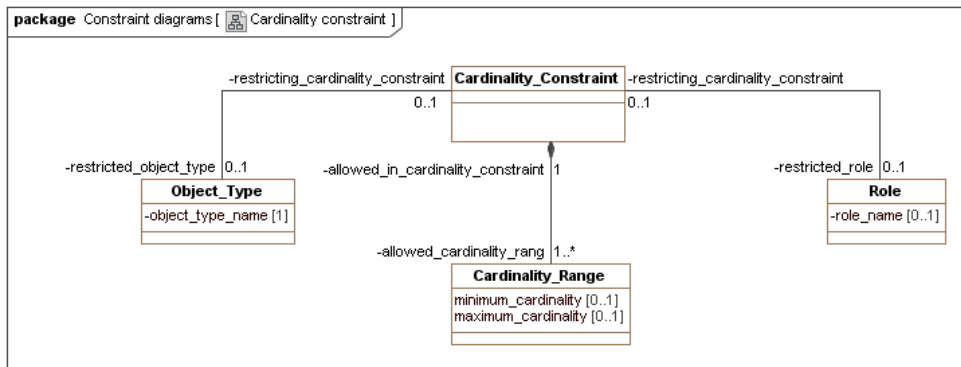


Figure 19 — Cardinality constraint

6.4.8.8.2 General

Cardinality_Constraint is a metaclass each instance of which represents a constraint on an object type or role that determines the possible number of instances that the object type or role may contain for any given state of the fact base.

Superclass

Constraint

| Attribute | DataType | Multiplicity | Description | Inverse | Precedence |
|---------------------------|-------------------|--------------|---|------------------------------------|------------|
| [none] | | | | | |
| restricted_role | Role | 0..1 | role restricted by the cardinality constraint | restricting_cardinality_constraint | |
| restricted_object_type | Object_Type | 0..1 | Object type restricted by the cardinality constraint | restricting_cardinality_constraint | |
| allowed_cardinality_range | Cardinality_Range | 1..* | reference to the range that specifies the allowed cardinalities | allowed_in_cardinality_constraint | |

6.4.8.8.3 Cardinality_Range

Cardinality_Range is a metaclass each instance of which represents a range, specified by means of a minimum and maximum, used by a cardinality constraint

Superclass

Model_Element (defined in MFI Core and Model Mapping)

| Attribute | DataType | Multiplicity | Description | Inverse | Precedence |
|---------------------|------------------------------|--------------|--|---------------------------------|------------|
| minimum_cardinality | integer | 0..1 | value that provides the minimum of the cardinality range | | |
| maximum_cardinality | integer | 0..1 | value that provides the maximum of the cardinality range | | |
| containing_model | Fact_Based_Information_Model | 1..1 | model containing the cardinality range | cardinality_range_model_element | |

| | | | | |
|------------------------|--------------|------|--|-----------------|
| allowed_in_ | Cardinality_ | 1..1 | cardinality | onwed_included_ |
| cardinality_constraint | Constraint | | constraint that uses the cardinality range | role_position |

6.4.8.9 Value_Constraint

6.4.8.9.1 Overview

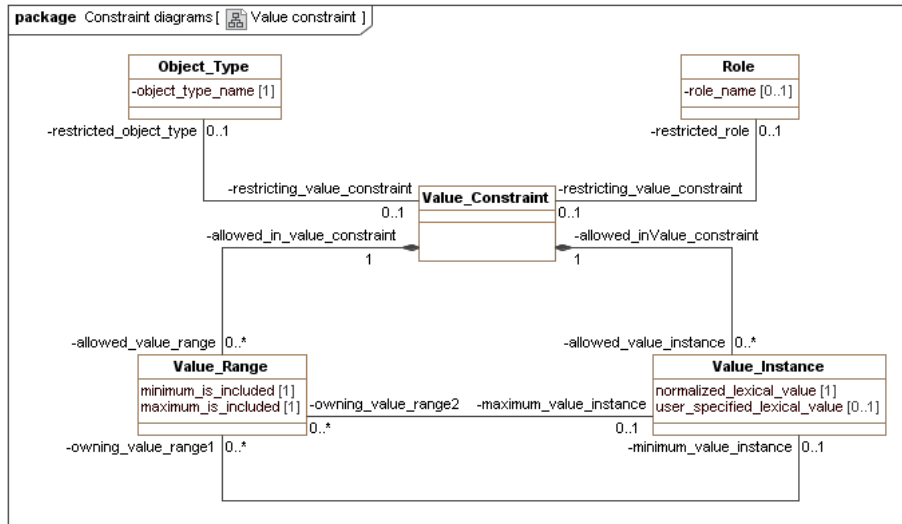


Figure 20 — Value constraint

6.4.8.9.2 General

Value_Constraint is a metaclass each instance of which represents a constraint that specifies the allowed possible values for the instances of a role or a lexical object type.

Superclass

Constraint

| Attribute | DataType | Multiplicity | Description | Inverse | Precedence |
|------------------------|----------------|--------------|--|------------------------------|------------|
| [none] | | | | | |
| restricted_role | Role | 0..1 | role restricted by the value constraint | restricting_value_constraint | |
| restricted_object_type | ObjectType | 0..1 | object type restricted by the value constraint | restricting_value_constraint | |
| allowed_value_range | Value_Range | 0..* | Value range containing the values that are allowed | allowed_in_value_constraint | |
| allowed_value_instance | Value_Instance | 0..* | Value instances that are allowed | allowed_inValue_constraint | |

6.4.8.9.3 Value_Range

Value_Range is a metaclass each instance of which represents an interval containing values that are allowed in the value range

Superclass

Model_Element (defined in MFI Core and Model Mapping)

| Attribute | Data Type | Multiplicity | Description |
|---------------------|-----------|--------------|---|
| minimum_is_included | Boolean | 1..1 | specifies whether or not the minimum value of the range is an allowed value (included in the range) |
| maximum_is_included | Boolean | 1..1 | specifies whether or not the maximum value of the range is an allowed value (included in the range) |

| Reference | Class | Multiplicity | Description | Inverse | Precedence |
|-----------------------------|------------------------------|--------------|---|---------------------------|------------|
| containing_model | Fact_Based_Information_Model | 1..1 | model to which the value range belongs | value_range_model_element | |
| minimum_value_instance | Value_Instance | 0..1 | value instance that is the minimum value of the range | owning_value_range1 | |
| maximum_value_instance | Value_Instance | 0..1 | value instance that is the maximum value of the range | owning_value_range2 | |
| allowed_in_value_constraint | Value_Constraint | 0..1 | Value constraint containing the value range | allowed_value_range | |

6.4.8.9.4 Value_Instance

Value_Instance is a metaclass each instance of which represents a value that is allowed in accordance to the associated value constraint

Superclass

Model_Element (defined in MFI Core and Model Mapping)

| Attribute | Data Type | Multiplicity | Description |
|------------------------------|-----------|--------------|-------------|
| normalized_lexical_value | string | 1..1 | xx |
| user_specified_lexical_value | string | 0..1 | xx |

| Reference | Class | Multiplicity | Description | Inverse | Precedence |
|-----------------------------|------------------------------|--------------|--|------------------------------|------------|
| containing_model | Fact_Based_Information_Model | 1..1 | model to which the value range belongs | value_instance_model_element | |
| owning_value_range1 | Value_Range | 0..* | value range for which the value instance specifies the minimum value | minimum_value_instance | |
| owning_value_range2 | Value_Range | 0..* | value range for which the value instance specifies the maximum value | maximum_value_instance | |
| allowed_in_value_constraint | Value_Constraint | 0..1 | Value constraint containing the value instance | allowed_value_instance | |

6.4.8.10 Ad_Hoc_Constraint

6.4.8.10.1 Overview

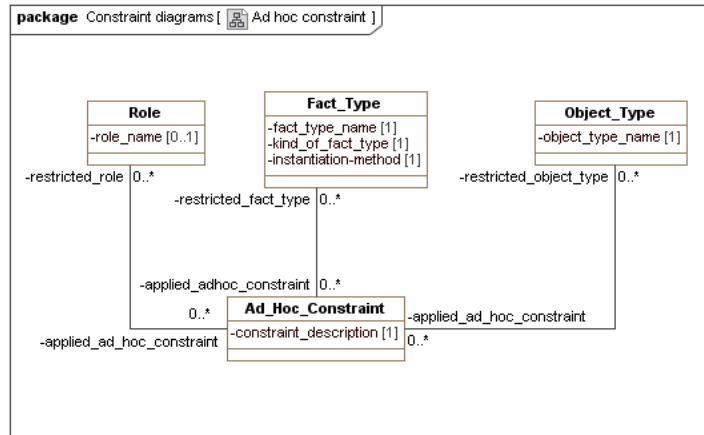


Figure 21 — Ad hoc constraint

6.4.8.10.2 Definition

Ad_Hoc_Constraint is a metaclass each instance of which represents a constraint on a role, fact type or object type by means of an informal description

Superclass

Constraint

| Attribute | DataType | Multiplicity | Description |
|------------------------|----------|--------------|----------------------|
| constraint_description | string | 1..1 | informal description |

| Reference | Class | Multiplicity | Description | Inverse | Precedence |
|--------------------|-------------|--------------|---|---------------------------|------------|
| owning_object_type | Object_Type | 0..* | object type restricted by the ad hoc constraint | applied_ad_hoc_constraint | |
| owning_role | Role | 0..* | role restricted by the ad hoc constraint | applied_ad_hoc_constraint | |
| owning_fact_type | Fact_Type | 0..* | fact type restricted by the ad hoc constraint | applied_ad_hoc_constraint | |

Bibliography

- Bakema, G., Zwart, J. & van der Lek, H. 2000, *Fully Communication Oriented Information Modelling*, Ten Hagen Stam, The Netherlands.
- ECSS 2011, *European Cooperation for Space Standardization*, ECSS-E-TM-10-23A, Space engineering, Space System data repository.
- Fillmore, C. 1988, 'The Case for Case', in *Universals in Linguistics*, Bach, E. and Harms, E.T. (eds.), pp. 1-88.
- Frege, G. 1879, *Begriffsschrift, eine der arithmetischen nachgebildete Formalsprache des reinen Denkens*, Halle a. S.
- Griethuysen, J. and Jardine, D. 1989, *Information Modelling with INFOMOD, Volume I – Concepts and Information Structure*, Philips, Eindhoven.
- Habrias, H. 1988, *Le modèle relationnel binaire: Méthode I.A. (NIAM)*, Eyrolles, Paris, ISBN 978-2212081695.
- Halpin, T. and Morgan, T. 2008, *Information Modeling and Relational Databases*, Morgan Kaufmann, San Francisco, ISBN 978-0-12-373568-3.
- ISO 1987, ISO/TC97/SC5/WG3 (J.J. van Griethuysen, ed.), *Concepts and Terminology for the Conceptual Schema and the Information Base*, Report No. ISO/TC97/TR9007, International Standards Organization, Central Secretariat, Geneva, Switzerland, 1987 (originally published in 1982).
- ISO/IEC 24707:2007, Common Logic, *Information technology - Common Logic (CL): a framework for a family of logic-based languages*, First Edition, Geneva, Switzerland: International Organization of Standards, October 1, 2007.
- Lemmens, I., Nijssen, M. and Nijssen, G. 2007, 'A NIAM2007 Conceptual Analysis of the ISO and OMG MOF Four Layer Metadata Architectures', *On the Move to Meaningful Internet Systems 2007: OTM 2007 Workshops*, Meersman, R., Tari Z., Herrero, P. (eds.), Springer-Verlag, ISBN 978-3-540-76887-6, pp. 613-623.
- Meersman, R. 1982, *The RIDL Conceptual Language*, Research Report, Int. Centre for Information Analysis Services, Control Data, Brussels, Belgium.
- Meersman, R. and Steel, T. (eds.) 1986, *Data semantics (DS-1) : proceedings of the IFIP WG 2.6 Working Conference on Data Semantics (DS-1), Hasselt, Belgium, 7-11 January, 1985*, North-Holland, Amsterdam, ISBN 0-444-70004-8.
- Nijssen, G. 1978, A Framework for Discussion in ISO/TC97/SC5/WG3, ISO/TC97/SC5/WG3 78.09/01.
- Verheijen, G. and Bekkum, J. van 1982, *NIAM: An Information Analysis Method*, Proceedings of the IFIP WG 8.1 Conference on Comparative Review of Information System Design Methodologies, North-Holland, pp. 537-590.

To be completed...