Committee Draft ISO/IEC CD	
Date: **2006-01-12**	Reference number: ISO/JTC 1/SC 32**N1402**
Supersedes document SC 32N1109	

THIS DOCUMENT IS STILL UNDER STUDY AND SUBJECT TO CHANGE. IT SHOULD NOT BE USED FOR REFERENCE PURPOSES.

ISO/IEC JTC 1/SC 32 Data Management and Interchange Secretariat: USA (ANSI)	Circulated to P- and O-members, and to technical committees and organizations in liaison for voting (P-members only) by: **2006-04-13** Please return all votes and comments in electronic form directly to the SC 32 Secretariat by the due date indicated.

ISO/IEC CD 19763-4:200x(E)

Title: Information technology — Framework for Metamodel interoperability Part 4: Metamodel for model mapping

Project: 1.32.22.01.04.00

Introductory note: The attached document is hereby submitted for a three-month letter ballot to the National Bodies of ISO/IEC JTC 1/SC 32. The ballot starts 2006-01-12.

Medium: E

No. of pages: 33

Address Reply to: SC 32 Secretary, ISO/IEC JTC 1/SC 32,
Farance Inc, Island Box 256, New York, NY 10044-0205, United States of America
Telephone: +1 212 486-4700; E-mail: SC32-Sec@JTC1SC32.org

ISO/IEC JTC 1/SC 32 WG2

Date: 2006-01-08

ISO/IEC CD19763-4:2006(E)-

ISO/IEC JTC 1/SC 32/WG 2

Secretariat:

Information Technology – Framework for metamodel interoperability -- Part-4: Metamodel for model mapping

Document type: International Standard
Document subtype:
Document stage: (50) WorkingDraft
Document language: E

Contents

Table of Figures

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 part of ISO/IECD 19763 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 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* — **Framework for metamodel interoperability**:

— Part 1: Reference model

— Part 2: Core model

— Part 3: Metamodel for ontology

— *Part 4: Metamodel for model mapping*

There are three Informative Annexes for this part of ISO/IEC CD19763-4

— Annex A – (informative) Transformation Language

— Annex B – (informative) MMF Registry and Model Mapping

— Annex C – (informative) Use Cases of Model Transformation

Introduction

Due to the spread of e-business and e-commerce over the Internet, the effective exchange of business transactions and other related information across countries and cultures has become a prime concern for people both inside and outside the IT industry. These people endeavor to standardize domain specific business process models, which represent the best practices of businesses, as well as standard modelling constructs such as data elements, entity profiles and value domains for each business domain.

Increasingly, companies are being called on to design and implement inter-enterprise e-commerce and e-business solutions with a global market outlook. In order to minimize the cost and time required for meeting these interoperability needs, enterprises must design their information systems using common, standardized business models.

In order to develop a common model within an industry, it is necessary for the various stakeholders in the industry to identify essential data and components common to the industry. Also, these industry stakeholders must establish a methodology for using a common industry model to enable interoperability among information systems.

Two modelling approaches specified by the Object Management Group (OMG) are of particular importance in pursuing these modelling efforts. These approaches consist of the Unified Modeling Language (UML) and the Model Driven Architecture (MDA).

Information Technology – Framework for metamodel interoperability – Part 4: Metamodel for model mapping

1 Scope

The primary purpose of *ISO/IEC CD19763* is to specify the *Framework for Metamodel Interoperability.* ISO/IEC CD19763-4 also specifies the registry of model mapping which is required to describe transformation rules between different registered objects.

The metamodel framework for model mapping provides a normative metamodel which allows the describing of differences regarding formats and types of transformations to be exchanged or shared. This metamodel framework also provides a capability for identifying transformation rules between different registered objects.

For describing model transformations, it is important to standardize the transformation description language. However, those standardizations are out of the scope of this part of ISO/IEC 19763.

2 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 11179-1, Information technology – Metadata registries (MDR) - Part 1: Framework

ISO/IEC 11179-2, Information technology – Metadata registries (MDR) - Part 2: Classification

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

ISO/IEC 11179-4, Information technology – Metadata registries (MDR) - Part 4: Formulation of data definitions

ISO/IEC 11179-5, Information technology – Metadata registries (MDR) - Part 5: Naming and identification principles

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

ISO/IEC 11404:1996, Information technology – Programming languages, their environments and system software interfaces – Language-independent datatypes

ISO 12620:1999, Computer applications in terminology – Data categories

ISO/IEC 19501-1: Information technology – Unified Modelling Language (UML) – Part 1: Specification

ISO/IEC 19502-1: Information technology – Meta Object Facility (MOF): Specification

ISO/IEC 19503-1: Information technology – XML Metadata Interchange (XMI)

ISO/IEC 8824-1:2002, Information Technology - Abstract Syntax Notation One (ASN.1): Specification of Basic Notation

ISO/IEC 20944, Information Technology -Metadata Registry Interoperability and Bindings (MDRIB)

3 Definitions

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

3.1 defines MOF QVT [7] terms used in specifying the MMF model.

3.2 defines general terms used in this document that are not included in 3.1.

3.1 MOF QVT Terms used in Specifying the Metamodel of MMF Model Mapping

3.1.1
mapping
a mapping is a potentially directed transformation implementation

3.1.2
MOF QVT (Query, View and Transformation)
a formal transformation language used to specify transformation rules

3.1.3
pattern
a pattern describes the 'shape' of an object it will be matched against

3.1.4
pattern matching
a process whereby parts of a model are matched against a pattern

3.1.5
query
a query takes as input a model and selects specific elements from that model

3.1.6
relation
relationship
a multi-directional transformation specification, relations are not executable in the sense that they are unable to create or alter a model; they can, however, check two or more models for consistency against one another.

3.1.7
transformation
a process of automatic generation of a target model from a source model according to a transformation definition

NOTE Transformation is the umbrella term for relation and mapping.

3.1.8
view
a view is a model that is derived from another model

3.2 General Terms used in this part of ISO/IEC 19763

3.2.1
Administered Item
a registry item for which administrative information is recorded in an Administration

[ISO/IEC 11179-3:2003,(3.3.1)]

3.2.2
mandatory
always required

NOTE 1 One of three obligation statuses applied to the attributes of metadata items, indicating the conditions under which the attribute is required. See also optional (3.2.10).

NOTE 2 Obligation statuses apply to metadata items with a Registration Status of "recorded" or higher.

3.2.3
metadata
data that defines and describes other data

3.2.4
metadata item
an instance of a metadata object

NOTE 1 In all parts of ISO/IEC 19763, this term is applied only to instances of metadata objects described by the metamodel in Clause 4 of ISO/IEC 19763-2. Examples include instances of Model Concept, Model Domain Profile, Model Instances, etc.

NOTE 2 A metadata item has associated attributes, as appropriate for the metadata object it instantiates.

3.2.5
metadata object
an object type defined by a metamodel

NOTE 1 In all parts of ISO/IEC 19763, this term is applied only to metadata objects described by the metamodel in Clause 4 of ISO/IEC 19763-2. Examples include Model Concept, Model Domain Profile, Model Instances, etc.

3.2.6
Metadata Registry
MDR
an information system for registering metadata

NOTE The associated information store or database is known as a metadata register.

3.2.7
metamodel construct
a unit of notation for modelling

3.2.8
Metamodel Framework
MMF
a framework for registering artefacts that are based on metamodel and model

3.2.9
name (of Administered item)
a name by which an Administered Item is designated within a specific Context

NOTE Metamodel construct is: Attribute of Designation.

3.2.10
optional
permitted but not required

NOTE 1 One of three obligation statuses applied to the attributes of metadata items, indicating the conditions under which the attribute is required. See also mandatory (3.2.2).

NOTE 2 Obligation statuses apply to metadata items with a Registration Status of "recorded" or higher.

3.2.11
registry item
a metadata item recorded in a Metadata Registry

3.2.12
Uniform Resource Identifier
URI
a formatted string that serves as an identifier for a resource, typically on the Internet

NOTE the syntax is designed to meet the recommendations laid out in "Functional Recommendations for Internet Resource Locators" [RFC1736] and "Functional Requirements for Uniform Resource Names" [RFC1737].

[IETF RFC 2396]

4 Structure of a MetaModel Framework

This standard is a part of the metamodel framework family of standards and is based on the MMF-2 core model.

This standard provides solutions to resolve problems typically found in a heterogeneous environment that consists of different software platforms and middleware. Even in a single environment, which consists of similar platforms, a modelling artefact might be implemented or installed in different formats or syntaxes. In such an environment, model mapping will be needed to exchange modelling artefacts and concrete data.

4.1 Overview of MMF Model Mapping

This part of ISO/IEC 19763 uses a metamodel to describe the structure of an MMF's metadata register. The MMF's registry metamodel is specified as a conceptual and abstract data model, i.e. one that describes how relevant information is structured in the natural world. Any implementation should follow MMF models to establish the common management of metamodels and their derived models.

Figure 1 shows the MMF Model Mapping Package. The packages of MMF-Core, MOF-QVT, ObjectByMOF, MDR-ByMOF and MOF, shown with non-shaded packages in Figure 1, have been identified based on specifications outside of this standard (see 2). The use of packages is for descriptive purposes only and has no other significance.

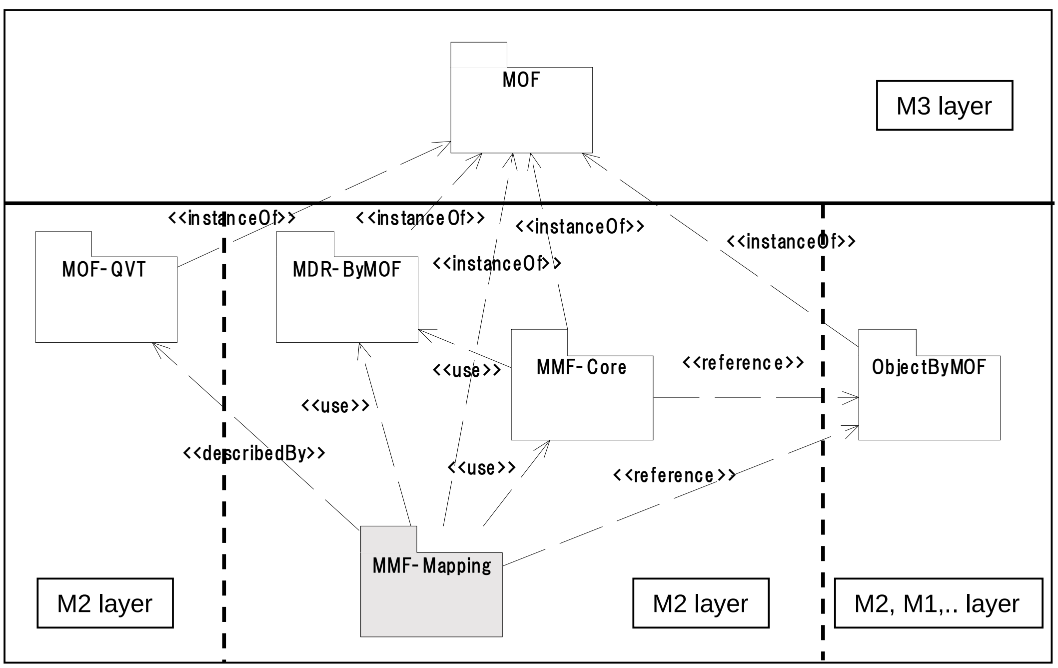


Figure 1- MMF Packages

4.2 Convention for Definition of MMF Model Mapping

The MMF Model Mapping registry model is specified using Administered Items as defined in the Metadata Registry (MDR), and conforming Meta Object Facility (MOF) standards. This standard uses the term "metamodel construct" for the model constructs it uses and the term "metadata objects" for the model constructs it specifies in the MOF. The metamodel constructs used include classes, relationships, association classes, attributes and references. These terms are defined in 3.1 of ISO/IEC 19763-2, and their model is described in Annex B of ISO/IEC 19763-2. The specified metadata objects are defined in this clause.

The MMF Model Mapping can be described by UML class diagrams, with the following conventions:

(1) Superclasses

<immediate inherited classes>

(2) Attributes

attribute name : Datatype and Cardinality <M> or <O>

<content and purpose>

(3) References

reference name : Class name and Cardinality <M> or <O>

<content and purpose>

(4) Constraints

<constraints if necessary, written in natural language>

The minimum and maximum cardinality for attributes and references can be attached. The maximum cardinality constraints are to be enforced for all registration status of the metadata item. The minimum cardinality constraints are to be enforced when the registration status is "recorded" or higher. In other words, a registration status of "recorded" or higher indicates that all mandatory attributes have been documented.

The requirement about attributes and references for each metaclass is also specified. “<M>” means that the Attribute/Reference is mandatory and “<O>” means that it is Optional.

There are the following metaclasses:

-LevelPair-Rule (see 4.3.1)

-MM-M-Rule (see 4.3.2)

-M-V-Rule (see 4.3.3)

-Transformation (see 4.3.4)

-MM-M-Transformation (see 4.3.5)

-M-V-Transformation (see 4.3.6)

The division of the model into regions is for descriptive purposes only and has no other significance.

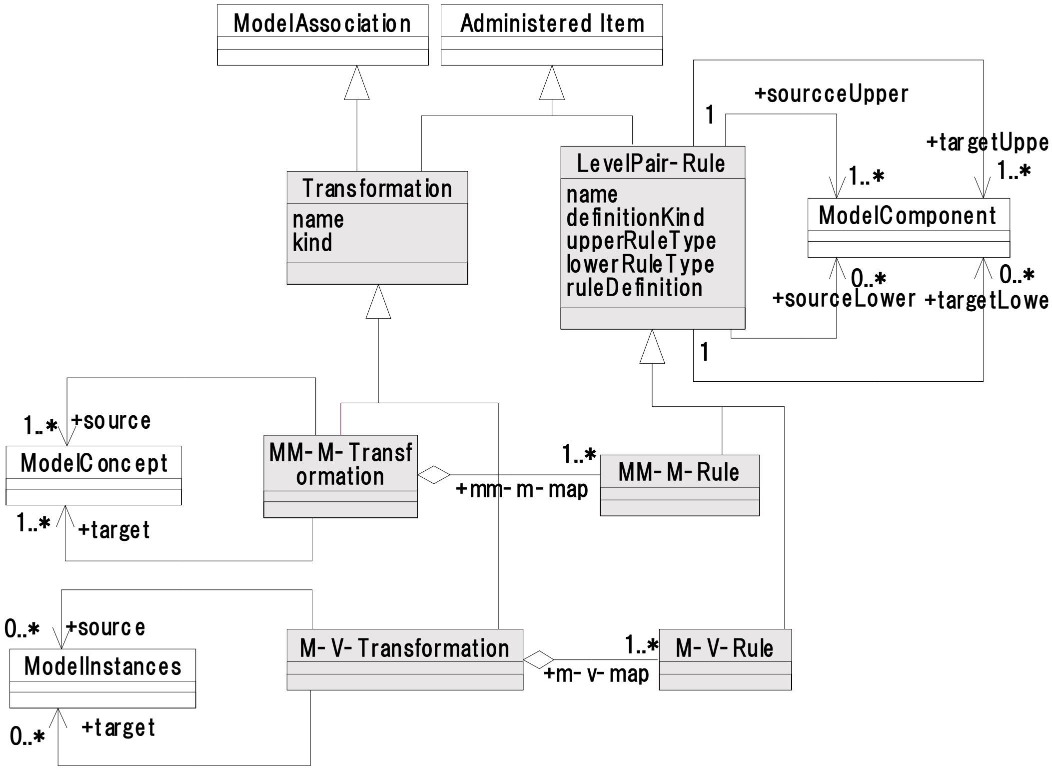


Figure 2- MMF-Model Mapping Package

4.3 MMF Model Mapping Elements

4.3.1 LevelPair-Rule

LevelPair-Rule is an abstract metamodel designating sets of transformation rules and transformed model components. Transformation rules are applied to four sets of model components including source upper, target upper, source lower and target lower.

Transformations are sets of constraints and derivations defined between upper/lower source and upper/lower target model components. The transformed lower model components are sets of objects, in MOF Extents, of the types of the registered packages only. Transformation rules are defined in the context of those model components.

In general, transformations have zero or more directions. Each direction defines a source and/or target of a transformation. A transformation can be executed in checking all constraints, or enforcing all derivations of that transformation in one chosen direction.

(1) Superclass

Administered Item (from MMF-Core)

(2) Attributes

name : *string[1..1] <M>*

The identifier of sets of transformation rules

definitionKind : *type code[1..1] <M>*

The kind of language describing the transformation rule, it may be specified using codes representing 'QVT-Relations', 'QVT-Core', 'QVT-Operational' and so on. (see Table 1 and Annex A).

Table 1- The Code Set of Definition Kind

Kind	Code	Description
QVT-Relations	QVT-1	Defined by the declarative language of MOF QVT.
QVT-Core	QVT-2	Defined by the pattern matching language of MOF QVT.
QVT-Operational	QVT-3	Defined by the mechanism of invoking the operational mapping of MOF QVT.
QVT-Black-box	QVT-4	Defined by non-standard Black-box MOF operation implement.
CWM	CWM	Defined by transformation framework of CWM.
XSLT	XSLT	Defined by the XSL transformation language XSLT.

upperRuleType : *type code[1..1] <M>*

The upperRuleType specifies the type of relationship between source upper and target upper model components. It may be specified using codes for 'Relationship', 'Mapping', or 'Computation'. (see Table 2 and 3).

lowerRuleType : *type code[1..1] <M>*

The lowerRuleType specifies the type of the relationship between source lower and target lower model components. It may be specified using codes for 'Relationship', 'Mapping', or 'Computation'. (see Table 2 and 3).

Table 2- Transformation Rule Types

Type	Code	Source model		Target model		Function
		Element	Instance	Element	Instance	
Relationship	REL	Should be specified	Pattern of Class / Associaiton	Should be specified	Pattern of Class / Association	Constraint
Mapping	MAP	Should be specified	Pattern of Class / Association and Object	Generated from source	Pattern of Class / Association and Object	Constraint/ Derivation
Computation	COM	Specified at execution time	Object (Value)	Generated from source	Object (Value)	Derivation

Table 3- Transformation Operation Types

Type	Operation	Description
Relationship	Renaming	Change a named element of source model component into another name of target model component
	Equivalence	Check if named elements between the source and target model component are equal
	Enhancement	Provide additional elements into the target model component
Mapping	Instantiation	Create instances of elements of the source model component into the target model component
	Generation	Compile the source model component and Issue codes for target model component
	Migration	Embed part of the source model component into the target model component
	Derivation	Extract elements of the soruce model commponet and build the target model component based on those elements
Computation	Aggregation	Apply arithemetic and logical operations for values of elements for the source model component and aggregate the results into values of the target model component
	Conversion	Convert values of data types of the soruce model component into appropriate ones of the target model component

ruleDefinition : *URI[1..1] <M>*

the URI of the file containing transformation rule definitions

(3) Referenes

sourceUpper : *ModelComponent[1..*] <M>*

the packages of transformed source upper model component

targetUpper : *ModelComponent[1..*] <M>*

the packages of transformed target upper model component

sourceLower : *ModelComponent[0..*] <M>*

the packages of transformed source lower model component

targetLower : *ModelComponent[0..*] <M>*

the packages of transformed target lower model component

(4) Constraints

sourceUpper may be in MM-level or M-level, and sourceLower should be in M-level or V-level

targetUpper may be in MM-level or M-level, and targetLower should be in M-level or V-level

4.3.2 MM-M-Rule

The MM-M-Rule is a subclass of LevelPair-Rule and a metaclass designating the metamodel transformation rule definition between source and target model component. The MM-M-Rule provides formal transformation rule at between MM-level and M-level.

(1) Superclass

LevelPair-Rule

(2) Attributes

definitionKind : *type code[1..1] <M>*

(from LevelPair-Rule)

upperRuleType : *type code[1..1] <M>*

(from LevelPair-Rule)

upperRuleType should be the code of 'Relationship'

lowerRuleType : *type code[1..1]*

lowerRuleType may be the code of 'Mapping' or 'Relationship'

ruleDefinition : *URI[1..1] <M>*

(from LevelPair-Rule)

(3) Referenes

sourceUpper : *ModelComponent[1..*] <M>*

(from LevelPair-Rule)

targetUpper : *ModelComponent[1..*] <M>*

(from LevelPair-Rule)

sourceLower : *ModelComponent[0..*] <M>*

(from LevelPair-Rule)

targetLower : *ModelComponent[0..*] <M>*

(from LevelPair-Rule)

(4) Constraints

sourceUpper should be in MM-level and sourceLower should be in M-level

targetUpper may be in MM-level, and targetLower should be in M-level

4.3.3 M-V-Rule

The M-V-Rule is a subclass of LevelPair-Rule and a metaclass designating the metamodel transformation rule definition between source and target model component. The M-V-Rule provides a formal transformation rule at between M-level and V-level.

(1) Superclass

LevelPair-Rule

(2) Attributes

definitionKind : *type code[1..1] <M>*

(from LevelPair-Rule)

upperRuleType : *type code[1..1] <M>*

(from LevelPair-Rule)

upperRuleType may be the code of ‘Mapping’ or ‘Relationship’

lowerRuleType : *type code[1..1] <M>*

lowerRuleType should be the code of ‘Computation’

ruleDefinition : *URI[1..1] <M>*

(from LevelPair-Rule)

(3) Referenes

sourceUpper : *ModelComponent[1..*] <M>*

(from LevelPair-Rule)

targetUpper : *ModelComponent[1..*] <M>*

(from LevelPair-Rule)

sourceLower : *ModelComponent[0..*] <M>*

(from LevelPair-Rule)

targetLower : *ModelComponent[0..*] <M>*

(from LevelPair-Rule)

(4) Constraints

sourceUpper should be in M-level and sourceLower should be in V-level respectively

targetUpper should be in M-level, and targetLower should be in V-level respectively

4.3.4 Transformation

Transformation is a subclass of AdministeredItem and ModelAssociation, and an abstract metaclass designating the transformation definition between source and target of ModelConcept / ModelInstances.

A mapping is part of one transformation and performs among zero or more ModelConcept / ModelInstances. Each ModelConcept / ModelInstances of a mapping has (is in) one direction of the transformation.

When a user executes a transformation to derive a model, all mappings that are part of that transformation are executed, deriving the ModelConcept / ModelInstances that are in one chosen direction.

(1) Superclass

AdministeredItem, ModelAssociation

(2) Attributes

name : *string[1..1] <M>*

The identifier of transformation between ModelConcept / ModelInstances

kind : *type code[1..1] <M>*

The type of transformation kind, it may be specified using a code for ‘Compilation type’, ‘Projection type I’, ‘Projection type II’ or ‘Metamorphose type’. (see Table 4).

Table 4- Transformation Kinds

Kind	code	Description
Compilation type	C	Compile the source into the target by generating
Projection type I	P1	Project subset of the source on the target by generating
Projection type II	P2	Project subset of the source on the target by matching
Metamorphose type	M	Transfer the modified source into the target by matchring

(3) References

none

(4) Constraints

none

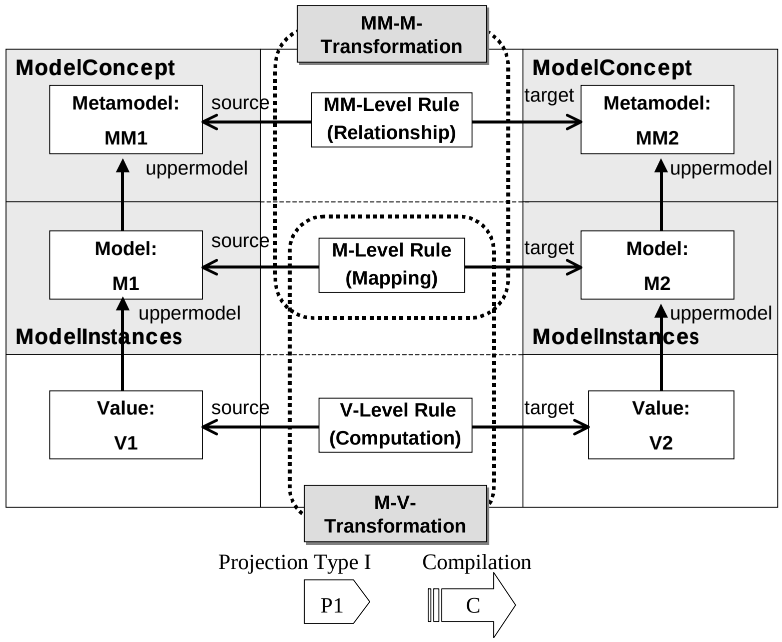


Figure 3- Transformation Kind (1)

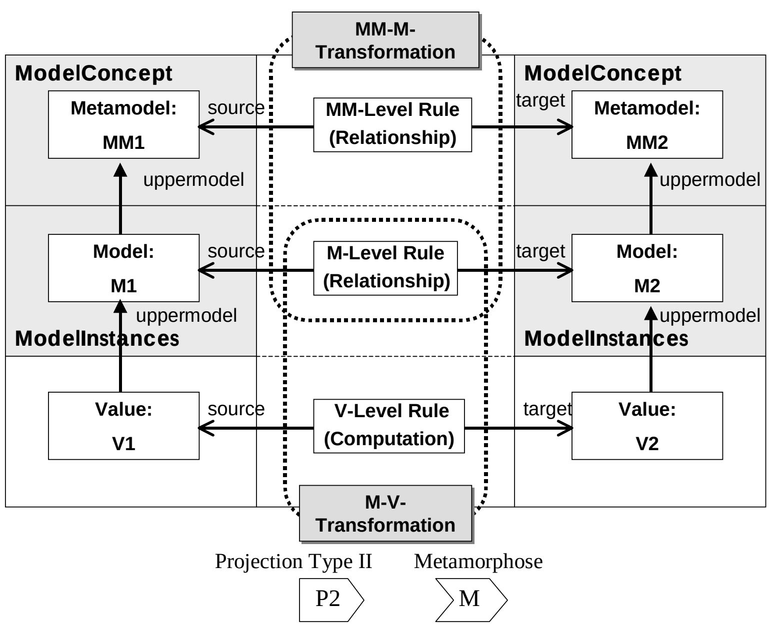


Figure 4- Transformation Kind (2)

4.3.5 MM-M-Transformation

M-V-Transformation is a subclass of Transformation and a metaclass designating the model transformation definition between source and target model component.

(1) Superclass

Transformation

(2) Attributes

name : *string[1..1] <M>*

(from Transformation)

kind : *type code[1..1] <M>*

(from Transformation)

mm-m-map : *MM-M-Rule[1..*] <M>*

the definition of mapping rule between source and target of model component for MM-level and M-level

(3) References

source : *ModelConcept[1..*] <M>*

the source ModelConcept of the model component, which is in MM-level

target : *ModelConcept[1..*] <M>*

the target ModelConcept of the model component, which is in MM-level

(4) Constraints

upperRuleType (MM-Level) and lowerRuleType (M-Level) of mm-m-map should be specified using the codes provided in table 5

Table 5- Transformation Kinds on MM-M-Transformation

Kind	code	MM-Level	M-Level	V-Level
Compilation type	C	Relationship	Mapping	N/A
Projection type I	P1	Relationship	Mapping	N/A
Projection type II	P2	Relationship	Relationship	N/A
Metamorphose type	M	Relationship	Relationship	N/A

4.3.6 M-V-Transformation

M-V-Transformation is a subclass of Transformation and a metaclass designating the value transformation definition between source and target value.

(1) Superclass

Transformation

(2) Attributes

name : *string[1..1] <M>*

(from Transformation)

kind : *type code[1..1] <M>*

(from Transformation)

m-v-map : *M-V-Rule[1..*] <M>*

the definition of mapping rule between source and target of model components for M-level and V-level

(3) References

source : *ModelInstances[0..*] <M>*

the source ModelInstances for each model component, which is in M-level

target : *ModelInstances[0..*] <M>*

the target ModelInstances of each model component, which is in M-level

(5) Constraints

upperRuleType (M-Level) and lowerRuleType (V-Level) of m-v-map should be specified using the codes provided in table 6

Table 6- Transformation Kinds on M-V-Transformation

Kind	code	MM-Level	M-Level	V-Level
Compilation type	C	N/A	Mapping	Computation
Projection type I	P1	N/A	Mapping	Computation
Projection type II	P2	N/A	Relationship	Computation
Metamorphose type	M	N/A	Relationship	Computation

4.4 Standard formats for interchanging models

In this part of ISO/IEC 19763, no concrete metadata format is specified. However, the XMI schema for MMF core model is a primary recommended metadata interchange format for interoperability. Another metadata format including bindings of ISO/IEC 20944 may be allowed if an appropriate XML schema is supplied as a profile.

5 Conformance

This part of ISO/IEC 19763 prescribes a conceptual metamodel, not a physical implementation. Therefore, the metamodel need not be physically implemented exactly as specified.

A conforming implementation shall:

- satisfy the requirements of 4.3;
- identify a Degree of Conformance (5.1); and
- identify a Level of Conformance (5.2).

5.1 Degree of Conformance

MMF Model Mapping conformance is specified along three orthogonal viewpoints; first is a value requirement viewpoint, second is interoperability viewpoint of metadata exchange, and third is conformance between registered content such as Uppermodel and Lowermodel.

1) Conformance on value requirements is specified in 5.3

2) Conformance on formats of interchanging model is specified in 5.4

NOTE Conformance on registered contents is not specified in this standard

The distinction between "Level 1" and "Level 2" implementations is necessary to address the simultaneous needs for interoperability and extensions. This part of ISO/IEC 19763 describes specifications that promote interoperability. Extensions are motivated by needs of users, vendors, institutions, and industries. The degree of conforming implementation is listed in Table7.

Table 7- Implementation Conformance Degree

	Extensions	Implementation	
		strictly conforming	**conforming**
a	are not directly specified by this part of ISO/IEC 19763	shall support all mandatory and optional attributes and references	shall support all mandatory and optional attributes and references
b	are specified and agreed to the other parts of ISO/IEC 19763	shall not use, test, access, or probe for any extension features nor extensions to attributes	as permitted by the implementation, may use, test, access, or probe for extension features or extensions to attributes
c	may serve as trial usage for future editions of this part of ISO/IEC 19763	shall not recognize, nor act on, nor allow the production of attributes that are dependent on any unspecified, undefined, or implementation-defined behavior	may recognize, act on, or allow the production of attributes that are dependent on implementation-defined behaviorbehaviour

NOTE The use of extensions to the metamodel or the basic attributes may cause undefined behavior.

A level 1 implementation may be limited in functionalities but is maximally interoperable with respect to this part of ISO/IEC 19763. A level 2 conforming implementation may have more functions, but may be less interoperable with respect to this part of ISO/IEC 19763.

5.2 Levels of Conformance

An implementation may conform to either of two levels of conformance to this standard:

Table 8- Level of Conformance

	Conformance View	Level of Conformance	
		Level 1	**Level 2**
1	**Value Requirements**	Only those metadata elements, relationships and properties specified in Clause 4 are supported and used	All metadata elements, relationships and properties specified in Clause 4 are supported and may be used.
2	**Interchange Format**	Only XMI format is supported and used	All format including bindings of ISO/IEC 20944 are supported and may be used

5.3 Obligation

One of two obligation states are applied to the attributes of metadata items, indicating the conditions under which the attribute is required. Obligation states apply to metadata items with a Registration Status of "recorded" or higher. Properties and relationships specified in this part of ISO/IEC 19763 are stated to be Mandatory or Optional.

For the purpose of conformance:

a) Mandatory properties and relationships shall exist, and these properties and relationships shall conform to the provisions of this part of ISO/IEC 19763.

b) Optional properties and relationships are not required to exist, but if they do exist they shall conform to the provisions of this part of ISO/IEC 19763.

Such obligations are enforced if and only if the Registration Status of the associated metadata items is Recorded or higher.

5.4 Implementation Conformance Statement

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

a) whether it conforms or strictly conforms (5.1):

b) whether conformance is to Level 1, Level 2 (5.2) or both;

c) what extensions are supported or used.

5.5 Roles and Responsibilities for Registration

Confomance needs to be considered in the context of the roles and responsibilities of registration authorities, as covered by ISO/IEC 11179-6: Registration of data elements. Extended conformance of systems requires formalisation of procedures, agreement of roles and responsibilities between parties, and guidelines addressing use of software products and conversions from other systems. The formalisation of these aspects must be consistent with the conformance requirements in the above Clauses and with the roles of registration authorities as set out in ISO/IEC 11179-6.

Annex A: (infomative) Transformation Languages

For interoperability of shared models, model transformation using a model pattern is a useful technique. Model transformation may be parameterized using a model pattern. The transformation definition also should be described with a formal transformation description language. Simply stated, a model compiler is a program that reads a transformation definition written in one formal language – the source model – and translates it into an equivalent model in another metamodel – the target model.

The tool based on MOF is useful for specifying and implementing the transformation rules. Such supporting tools use specialized transformation description languages. A transformation definition tool provides support for creating and modifying transformation definitions. A formal transformation description language is needed to define complex transformation rule and to share transformations among different tools.

For example, OMG MOF QVT (Query, View and Transformation) is a formal transformation language used to specify transformation rules precisely and availably. A transformation engine such as a model compiler generates simple code from a model. An important function supporting MDA is a function for completing transformations between one model and another model.

Kinds of Transformation Languages

-QVT-Relations : [7]

A declarative specification of the relationships between MOF models. The Relations language supports complex object pattern matching, and implicitly creates trace classes and their instances to record what occurred during a transformation execution. Relations can assert that other relations also hold between particular model elements matched by their patterns.

-QVT-Core : [7]

This is a small model/language which only supports pattern matching over a flat set of variables by evaluating conditions over those variables against a set of models. It treats all of the model elements of source, target and trace models symmetrically. In addition, the trace models must be explicitly defined, and these trace models are not deduced from the transformation description, as is the case with Relations. The core model may be implemented directly, or it may be simply used as a reference for the semantics of Relations, which are mapped to the Core, using the transformation language itself.

-QVT-Operational, QVT-Black-box: [7]

In addition to the declarative Relations and Core Languages which embody the same semantics at two different levels of abstraction, there are two mechanisms for invoking imperative implementations of transformations from Relations or Core: one standard language, Operational Mappings, as well as non-standard Black-box MOF Operation implementations. Each relation defines a class which will be instantiated to trace between model elements being transformed, and it has a one-to-one mapping to an Operation signature that the Operational Mapping or Black-box implements.

-CWM:

The transformation framework defined in the OMG's Common Warehouse Metamodel (CWM) Specification and transformation implemented using XSLT. The CWM transformation framework provides a mechanism for linking source and target elements, but the derivation of the target elements has to be implemented in some concrete language, which is not prescribed by CWM. Effectively, CWM gives a general model, but no actual mechanism to implement model transformations.

-XSLT:

XSLT is another model transformation langaugae. As MOF models can be serialized as XML using the XML Metadata Interchange (XMI), implementing model transformations using XSLT is possible.

Annex B: (infomative) MMF Registory and Model Mapping

Specialized diagram languages like UML are used in virtually every area of information technology. Source and Target modeles are equally as diversified; a target model may be another model or programming language, or the machine language of any computer between a microprocessor and a supercomputer.

There are many use cases of MMF registry and model mapping.

Figure B1 illustrates an example of a health care application. MMF R&R (Registry and Repository) holds registered model components and model transformation including minimal set of metamodel for representing model mapping. Model mappings are required between different systems as shown in the figure.

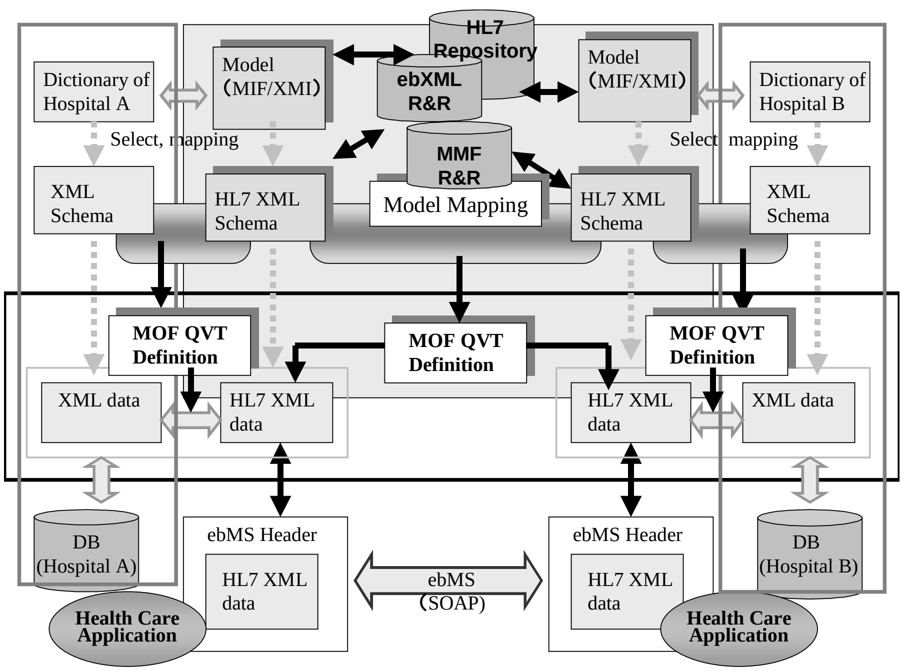


Figure B1- An Environment of MMF Registry and Model Mapping

Annex C: (infomative) Use Cases of Model Transformation

There are several cases of model transformation as follows:

-Case 1: Transformation on HL7 and ebXML (see Figure B1)

-Case 2: Transformation on MDA (see Figure B2)

-Case 3: Transformation on CWM and XSLT (see Figure B3)

(1) Case 1

HL7HDF (HL7 Development Framework) provides model transformation from DMIM (Domain Message Information Model) to RMIM (Refined Message Information Model).

Another approach of building a message model is ebXML, which was developed by OASIS and UN/CEFACT. For example, the mapping from CC (Core Component) to BIE (Business Information Entity) based on the metamodel of CCLib (Core Component Library) is a typical model mapping.

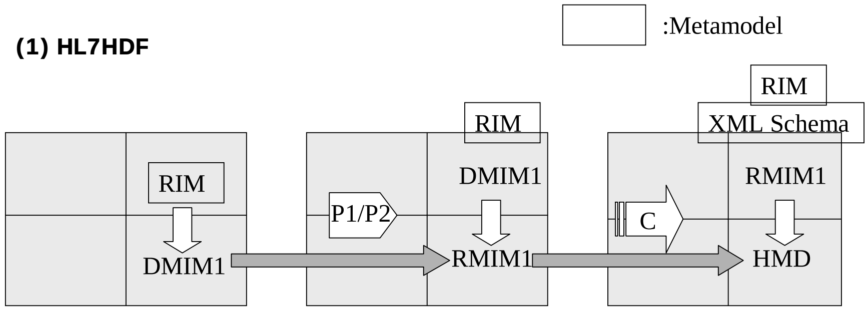
(2) Case 2

Case 2 is a scenario of MDA. MDA provides the transformation function from an upper level model with higher abstraction level into several implementation models. A transformation from PIM (Platform independent model) into PSM (Platform specific model) in MDA is a typical example.

(3) Case 3

A model transformation between different metamodels in CWM (Common Warehouse Metamodel) is shown. For example, the ETL (Extract, Transform and Load) tool provides model transformation between SQL tables. Also, the transformation related to XSL, XPath, XSLT is a typical example of this type transformation.

(1) HL7HDF



(2) ebXML/UMM

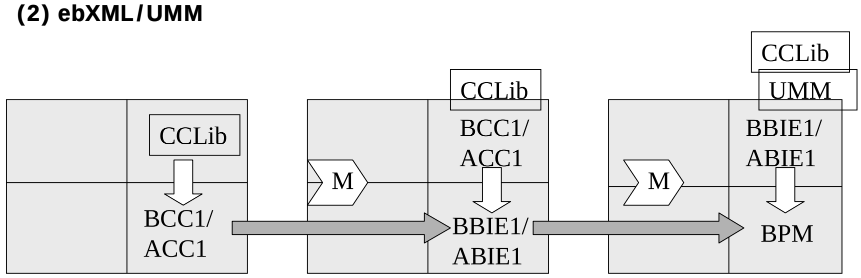
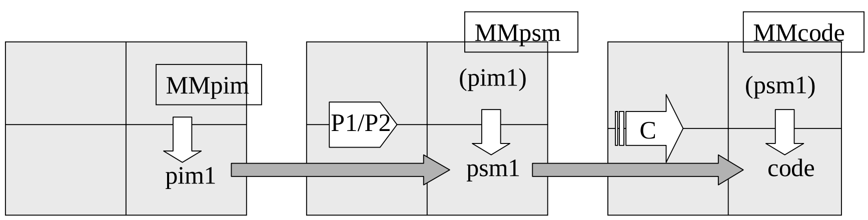


Figure C1- Transformation on HL7 and ebXML

(3) MDA (PIM to PSM)



(4) MDA (PIM to PIM)

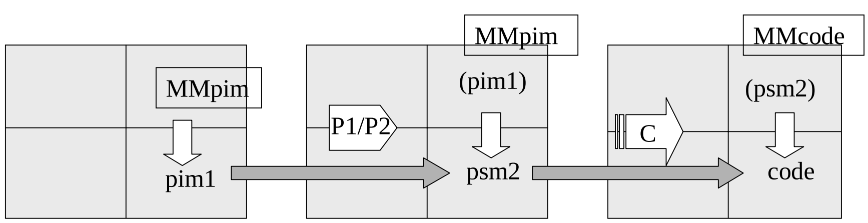
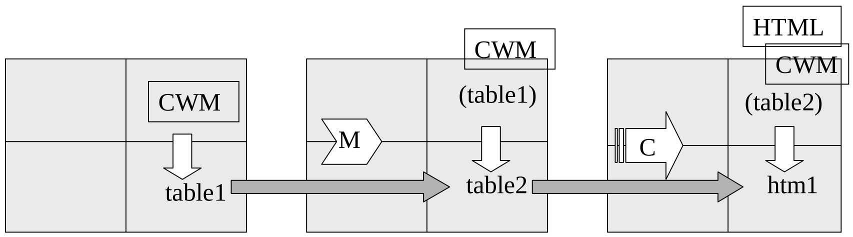


Figure C2- Transformation on MDA

(5) CWM



(6) XSL, XPATH, XSLT

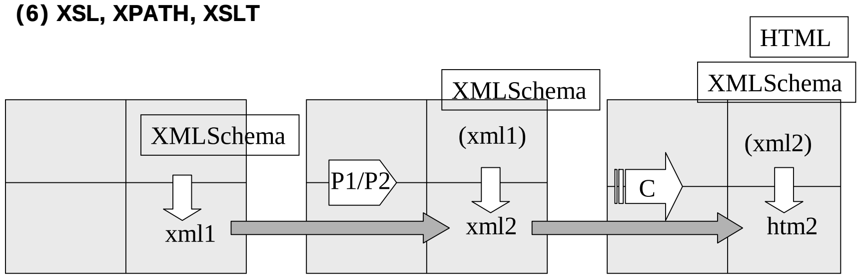


Figure C3- Transformation on CWM and XSLT

Bibliography

[1] ISO/IEC TR 9007:1987, *Information processing systems – Concepts and terminology for the conceptual schema and the information base*

TR 9007 provides information on conceptual modelling.

[2] ISO/IEC 10027:1990, *Information technology – Information Resource Dictionary System (IRDS) Framework*

ISO/IEC 10027 describes the concept of levels of modelling.

[3] ISO/IEC TR 15452:2000, *Information technology – Specification of data value domains*

TR 15452 describes the specification of value domains. It is expected to be replaced by ISO/IEC TR 20943-3.

[4] ISO/IEC TR 20943-1:200n (to be published), *Information technology – Achieving metadata registry content consistency – Part 1:Data elements*

[5] ISO/IEC TR 20943-3:200n (to be published), *Information technology – Achieving metadata registry content consistency – Part 3:Value domains*

[6] *Object Management Group, MOF 2.0 Query / Views / Transformations RFP, OMG Document: ad/2002-04-10,* revised on April 24, 2002

[7] *Adaptive, Alcatel, UMTP, Compuware, DSTC, IBM, Interactive Objects, Kennedy Carter, Softeam, Sun Microsystems, Tata Consultancy Services, Thales, TNIValiosys*, MOF 2.0Query/View/Transformation RFP (ad/2002-04-10) QVT-Merge Group version 2.0 ad/2005-03-02

[8] QVT-Partners. MOF Query/Views/Transformations, Revised Submission. OMG Document: ad/2003-08-08

[9] CBOP, DSTC, and IBM. MOF Query/Views/Transformations, Revised Submission. OMG Document: ad/03-08-03

[10] OMG, The Common Warehouse Model 1.1., OMG Document: formal/2003-02-03,

[11] W3C, XSL Transformations (XSLT) Version 1.0, November 1999, http://www.w3.org/TR/xslt

