INTERNATIONAL STANDARD

ISO/IEC 10165-7

First edition
1996-09-01

Information technology — Open Systems Interconnection — Structure of management information: General relationship model

Technologies de l'information — Interconnexion de systèmes ouverts (OSI) — Structure des informations de gestion: Modèle général relationnel

Reference number
ISO/IEC 10165-7:1996(E)

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

Contents

1 Scope 1

2 Normative references 1

2.1 Identical Recommendations | International Standards 1

2.2 Paired Recommendations | International Standards equivalent in technical content 2

3 Definitions 2

3.1 Management framework definitions 2

3.2 Systems management overview definitions 2

3.3 CMIS definitions 3

3.4 Management information model definitions 3

3.5 Guidelines for the definition of managed objects definitions 3

3.6 Requirement and guidelines for implementation conformance statement proformas associated with OSI management definitions 3

3.7 State management function definitions 4

3.8 Additional definitions 4

4 Abbreviations 4

5 Conventions 5

6 Requirements 5

7 Model 5

7.1 Managed relationships 5

7.2 Relationship mappings 7

7.3 Re-usable specifications 8

7.4 Representation and management of managed relationships 8

8 Generic definitions 10

8.1 Relationship management operations and notification 11

8.2 Managed object class – genericRelationshipObject 11

8.3 Name binding – genericRelationshipObject-system 11

8.4 Attributes 11

8.5 Attribute group – relationships 12

8.6 Parameters 12

Annex A – Relationship templates 13

A.1 Relationship class template 13

A.2 Relationship mapping template 17

Annex B – Definition of management information 21

B.1 Allocation of object identifiers 21

© ISO/IEC 1996

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the publisher.

ISO/IEC Copyright Office • Case postale 56 • CH-1211 Genève 20 • Switzerland

Printed in Switzerland

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

B.2 Definition of managed object classes 21
B.3 Definition of name bindings 21
B.4 Definition of attributes 21
B.5 Definition of parameters 22
B.6 Abstract syntax definitions 22
Annex C – Managed relationship conformance statement proforma for General Relationship Model 23
C.1 Introduction 23
C.2 Instructions for completing the MRCS proforma 23
C.3 Symbols, abbreviations and terms 23
C.4 Managed relationship support 23
Annex D – MIDS (attribute) proforma 25
D.1 Introduction 25
D.2 Attributes 25
D.3 Parameters 25
Annex E – Illustration of representation methods 27
Annex F – Examples of use of templates 29
F.1 Allocation of object identifiers 29
F.2 Symmetric relationship example 29
F.3 Dependency relationship example 30
F.4 General composition relationship example 36
F.5 Access control domain example 37
Annex G – Commentary 41
G.1 Introduction 41
G.2 Dependency between managed objects in a managed relationship 41
G.3 Consistency of views 41
G.4 Expression of relationship management operations and notifications 41
G.5 Generic management 42
G.6 Relationship awareness 42
G.7 Role specification 42
G.8 Re-use of specifications 42
G.9 AND SUBCLASSES 42
G.10 Relationship between relationships 42
G.11 Naming Scope of relationship objects 43
G.12 Allowable representation methods 43

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

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. 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.

International Standard ISO/IEC 10165-7 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 21, *Open Systems Interconnection, data management and open distributed processing*, in collaboration with ITU-T. The identical text is published as ITU-T Recommendation X.725.

ISO/IEC 10165 consists of the following parts, under the general title *Information technology — Open Systems Interconnection — Structure of management information*:

— *Part 1: Management Information Model*

— *Part 2: Definition of management information*

— *Part 4: Guidelines for the definition of managed objects*

— *Part 5: Generic management information*

— *Part 6: Requirements and guidelines for implementation conformance statement proformas associated with OSI management*

— *Part 7: General relationship model*

Annexes A to D form an integral part of this part of ISO/IEC 10165. Annexes E to G are for information only.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Introduction

This Recommendation | International Standard provides a model for the definition, representation and management of relationships between resources and the notational "tools" for specifying these. In addition, the definitions of general management information that may be used in the representation of relationships are specified. Finally guidelines for the development of conformance statement proformas are provided. The capability afforded by this Recommendation | International Standard is important for those concerned with specifying a management information model.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

INTERNATIONAL STANDARD

ITU-T RECOMMENDATION

INFORMATION TECHNOLOGY – OPEN SYSTEMS INTERCONNECTION – STRUCTURE OF MANAGEMENT INFORMATION: GENERAL RELATIONSHIP MODEL

1 Scope

This Recommendation | International Standard provides:

a) a model for reasoning about, representing, managing, and developing re-usable specifications for relationships between resources;

b) notational tools for specifying relationships, their representation, and management;

c) definitions of generic management information that may be used in the representation and management of relationships;

d) guidelines for the development of conformance statement proforma;

e) example definitions.

The general relationship model is specified in clause 7. The notation tools are specified in Annex A. The generic management information is defined in clause 8 and Annex B. The guidelines for the specification of implementation conformance statement proforma are given in Annexes C and D. An illustration of the representation methods and example definitions are presented in Annex E and Annex F respectively. A commentary on the text is included in Annex G.

This Recommendation | International Standard does not provide a mechanism for the maintenance of consistency between resources that is implied by a relationship.

CCITT Rec. X.732 | ISO/IEC 10164-3 specifies a model of relationships represented by attributes and a set of generic attributes for representing specific types of relationships. The modelling concepts and specification tools defined in this Recommendation | International Standard are applicable to the definition of relationships in general, and hence, are also applicable to relationships represented by attributes as modelled in CCITT Rec. X.732 | ISO/IEC 10164-3.

2 Normative references

The following Recommendations and International Standards contain provisions which, through reference in this text, constitute provisions of this Recommendation | International Standard. At the time of publication, the editions indicated were valid. All Recommendations and Standards are subject to revision, and parties to agreements based on this Recommendation | International Standard are encouraged to investigate the possibility of applying the most recent editions of the Recommendations and Standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards. The Telecommunication Standardization Bureau of the ITU maintains a list of the currently valid ITU-T Recommendations.

2.1 Identical Recommendations | International Standards

- CCITT Recommendation X.701 (1992) | ISO/IEC 10040:1992, *Information technology – Open Systems Interconnection – Systems management overview.*
- CCITT Recommendation X.720 (1992) | ISO/IEC 10165-1:1993, *Information technology – Open Systems Interconnection – Structure of management information: Management Information Model.*
- CCITT Recommendation X.721 (1992) | ISO/IEC 10165-2:1992, *Information technology – Open Systems Interconnection – Structure of management information: Definition of management information.*

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

- CCITT Recommendation X.722 (1992) | ISO/IEC 10165-4:1992, *Information technology – Open Systems Interconnection – Structure of management information: Guidelines for the definition of managed objects.*
- ITU-T Recommendation X.724 (1993) | ISO/IEC 10165-6:1994, *Information technology – Open Systems Interconnection – Structure of management information: Requirements and guidelines for implementation conformance statement proformas associated with OSI management.*
- CCITT Recommendation X.731 (1992) | ISO/IEC 10164-2:1993, *Information technology – Open Systems Interconnection – Systems management: State management function.*
- CCITT Recommendation X.732 (1992) | ISO/IEC 10164-3:1993, *Information technology – Open Systems Interconnection – Systems management: Attributes for representing relationships.*

2.2 Paired Recommendations | International Standards equivalent in technical content

- CCITT Recommendation X.208 (1988), *Specification of Abstract Syntax Notation One (ASN.1).*

  ISO/IEC 8824:1990, *Information technology – Open Systems Interconnection – Specification of Abstract Syntax Notation One (ASN.1).*
- CCITT Recommendation X.291 (1992), *OSI conformance testing methodology and framework for protocol Recommendations for CCITT applications – Abstract test suite specification.*

  ISO/IEC 9646-2:1994, *Information technology – Open Systems Interconnection – Conformance testing methodology and framework – Part 2: Abstract Test Suite specification.*
- ITU-T Recommendation X.296 (1995), *OSI conformance testing methodology and framework for protocol Recommendations for ITU-T applications – Implementation conformance statements.*

  ISO/IEC 9646-7:1995, *Information technology – Open Systems Interconnection – Conformance testing methodology and framework – Part 7: Implementation Conformance Statements.*
- CCITT Recommendation X.700 (1992), *Management framework for Open Systems Interconnection (OSI) for CCITT applications.*

  ISO/IEC 7498-4:1989, *Information processing systems – Open Systems Interconnection – Basic Reference Model – Part 4: Management framework.*
- CCITT Recommendation X.710 (1991), *Common management information service definition for CCITT applications.*

  ISO/IEC 9595:1991, *Information technology – Open Systems Interconnection – Common management information service definition.*
- CCITT Recommendation X.711 (1991), *Common management information protocol for CCITT applications.*

  ISO/IEC 9596-1:1991, *Information technology – Open Systems Interconnection – Common management information protocol – Part 1: Specification.*

3 Definitions

For the purposes of this Recommendation | International Standard, the following definitions apply.

3.1 Management framework definitions

This Recommendation | International Standard makes use of the following terms as defined in CCITT Rec. X.700 | ISO/IEC 7498-4:

- managed object.

3.2 Systems management overview definitions

This Recommendation | International Standard makes use of the following terms as defined in CCITT Rec. X.701 | ISO/IEC 10040:

a) managed object class;

b) manager;

c) MOCS;

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

d) MOCS proforma;

e) notification;

f) (systems management) operation.

3.3 CMIS definitions

This Recommendation | International Standard makes use of the following terms as defined in CCITT Rec. X.710 | ISO/IEC 9595:

– attribute.

3.4 Management information model definitions

This Recommendation | International Standard makes use of the following terms as defined in CCITT Rec. X.720 | ISO/IEC 10165-1:

a) action;

b) attribute group;

c) attribute type;

d) behaviour;

e) characteristic;

f) containment;

g) inheritance;

h) invariant;

i) multiple inheritance;

j) name binding;

k) naming tree;

l) packages;

m) parameter;

n) post-condition;

o) pre-condition;

p) specialization;

q) subclass;

r) subordinate object;

s) superclass;

t) superior object.

3.5 Guidelines for the definition of managed objects definitions

This Recommendation | International Standard makes use of the following terms as defined in CCITT Rec. X.722 | ISO/IEC 10165-4:

a) managed object class definition;

b) template.

3.6 Requirement and guidelines for implementation conformance statement proformas associated with OSI management definitions

This Recommendation | International Standard makes use of the following terms as defined in ITU-T Rec. X.724 | ISO/IEC 10165-6:

a) Managed Relationship Conformance Summary (MRCS);

b) MRCS proforma;

c) Management Information Definition Statement (MIDS);

d) MIDS proforma.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

3.7 State management function definitions

This Recommendation | International Standard makes use of the following terms as defined in CCITT Rec. X.731 | ISO/IEC 10164-2:

a) administrative state;

b) operational state;

c) usage state.

3.8 Additional definitions

3.8.1 binding: The associating of managed objects with a given role of a managed relationship.

3.8.2 binding support: The ability of a managed relationship to support the binding of managed objects, into a given role, during the existence of the managed relationship.

3.8.3 consistency (of a subclass): A refinement of a managed relationship class such that instances of the subclass may be substituted for instances of the superclass without affecting the function of a managing system.

3.8.4 managed relationship: A collection of managed objects together with an invariant referring to the properties of the managed objects.

3.8.5 managed relationship class: A named set of managed relationships sharing the same definition.

3.8.6 participant: A managed object fulfilling a role in a managed relationship.

3.8.7 participant pointer: An attribute that identifies participants in a particular role in a managed relationship.

3.8.8 relationship cardinality: The number of instances of the same managed relationship class in which a managed object participates in the same role.

3.8.9 relationship cardinality constraint: A set of values to which the relationship cardinality is restricted.

3.8.10 relationship management notification: A notification from a managed relationship that is mapped onto one or more systems management notifications.

3.8.11 relationship management operation: An operation applied to a managed relationship that is mapped onto one or more systems management operations.

3.8.12 role cardinality: The number of managed objects participating in a given role of a managed relationship.

3.8.13 role cardinality constraint: A set of values to which the role cardinality of a given role is restricted.

3.8.14 relationship mapping specification: A named specification of the mapping of the characteristics of a managed relationship class to the characteristics of one or more managed object classes.

3.8.15 relationship class specification: A named specification of the characteristics of a managed relationship.

3.8.16 role: The properties common to a particular kind of participant in a managed relationship.

3.8.17 unbinding: The disassociating of managed objects from a given role of a managed relationship.

3.8.18 unbinding support: The ability of a managed relationship to support the unbinding of managed objects, from a given role, during the existence of the managed relationship.

4 Abbreviations

For the purposes of this Recommendation | International Standard, the following abbreviations apply:

ASN.1 Abstract Syntax Notation One (see CCITT Rec. X.208 | ISO/IEC 8824)

CMIS Common Management Information Service (see CCITT Rec. X.710 | ISO/IEC 9595)

GDMO Guidelines for the Definition of Managed Objects (see CCITT Rec. X.722 | ISO/IEC 10165-4)

MIM Management Information Model (see CCITT Rec. X.720 | ISO/IEC 10165-1)

MRCS Managed Relationship Conformance Statement

MOCS Managed Object Conformance Statement

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

OSI Open Systems Interconnection

SMI Structure of Management Information

MIDS Management Information Definition Statement

5 Conventions

A distinctive combination of type weight and size is used throughout this Recommendation | International Standard where the text makes use of **ASN.1 notation**, **GDMO notation**, or the **notational tools** defined in Annex A.

The notational tools within this Recommendation | International Standard are specified according to the conventions defined in CCITT Rec. X.722 | ISO/IEC 10165-4.

6 Requirements

In the context of systems management, there can exist relationships between resources in the sense that resources may affect each other. A manager needs the ability to manage such relationships and, in particular, requires:

- a model of relationships between resources independent of the location of the resources and independent of the method of representing the relationships;
- notational tools for the specification of relationships;
- a model for the representation and management of relationships within the context of OSI Systems Management;
- notational tools for the specification of the representation and management of relationships within the context of OSI Systems Management;
- a model for the development of re-usable specifications.

7 Model

For the purposes of systems management, resources are modelled as managed objects, thus relationships between resources are modelled as **managed relationships** between managed objects. A managed relationship is defined as a collection of managed objects together with an invariant referring to the properties of the managed objects. Examples of such invariants are:

a) a managed object must remain in the enabled operational state to fulfil the supplier role in a supplier-consumer managed relationship;

b) a managed object must be in the enabled operational state, the idle usage state, and the unlocked administrative state to fulfil the backing-up role in a back-up managed relationship;

c) there must be at least one managed object in the subordinate role in a superior-subordinate managed relationship.

Managed relationships are additional information modelling concepts in the Structure of Management Information but are represented and managed by existing constructs of the Management Information Model (see CCITT Rec. X.720 | ISO/IEC 10165-1). Managed relationships that share the same definition are grouped into **managed relationship classes**; a notation for specifying managed relationship classes is defined in Annex A.

The model specifically recognizes that the same managed relationship class may be represented in different ways within the Management Information Model; a **relationship mapping** describes a particular representation. Relationship mappings are discussed in detail in 7.4. A notation for specifying relationship mappings is defined in Annex A.

7.1 Managed relationships

A managed relationship models the representation-independent properties of a relationship between managed objects in terms of roles, behaviour, relationship management operations and notifications, inheritance, and qualifying properties. The roles are modelled in terms of compatible managed object classes, role cardinality, relationship cardinality, and support for binding and unbinding operations. The modelling concepts are detailed in the following subclauses.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

7.1.1 Relationship management operations and notifications

Relationship management operations and notifications model the representation-independent operations and notifications that a managed relationship supports. They are expressed in terms of the following prototypical operations and notification:

ESTABLISH	establish a managed relationship;
TERMINATE	terminate a managed relationship;
BIND	bind a managed object into a role in a managed relationship;
UNBIND	release a managed object from a role in a managed relationship;
QUERY	request information about a managed relationship;
NOTIFY	report events concerning a managed relationship;
USER DEFINED	user-defined operation, the semantics of which are modelled in the behaviour of the related managed relationship.

The semantics of these prototypical operations and notification are given in clause 8.

A managed relationship models the actual relationship operations and notifications supported and any semantics additional to those defined in clause 8.

A particular managed relationship may model a number of actual relationship management operations or notifications with respect to a single prototype. For example, a particular managed relationship could model an unbind operation of a participant that requires all other participants to be unbound and deleted; it could model a second unbind operation that requires that all other participants remain unaffected. A managed relationship need not model a relationship operation or notification with respect to each of the prototypes.

7.1.2 Managed relationship behaviour

Managed relationship behaviour models the representation-independent behaviour of a managed relationship in terms of invariants over participant roles and invariants, pre-conditions, and post-conditions over relationship management operations and notifications.

7.1.2.1 invariant: A logical predicate that must remain true during some scope; a scope might be the lifetime of a managed relationship or the execution of a relationship management operation.

7.1.2.2 pre-condition (for a relationship management operation or notification): A logical predicate that must be true immediately before the execution of a relationship management operation or immediately before the emission of a relationship management notification.

7.1.2.3 post-condition (for a relationship management operation or notification): A logical predicate that must be true immediately after the execution of a relationship management operation or immediately after the emission of a relationship management notification.

7.1.3 Relationship qualification

Relationship qualification models attributes that are properly associated with the managed relationship as a whole and are made available in an implementation irrespective of the representation method used. For example, a telephone call could be modelled as a managed relationship between two subscriber-role managed objects, in which case the call duration is properly a property of the call rather than a property of either subscriber. However, in a particular implementation and depending on the representation method used, the call-duration attribute could be mapped to either of the subscriber managed objects or to a relationship object.

7.1.4 Roles

Each managed object bound in a managed relationship is known as a participant and fulfils one or more roles in the relationship. A role imposes requirements on a participant and on the managed relationship. A participating managed object is required to posses certain properties to fulfil a role; a managed relationship is required to comply with the requirements of a role.

Managed objects of the same class may fulfil different roles in the same managed relationship. A managed object may fulfil more than one role in a managed relationship. A managed object may participate in more than one instance of a managed relationship.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

7.1.4.1 Participant properties

The properties that a managed object must posses to fulfil a particular role are modelled in terms of a compatible[1] managed object class. In general, the compatible class will model only those properties that are peculiar to the role. In a particular implementation, the managed object fulfilling the role may posses additional properties but it must posses at least the properties of the compatible class and hence must be allomorphic to the compatible class.

7.1.4.2 Role cardinality

In general, a number of managed objects may participate in a managed relationship in a given role; this number is termed the cardinality of the role. An implementation of a managed relationship is required to comply with two types of constraints on role cardinality: permitted role cardinality and required role cardinality. Each constraint is modelled in terms of a value-set – a set of non-negative integers that is often a contiguous range of values.

The permitted role cardinality constraint is a constraint on the role cardinality that an implementation is permitted to support whilst the required role cardinality constraint is a constraint on the role cardinality that an implementation is required to support. The required role cardinality constraint value-set must be a subset of, or equal to, the permitted role cardinality constraint value-set.

7.1.4.3 Support of binding and unbinding

A managed relationship may support, on a role-by-role basis, the binding and unbinding of managed objects during the existence of the relationship. Thus, such a managed relationship supports the relationship management operations **BIND** and **UNBIND**, respectively.

Where a managed relationship supports binding, managed objects may become participants in the relationship during the existence of the relationship provided that the role cardinality constraints are not violated. An attempt to violate such constraints will cause the bind request to fail.

Where a relationship supports unbinding, participants can be released from a managed relationship during the existence of the relationship provided that the role cardinality constraints are not violated. An attempt to violate such constraints will cause the unbind request to fail.

7.1.4.4 Relationship cardinality

A managed object may participate in the same role in a number of instances of the same managed relationship class. The number of instances is termed the relationship cardinality for the role. An implementation of a managed relationship is required to comply with a single constraint on relationship cardinality – the permitted relationship cardinality constraint. The constraint is modelled in terms of a value-set – a set of non-negative integers that is often a contiguous range of values. The permitted relationship cardinality constraint is a restriction on the relationship cardinality that an implementation is permitted to support.

7.2 Relationship mappings

A relationship mapping models the representation of a managed relationship in terms of the properties of one or more managed objects, namely:

- the mapping of relationship roles and relationship qualifications to candidate object classes;
- the mapping of relationship operations and notifications to systems management operations and notifications respectively;
- relationship objects;
- participant pointers.

There may be more than one relationship mapping associated with a particular managed relationship class.

7.2.1 Participant pointers

Participants in the managed relationship and their respective roles may be identified by means of **participant pointer** attributes. The value of a participant pointer attribute identifies the participating managed object(s) whilst the type of the attribute indicates the role fulfilled by the managed object(s). Modification of these attribute values, either by attribute-based operations or by object-based operations, may be used to alter managed-object participation, subject to any constraints associated with the particular managed relationship or relationship mapping. Participant pointer definitions are derived from the definition of the attribute **participantPointer** in Annex B.

---

[1] The concept of compatibility is discussed in 5.2 of MIM.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

7.2.2 Relationship management operations and notifications

This Recommendation | International Standard does not prescribe the mapping of relationship management operations and notifications to systems management operations and notifications. However, Annex A provides templates for defining such mappings and Annex B defines suitable attributes for identifying names of managed relationships, relationship-class membership, and the relationship mapping in effect.

The potential mappings of relationship management operations and notifications may be constrained by the chosen representation method; a relationship mapping models the mappings for a particular representation of a managed relationship class. Subclause 7.4 gives more details of the constraints imposed by particular representation methods. A single relationship management operation or notification may be mapped to one or more systems management operations and notifications.

The mapping of relationship management operations and notifications is such that the pre- and post-conditions and invariant for the relationship management operations or notifications and the invariant for the managed relationship are respected by the systems management operations and notifications. The relationship mapping models the precise mechanism for meeting this requirement. For example in the case of a superior-subordinate managed relationship that requires at least one managed object in the subordinate role, a relationship mapping could model the mapping of **ESTABLISH** to:

- explicit create operations for the managed objects fulfilling the superior and subordinate roles and attribute-oriented operations to adjust the participant pointers; or
- a single create operation for the managed object fulfilling the superior role and then rely on the managed system to create the managed object fulfilling the subordinate role and to adjust the participant pointers.

7.2.3 Behaviour

The relationship-mapping behaviour models the mapping of representation-independent behaviour of a managed relationship and its associated relationship management operations and notifications to representation-dependent behaviour in terms of invariants over participants and invariants, pre-conditions, and post-conditions over systems management operations and notifications related to participants. It also models any additional behaviour related to the representation method.

7.3 Re-usable specifications

Managed relationship classes, inheritance, and specialization form the model for the development of re-usable specifications. The essence of the model is specialization – the derivation of classes from existing managed relationship classes by means of inheritance and incremental specification.

A managed relationship class may be specialized by combining characteristics inherited from one or more managed relationship classes with characteristics specified in the managed relationship class template. The specialized class is referred to as the subclass of the original class(es); the original class(es) are referred to as the superclass(es) of the specialized class. The rules for specialization defined in Annex A ensure that a subclass of a managed relationship is **consistent** with its superclass(es). The consistency of a subclass with respect to its superclass(es) is such that an instance of a subclass of a managed relationship may be substituted for an instance of one of its superclasses without affecting the function of a managing system.

7.4 Representation and management of managed relationships

Managed relationships may be represented by the following methods based on constructs defined in the Management Information Model (see CCITT Rec X.720 | ISO/IEC 10165-1):

- naming;
- participant pointers;
- relationship objects;
- systems management operations.

Not all categories of managed relationships can be represented by all of the representation methods. A single relationship mapping may use a combination of these representation methods. For example, a relationship mapping of a managed relationship with three roles could represent two roles by naming and the third by participant pointers.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

7.4.1 Representation and management by means of naming

A relationship mapping may model the representation and management of a managed relationship by means of naming. A number of subordinate managed objects are named within the naming scope of a superior managed object. The relationship mapping indicates the name binding associated with the managed relationship.

Relationship management operations may be mapped to systems management operations on either the superior or subordinate managed objects. Table 1 enumerates the possible mappings; mappings for a particular representation are modelled by the associated relationship mapping.

The participants in the managed relationship may be discovered by analysis of the components of the distinguished name of the subordinates for a given name binding.

Table 1 – Operation mappings for naming relationship

Prototypical relationship management operation	Candidate systems management operations on the superior managed object	Candidate systems management operations on the subordinate managed objects
BIND	Create, Action	Create
UNBIND	Delete, Action	Delete
QUERY	Action	Get + name analysis, Action
ESTABLISH	Create, Action	Create, Action
TERMINATE	Delete, Action	Delete, Action

7.4.2 Representation and management by means of participant pointers

A relationship mapping may model the representation and management of a managed relationship by means of participant-pointer attributes exhibited by the participants in the managed relationship. Relationship management operations may be mapped to attribute-based operations on participant pointers or object-based operations on participating managed objects. Table 2 enumerates the possible mappings; mappings for a particular representation are modelled by the associated relationship mapping.

Table 2 – Operation mappings for relationship attributes

Prototypical relationship management operation	Candidate attribute-based systems management operations on participating managed objects	Candidate object-based systems management operations on participating managed objects
BIND	Replace, Add	Create, Action
UNBIND	Replace, Remove	Delete, Action
QUERY	Get	Action
ESTABLISH	Replace, Add	Create, Action
TERMINATE	Replace, Remove	Delete, Action

7.4.3 Representation and management by means of a relationship object

A relationship mapping may model the representation and management of a managed relationship by means of a managed object – such a managed object is called a **relationship object.** The superclass of all relationship object classes, **genericRelationshipObject**, has the following attributes:

a) relationship name – which identifies the name of the managed relationship;

b) relationship class – which identifies the class of the managed relationship;

c) relationship mapping – which identifies the relationship mapping in effect.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Relationship object classes include a participant-pointer attribute for each role defined in the relationship class to identify participants in an instance of a managed relationship.

The **relationships** attribute group (see CCITT Rec. X.732 | ISO/IEC 10164-3) is exhibited by **genericRelationshipObject**. All participant-pointer attributes may be included in this attribute group.

Relationship management operations are mapped to object-based or attribute-based systems management operations on the relationship object. Table 3 enumerates the possible mappings; mappings for a particular representation are modelled by the associated relationship mapping.

Table 3 – Operation mappings for relationship object

Prototypical relationship management operation	Candidate attribute-based systems management operations on the relationship object	Candidate object-based systems management operations on the relationship object
BIND	Replace, Add	Create, Action
UNBIND	Replace, Remove	Delete, Action,
QUERY	Get	Action
ESTABLISH	Replace, Add	Create, Action
TERMINATE	Replace, Remove	Delete, Action

7.4.4 Representation and management by means of systems management operations

A relationship mapping may model the representation and management of a managed relationship by means of object-based systems management operations on participating managed objects. Relationship management operations are mapped to object-based systems management operations on participating objects. Table 4 enumerates the possible mappings; mappings for a particular representation are modelled by the associated relationship mapping.

Table 4 – Operation mappings for relationships represented by management operation

Prototypical relationship management operation	Candidate object-based systems management operations on participating managed objects
BIND	Create, Action
UNBIND	Delete, Action
QUERY	Action
ESTABLISH	Create, Action
TERMINATE	Delete, Action

8 Generic definitions

This Recommendation | International Standard defines the semantics of generic management information and prototypical relationship management operations and notification. The formal specification of the syntax of the generic management information is given in Annex B.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

8.1 Relationship management operations and notification

8.1.1 ESTABLISH

Invariant: the role and relationship cardinality constraints are not violated.

Pre-condition: the managed relationship does not exist; managed objects specified in the establish operation to be bound are of a class permitted to take on the role.

Post-condition: the managed relationship exists; managed objects specified in the establish operation exist and are bound into the managed relationship.

8.1.2 TERMINATE

Pre-condition: the managed relationship exists.

Post-condition: the managed relationship does not exist; managed objects that were bound in the relationship are not bound in the relationship.

8.1.3 BIND

Invariant: the managed relationship exists; the role and relationship cardinality constraints are not violated.

Pre-condition: the classes of managed objects specified in the bind operations are those permitted to take on the role; and the managed relationship supports the bind operation for the role.

Post-condition: managed objects specified in the bind operation exist and are bound into the relationship.

8.1.4 UNBIND

Invariant: the managed relationship exists; the role and relationship cardinality constraints are not violated;

Pre-condition: the managed objects specified in the unbind operation exist and are bound into the managed relationship; the managed relationship supports the unbind operation for the role;

Post-condition: the managed objects specified in the unbind operation are not bound into the managed relationship.

8.1.5 QUERY

Pre-condition: TRUE;

Post-condition: the managed relationship remains unchanged.

8.1.6 NOTIFY

Pre-condition: TRUE;

Post-condition: the managed relationship remains unchanged.

8.1.7 USER DEFINED

This Recommendation | International Standard defines no semantics for this prototype.

8.2 Managed object class – genericRelationshipObject

All relationship object classes shall be specialized from **genericRelationshipObject**; it includes the **relationshipMapping**, **relationshipClass**, and **relationshipName** attributes. A relationship object class for a particular relationship class shall include an attribute, derived from **participantPointer** attribute, for each role defined in the managed relationship class.

8.3 Name binding – genericRelationshipObject-system

This name binding shall be used to name relationship objects with respect to the system managed object using the attribute **relationshipName**.

8.4 Attributes

8.4.1 Attribute – relationshipName

This attribute shall be used for naming managed relationships and for naming relationship objects.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

8.4.2 Attribute – relationshipClass

This attribute shall be used to identify the class of the managed relationship. It shall be that value assigned in the respective managed relationship template.

8.4.3 Attribute – relationshipMapping

This attribute shall be use to identify the relationships mapping in effect. It shall be that value assigned in the respective relationship mapping template.

8.4.4 Attribute – participantPointer

The **participantPointer** attribute is an unregistered attribute that shall serve as the prototype for all participant pointer attributes. The syntax of the attribute is set of managed-object names; the attribute matches for equality, set comparison, and set intersection. The attribute supports the role cardinality violation, relationship cardinality violation, no such object, and role instance conflict specific errors.

The value of a derived participant pointer attributes shall indicate the managed object(s) currently fulfilling the role; the type of the attribute shall indicate the role.

8.5 Attribute group – relationships

This attribute is defined in CCITT Rec. X.732 | ISO/IEC 10164-3 and shall be used to group together all participant pointer attributes.

8.6 Parameters

8.6.1 Parameter – noSuchObject

This specific error shall be used to signal that a relationship management bind operation specified the name of a managed object that is not known to the performer. The value of this parameter shall be that name specified in the bind operation.

8.6.2 Parameter – roleCardinalityViolation

This specific error shall be used to signal that a relationship management bind or unbind operation would have violated one of the role cardinality constraints of a managed relationship. The value of this parameter shall be null.

8.6.3 Parameter – roleInstanceConflict

This specific error shall be used to signal that a relationship management bind operation specified the name of a managed object of a class that is not permitted by the relationship mapping of the managed relationship. The value of this parameter shall be the name specified in the bind operation.

8.6.4 Parameter – relationshipCardinalityViolation

This specific error shall be used to signal that a relationship-management bind or unbind operation would have violated the relationship cardinality constraint of a managed relationship. The value of this parameter shall be null.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Annex A

Relationship templates

(This annex forms an integral part of this Recommendation I International standard)

A.1 Relationship class template

A.1.1 Overview

The relationship class template forms the basis of the formal definition of a managed relationship. Constructs in the template allow the various characteristics of the managed relationship class to be defined, namely:

a) relationship inheritance;

b) relationship qualification;

c) relationship behaviour;

d) role compatibility;

e) role cardinality constraints;

f) bind and unbind support;

g) relationship cardinality constraints.

The following template labels and supporting definitions used in the relationship class template are defined in GDMO:

<behaviour-label>

<class-label>

<attribute-label>

object-identifier

type-reference

The following supporting definitions, used in the relationship class template, are defined in ASN.1:

– identifier.

Labels values shall be unique within the assigning document.

A.1.1.1 Inheritance

The managed relationship class template permits the specification of the managed relationship superclass(es) from which a managed relationship class has been derived. Characteristics of the superclass(es) are inherited by the subclass. The specialization of a subclass is such that a subclass of a managed relationship is consistent with its superclass(es).

A.1.1.2 Relationship qualification

The managed relationship class template permits the definition of characteristics that qualify the relationship as a whole and are independent of the particular representation method.

A.1.1.3 Behaviour

The managed relationship class template requires the specification of behaviour of a managed relationship that is independent of the particular representation method. Behaviour that is dependent on the particular representation method shall be specified in the relationship mapping template.

A.1.1.4 Roles

The managed relationship class template permits the definition of the roles of the relationship and their associated characteristics.

A.1.1.5 Relationship class identifier

The managed relationship class template requires the specification of an object identifier which may be used to reference the relationship class in management protocol.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

A.1.2 Template structure

<relationship-class-label> RELATIONSHIP CLASS
[DERIVED FROM <relationship-class-label>
[, <relationship-class-label>]* ;]
BEHAVIOUR <behaviour-label> [, <behaviour-label]*;
[SUPPORTS supported [, supported]*;]
[QUALIFIED BY <attribute-label> [, <attribute-label>]*;]
[role-specifier]*;
REGISTERED AS object-identifier;

supporting productions

supported–>
ESTABLISH [operation-name]
I TERMINATE [operation-name]
I QUERY [operation-name]
I NOTIFY [notification-name]
I USER DEFINED [operation-name]

role-specifier–>
ROLE role-name
[COMPATIBLE-WITH <class-label>]
[PERMITTED-ROLE-CARDINALITY-CONSTRAINT type-reference]
[REQUIRED-ROLE-CARDINALITY-CONSTRAINT type-reference]
[BIND-SUPPORT [operation-name]]
[UNBIND-SUPPORT [operation-name]]
[PERMITTED-RELATIONSHIP-CARDINALITY-CONSTRAINT type-reference]
[REGISTERED AS object-identifier]

role-name –> <identifier>
operation-name –> <identifier>
notification-name –> <identifier>

A.1.3 Supporting definitions

A.1.3.1 DERIVED FROM <relationship-class-label> [, <relationship-class-label>]*

This construct shall be used to specify the superclass(es) from which the managed relationship class inherits its characteristics including any which may, in turn, have been inherited from other managed relationship class(es). The managed relationship class is a specialization of the inherited characteristics and those specified in the balance of the completed template; the specialization is such that the subclass is consistent with its superclass(es). If this construct is absent, the managed relationship class is not specialized from other relationship classes.

Specification of characteristics that are inherited from other managed relationship classes shall not be repeated in the specification of the subclass unless one of the techniques described in CCITT Rec. X.722 I ISO/IEC 10165-4 for extending a specification inherited from a superclass is being used.

The rules for specifying managed relationship subclasses to ensure consistency are as follows:

a) **SUPPORTS:** The specialized relationship management operations shall be the union of the relationship management operations of the superclasses and those specified in the subclass; inheritance and specialization shall not introduce additional relationship management notifications into a subclass.

b) **QUALIFIED BY:** Permitted and required value-sets of attribute ranges shall not be changed in a subclass.ehaviou

c) **BEHAVIOUR:** The br of a subclass shall be:

- the disjunctive combination of the pre-conditions inherited from its superclass(es) and those specified in the subclass;
- the conjunctive combination of the post-conditions inherited from its superclass(es) and those specified in the subclass;
- the conjunctive combination of the invariants inherited from its superclass(es) and those specified in the subclass; if the invariants are mutually contradictory, a subclass cannot be specified.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

d) **ROLE:**

- Additional role specifications may be included in the subclass definition.
- A managed objects class introduced by the **COMPATIBLE WITH** clause in the subclass shall be compatible[2] to those referenced in similar clauses in the superclass(es).
- The inherited **PERMITTED-ROLE-CARDINALITY-CONSTRAINT** value of a role inherited from more than one superclass shall be the set intersection of the values specified for that role in the superclasses; any permitted role cardinality constraint value specified in the subclass shall be a subset of, or equal to, the inherited permitted role cardinality constraint value; the specialized permitted role cardinality constraint value shall be the set intersection of the inherited values and that specified in the subclass.
- The inherited **REQUIRED-ROLE-CARDINALITY CONSTRAINT** value of a role inherited from more than one superclass shall be the set union of the values specified for the role in the superclasses set-intersected with the inherited permitted role cardinality constraint value; any required role cardinality constraint value specified in the subclass shall be a superset of, or equal to, the inherited required role cardinality constraint value; the specialized required role cardinality constraint value shall be the set union of the inherited value and that specified in the subclass set-intersected with the value of the specialized permitted role cardinality constraint.
- **BIND-SUPPORT** may be added in the subclass specification.
- **UNBIND-SUPPORT** may be added in the subclass specification.
- The inherited **PERMITTED-RELATIONSHIP-CARDINALITY-CONSTRAINT** value of a role inherited from more than one superclass shall be the set intersection of the values specified for the role in the superclasses; any permitted relationship cardinality constraint value specified in the subclass shall be a subset of, or equal to, the inherited permitted relationship cardinality constraint value; the specialized permitted relationship cardinality constraint value shall be the set intersection of the inherited value and that specified in the subclass.

e) **REGISTERED AS:** The subclass registration shall replace any registration inherited from other definitions.

A.1.3.2 BEHAVIOUR <behaviour-label>[, <behaviour-label>]

This construct shall be used to specify the representation-independent behaviour of the managed relationship. It shall be stated in terms of an invariant over the managed relationship and invariant and pre- and post-conditions for the relationship management operations and notification. The construct references behaviour templates as defined in CCITT Rec. X.722 | ISO/IEC 10165-4.

A.1.3.3 SUPPORTS supported [, supported]*

This construct shall be used to define the relationship management operations and notifications that a managed relationship supports. The **supported** supporting production shall be used to specify the prototypical operation or notification on which the relationship management operation or notification is based, namely:

- **ESTABLISH [operation-name];**
- **TERMINATE [operation-name];**
- **QUERY [operation-name];**
- **NOTIFY [notification-name];**
- **USER DEFINED [operation-name].**

The **operation-name** and **notification-name** shall be used, where necessary, to:

- provide a link to an optional specification, in behaviour templates referenced by the **BEHAVIOUR** construct, of behaviour additional to that specified for the referenced prototypical operation;
- disambiguate relationship management operations or notifications that are based on the same prototypical operation or notification respectively;
- provide a link to the related systems management operations and notifications specified in the relationship mapping template.

---

2) The concept of compatibility is discussed in 5.2 of MIM.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

A.1.3.4 QUALIFIED BY <attribute-label>[, <attribute-label>]*

This construct shall be used to specify attributes that are associated with the managed relationship as a whole. Qualifying attributes shall be made available in all implementations of the managed relationship irrespective of the representation method used. The relationship mapping template shall be used to specify how these attributes are made available by a particular representation.

A.1.3.5 ROLE role-name

This construct shall be used to specify the roles associated with the managed relationship class; the label role-name shall be used as a reference name to the role.

A.1.3.5.1 COMPATIBLE WITH <class-label>

This construct shall be used to specify the characteristics required of a managed object to fulfil the requirements of the role; the characteristics shall be specified in terms of a compatible[3] managed object class. If the construct is not present the characteristics of **top** (see CCITT Rec. X.721 | ISO/IEC 10165-2) are assumed. The role specification is independent of the representation method.

A.1.3.5.2 PERMITTED-ROLE-CARDINALITY-CONSTRAINT type-reference

This construct shall be used to specify any restriction on the number of managed objects that a managed relationship is permitted to support in the role. It shall reference an ASN.1 subtype value set of non-negative integers.

For example, if the construct specifies a value set of **INTEGER (1..3)**, a managed relationship is permitted to support either 1, 2, or 3 managed objects in the role but it is not permitted to support more than 3 managed objects in the role. An implementation is required to enforce the constraint.

If the value set contains 0, the role is optional; however an optional role does not imply support of either the bind- or unbind-operations. If the construct is absent, the inherited permitted role cardinality constraint shall be used as the default; if no constraint has been inherited, a value set **INTEGER (0..MAX)** is assumed for the constraint.

The **PERMITTED-ROLE-CARDINALITY-CONSTRAINT** value set shall be a superset of, or equal to, the **REQUIRED-ROLE-CARDINALITY-CONSTRAINT** value set.

A.1.3.5.3 REQUIRED-ROLE-CARDINALITY-CONSTRAINT type-reference

This construct shall be used to specify any restriction on the number of managed objects that a managed relationship is required to support in the referenced role. The constraint shall be specified in terms of an ASN.1 subtype value set of non-negative integers. For example, if the construct specifies a value set of **INTEGER (1, 3, 4)**, a managed relationship is required to support either 1, 3, or 4 managed objects in the role but it is not required to support either 2 managed objects or more than 4 managed objects in the role. An implementation is required to enforce the constraint.

If the value set contains 0, the role is optional; however an optional role does not imply support of either the bind- or unbind-operations. If the construct is absent, the inherited required role cardinality constraint value shall be used as the default; if no value has been inherited, the managed relationship is not required to support the constraint.

The **REQUIRED-ROLE-CARDINALITY-CONSTRAINT** value set shall always be a subset of, or equal to, the **PERMITTED-ROLE-CARDINALITY-CONSTRAINT** value set.

A.1.3.5.4 BIND-SUPPORT [operation-name]

This construct shall be used to specify that managed objects may become participants in the role during the existence of the relationship provided that role cardinality constraints are not violated. Absence of this construct implies that managed objects may not become participants in the role during the existence of the relationship.

The operation-name shall be used, where necessary, to:

- provide a link to an optional specification, in behaviour templates referenced by the **BEHAVIOUR** construct, of behaviour additional to that specified for the referenced **BIND** prototypical operation;
- disambiguate multiple relationship management operations that are based on the **BIND** prototypical operation;
- provide a link to the related systems management operations specified in the relationship mapping template.

---

[3] The concept of compatibility is discussed in 5.2 of MIM.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

A.1.3.5.5 UNBIND-SUPPORT [operation-name]

This construct shall be used to specify that participants may be released from the role during the existence of the relationship provided that role cardinality constraints are not violated. Absence of this construct implies that participants may not be released from the role during the existence of the relationship.

The operation-name shall be used, where necessary, to:

- provide a link to an optional specification, in behaviour templates referenced by the **BEHAVIOUR** construct, of behaviour additional to that specified for the referenced **UNBIND** prototypical operation;
- disambiguate multiple relationship management operations that are based on the **UNBIND** prototypical operation;
- provide a link to the related systems management operations specified in the relationship mapping template.

A.1.3.5.6 PERMITTED-RELATIONSHIP-CARDINALITY-CONSTRAINT type-reference

This construct shall be used to specify a restriction on the number of relationships of the referenced class in which managed object is permitted to participate in the referenced role. The constraint shall be specified in terms of an ASN.1 subtype value set of non-negative integers. For example, if the construct specifies a value set of **INTEGER (0..3)**, a managed object is permitted to participate in up to and including, but not more than, three instances of the referenced managed relationship class in the given role. An implementation is required to enforce the constraint. If the construct is absent, the inherited permitted relationship cardinality constraint shall be used as the default; if no constraint has been inherited, a value set **INTEGER (0..MAX)** is assumed for the constraint.

A.1.3.5.7 REGISTERED AS object-identifier

This construct shall be used to specify a globally-unique identifier which registers the role; the identifier may be used in protocol to unambiguously identify the role. If the role has been inherited, this construct shall not be present.

A.1.3.6 REGISTERED AS object-identifier

This construct shall be used to specify a globally-unique identifier which registers the managed relationship class; the identifier may be used in protocol to unambiguously identify the managed relationship class.

A.2 Relationship mapping template

A.2.1 Overview

The relationship mapping template forms the basis of the formal definition of a relationship mapping. Constructs in the template allow the various elements of the representation to be defined, namely:

a) relationship mapping behaviour;

b) relationship objects;

c) candidate classes from which managed objects may be drawn to fulfil roles;

d) representational methods;

e) qualification attributes;

f) operation and notification mappings.

The following template labels and supporting definitions used in the relationship mapping template are defined in GDMO:

<action-label>	<name-binding-label>
<attribute-label>	<notification-label>
<behaviour-label>	<parameter-label>
<class-label>	object-identifier

The following supporting definition, used in the relationship mapping template, is defined in ASN.1:

- identifier.

Labels values shall be unique within the assigning document.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

A.2.1.1 Behaviour

The relationship mapping template specifies any behaviour that is peculiar to the representation method defined in the template.

A.2.1.2 Representation methods

The relationship mapping template requires the specification of the method used to represent a managed relationship and any relevant management information associated with the role representation.

A.2.1.3 Roles

The relationship mapping template requires the specification of the mapping of roles and relationship qualifications to managed object classes.

A.2.2 Template structure

<relationship-mapping-label> RELATIONSHIP MAPPING
RELATIONSHIP CLASS <relationship-class-label> ;
BEHAVIOUR <behaviour-label> [, <behaviour-label>]*;
[RELATIONSHIP OBJECT <class-label> [QUALIFIES <attribute-label>
[, <attribute-label>]*];]
role-mapping-specification [, role-mapping-specification]*;
[OPERATIONS MAPPING relationship-operation maps-to
[, relationship-operation maps-to]* ;]
REGISTERED AS object-identifier;

supporting productions

role-mapping-specification –>
ROLE role-name RELATED-CLASSES <class-label> [<class-label>]*
[REPRESENTED-BY representation]
[QUALIFIES <attribute-label> [<attribute-label>]*]

representation ->
NAMING <name-binding-label> USING superiorOrSubordinate
| ATTRIBUTE <attribute-label>
| RELATIONSHIP-OBJECT-USING-POINTER <attribute-label>
| OPERATION

superiorOrSubordinate –>
SUPERIOR|SUBORDINATE

relationship-operation –>
ESTABLISH [operation-name]
| TERMINATE [operation-name]
| BIND [operation-name] [role-name]
| UNBIND [operation-name] [role-name]
| QUERY [operation-name] [role-name]
| NOTIFY [notification-name]
| USER DEFINED [operation-name]

maps-to –>
MAPS-TO-OPERATION systems-management-operation
OF role-or-relObject [systems-management-operation
OF role-or-relObject]*

systems-management-operation –>
GET <attribute-label> [<parameter-label>]*
| REPLACE <attribute-label> [<parameter-label>]*
| ADD <attribute-label> [<parameter-label>]*
| REMOVE <attribute-label> [<parameter-label>]*
| CREATE [<class-label>] [<parameter-label>]*
| DELETE [<parameter-label>]*
| ACTION <action-label> [<parameter-label>]*
| NOTIFICATION <notification-label> [<parameter-label>]*

role-or-relObject –> role-name | RELATIONSHIP OBJECT
role-name –> <identifier>
operation-name –> <identifier>
notification-name –> <identifier>

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

A.2.3 Supporting definitions

A.2.3.1 RELATIONSHIP CLASS <relationship-class-label>

This construct shall be used to specify the managed relationship class to which this relationship mapping is related.

A.2.3.2 BEHAVIOUR <behaviour-label> [, <behaviour-label>]

This construct shall be used to specify the representation-dependent behaviour of the managed relationship and its relationship operations and notifications. It shall be stated in terms of an invariant over the participating managed objects and an invariant and pre- and post-conditions over systems management operations and notifications related to participating managed objects. The construct shall not specify behaviour in addition to that already exhibited by the participating managed objects.

A.2.3.3 RELATIONSHIP OBJECT <class-label> [QUALIFIES <attribute-label> [, <attribute-label>]*]

This construct is present in templates which specify the representation of a managed relationship by means of a relationship object. The **<class-label>** shall be used to indicate the class of the relationship object; in a real implementation the class of the relationship object shall that referenced by **<class-label>** or a subclass thereof. The managed object class referenced by **<class-label>** shall be a subclass of **genericRelationshipObject** and shall exhibit participant pointer attributes for each of the roles specified in the associated managed relationship class template.

The **QUALIFIES <attribute-label> [, <attribute-label>]*** construct shall be used to specify the relationship qualification attributes, defined in the referenced relationship class template, that are to be realized by the relationship object.

A.2.3.4 ROLE role-name RELATED-CLASSES <class-label> [<class-label>]* [REPRESENTED-BY representation] [QUALIFIES <attribute-label> [<attribute-label>]*]

This construct shall be used to identify candidate managed object classes, referenced by **<class-label> [<class-label>]***, that may fulfil the role, referenced by **role-name**. Therole shall be one of the roles specified in the referenced managed relationship class template; the classes shall be compatible with that referenced in the **COMPATIBLE WITH** clause of the referenced relationship class template. Only managed objects of the classes specified in the **<class-label> [, <class-label>]*** construct and their subclasses shall be permitted to fulfil the role in an instance of the referenced relationship class that uses this mapping.

The **representation** supporting definition shall specify the method by which the referenced role is to be represented and any associated management information. A choice of one of the following productions shall be used to specify representation by naming, participant pointers, relationship object, or systems management operations respectively:

- **NAMING <name-binding-label> USING superiorOrSubordinate**: The role referenced by role-name shall be represented by an object of either the **SUPERIOR OBJECT CLASS** or the **SUBORDINATE OBJECT CLASS** indicated in the name binding referenced by **<name-binding-label>**; the expansion of the **superiorOrSubordinate** supporting production, **SUPERIOR** or **SUBORDINATE**, shall indicate either the **SUPERIOR OBJECT CLASS** or the **SUBORDINATE OBJECT CLASS** respectively.
- **ATTRIBUTE <attribute-label>**: The type of the attribute referenced by **<attribute-label>** shall indicate the referenced role; the value of the attribute shall specify participant(s) fulfilling that role.
- **RELATIONSHIP-OBJECT-USING-POINTER <attribute-label>**: The type of the attribute referenced by **<attribute-label>** shall indicate the referenced role; the value of the attribute shall specify participant(s) fulfilling that role.
- **OPERATION**: The mapping of relationship management operations to systems management operations shall be specified in the **OPERATIONS MAPPING** construct.

The **QUALIFIES <attribute-label> [<attribute-label>]*** construct identifies relationship qualification attributes, defined in the referenced relationship class template, that are to be realized by the referenced managed object classes.

A.2.3.5 OPERATIONS MAPPING relationship-operation maps-to [, relationship-operation maps-to]*

This construct shall be used to specify the mapping of a relationship management operation to one or more systems management operations.

The **relationship-operation** supporting definition specifies a choice of one of the following productions which shall be used to indicate the respective relationship management operation or notification and the role to which it refers:

- **ESTABLISH [operation-name];**
- **TERMINATE [operation-name];**
- **BIND [operation-name][role-name];**

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

- **UNBIND [operation-name] [role-name];**
- **QUERY [operation-name] [role-name];**
- **NOTIFY [notification-name];**
- **USER DEFINED [operation-name].**

The **operation-name** or **notification-name** specified shall be one of those defined in the related relationship class template and shall form, where required, the link between the semantics of the relationship management operations and notifications and their representation in terms of systems management operations and notifications. Where a managed relationship defines only one role, the specification of a **role-name** is optional.

The **maps-to** supporting definition specifies the following production:

- **MAPS TO OPERATION systems-management-operation OF role-or-relObject [systems-management-operation OF role-or-relObject]***

The **systems-management-operation** supporting definition specifies a choice of one of the following productions each of which indicates the respective systems management operation or notification and related systems management information; the **[<parameter-label>]** shall be used to specify any parameters to be associated with the systems management operation or notification:

- **GET <attribute-label> [<parameter-label>]***–The attribute referenced by **<attribute-label>** shall specify the attribute value to be retrieved.
- **REPLACE <attribute-label> [<parameter-label>]***–The attribute referenced by **<attribute-label>** shall specify the attribute value to be replaced.
- **ADD <attribute-label> [<parameter-label>]***–The attribute referenced by **<attribute-label>** shall specify the attribute to which the value is to be added.
- **REMOVE <attribute-label> [<parameter-label>]***–The attribute referenced by **<attribute-label>** shall specify the attribute from which the value is to be removed.
- **CREATE [<class-label>] [<parameter-label>]***–The class referenced by the **<class-label>** shall specify the class to which the created managed object is to belong.
- **DELETE [<parameter-label>]*.**
- **ACTION <action-label> [<parameter-label>]***–The action referenced by the **<action-label>** shall specify the action to be issued.
- **NOTIFICATION <notification-label> [<parameter-label>]***–The notification referenced by the **<notification-label>** shall specify the notification to be issued.

The **role-or-relObject** supporting definition specifies target or source managed objects for the referenced systems management operation. A choice of one of the following productions is permitted which shall be used to specify either the managed object fulfilling the role referenced in role-name or the relationship object respectively:

- **role-name;**
- **RELATIONSHIP-OBJECT.**

A.2.3.6 REGISTERED AS object-identifier

This construct shall be used to specify a globally-unique identifier which registers the relationship mapping; the identifier may be used in protocol to unambiguously identify the relationship mapping.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Annex B

Definition of management information[4)]

(This annex forms an integral part of this Recommendation | International Standard)

B.1 Allocation of object identifiers

This Recommendation | International Standard allocates the following object identifiers:

```
GRMD {joint-iso-itu-t ms(9) smi(3) part7(7) asn1Module(2) 1}

DEFINITIONS ::= BEGIN

grm-Object OBJECT IDENTIFIER ::= {joint-iso-itu-t ms(9) smi(3) part7(7) managedObjectClass(3)}
grm-Package OBJECT IDENTIFIER ::= {joint-iso-itu-t ms(9) smi(3) part7(7) package(4)}
grm-Parameter OBJECT IDENTIFIER ::= {joint-iso-itu-t ms(9) smi(3) part7(7) parameter(5)}
grm-NameBinding OBJECT IDENTIFIER ::= {joint-iso-itu-t ms(9) smi(3) part7(7) nameBinding(6)}
grm-Attribute OBJECT IDENTIFIER ::= {joint-iso-itu-t ms(9) smi(3) part7(7) attribute(7)}
grm-RelationshipClass OBJECT IDENTIFIER ::= {joint-iso-itu-t ms(9) smi(3) part7(7) relationshipClass(11)}
grm-RelationshipMapping OBJECT IDENTIFIER ::= {joint-iso-itu-t ms(9) smi(3) part7(7) relationshipMapping(12)}
grm-RelationshipRole OBJECT IDENTIFIER ::= {joint-iso-itu-t ms(9) smi(3) part7(7) relationshipRole(13)}

END
```

B.2 Definition of managed object classes

```
genericRelationshipObject    MANAGED OBJECT CLASS
        DERIVED FROM "CCITT Rec. X.721 | ISO/IEC 10165-2":top;
        CHARACTERIZED BY genericRelationshipObjectPackage PACKAGE
            ATTRIBUTES    relationshipName GET,
                          relationshipClass GET,
                          relationshipMapping GET;
            ATTRIBUTE GROUPS
        "CCITT Rec. X.721 | ISO/IEC 10165-2":relationships;;;
REGISTERED AS {GRMD.grm-Object 1};
```

B.3 Definition of name bindings

```
genericRelationshipObject-system    NAME BINDING
    SUBORDINATE OBJECT CLASS    genericRelationshipObject AND SUBCLASSES;
    NAMED BY SUPERIOR OBJECT CLASS  "CCITT Rec. X.721 | ISO/IEC 10165-2":system AND SUBCLASSES;
    WITH ATTRIBUTE  relationshipName;
REGISTERED AS {GRMD.grm-NameBinding 1};
```

B.4 Definition of attributes

```
relationshipName    ATTRIBUTE
    WITH ATTRIBUTE SYNTAX    GRM-ASN1Module.SimpleNameType;
REGISTERED AS {GRMD.grm-Attribute 1};

relationshipClass    ATTRIBUTE
        WITH ATTRIBUTE SYNTAX   GRM-ASN1Module.RelationshipClass;
        MATCHES FOR EQUALITY;
REGISTERED AS {GRMD.grm-Attribute 2};

relationshipMapping    ATTRIBUTE
        WITH ATTRIBUTE SYNTAX    GRM-ASN1Module.RelationshipMapping;
        MATCHES FOR EQUALITY;
REGISTERED AS {GRMD.grm-Attribute 3};
```

4) Users of this Recommendation | International Standard may freely reproduce the contents of this annex so that it can be used for its intended purpose.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
participantPointer ATTRIBUTE
        WITH ATTRIBUTE SYNTAX  GRM-ASN1Module.GroupObjects;
        MATCHES FOR EQUALITY, SET-INTERSECTION, SET-COMPARISON;
        PARAMETERS    noSuchObject,
                roleInstanceConflict,
                roleCardinalityViolation,
                relationshipCardinalityViolation;;
-- An implementation may choose to apply ASN.1 subtyping restrictions to the attribute syntax of the
-- participantPointer attribute to reflect the permitted role cardinality constraints defined in a
-- specification.
```

B.5 Definition of parameters

```
noSuchObject PARAMETER
        CONTEXT SPECIFIC-ERROR;
        WITH SYNTAX GRM-ASN1Module.ObjectInstance;
REGISTERED AS {GRMD.grm-Parameter 1};

roleCardinalityViolation PARAMETER
        CONTEXT SPECIFIC-ERROR;
        WITH SYNTAX GRM-ASN1Module.Null;
REGISTERED AS {GRMD.grm-Parameter 3};

roleInstanceConflict PARAMETER
        CONTEXT SPECIFIC-ERROR;
        WITH SYNTAX GRM-ASN1Module.ObjectInstance;
REGISTERED AS {GRMD.grm-Parameter 2};

relationshipCardinalityViolation PARAMETER
        CONTEXT SPECIFIC-ERROR;
        WITH SYNTAX GRM-ASN1Module.Null;
REGISTERED AS {GRMD.grm-Parameter 4};
```

B.6 Abstract syntax definitions

```
GRM-ASN1Module {joint-iso-itu-t ms(9) smi(3) part7(7) asn1Module(2) 2}
DEFINITIONS ::= BEGIN

IMPORTS  ObjectInstance  FROM CMIP-1 {joint-iso-itu-t ms(9) cmip(1) version(1) protocol(3) }
SimpleNameType, GroupObjects
        FROM Attribute-ASN1Module {joint-iso-itu-t ms(9) smi(3) part2(2) asn1Module(2) 1}
RelationshipClass ::= OBJECT IDENTIFIER
RelationshipMapping ::= OBJECT IDENTIFIER
Null  ::=  NULL

END
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Annex C

Managed relationship conformance statement proforma[5] for General Relationship Model

(This annex forms an integral part of this Recommendation | International Standard)

C.1 Introduction

The purpose of the proforma in this annex is to provide guidelines the for Managed Relationship Conformance Statement (MRCS) so that a supplier of an implementation which claims to conform to a managed relationship class can provide conformance information in a standard form. The proforma defined in this annex is an additional proforma to that specified in ITU-T Rec. X.724 | ISO/IEC 10165-6.

C.2 Instructions for completing the MRCS proforma

The MRCS proforma contained in this annex is comprised of information in tabular form. The supplier of the implementation shall state which items are supported in Tables C.1 to C.3 and if necessary provide additional information.

C.3 Symbols, abbreviations and terms

The following common notations defined in CCITT Rec. X.291 | ISO/IEC 9646-2 are used for the status columns:

- m Mandatory
- o Optional
- c Conditional
- x Prohibited
- – Not applicable

The following common notations, defined in CCITT Rec. X.291 | ISO/IEC 9646-2 and ITU-T Rec. X.296 | ISO/IEC 9646-7 are used for support columns:

- Y Implemented
- N Not implemented
- – No answer required
- Ig The item is ignored (i.e. processed syntactically but not semantically)

C.4 Managed relationship support

The supplier of the implementation shall state the managed relationship class and the relationship mappings supported using Table C.1.

Table C.1 – Managed relationship support

Index	Relationship class template label	Value of object identifier for relationship class	Relationship mapping template label	Value of object identifier for relationship mapping	Status	Support	Additional information
1							

C.4.1 Roles support

For each role identified in the relationship mapping, the supplier of the implementation shall indicate support using Table C.2.

---

5) Users of this Recommendation | International Standard may freely reproduce the MRCS proforma in this annex so that it can be used for its intended purpose.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Table C.2 – Roles support

Index	Role label	Constraints and values	Status	Supported	Value of object identifier for actual participants managed object class	MOCS reference for actual participants managed object class	Additional information
1							
2							

C.4.1.1 Relationship management operations, notifications, and parameters support

The supplier of the implementation shall indicate the relationship management operations and notifications supported using Table C.3.

The supplier of the implementation shall indicate support for the parameters, if any, specified in the relationship mapping template by using the parameter support table specified in Annex D.

Table C.3 – Relationship management operations and notifications support

Index	Relationship management operation or notification	Systems management operation or notification	Constraints and values	Status	Support	Additional information
1						
2						

C.4.2 Relationship object support

The supplier of the implementation shall indicate support for the relationship object class, if any, specified in the relationship mapping template by using the MOCS proforma defined in ITU-T Rec. X.724 | ISO/IEC 10165-6 and MIDS proforma defined in Annex D. The relationship object class shall be a subclass of **genericRelationshipObject**.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Annex D

MIDS (attribute) proforma[6)]

(This annex forms an integral part of this Recommendation I International Standard)

D.1 Introduction

The purpose of the proforma in this annex is to provide guidelines for the Management Information Definition Statement (MIDS) so that a supplier of an implementation which claims to conform to a managed relationship class can provide conformance information in a standard form.

D.2 Attributes

See Table D.1.

Table D.1 – Attribute support

				Set by create		Get		Replace	
Index	Attribute template label	Value of object identifier for attribute	Constraints and values	Status	Support	Status	Support	Status	Support
1	relationshipName	{joint-iso-itu-t ms(9) smi(3) part7(7) attribute(7) 1}		o		m		x	
2	relationshipClass	{joint-iso-itu-t ms(9) smi(3) part7(7) attribute(7) 2}		o		m		x	
3	roleMapping	{joint-iso-itu-t ms(9) smi(3) part7(7) attribute(7) 3}		o		m		x	
4	participantPointer	–		o		o		o	

Table D.1 *(concluded)* **– Attribute support**

	Add		Remove		Set to default		
Index	Status	Support	Status	Support	Status	Support	Additional information
1	–		–		–		
2	–		–		–		
3	–		–		–		
4	o		o		–		

D.3 Parameters

See Table D.2.

6) Users of this Recommendation I International Standard may freely reproduce the MIDS proforma in this annex so that it can be used for its intended purpose. Instructions for constructing MIDS (attribute) proforma are specified in ITU-T Rec. X.724 I ISO/IEC 10165-6.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Table D.2 – Parameter support

Index	Parameter template label	Value of object identifier for parameter	Constraints and values	Status	Support	Additional information
1	noSuchObject	{joint-iso-itu-t ms(9) smi(3) part7(7) parameter(5) 1}		o		
2	roleCardinalityViolation	{joint-iso-itu-t ms(9) smi(3) part7(7) parameter(5) 3}		o		
3	roleInstanceConflict	{joint-iso-itu-t ms(9) smi(3) part7(7) parameter(5) 2}		o		
4	relationshipCardinalityViolation	{joint-iso-itu-t ms(9) smi(3) part7(7) parameter(5) 4}		o		

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Annex E
Illustration of representation methods

(This annex does not form an integral part of this Recommendation | International Standard)

This annex presents a graphical interpretation of the layout and use of the Relationship Class and Relationship Mapping Templates (see Figures E.1 and E.2).

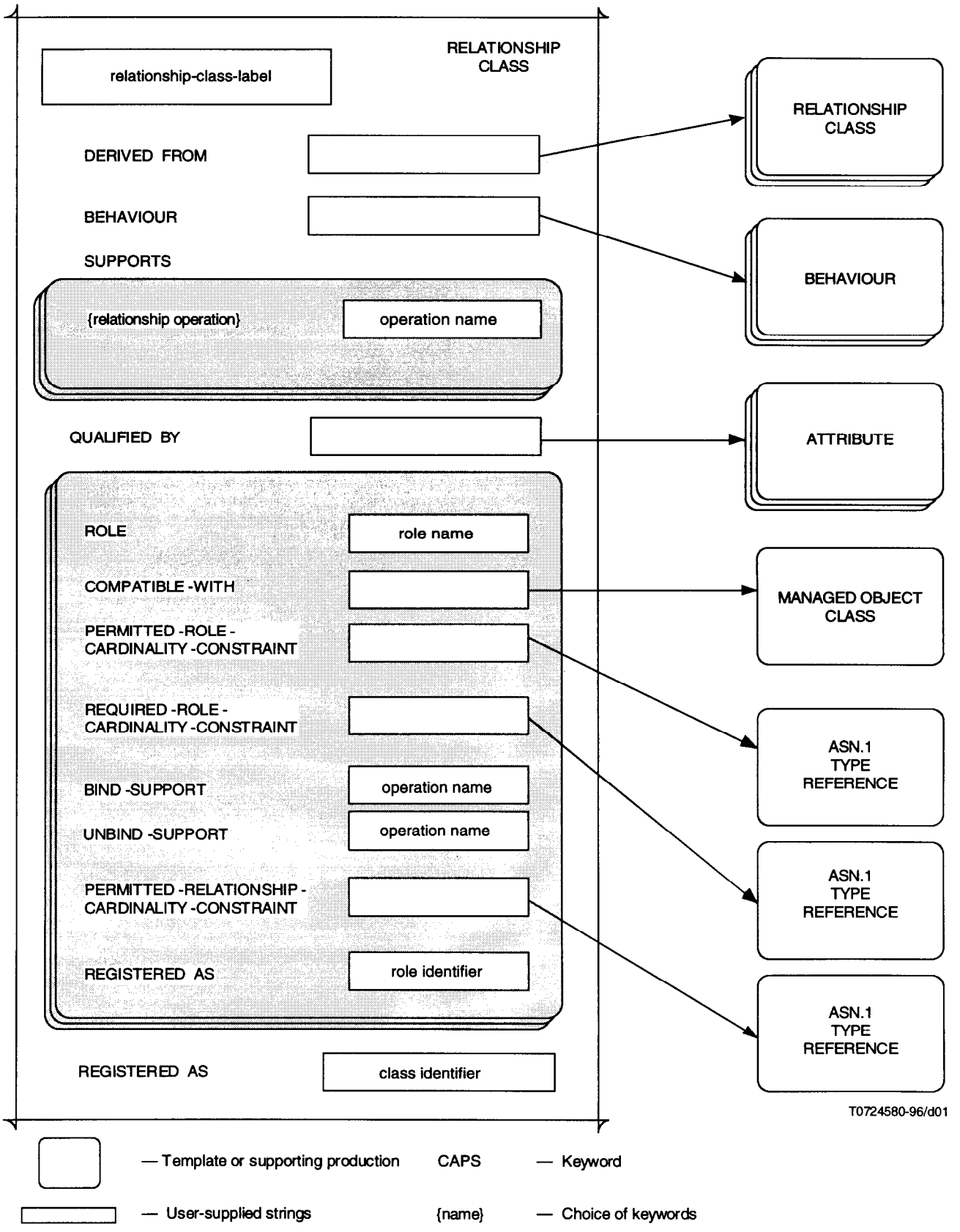

Figure E.1 – Relationship Class Template

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

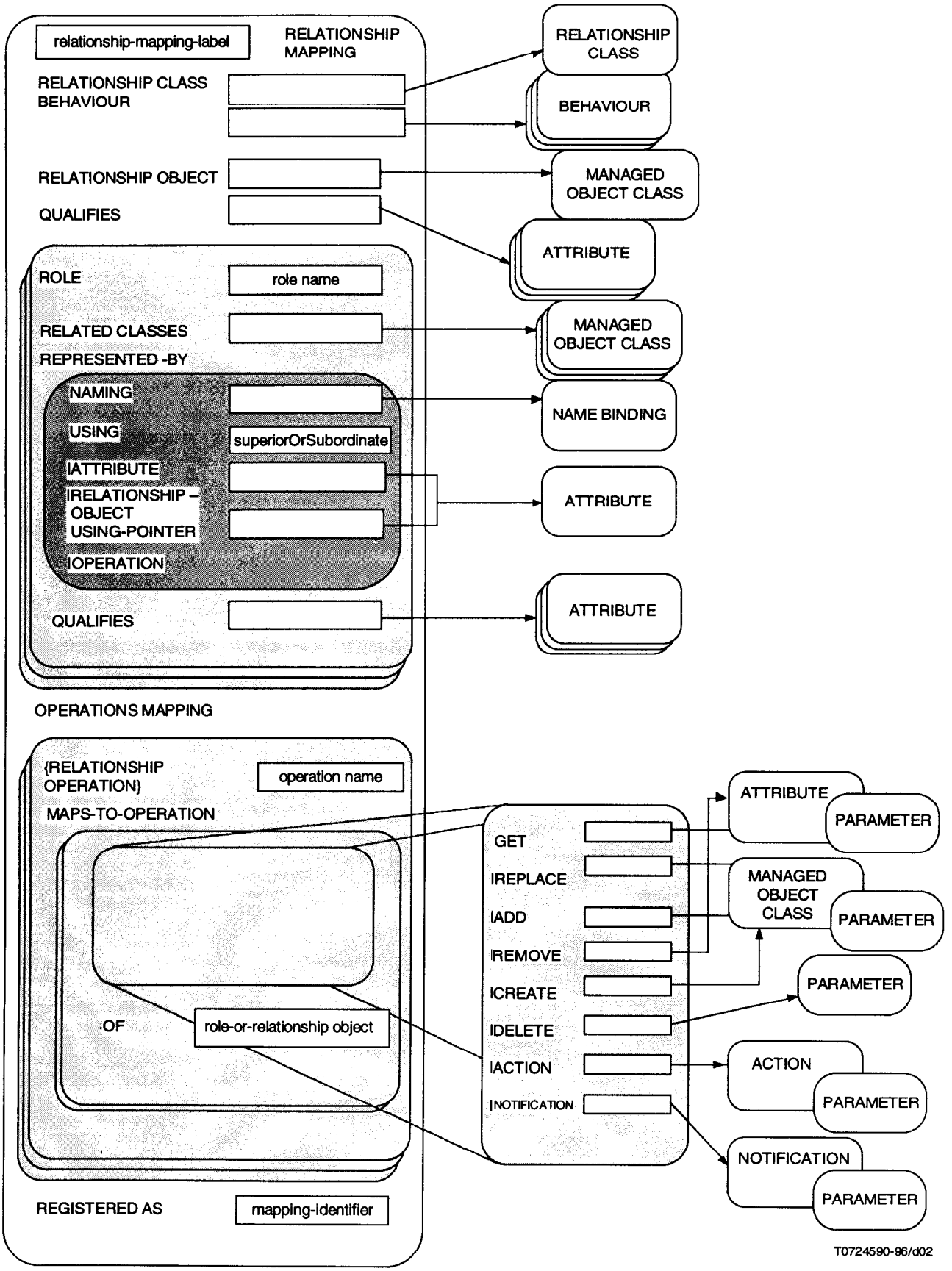

Figure E.2 – Relationsphip Mapping Template

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Annex F

Examples of use of templates

(This annex does not form an integral part of this Recommendation | International Standard)

The examples shown in this annex are intended to provide illustration of the concepts identified in this Recommendation | International Standard and to give examples of the use of the **RELATIONSHIP CLASS** and **RELATIONSHIP MAPPING** template notations. These examples are not intended to provide definitions which are necessarily useful in real implementations.

F.1 Allocation of object identifiers

```
GRMExample {joint-iso-itu-t ms(9) smi(3) part7(7) asn1Module(2) exampleASN1(99)}

DEFINITIONS ::=
BEGIN
    grmEx-Role OBJECT IDENTIFIER ::=
        {joint-iso-itu-t ms(9) smi(3) part7(7) grm-Role(13) exampleRole(99)}
    grmEx-RelationshipClass OBJECT IDENTIFIER ::=
        {joint-iso-itu-t ms(9) smi(3) part7(7) grm-RelationshipClass(11)exampleRelationshipClass(99)}
    grmEx-RelationshipMapping OBJECT IDENTIFIER ::=
        {joint-iso-itu-t ms(9) smi(3) part7(7) grm-RelationshipMapping(12) exampleRelationshipMapping(99)}
    grmEx-Object OBJECT IDENTIFIER ::=
        {joint-iso-itu-t ms(9) smi(3) part7(7) managedObjectClass(3) exampleObjectClass(99)}
    grmEx-Attribute OBJECT IDENTIFIER ::=
        {joint-iso-itu-t ms(9) smi(3) part7(7) attribute(7) exampleAttribute(99)}
    grmEx-NameBinding OBJECT IDENTIFIER ::=
        {joint-iso-itu-t ms(9) smi(3) part7(7) nameBinding(6) exampleNameBinding(99)}
    grmEx-Package OBJECT IDENTIFIER ::=
        {joint-iso-itu-t ms(9) smi(3) part7(7) package(4) examplePackage(99)}

    PersonName ::=      GraphicString
    SingleValued ::=    GroupObjects (SIZE (1))

    ZeroToTwo ::=       INTEGER (0..2)
    One ::=             INTEGER (1..1)
    OneToFive ::=       INTEGER (1..5)
    OneToMax ::=        INTEGER (1..MAX)
    Two ::=             INTEGER (2..2)
    TwoToMax ::=        INTEGER (2..MAX)

END
```

F.2 Symmetric relationship example

The following example shows how the relationship class template may be used to define a generic, single-role relationship between objects of the same class and how the relationship mapping template may be used to define a representation.

F.2.1 Symmetric relationship class definition

```
symmetricRelationship   RELATIONSHIP CLASS
    BEHAVIOUR   symmetricRelationshipBehaviour;

    SUPPORTS
        ESTABLISH   establishSymmetricRelationship,
        TERMINATE   terminateSymmetricRelationship,
        QUERY       querySymmetricRelationship;

    ROLE        peerRole
        PERMITTED-ROLE-CARDINALITY-CONSTRAINT                   GRMExample.TwoToMax
        REQUIRED-ROLE-CARDINALITY-CONSTRAINT                    GRMExample.Two
        PERMITTED-RELATIONSHIP-CARDINALITY-CONSTRAINT           GRMExample.One
    REGISTERED AS {GRMExample.grmEx-Role x};
REGISTERED AS {GRMExample.grmEx-RelationshipClass x};
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

symmetricRelationshipBehaviour BEHAVIOUR DEFINED AS "

INVARIANT: This relationship has one role – the peer role – for which the minimum permitted and minimum required role cardinality constraint is 2. The existence of an instance of this relationship class implies the existence of at least two corresponding managed objects fulfilling the peer role.

OPERATIONS:

ESTABLISH establishSymmetricRelationship

Signature: The class and identity of the proposed participants in the peer role to be bound in a new instance of the Symmetric Relationship class.

Precondition: The instance of the Symmetric Relationship relationship class does not exist.

Postcondition: The participants in the peer role exist; the instance of the Symmetric Relationship relationship class exists; the participants in the peer role referenced in the signature are bound in this instance of the Symmetric Relationship class.

TERMINATE terminateSymmetricRelationship

Signature: The identity of the Symmetric Relationship relationship instance to be terminated.

Precondition: The instance of the Symmetric Relationship relationship class referenced in the signature exists; the participants in the peer role bound in this instance of the Symmetric Relationship class exist.

Postcondition: The referenced instance of the Symmetric Relationship relationship class does not exist; the participants in the peer role which were bound in this instance of the Symmetric Relationship class exist.";

F.2.2 Symmetric relationship represented by a relationship object

symmetricRelationshipMapping RELATIONSHIP MAPPING
RELATIONSHIP CLASS symmetricRelationship;

BEHAVIOUR symmetricRelationshipMappingBehaviour BEHAVIOUR DEFINED AS "

This representation of the symmetric relationship uses a relationship object to represent the relationship. Objects fulfilling the peer role are identified by the peerPointer attribute of the symmetric relationship managed object.";;

RELATIONSHIP OBJECT symmetricRelationshipObject;

ROLE peerRole RELATED-CLASSES "CCITT Rec. X.721 | ISO/IEC 10165-2":top
REPRESENTED-BY RELATIONSHIP-OBJECT-USING-POINTER peerPointer;

OPERATIONS MAPPING
ESTABLISH establishSymmetricRelationship
MAPS-TO-OPERATION CREATE OF RELATIONSHIP OBJECT,
TERMINATE terminateSymmetricRelationship
MAPS-TO-OPERATION DELETE OF RELATIONSHIP OBJECT,
QUERY querySymmetricRelationship
MAPS-TO-OPERATION GET peerPointer OF RELATIONSHIP OBJECT;
REGISTERED AS {GRMExample.grmEx-RelationshipMapping x};

symmetricRelationshipObject MANAGED OBJECT CLASS
DERIVED FROM genericRelationshipObject;
CHARACTERIZED BY symmetricRelationshipPackage PACKAGE
ATTRIBUTES peerPointer GET-REPLACE ADD-REMOVE;;;
REGISTERED AS {GRMExample.grmEx-Object x};

peerPointer ATTRIBUTE
DERIVED FROM participantPointer;
REGISTERED AS {GRMExample.grmEx-Attribute x};

F.3 Dependency relationship example

The following example illustrates a relationship of dependency of one or more objects that assume a dependency role on a single object that assumes a parent role. The example illustrates mappings in terms of participant pointers, a relationship object, and naming.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS Not for Resale

The dependency relationship class might be useful to represent a directed acyclic graph by means of relationship specialization. In such a DAGDependency relationship class, the level of a dependent relative to its parent in the graph should be introduced and represented by the addition of an appropriate attribute. An invariant should be added stating that the value of the level attribute in a dependent must always be greater than that in its parents. The dependency relationship class might also be useful to represent a family relationship by the specialization of the person managed object class into three subclasses:

- parent;
- son; and
- daughter.

F.3.1 Dependency relationship class definition

dependency RELATIONSHIP CLASS
BEHAVIOUR dependencyBehaviour;

SUPPORTS
ESTABLISH establishDependency,
TERMINATE terminateDependency,
QUERY queryDependents,
QUERY queryParent;

QUALIFIED-BY timeOfEstablishment;

ROLE parentRole
PERMITTED-ROLE-CARDINALITY-CONSTRAINT GRMExample.One
REQUIRED-ROLE-CARDINALITY-CONSTRAINT GRMExample.One
PERMITTED-RELATIONSHIP-CARDINALITY-CONSTRAINT GRMExample.One
REGISTERED AS {GRMExample.grmEx-Role x},

ROLE dependentRole
PERMITTED-ROLE-CARDINALITY-CONSTRAINT GRMExample.OneToMax
REQUIRED-ROLE-CARDINALITY-CONSTRAINT GRMExample.One
PERMITTED-RELATIONSHIP-CARDINALITY-CONSTRAINT GRMExample.One
BIND-SUPPORT bindDependent
UNBIND-SUPPORT unbindDependent
REGISTERED AS {GRMExample.grmEx-Role x};
REGISTERED AS {GRMExample.grmEx-RelationshipClass x};

dependencyBehaviour BEHAVIOUR DEFINED AS "

INVARIANT: There exist two roles in this relationship class – parent role and dependent role. The existence of a participant in the dependent role implies the existence of at least one corresponding participant in the parent role. A managed object may not participate in both roles.

COMMENTS: An object instance fulfilling the dependent role may only participate in one instance of this dependency relationship, that is, the relationship cardinality is equal to one. An object instance able to fulfil the parent role may exist outside a dependency relationship an object fulfilling a dependent role shall not. The qualifying attribute, timeOfEstablishment, indicates the time, in UTC time format, of establishment of the relationship.

OPERATIONS:

ESTABLISH establishDependency

Signature: The class and identity of the proposed participant object in the dependent role to be created by the ESTABLISH operation; the class and identity of the proposed participant in the parent role.

Precondition: The proposed participant in the dependent role does not exist; the proposed participant in the parent role exists.

Postcondition: A new instance of the dependency relationship class exists; the participants in the parent role and the dependent role proposed in the ESTABLISH signature exist and are bound in the new instance of the dependency relationship class. The qualifying attribute, timeOfEstablishment, is set to the current value of UTC time.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

BIND bindDependent

Signature: The class and identity of a participant in the parent role; the class and identity of the proposed participant to be created in the dependent role.

Precondition: The participant in the parent role exists and is bound into an instance of the dependency relationship class; the proposed participant in the dependent role does not exist.

Postcondition: The participant in the dependent role referenced in the BIND signature exists and is bound into the same dependency relationship as that in which the participant in the parent role referenced in the BIND signature is bound.

UNBIND unbindDependent

Signature: The class and identity of a participant in the parent role; the class and identity of the participant in a dependent role.

Precondition: The two participants identified in the UNBIND signature exist and are bound into the same instance of a dependency relationship; the dependency relationship exists; there exists at least one other participant in the dependent role bound into the relationship.

Postcondition: The participant in the dependent role referenced in the UNBIND signature does not exist; all other participants bound into the instance of dependency relationship class exist and remain bound in the instance of the dependency relationship class.

TERMINATE terminateDependency

Signature: The identity of a dependency relationship instance to be terminated.

Precondition: The instance of the dependency relationship class identified in the signature exists; only a single participant in dependent role is bound into the identified dependency relationship.

Postcondition: The instance of the dependency relationship class referenced in the signature does not exists; the participant that was in the parent role exists. The participant(s) in the dependent role do not exist. The value of the qualifying attribute, timeOfEstablishment, is undefined.";

```
person MANAGED OBJECT CLASS
    DERIVED FROM "CCITT Rec. X.721 | ISO/IEC 10165-2":top;
    CHARACTERIZED BY
        personPackage    PACKAGE
            ATTRIBUTES personName GET;;;
REGISTERED AS {GRMExample.grmEx-Object x};

personName ATTRIBUTE
    WITH ATTRIBUTE SYNTAX GRMExample.PersonName;
REGISTERED AS {GRMExample.grmEx-Attribute x};

timeOfEstablishment    ATTRIBUTE
    WITH ATTRIBUTE SYNTAX UTCTime;
REGISTERED AS {GRMExample.grmEx-Attribute x};
```

F.3.2 Dependency relationship class represented by means of conjugate pointers

```
dependencyAttributeRepresentation    RELATIONSHIP MAPPING
    RELATIONSHIP CLASS dependency;
    BEHAVIOUR    dependencyAttributeRepresentationBehaviour;

    ROLE    parentRole
        RELATED-CLASSES                 aPerson
        REPRESENTED-BY ATTRIBUTE        parent
        QUALIFIES timeOfEstablishment,

    ROLE      dependentRole
        RELATED-CLASSES                 bPerson
        REPRESENTED-BY ATTRIBUTE        dependents;
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

OPERATIONS MAPPING
ESTABLISH establishDependency
MAPS-TO-OPERATION CREATE OF dependentRole,
TERMINATE terminateDependency
MAPS-TO-OPERATION DELETE OF dependentRole,
BIND bindDependent
MAPS-TO-OPERATION CREATE OF dependentRole ,
UNBIND unbindDependent
MAPS-TO-OPERATION DELETE OF dependentRole,
QUERY queryParent parentRole
MAPS-TO-OPERATION GET parent OF dependentRole,
QUERY queryDependents dependentRole
MAPS-TO-OPERATION GET dependents OF parentRole;
REGISTERED AS {GRMExample.grmEx-RelationshipMapping x};

dependencyAttributeRepresentationBehaviour BEHAVIOUR DEFINED AS "

This representation of the dependency relationship class uses conjugate participant pointers to represent an instance of the relationship; participant pointer consistency is to be maintained.

The relationship management operations ESTABLISH establishDependency and BIND bindDependent both map to a create of a participant in the dependent role: the distinction being that the relationship management operation ESTABLISH establishDependency is used when a participant is the first to fulfil the dependent role; the relationship management operation BIND bindDependent is used when there is at least one other participant in the dependent role at the time of binding. After creation of an object of class bPerson with the attribute, parent, identifying an object of class aPerson, the value of the attribute, dependents, in the object of class aPerson identifies the corresponding object of class bPerson.

Similarly, the relationship management operations TERMINATE terminateDependency and UNBIND unbindDependent both map to a delete of a participant in the dependentRole: the distinction being that the relationship management operation TERMINATE terminateDependency is used if there is only one participant fulfilling the dependentRole; the relationship management operation UNBIND unbindDependent is used if there is more than one participant fulfilling the dependentRole at time of deletion. Upon deletion of an object of class bPerson fulfilling the dependentRole, the value of the attribute, dependents, in the object of class aPerson object fulfilling the parentRole is modified by removing the identity of the corresponding object of class bPerson.

The QUERY queryDependents relationship management operation maps to a GET of the dependents attribute in the aPerson object fulfilling a parentRole; the QUERY queryParent operation maps to a GET of the parent attribute in the bPerson object fulfilling a dependentRole.

The creation of a bPerson managed object class (or bPerson subclass) results in the establishment of an instance of the dependency relationship with dependencyAttributeRepresentation RELATIONSHIP MAPPING when the value of the parent attribute in the object of class bPerson is set-by-create to an instance of a managed object of class aPerson and the value of the dependents attribute in the object of class aPerson is an empty set.

The creation of a bPerson (or bPerson subclass) managed object results in its being bound to an instance of the dependency relationship with dependencyAttributeRepresentation RELATIONSHIP MAPPING when the value of the parent attribute in the bPerson object is set-by-create to an instance of aPerson managed object class and the dependents attribute in the aPerson object is a non-empty set.

The deletion of a bPerson (or bPerson subclass) managed object results in its being unbound from an instance of the dependency relationship with dependencyAttributeRepresentation RELATIONSHIP MAPPING when the value of the dependents attribute in the aPerson object is not empty after the deletion and associated update of the dependents attribute.

The deletion of a bPerson (or bPerson subclass) managed object results in the termination of an instance of the dependency relationship with dependencyAttributeRepresentation RELATIONSHIP MAPPING when the value of the dependents attribute in the aPerson object is empty after the deletion and the associated update of the dependents attribute.";

aPerson MANAGED OBJECT CLASS
DERIVED FROM person;
CHARACTERIZED BY parentPackage PACKAGE
ATTRIBUTES dependents GET,
timeOfEstablishment GET;;;
REGISTERED AS {GRMExample.grmEx-Object x};

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
bPerson MANAGED OBJECT CLASS
    DERIVED FROM person;
    CHARACTERIZED BY
        dependentPackage PACKAGE
            ATTRIBUTES      parent PERMITTED VALUES GRMExample.SingleValued GET;;;
REGISTERED AS {GRMExample.grmEx-Object x};

dependent ATTRIBUTE
    DERIVED FROM participantPointer;
REGISTERED AS {GRMExample.grmEx-Attribute x};

parent ATTRIBUTE
    DERIVED FROM participantPointer;
REGISTERED AS {GRMExample.grmEx-Attribute x};
```

F.3.3 Dependency relationship class represented by means of a relationship object

```
dependencyObjectRepresentation    RELATIONSHIP MAPPING
    RELATIONSHIP CLASS dependency;
    BEHAVIOUR dependencyObjectRepresentationBehaviour;
    RELATIONSHIP OBJECT dependencyRelationshipObject
        QUALIFIES    timeOfEstablishment;

    ROLE parentRole
        RELATED-CLASSES person
        REPRESENTED-BY RELATIONSHIP-OBJECT-USING-POINTER parent,

    ROLE dependentRole
        RELATED-CLASSES person
        REPRESENTED-BY RELATIONSHIP-OBJECT-USING-POINTER dependents;

    OPERATIONS MAPPING
        ESTABLISH establishDependency
            MAPS-TO-OPERATION CREATE OF RELATIONSHIP OBJECT,
        TERMINATE terminateDependency
            MAPS-TO-OPERATION DELETE OF RELATIONSHIP OBJECT,
        BIND bindDependent dependentRole
            MAPS-TO-OPERATION ADD dependents OF RELATIONSHIP OBJECT,
        UNBIND unbindDependent dependentRole
            MAPS-TO-OPERATION REMOVE dependents OF RELATIONSHIP OBJECT,
        QUERY queryDependents dependentRole
            MAPS-TO-OPERATION GET dependents OF RELATIONSHIP OBJECT,
        QUERY queryParents parentRole
            MAPS-TO-OPERATION GET parent OF RELATIONSHIP OBJECT;
REGISTERED AS {GRMExample.grmEx-RelationshipMapping x};

dependencyObjectRepresentationBehaviour BEHAVIOUR DEFINED AS"
```

This representation of the dependency relationship uses a relationship object to represent an instance of the relationship and to relate the participants. The relationship management operation ESTABLISH establishDependency maps to a CREATE of a dependencyRelationshipObject object and the relationship management operation TERMINATE terminateDependency maps to a DELETE of the dependencyRelationshipObject object. The relationship management operation BIND bindDependent maps to an ADD operation on the dependents attribute in a dependencyRelationshipObject object. The relationship management operation UNBIND unbindDependent maps to a REMOVE operation on the dependents attribute in the dependencyRelationshipObject object.

The creation of a DependencyRelationshipObject object results in the establishment of a dependency relationship with the dependencyObjectRepresentation RELATIONSHIP MAPPING. Because the parent role is not dynamic (i.e. BIND-SUPPORT and UNBIND-SUPPORT are not defined for the parent role), the parent attribute within the DependencyRelationshipObject must be set-by-create to the value of exactly one instance of person object fulfilling the parentRole; the value of the parent attribute cannot be changed during the lifetime of the dependency operation.

The addition of a value representing a person object to the dependents attribute of a dependencyRelationshipObject object results in the person object's being bound into the relationship corresponding to the dependencyRelationshipObject object in the dependentRole.

The removal of a value representing a person object from the dependents attribute of a dependencyRelationshipObject object, results in the person object's being unbound from the relationship corresponding to the dependencyRelationshipObject object.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

The deletion of a dependencyRelationshipObject results in the termination of the corresponding dependency relationship with the dependencyObjectRepresentation RELATIONSHIP MAPPING.";

```
dependencyRelationshipObject      MANAGED OBJECT CLASS
    DERIVED FROM genericRelationshipObject;
    CHARACTERIZED BY
    dependencyRelationshipObjectPackage PACKAGE
        ATTRIBUTES
            dependents   GET-REPLACE ADD-REMOVE,
            parent     GET,
            timeOfEstablishment GET;
    REGISTERED AS {GRMExample.grmEx-Package x};;
REGISTERED AS {GRMExample.grmEx-Object x};
```

F.3.4 Dependency relationship represented by means of naming

```
dependencyNamingRepresentation    RELATIONSHIP MAPPING
    RELATIONSHIP CLASS dependency;
    BEHAVIOUR dependencyNamingRepresentationBehaviour ;

    ROLE parentRole
            RELATED-CLASSES cPerson
            REPRESENTED-BY NAMING aNameBinding USING SUPERIOR
            QUALIFIES    timeOfEstablishment,

    ROLE dependentRole
            RELATED-CLASSES person
            REPRESENTED-BY NAMING aNameBinding USING SUBORDINATE;

    OPERATIONS MAPPING
        ESTABLISH establishDependency
            MAPS-TO-OPERATION CREATE OF dependentRole,
        BIND bindDependent dependentRole
            MAPS-TO-OPERATION CREATE OF dependentRole,
        UNBIND unbindDependent dependentRole
            MAPS-TO-OPERATION DELETE OF dependentRole,
        TERMINATE terminateDependency
            MAPS-TO-OPERATION DELETE OF dependentRole,
        QUERY queryDependents dependentRole MAPS-TO-OPERATION GET
                "CCITT Rec. X.721 | ISO/IEC 10165-2":nameBinding OF dependentRole,
        QUERY queryParent parentRole MAPS-TO-OPERATION GET
                "CCITT Rec. X.721 | ISO/IEC 10165-2":nameBinding OF dependentRole;
REGISTERED AS {GRMExample.grmEx-RelationshipMapping x};

dependencyNamingRepresentationBehaviour BEHAVIOUR DEFINED AS "
```

This representation of the dependency relationship uses naming to represent an instance of the relationship.

The relationship management operations ESTABLISH establishDependency and BIND bindDependent both map to a create of a person (or person subclass) object participant in the dependentRole using a name binding with a cPerson (or cPerson subclass) object as the superior object in the parentRole. The distinction is that: the relationship management operation ESTABLISH establishDependency is used when the proposed participant in the dependent role would be the first object in the role; the relationship management operation BIND bindDependent is used when there is at least one other participant in the dependent role at the time of creation.

Similarly, the relationship management operations TERMINATE terminateDependency and UNBIND unbindDependent both map to a delete of a participant in the dependent role, the distinction being that: the relationship management operation TERMINATE terminateDependency is used if the participant is the only one fulfilling the dependentRole and the relationship management operations UNBIND unbindDependent is used if at least one other participant remains fulfilling the dependent role after deletion.

The QUERY queryDependents relationship management operation maps to a scoped get of the nameBinding attribute with a scope level of one on the person object in the parent role to determine the contained person objects that have the value of their name binding attribute equal to aNameBinding; such objects are fulfilling the dependents role.

The QUERY queryParent relationship management operation maps to a get of the nameBinding attribute of the subordinate object to determine that the value of its name binding attribute is equal to aNameBinding; subsequent analysis of the RDN of the subordinate object name will indicate the parent object.

--`,,```,,,,````-`-`,,`,,`,`,,`---

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

The creation of a person (or person subclass) managed object as a subordinate to a cPerson (or cPerson subclass) object with aNameBinding name binding results in the establishment of an instance of the dependency relationship with dependencyNamingRepresentation RELATIONSHIP MAPPING if there are no other subordinates with aNameBinding name binding.

The creation of a person (or person subclass) managed object as a subordinate of a cPerson (or cPerson subclass) object with aNameBinding name binding results in the binding of the created object into a dependency relationship with the dependencyNamingRepresentation RELATIONSHIP MAPPING if there is at least one other subordinate with aNameBinding name binding.

The deletion of a person (or person subclass) managed object bound in the dependent role of a dependency relationship with the dependencyNamingRepresentation RELATIONSHIP MAPPING, results in the unbinding of the deleted object from the dependency relationship when at least one other dependents with aNameBinding will exist after the deletion.

The deletion of a person (or person subclass) managed object bound in the dependent role of a dependency relationship with the dependencyNamingRepresentation RELATIONSHIP MAPPING, results in the termination of the dependency relationship when there will exist no other dependents with aNameBinding after the deletion.";

```
aNameBinding NAME BINDING
    SUBORDINATE OBJECT CLASS person AND SUBCLASSES;
    NAMED BY SUPERIOR OBJECT CLASS cPerson AND SUBCLASSES;
        WITH ATTRIBUTE personName;
        CREATE;
        DELETE;
REGISTERED AS {GRMExample.grmEx-NameBinding x};

cPerson    MANAGED OBJECT CLASS
    DERIVED FROM person;
    CHARACTERIZED BY
        timePackage    PACKAGE
            ATTRIBUTES timeOfEstablishment GET;;;
REGISTERED AS {GRMExample.grmEx-Object x};
```

F.4 General composition relationship example

This example illustrates the use of the relationship class template to define a generic composition relationship between a single object in a composite role and one or more objects in a component role and how the template might be refined. Such a relationship might be useful for modelling an assembly/sub-assembly relationship.

```
generalCompositionRelationship      RELATIONSHIP CLASS
    BEHAVIOUR generalCompositionRelationshipBehaviour;

    SUPPORTS
        ESTABLISH establishGeneralComposition,
        TERMINATE terminateGeneralComposition;

    ROLE    compositeRole
            PERMITTED-ROLE-CARDINALITY-CONSTRAINT    GRMExample.OneToOne
            REQUIRED-ROLE-CARDINALITY-CONSTRAINT     GRMExample.OneToOne
    REGISTERED AS {GRMExample.grmEx-Role x},

    ROLE    componentRole
            PERMITTED-ROLE-CARDINALITY-CONSTRAINT    GRMExample.OneToMax
            REQUIRED-ROLE-CARDINALITY-CONSTRAINT     GRMExample.OneToOne
            BIND-SUPPORT bindComponent
            UNBIND-SUPPORT unbindComponent
    REGISTERED AS {GRMExample.grmEx-Role x};
REGISTERED AS {GRMExample.grmEx-RelationshipClass x};

generalCompositionRelationshipBehaviour BEHAVIOUR DEFINED AS "
```

INVARIANT: The existence of an instance of this relationship class implies the existence of exactly one participant in the composite role and one or more participants in the component role. At least one property of the composite participant is such that it depends upon properties of the components. At least the identity of the composite participant is such that it is independent of the existence or properties of the components; that is, creating, updating, or deleting any component does not change the identity of the composite.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

OPERATIONS:

ESTABLISH establishGeneralComposition

Signature: The class and identity of the proposed participant in the composite role and the class and identity of the proposed participant(s) in the component role to be bound in an instance of the generalCompositionRelationship.

Precondition: The proposed participants are not already bound in the same instance of the generalCompositionRelationship class or a subclass thereof.

Postcondition: An instance of the generalCompositionRelationship class exists; the participants referenced in the signature are bound into this instance of the generalCompositionRelationship class.

BIND bindComponent

Signature: The class and identity of a proposed participant in the component role; the identity of a generalCompositionRelationship.

Precondition: The referenced instance of the generalCompositionRelationship class exists; the proposed participant in the component role is not bound into this instance of generalCompositionRelationship class; there exists at least one other participant in the component role bound into this instance of the generalCompositionRelationship class.

Postcondition: The participant in the component role referenced in the signature exists and is bound in this instance of the generalCompositionRelationship class.

UNBIND unbindComponent

Signature: The class and identity of a participant in the component role; the identity of a generalCompositionRelationship.

Precondition: The instance of the generalCompositionRelationship class referenced in the signature exists; the participant in the component role referenced in the signature is bound into the referenced instance of generalCompositionRelationship class; there exists at least one other participant in the component role bound into the referenced instance of the generalCompositionRelationship class.

Postcondition: The referenced participant in the component role exists but is not bound into the referenced instance of the generalCompositionRelationship class; the referenced instance of the generalCompositionRelationship class exists.

TERMINATE terminateGeneralComposition

Signature: The identity of a generalCompositionRelationship instance.

Precondition: The referenced instance of the generalCompositionRelationship class exists.

Postcondition: The referenced instance of the generalCompositionRelationship class does not exist; the participants in the composite role and in the component role that were bound into the relationship exist.";

F.4.1 Subclass of general composition relationship

subclassedCompositionRelationship RELATIONSHIP CLASS
DERIVED FROM generalCompositionRelationship;
BEHAVIOUR subclassedCompositionRelationshipBehaviour
BEHAVIOUR DEFINED AS"

This relationship class refines the required role cardinality of the component role of the generalCompositionRelationship class to be the range 1 to 5; all other characteristics of this relationship class are inherited from the generalCompositionRelationship class.";;

ROLE componentRole
REQUIRED-ROLE-CARDINALITY-CONSTRAINT GRMExample.OneToFive;
REGISTERED AS {GRMExample.grmEx-Object x};

F.5 Access control domain example

accessControlDomain RELATIONSHIP CLASS
BEHAVIOUR accessControlDomainBehaviour BEHAVIOUR DEFINED AS"

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

This relationship class binds managed objects which are subject to access control (memberObjectRole) to managed objects representing the access enforcement function (aefRole) and access decision function (adfRole) respectively.";;

SUPPORTS QUERY queryAccessControlDomain;

ROLE memberObjectRole
REQUIRED-ROLE-CARDINALITY-CONSTRAINT GRMExample.OneToTwo
BIND-SUPPORT bindMember
UNBIND-SUPPORT unbindMember
REGISTERED AS {GRMExample.grmEx-Role memberObjectRoleArc(1) },

ROLE aefRole
COMPATIBLE-WITH "ITU-T Rec. X.741 | ISO/IEC 10164-9":notificationEmitter
PERMITTED-ROLE-CARDINALITY-CONSTRAINT GRMExample.OneToOne
REQUIRED-ROLE-CARDINALITY-CONSTRAINT GRMExample.OneToOne
REGISTERED AS {GRMExample.grmEx-Role aefRoleArc(2) },

ROLE adfRole
COMPATIBLE-WITH "ITU-T Rec. X.741 | ISO/IEC 10164-9":accessControlRules
PERMITTED-ROLE-CARDINALITY-CONSTRAINT GRMExample.OneToOne
REQUIRED-ROLE-CARDINALITY-CONSTRAINT GRMExample.OneToOne
REGISTERED AS {GRMExample.grmEx-Role adfRoleArc(3) };
REGISTERED AS {GRMExample.grmEx-RelationshipClass accessControlDomainArc(1) };

F.5.1 Access control domain relationship represented by attributes and naming

simpleAccessControlDomain RELATIONSHIP MAPPING
RELATIONSHIP CLASS accessControlDomain;
BEHAVIOUR simpleAccessControlDomainBehaviour BEHAVIOUR DEFINED AS"

In this mapping of the accessControlDomain managed relationship class, the accessControlDomainObject class (a subclass of the accessControlRules class) participates in the adfRole and the notificationEmitter class participates in the aefRole; any managed object may participate in the memberObjectRole. The memberObjectAttribute in the accessControlDomainObject identifies the participants in the memberObjectRole and the notificationEmitter-accessControlRules name binding contains the aef participant within the adf participant.

The QUERY queryAccessControlDomain relationship management operation maps to two operations, namely :

(a) a GET of the memberObjectAttribute of the object fulfilling the adfRole; followed by

(b) a scoped GET of the nameBinding attribute with a scope level of one on the object fulfilling the adf role to determine the contained objects that have the value of their name binding attribute equal to "ITU-T Rec. X.741 | ISO/IEC 10164-9":notificationEmitter-accessControlRules.";;

ROLE memberObjectRole RELATED-CLASSES "ITU-T Rec. X721 | ISO/IEC 10165-2":top,
REPRESENTED-BY ATTRIBUTE memberObjectAttribute ;

ROLE aefRole
RELATED-CLASSES "ITU-T Rec. X.741 | ISO/IEC 10164-9":notificationEmitter
REPRESENTED-BY NAMING
"ITU-T Rec. X.741 | ISO/IEC 10164-9":notificationEmitter-accessControlRules USING
SUBORDINATE,

ROLE adfRole
RELATED-CLASSES accessControlDomainObject
REPRESENTED-BY NAMING
"ITU-T Rec. X.741 | ISO/IEC 10164-9":notificationEmitter-accessControlRules USING
SUPERIOR,

OPERATIONS MAPPING
BIND bindMember memberObjectRole
MAPS-TO-OPERATION ADD memberObjectAttribute OF adfRole,
UNBIND unbindMember memberObjectRole
MAPS-TO-OPERATION REMOVE memberObjectAttribute OF adfRole,
QUERY queryAccessControlDomain memberObjectRole
MAPS-TO-OPERATION GET memberObjectAttribute OF adfRole
MAPS-TO-OPERATION GET
"CCITT Rec. X.721 | ISO/IEC 10165-2":nameBinding OF adfRole;
REGISTERED AS {GRMExample.grmEx-RelationshipMapping simpleAccesControlDomainArc(1) };

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
accessControlDomainObject MANAGED OBJECT CLASS
   DERIVED FROM "ITU-T Rec. X.741 | ISO/IEC 10164-9":accessControlRules;
      CHARACTERIZED BY accessControlDomainPackage PACKAGE
         BEHAVIOUR accessControlDomainBehaviour
            BEHAVIOUR DEFINED AS "
```

Membership of the access control domain is identified and modified by operations upon the memberObjectAttribute.";;

```
         ATTRIBUTES memberObjectAttribute GET-REPLACE ADD-REMOVE;;;
REGISTERED AS {GRMExample.grmEx-Object accessControlDomainObjectArc(1) };
```

F.5.2 Access control domain relationship representation using a relationship object

```
coordinatedAccessControlDomain RELATIONSHIP MAPPING
   RELATIONSHIP CLASS accessControlDomain;
   BEHAVIOUR coordinatedAccessControlDomainBehaviour
      BEHAVIOUR DEFINED AS"
```

In this mapping of the accessControlDomain managed relationship class, the accessControlRules class participates in the adfRole and the notificationEmitter class participates in the aefRole; any managed object may participate in the memberObjectRole. The relationship is represented by the accessControlDomainCoordinator, a subclass of the genericRelationshipObject, using the memberObjectAttribute, aefAttribute, and adfAttribute attributes.

The QUERY queryAccessControlDomain relationship management operation maps to three GET operations on the relationship object, namely:

(a) a GET of the memberObjectAttribute;

(b) a GET of the aefAttribute; and

(c) a GET of the adfAttribute.";;

```
RELATIONSHIP OBJECT accessControlDomainCoordinator;

   ROLE     memberObjectRole
         RELATED-CLASSES "CCITT Rec. X.721 | ISO/IEC 10165-2":top
         REPRESENTED-BY RELATIONSHIP-OBJECT-USING-POINTER memberObjectAttribute,

   ROLE     aefRole
         RELATED-CLASSES
         "ITU-T Rec. X.741 | ISO/IEC 10164-9":notificationEmitter
         REPRESENTED-BY RELATIONSHIP-OBJECT-USING-POINTER aefAttribute,

   ROLE     adfRole
         RELATED-CLASSES "ITU-T Rec. X.741 | ISO/IEC 10164-9":accessControlRules
         REPRESENTED-BY RELATIONSHIP-OBJECT-USING-POINTER adfAttribute;

   OPERATIONS MAPPING
      BIND bindMember
         MAPS-TO-OPERATION ADD memberObjectAttribute OF RELATIONSHIP OBJECT,
      UNBIND unbindMember
         MAPS-TO-OPERATION REMOVE memberObjectAttribute OF RELATIONSHIP OBJECT,
      QUERY queryAccessControlDomain
         MAPS-TO-OPERATION GET memberObjectAttribute OF RELATIONSHIP OBJECT
         MAPS-TO-OPERATION GET aefAttribute OF RELATIONSHIP OBJECT
         MAPS-TO-OPERATION GET adfAttribute OF RELATIONSHIP OBJECT;
REGISTERED AS
   {GRMExample.grmEx-RelationshipMapping coordinatedAccessControlDomainArc(2)};

accessControlDomainCoordinator MANAGED OBJECT CLASS
   DERIVED FROM genericRelationshipObject;
   CHARACTERIZED BY accessControlDomainCoordinatorPackage PACKAGE

      ATTRIBUTES
         memberObjectAttribute
            ATTRIBUTE DERIVED FROM participantPointer;
            REGISTERED AS { GRMExample.grmEx-Attribute memberObjectAttributeArc(1) };
               GET-REPLACE ADD-REMOVE,
```

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

```
        aefAttribute
            ATTRIBUTE DERIVED FROM participantPointer;
            REGISTERED AS { GRMExample.grmEx-Attribute aefAttributeArc(1) }; GET,
        adfAttribute
            ATTRIBUTE DERIVED FROM participantPointer;
            REGISTERED AS { GRMDExample.grmEx-Attribute adfAttributeArc(1) }; GET;
REGISTERED AS {GRMExample.grmEx-Object accessControlDomainCoordinatorArc(1)};
```


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Annex G

Commentary

(This annex does not form an integral part of this Recommendation | International Standard)

G.1 Introduction

The following commentary has been developed from the list of issues that was maintained over the development of the standard.

G.2 Dependency between managed objects in a managed relationship

- **Issue:** The essence of the GRM is that managed objects participating in a managed relationship affect one another; this is expressed as an invariant over the properties of the participants. How should this invariant be specified?

- **Commentary:** Previous drafts of the GRM have attempted to single out, and provide notational support for, various types of invariants such as attribute-value constraints or existence dependency. Recognizing that behaviour templates can potentially express all types of invariants, the inclusion of notational support for particular invariant types has not retained consistent NB support. Hence all invariants are expressed in terms of managed relationship behaviour. The invariant is specified in terms of properties of the managed relationship (roles, relationship management operations, etc.). The relationship mapping template may provide a mapping of the invariant in terms of the representation method (participating managed objects, relationship objects, participant pointers, etc.).

Invariants are, by definition, requirements and conformant implementations must meet these requirements. The GRM prescribes no general mechanism for meeting these requirements though the relationship mapping template does provide the tools for managed relationship specifiers to prescribe such mechanisms in particular cases of relationship mapping.

G.3 Consistency of views

- **Issue:** A representation method may specify management information (e.g. participant pointers, relationship objects) that is related solely to the representation method. How is this information to be kept consistent?

- **Commentary:** It is a fundamental concept of the GRM that the semantics of the managed relationship be consistently expressed in the elements of an implementation; in other words the relationship drives the representation, not the other way round. Thus, if a relationship mapping chooses to represent the semantics of managed object participation as conjugate pointers in the participant objects, then an implementation must ensure that the pointers are always consistent. Furthermore, if a relationship mapping chooses to represent the BIND operation as an attribute-based addOperation on one of a pair of conjugate participant pointers, an implementation is required to adjust the other pointer to maintain consistency. The GRM only specifies requirements for consistency of information; it does not specify mechanisms for maintaining consistency either within a single managed system or across multiple managed systems.

G.4 Expression of relationship management operations and notifications

- **Issue:** How are relationship management operations and notifications expressed and how are they mapped to systems management operations?

- **Commentary:** Relationship management operations and notifications are expressed in terms of a number of prototypical operations and a notification which are subsequently mapped to systems management operations and notifications. The final text gives full details and examples of the technique.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

G.5 Generic management

- **Issue:** Can mechanisms be defined to permit the management of a broad range of managed relationship types?

- **Commentary:** A companion standard, the General Relationship Management Function, was raised in parallel to the GRM. However, given the broad range of relationship types that could be defined, subsequent investigation indicated that generic management tools for managing relationships across the board would be of limited use. It was thus agreed to provide managed-relationship specifiers tools to specify such mechanisms on a relationship-by-relationship basis. The GRM defines a template for mapping managed relationship operations and notifications and defines generic management information.

However, managed relationship subclasses are consistent with their superclasses and, in this sense, generic management is provided within an inheritance hierarchy.

G.6 Relationship awareness

- **Issue:** How does a managed object "know" that it is in a managed relationship?

- **Commentary:** An anthropomorphic view of a relationship is not helpful. A managed object must fulfil the requirements of the role as modelled by the managed relationship. In the final analysis, an implementation must ensure that the semantics of the relationship are preserved and that implementations of managed objects fulfil the requirements of the role.

G.7 Role specification

- **Issue:** Should a role be specified out of line?

- **Commentary:** Initially roles were seen as independent, re-usable specifications. Subsequent reflection has indicated that roles are intimately connected with their managed relationship and out-of-line specification is of limited value.

G.8 Re-use of specifications

- **Issue:** Re-use of specifications is an important facet of OSI systems management; how is it implemented in the GRM?

- **Commentary:** Subclasses of managed relationship classes are *consistent* with their supertypes in that an instance of a subclass can be substituted for an instance of a superclass without affect the operation of the managing system. Subclasses are, in fact, subtypes within the Open Distributed Processing definition of the term. Thus the inheritance and specialization tools provide a mechanism for re-use of specifications.

G.9 AND SUBCLASSES

- **Issue:** The **AND SUBCLASSES** clause was not carried over from **GDMO name-binding template** to the **role-mapping-specification** supporting production of the **RELATIONSHIP MAPPING** template.

- **Commentary:** The ability of a subclass to support of a role is regarded as a fundamental property of a managed object and should thus be unconditionally inherited.

G.10 Relationship between relationships

- **Issue:** How can relationships between relationships be modelled?

- **Commentary:** Whilst the GRM models relationships between managed objects, if relationships are represented by relationship objects, then there is no reason why the GRM cannot model relationships between relationships. The GRM provides no particular support for this, but any additional semantics could be specified in the BEHAVIOUR template.

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

G.11 Naming Scope of relationship objects

- **Issue:** What should be the scope for the naming of relationship objects?
- **Commentary:** There was discussion regarding the naming of all relationship objects in a managed system within the scope of a single object of a particular class – often referred to as the *anchor object* class – particularly with a view to being able to discover all relationship objects in a managed system by means of CMIS scoping. It was concluded that, since existing management standards regard naming structure as a local matter, it would be inconsistent for the GRM to prescribe a particular structure.

G.12 Allowable representation methods

- **Issue:** Can representation methods represent all types of relationships?
- **Commentary:** No; some representation methods are inherently restricted in the type of the relationships they can represent. Table G.1 shows the types of relationships that can be represented by the various methods.

Table G.1 – Allowable representation methods

	Relationship Cardinality = 1			Relationship Cardinality > 1		
	Role Cardinality			Role Cardinality		
Representation Method	1:n	n:m	n:m:p	1:n	n:m	n:m:p
Naming	Yes	No	No	No	No	No
Participant Pointers	Yes	Yes	Yes	No	No	No
Relationship Object	Yes	Yes	Yes	Yes	Yes	Yes
Systems Management Operations	Yes	Yes	Yes	Yes	Yes	Yes

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

ICS 35.100.70

Descriptors: data processing, information interchange, network interconnection, open systems interconnection, application layer, management, information, models.

Price based on 43 pages

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale