INTERNATIONAL STANDARD

ISO/IEC 10164-16

First edition
1997-05-15

AMENDMENT 1
1998-12-15

Information technology — Open Systems Interconnection — Systems Management: Management knowledge management function

AMENDMENT 1: Extension for General Relationship Model

Technologies de l'information — Interconnexion de systèmes ouverts (OSI) — Gestion-systèmes: Fonction de gestion de connaissance de gestion

AMENDEMENT 1: Extension pour modèle général de relation

Reference number
ISO/IEC 10164-16:1997/Amd.1:1998(E)

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.

Amendment 1 to ISO/IEC 10164-16:1997 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 33, *Distributed application services*, in collaboration with ITU-T. The identical text is published as ITU-T Rec. X.750/Amd.1.

INTERNATIONAL STANDARD

ITU-T RECOMMENDATION

INFORMATION TECHNOLOGY – OPEN SYSTEMS INTERCONNECTION – SYSTEMS MANAGEMENT: MANAGEMENT KNOWLEDGE MANAGEMENT FUNCTION

AMENDMENT 1

Extension for General Relationship Model

1) Subclause 3.6

Add references to these SMO-defined terms inserted among existing terms in alphabetical order:

e) managed object class

and renumber the list as from the existing e) *which becomes* f), *to the end.*

2) Subclause 3.11

Add references to these GRM-defined terms inserted before and after existing term in alphabetical order:

binding;

binding support;

managed relationship class;

relationship cardinality;

relationship management notification;

relationship management operation;

role cardinality;

relationship mapping specification;

relationship class specification;

role;

unbinding;

unbinding support.

3) Clause 4

Insert the following abbreviation by alphabetical order:

GRM General relationship model

4) Clause 6

In this clause change:

– *Definition knowledge*: Information on the formal specification of managed object classes, name bindings, etc., e.g. templates for classes, name bindings.

to:

– *Definition knowledge*: Information on the formal specification of managed object classes and managed relationship classes and related specifications, e.g. templates for classes, attributes, name bindings.

Add an X *in the Table 1 cell in the* Definition knowledge *column in the* Relationship knowledge *row.*

5) Subclause 6.3

In the 2nd item:

relationship role bindings

to:

relationship mapping specifications

6) Subclauses 6.3, 6.5, 7.1.1.1, 8.1.1.2, Table 8, A.2.1 and A.2.3

In subclauses 6.3, 6.5, 7.1.1.1, 8.1.1.2, Table 8, and in behaviour text in A.2.1 and A.2.3, replace each occurrence of the term relationship class *that is not preceded by* managed *with the term defined in GRM* managed relationship class *but do not change the label* **supportedRelationshipClassList**.

7) Subclause 7.1.2

In this subclause change GDMO and ASN.1 specifications *to* GDMO, GRM and ASN.1 specifications.

8) Subclauses 7.1.2, 8.1.2.12, 8.6.2 and A.3.3

In subclauses 7.1.2, 8.1.2.12, 8.6.2 and A.3.3, change all occurrences of GDMO template(s) *to* template(s).

9) Subclause 7.3

Add an X *in the four Table 2 cells that are in either the* Definition managed objects *or* Definition Directory objects *columns and that are in either the* Managed object class knowledge *or* Relationship knowledge *rows*. *Change both occurrences of* X (Note 7) *to* X *and eliminate Note 7*.

10) New subclauses 8.1.2.12 and 8.1.2.13

Renumber subclause 8.1.2.12 as 8.1.2.14 and insert two new subclauses before it:

8.1.2.12 Relationship class template managed object

The relationship class template managed object represents a GRM relationship class specification.

The relationship class template managed object class is defined as a subclass of the template managed object class. Its mandatory, read-only attributes are:

a) derived from;

b) behaviour.

Its optional, read-only attributes are:

a) supports;

b) qualified by;

c) role specifier.

8.1.2.13 Relationship mapping template managed object

The relationship mapping template managed object represents a GRM relationship mapping specification.

The relationship mapping template managed object class is defined as a subclass of the template managed object class. Its mandatory, read-only attributes are:

a) mapped relationship class;

b) behaviour;

c) role mapping specification;

d) registered as.

Its optional, read-only attributes are:

a) relationship object;

b) operations mapping.

11) Subclause 8.6.2

In this subclause, change the two occurrences of GDMO specifications *to* GDMO, GRM or ASN.1 specifications.

Append to this subclause the following paragraphs:

The Directory auxiliary class relationship class template is defined for inclusion in Directory entries of Directory structural class Registered information and represents a GRM relationship class specification. It must contain the attributes Derived from and Behaviour. It may contain the attributes Supports, Qualified by, and Role specifier.

The Derived from attribute identifies the immediate superclasses of a managed relationship class. The Behaviour attribute specifies the behaviour associated with the managed relationship class. The Supports attribute defines the relationship management operations and notifications that a managed relationship supports. The Qualified by attribute specifies the attributes that are associated with the managed relationship as a whole. The Role specifier attribute identifies the roles associated with the managed relationship class.

The Directory auxiliary class Relationship mapping template is defined for inclusion in Directory entries of Directory structural class Registered information and represents a GRM relationship mapping specification. It must contain the attributes Behaviour, Mapped relationship class, Role mapping specification set, and Registered as. It may contain the attributes Relationship object and Operations mapping.

The Behaviour attribute defines the behaviour associated with the relationship mapping. The Mapped relationship class attribute specifies the managed relationship class to which the represented relationship mapping is related. The Role mapping specification set attribute identifies candidate managed object classes that may fulfil a given role. The Registered as attribute contains the object identifier of the represented relation mapping. The Relationship object attribute indicates the class of a relationship object that shall represent the managed relationship. The Operations mapping attribute specifies the mapping of a relationship management operation to one or more systems management operations.

12) Subclause 11.1.1.2

Add entries to Table 6 for relationship class template and relationship mapping template *(inserting before the entry for the template class)*.

13) Subclause 11.1.5.1

Add entries to Table 9 for the new attributes mentioned in the new subclauses 8.1.2.12 and 8.1.2.13.

14) Subclause 11.1.5.2

Insert in Table 14 the following entries before the entry for the template class:

Relationship class template	managementRelationshipClass
Relationship mapping template	managementRelationshipMapping

15) Subclause 11.1.6.1

Insert in Table 15 the following entries before the entry for the template class:

relationshipClassTemplate	managementRelationshipClass
relationshipMappingTemplate	managementRelationshipMapping

16) Subclause 11.2.1.2

Add to Table 17 the following entries (inserting in alphabetical order):

Mapped relationship class	mappedRelationshipClass
Operations mapping	operationsMapping
Qualified by	qualifiedBy
Relationship object	relationshipObject
Role mapping specification set	roleMappingSpecificationSet
Role specifier	roleSpecifier
Supports	supports

17) Subclause A.1

In this subclause change:

--%PRAGMA version BIT STRING {v1990(0), v1994(1)} ::= {v1990, v1994}

to:

--<ASN1.Version 1990, 1994 MKMD>--

18) Subclause A.2.4

In this subclause change:

--%PRAGMA version BIT STRING {v1990(0), v1994(1)} ::= v1990

to:

--<ASN1.Version 1990 RepertoireASN1Module >--

Also in this subclause change:

FROM Attribute-ASN1Module {joint-iso-ccitt(2) ms(9) smi(3) part2(2) asn1Module(2)}

to:

FROM Attribute-ASN1Module {joint-iso-itu-t(2) ms(9) smi(3) part2(2) asn1Module(2) 1}

19) Subclause A.3.1

Add to this subclause these managed object class definitions (inserting in alphabetical order):

```
relationshipClassTemplate MANAGED OBJECT CLASS
        DERIVED FROM          template;
        CHARACTERIZED BY  relationshipClassPackagePACKAGE
        BEHAVIOUR relationshipClassBehaviour BEHAVIOUR DEFINED AS !
            A managed object with this behaviour represents a RELATIONSHIP CLASS template.
            !;;
        ATTRIBUTES
            derivedFrom GET,
            behaviour GET;;;
        CONDITIONAL PACKAGES
            relationshipClassSupportsPackage          PACKAGE
                BEHAVIOUR relationshipClassSupportsBehaviour BEHAVIOUR DEFINED AS !
                    A managed object with this behaviour represents a template that was defined with the SUPPORTS
                    construct.
                    !;;
                ATTRIBUTES
                    supports GET;
            REGISTERED AS {MKMD.mkmPackage 5}; PRESENT IF "managed object represents a template that was
                    defined with the SUPPORTS construct",
            relationshipClassQualifiedByPackage      PACKAGE
                BEHAVIOUR relationshipClassQualifiedByBehaviour BEHAVIOUR DEFINED AS !
                    A managed object with this behaviour represents a template that was defined with the QUALIFIED
                    BY construct.
                    !;;
                ATTRIBUTES
                    qualifiedBy GET;
            REGISTERED AS {MKMD.mkmPackage 6}; PRESENT IF "managed object represents a template that was
                    defined with the QUALIFIED BY construct",
            relationshipClassRoleSpecifierPackage    PACKAGE
                BEHAVIOUR relationshipClassRoleSpecifierBehaviour BEHAVIOUR DEFINED AS !
                    A managed object with this behaviour represents a template that was defined with the role-specifier
                    supporting production.
                    !;;
                ATTRIBUTES
                    roleSpecifier GET;
            REGISTERED AS {MKMD.mkmPackage 7}; PRESENT IF "managed object represents a template that was
                    defined with the role-specifier supporting production";
REGISTERED AS {MKMD.mkmMObjectClass 16};

        relationshipMappingTemplate MANAGED OBJECT CLASS
            DERIVED FROM          template;
            CHARACTERIZED BY  relationshipMappingPackage PACKAGE
            BEHAVIOUR relationshipMappingBehaviour BEHAVIOUR DEFINED AS !
                A managed object with this behaviour represents a RELATIONSHIP MAPPING template.
                !;;
            ATTRIBUTES
                mappedRelationshipClass GET,
                behaviour GET,
                roleMappingSpecificationSet GET,
                registeredAs GET;;;
            CONDITIONAL PACKAGES
                relationshipMappingRelationshipObjectPackage PACKAGE
                    BEHAVIOUR      relationshipMappingRelationshipObjectBehaviour      BEHAVIOUR DEFINED AS !
                        A managed object with this behaviour represents a template that was defined with the
                        RELATIONSHIP OBJECT construct.
                        !;;
                    ATTRIBUTES
                        relationshipObject GET;
                REGISTERED AS {MKMD.mkmPackage 8}; PRESENT IF "managed object represents a template that
                        was defined with the RELATIONSHIP OBJECT construct",
```

```
        relationshipMappingOperationsMappingPackage PACKAGE
            BEHAVIOUR relationshipMappingOperationsMappingBehaviour BEHAVIOUR DEFINED AS !
                A managed object with this behaviour represents a template that was defined with the
                OPERATIONS MAPPING construct.
                !;;
            ATTRIBUTES
                operationsMapping GET;
        REGISTERED AS {MKMD.mkmPackage 9}; PRESENT IF "managed object represents a template that
                was defined with the OPERATIONS MAPPING construct";
REGISTERED AS {MKMD.mkmMObjectClass 17};
```

20) Subclause A.3.3

In this subclause change:

```
BEHAVIOUR derivedFromBehaviour BEHAVIOUR DEFINED AS !
    This set-valued attribute identifies the immediate superclasses of a managed object class.
    !;;
```

to:

```
BEHAVIOUR derivedFromBehaviour BEHAVIOUR DEFINED AS !
    This set-valued attribute identifies the immediate superclasses of a managed object class or managed
    relationship class.
    !;;
```

Add to subclause A.3.3 these attribute definitions (inserting in alphabetical order):

```
mappedRelationshipClass ATTRIBUTE
    WITH ATTRIBUTE SYNTAX GrmAttributeDefinitionModule.MappedRelationshipClass;
    MATCHES FOR EQUALITY;
REGISTERED AS {MKMD.mkmAttribute 46};

operationsMapping ATTRIBUTE
    WITH ATTRIBUTE SYNTAX GrmAttributeDefinitionModule.OperationsMapping;
    MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
REGISTERED AS {MKMD.mkmAttribute 47};

qualifiedBy ATTRIBUTE
    WITH ATTRIBUTE SYNTAX GrmAttributeDefinitionModule.QualifiedBy;
    MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
REGISTERED AS {MKMD.mkmAttribute 48};

relationshipObject ATTRIBUTE
    WITH ATTRIBUTE SYNTAX GrmAttributeDefinitionModule.RelationshipObject;
    MATCHES FOR EQUALITY;
REGISTERED AS {MKMD.mkmAttribute 49};

roleMappingSpecificationSet ATTRIBUTE
    WITH ATTRIBUTE SYNTAX GrmAttributeDefinitionModule.RoleMappingSpecificationSet;
    MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
REGISTERED AS {MKMD.mkmAttribute 50};

roleSpecifier ATTRIBUTE
    WITH ATTRIBUTE SYNTAX GrmAttributeDefinitionModule.RoleSpecifier;
    MATCHES FOR EQUALITY;
REGISTERED AS {MKMD.mkmAttribute 51};

supports ATTRIBUTE
    WITH ATTRIBUTE SYNTAX GrmAttributeDefinitionModule.Supports;
    MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
REGISTERED AS {MKMD.mkmAttribute 52};
```

21) Subclause A.3.5

In this subclause change:

--%PRAGMA version BIT STRING {v1990(0), v1994(1)} ::= {v1990, v1994}

to:

--<ASN1.Version 1990, 1994 DefinitionASN1Module >--

Also in subclause A.3.5 change:

FROM Attribute-ASN1Module {joint-iso-ccitt(2) ms(9) smi(3) part2(2) asn1Module(2)}

to:

FROM Attribute-ASN1Module {joint-iso-itu-t(2) ms(9) smi(3) part2(2) asn1Module(2) 1}

Add to the same subclause this new ASN.1 module (after END of DefinitionASN1Module):

```
-- <ASN1.Version 1990, 1994 GrmAttributeDefinitionModule >--
GrmAttributeDefinitionModule {joint-iso-itu-t(2) ms(9) function(2) part16(16) asn1Modules(2) 6}
DEFINITIONS IMPLICIT TAGS ::=
BEGIN
-- EXPORTS everything;
IMPORTS
DefinedType, Identifier, OptionallyRegisteredAs, TemplateLabel, TemplateList
    FROM DefinitionASN1Module {joint-iso-itu-t(2) ms(9) function(2) part16(16) asn1Modules(2) 1}
    -- this Recommendation | International Standard
;
LabelAndParameters          ::= SEQUENCE {
                                    label                       TemplateLabel,
                                    parameters                  TemplateList OPTIONAL}
MappedRelationshipClass     ::= TemplateLabel
MappingPair                 ::= SEQUENCE {
                                    relationshipOperation       RelationshipOperation,
                                    mapsTo                      MapsTo}
MapsTo                      ::= SET OF MapsToPair
MapsToPair                  ::= SEQUENCE {
                                    systemsMgtOperation         SystemsMgtOperation,
                                    roleOrRelationshipObject    RoleOrRelationshipObject}
OperationsMapping           ::= SET OF MappingPair
OptionalIdentifier          ::= CHOICE {
                                    present                     Identifier,
                                    absent                      NULL}
OptionalIdentifierAndRole   ::= SEQUENCE {
                                    optionalOperationName       OptionalIdentifier,
                                    optionalRoleName            Identifier OPTIONAL}
OptionalLabelAndParameters  ::= SEQUENCE {
                                    label                       TemplateLabel OPTIONAL,
                                    parameters                  TemplateList OPTIONAL}
QualifiedBy                 ::= TemplateList
RelationshipObject          ::= SEQUENCE {
                                    class                       TemplateLabel,
                                    qualifies                   TemplateList OPTIONAL}
RelationshipOperation       ::= CHOICE {
                                    establish                   [0] EXPLICIT OptionalIdentifier,
                                    terminate                   [1] EXPLICIT OptionalIdentifier,
                                    query                       [2] OptionalIdentifierAndRole,
                                    notify                      [3] EXPLICIT OptionalIdentifier,
                                    userDefined                 [4] EXPLICIT OptionalIdentifier,
                                    bind                        [5] OptionalIdentifierAndRole,
                                    unbind                      [6] OptionalIdentifierAndRole}
```

```
Representation                ::= CHOICE {
                                      nameBindingSuperior                [0] TemplateLabel,
                                      nameBindingSubordinate             [1] TemplateLabel,
                                      attribute                          [2] TemplateLabel,
                                      relationshipObjectPointer          [3] TemplateLabel,
                                      operation                          NULL}
RoleMappingSpecification      ::= SEQUENCE {
                                      role                               Identifier,
                                      relatedClasses                     TemplateList,
                                      representedBy                      [0] EXPLICIT Representation OPTIONAL,
                                      qualifies                          [1] TemplateList OPTIONAL}
RoleMappingSpecificationSet   ::= SET OF RoleMappingSpecification
RoleOrRelationshipObject      ::= CHOICE {
                                      role                               Identifier,
                                      relationshipObject                 NULL}
RoleSpecifier                 ::= SEQUENCE {
                                      role                               Identifier,
                                      compatibleWith                     [0] TemplateLabel OPTIONAL,
                                      permittedRoleCardinality           [1] DefinedType OPTIONAL,
                                      requiredRoleCardinality            [2] DefinedType OPTIONAL,
                                      bindingSupport                     [3] EXPLICIT OptionalIdentifier OPTIONAL,
                                      unbindingSupport                   [4] EXPLICIT OptionalIdentifier OPTIONAL,
                                      permittedRelationshipCardinality   [5] DefinedType OPTIONAL,
                                      optionalRegisteredAs               OptionallyRegisteredAs}
SupportedChoice               ::= CHOICE {
                                      establish                          [0] EXPLICIT OptionalIdentifier,
                                      terminate                          [1] EXPLICIT OptionalIdentifier,
                                      query                              [2] EXPLICIT OptionalIdentifier,
                                      notify                             [3] EXPLICIT OptionalIdentifier,
                                      userDefined                        [4] EXPLICIT OptionalIdentifier}
Supports                      ::= SET OF SupportedChoice
SystemsMgtOperation           ::= CHOICE {
                                      get                                [0] LabelAndParameters,
                                      replace                            [1] LabelAndParameters,
                                      add                                [2] LabelAndParameters,
                                      remove                             [3] LabelAndParameters,
                                      create                             [4] OptionalLabelAndParameters,
                                      delete                             [5] TemplateList,
                                      action                             [6] LabelAndParameters,
                                      notification                       [7] LabelAndParameters}
END
```

22) Subclause A.4.1

In this subclause, in the **DERIVED FROM** *section, remove the extraneous blank between the* double-quote *and* **CCITT**.

Also, in this subclause, in the **ATTRIBUTES** *section, change:*

discoveryId;

to:

discoveryId GET;

23) Subclause A.4.5

In this subclause change:

--%PRAGMA version BIT STRING {v1990(0), v1994(1)} ::= {v1990, v1994}

to:

--<ASN1.Version 1990, 1994 DiscoveryASN1Module >--

Also in this subclause change:

FROM InformationFramework {joint-iso-ccitt(2) ds(5) modules(1) informationFramework(1)}
-- ITU-T Rec.X.501 | ISO/IEC 9594-2

to:

FROM InformationFramework {joint-iso-itu-t(2) ds(5) modules(1) informationFramework(1) 2}
-- NOTE – This Recommendation | International Standard imports
-- RelativeDistinguishedName from CCITT Rec. X.501 (1988) | ISO/IEC 9594-2:1990.
-- The specification for this syntax can now be found in an informative
-- annex of ITU-T Rec. X.711 (1997) | ISO/IEC 9596-1:1998.

In the same subclause, in the production for **MITSearch***, ensure that blank space is apparent between* **discoveryScope** *and* **Scope.**

24) Subclause B.1

In this subclause change:

--%PRAGMA version BIT STRING {v1990(0), v1994(1)} ::= v1994

to:

--< ASN1.Version 1994 RepertoireDirectoryASN1Module >--

25) Subclause B.2

In this subclause change:

--%PRAGMA version BIT STRING {v1990(0), v1994(1)} ::= v1994

to:

--< ASN1.Version 1994 DefinitionDirectoryASN1Module >--

Add to this subclause this new ASN.1 module (after **END** *of DefinitionDirectoryASN1Module):*

--<ASN1.Version 1994 GrmDefinitionDirectoryASN1Module >--
GrmDefinitionDirectoryASN1Module {joint-iso-itu-t(2) ms(9) function(2) part16(16) asn1Modules(2) 7}
DEFINITIONS IMPLICIT TAGS ::=
BEGIN
-- EXPORTS everything;
IMPORTS
ATTRIBUTE, OBJECT-CLASS, CONTENT-RULE
FROM InformationFramework {joint-iso-itu-t(2) ds(5) modules(1) informationFramework(1) 2}
-- ITU-T Rec. X.501 | ISO/IEC 9594-2
behaviour, derivedFrom, managementTemplate, registeredAs, registeredInformation
FROM DefinitionDirectoryASN1Module
{joint-iso-itu-t(2) ms(9) function(2) part16(16) asn1Modules(2) 4}
-- this Recommendation | International Standard
MappedRelationshipClass, MappingPair, RelationshipObject, RoleMappingSpecification, RoleSpecifier, SupportedChoice
FROM GrmAttributeDefinitionModule
{joint-iso- itu-t(2) ms(9) function(2) part16(16) asn1Modules(2) 6}
-- this Recommendation | International Standard
mkmDirectoryObjectClass, mkmDirectoryAttributeType
FROM MKMD {joint-iso- itu-t(2) ms(9) function(2) part16(16) asn1Modules(2) 5}
-- this Recommendation | International Standard
;

```
managementRelationshipClass OBJECT-CLASS    ::=   {
            SUBCLASS OF       managementTemplate
            KIND              auxiliary
            MUST CONTAIN      { derivedFrom |
                              behaviour }
            MAY CONTAIN       { supports |
                              qualifiedBy |
                              roleSpecifier }
            ID                { mkmDirectoryObjectClass 16 }}

managementRelationshipMapping OBJECT-CLASS      ::=   {
            SUBCLASS OF       managementTemplate
            KIND              auxiliary
            MUST CONTAIN      { mappedRelationshipClass |
                              behaviour |
                              roleMappingSpecificationSet |
                              registeredAs }
            MAY CONTAIN       { relationshipObject |
                              operationsMapping }
            ID                { mkmDirectoryObjectClass 17 }}

-- Definition of DIT content rules
-- Only one of the specified auxiliary object classes can be
-- included in a given entry of class registeredInformation
registeredManagementRelationshipInformationCR       CONTENT-RULE ::= {
            STRUCTURAL OBJECT CLASS          registeredInformation
            AUXILIARY OBJECT CLASS           { managementRelationshipClass |
                                             managementRelationshipMapping }}

-- Definition of used attributes
mappedRelationshipClass ATTRIBUTE ::=   {
            WITH SYNTAX       MappedRelationshipClass
            SINGLE VALUE      TRUE
            ID                { mkmDirectoryAttributeType 47 }}

operationsMapping ATTRIBUTE   ::=   {
            WITH SYNTAX       MappingPair
            ID                { mkmDirectoryAttributeType 48 }}

qualifiedBy ATTRIBUTE         ::=   {
            WITH SYNTAX       TemplateLabel
            ID                { mkmDirectoryAttributeType 49 }}

relationshipObject ATTRIBUTE    ::=   {
            WITH SYNTAX       RelationshipObject
            SINGLE VALUE      TRUE
            ID                { mkmDirectoryAttributeType 50 }}

roleMappingSpecificationSet ATTRIBUTE          ::=   {
            WITH SYNTAX       RoleMappingSpecification
            ID                { mkmDirectoryAttributeType 51 }}

roleSpecifier ATTRIBUTE       ::=   {
            WITH SYNTAX       RoleSpecifier
            SINGLE VALUE      TRUE
            ID                { mkmDirectoryAttributeType 52 }}

supports ATTRIBUTE    ::=   {
            WITH SYNTAX       SupportedChoice
            ID                { mkmDirectoryAttributeType 53 }}
END
```

26) Annexes A to F

Add this footnote to the titles of Annexes A to F (and renumber the existing two footnotes):

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

27) Subclause C.1.3

In this subclause, add the following text after the line:

– Not applicable or out of scope

The value of 'm' in Status column for the receiving of parameters, of tables of type MICS or PICS, indicates that there is a minimum requirement for the implementation to be able to receive the parameter. The Additional information column shall be used to state whether the implementation provides support for more than the minimum requirement.

28) Subclause C.4

In Table C.3, add these rows in index order (after existing rows):

17	Relationship class template managed object class	c6		
18	Relationship mapping template managed object class	c6		

In Table C.5, change the status from m *to* o *and add the following Note to the table:*

NOTE – The supplier of implementation shall indicate the application contexts supported.

In Table C.6, add these rows in index order (after existing rows):

17	ITU-T Rec. X.750 | ISO/IEC 10164-16	Tables E.85-E.90	relationship class template managed object class	–	c26			
18	ITU-T Rec. X.750 | ISO/IEC 10164-16	Tables E.91-E.96	relationship mapping template managed object class	–	c27			

Following Table C.6, add these conditions after existing ones:

c26: if C.3/17a then m else –

c27: if C.3/18a then m else –

29) Subclause D.3.1

In Table D.1, add these rows in index order (after existing rows for first set of columns and before second set of columns):

50	mappedRelationshipClass	{MKMD.mkmAttribute 46}	–	–		o.4	
51	operationsMapping	{MKMD.mkmAttribute 47}	–	–		o.4	
52	qualifiedBy	{MKMD.mkmAttribute 48}	–	–		o.4	
53	relationshipObject	{MKMD.mkmAttribute 49}	–	–		o.4	
54	roleMappingSpecificationSet	{MKMD.mkmAttribute 50}	–	–		o.4	
55	roleSpecifier	{MKMD.mkmAttribute 51}	–	–		o.4	
56	supports	{MKMD.mkmAttribute 52}	–	–	–	o.4	

In Table D.1, add these rows in index order (after existing rows for second set of columns):

50	–		–		–		–		
51	–		–		–		–		
52	–		–		–		–		
53	–		–		–		–		
54	–		–		–		–		
55	–		–		–		–		
56	–		–		–		–		

30) New subclauses E.3.16 and E.3.17

At the end of Annex E (after Table E.84), add the new subclauses E.3.16 and E.3.17 and the following tables:

E.3.16 Relationship class template managed object class

Table E.85 – Relationship class template managed object class support

Index	Managed object class template label	Value of object identifier for class	Support of all mandatory features? (Y/N)	Is the actual class the same as the managed object class to which conformance is claimed? (Y/N)
1	relationshipClassTemplate	{MKMD.mkmMObjectClass 16}		

If the answer to the actual class question in Table E.85 is No, the supplier of the implementation shall fill in the actual class support Table E.86.

Table E.86 – Actual class support

Index	Actual managed object class template label	Value of object identifier for actual class	Additional information

Table E.87 – Package support

Index	Action type template label	Value of object identifier for action type	Constraints and values	Status	Support	Additional information
1	templateDefinitionPackage	{MKMD.mkmPackage 3}	–	o		
2	relationshipClassSupports Package	{MKMD.mkmPackage 5}	–			
3	relationshipClassQualifiedByPackage	{MKMD.mkmPackage 6}	–			
4	relationshipClassRoleSpecifierPackage	{MKMD.mkmPackage 7}	–	o		

Table E.88 – Relationship class template managed object attribute support

				Set by create		Get	
Index	Attribute template label	Value of object identifier for attribute	Constraints and values	Status	Support	Status	Support
1	objectClass	{MKMD.dmiAttribute 65}	–	x		m	
2	nameBinding	{MKMD.dmiAttribute 63}	–	x		m	
3	packages	{MKMD.dmiAttribute 66}	–	x		m	
4	allomorphs	{MKMD.dmiAttribute 50}	–	x		c76	
5	templateName	{MKMD.mkmAttribute 7}	–	x		m	
6	templateDefinition	{MKMD.mkmAttribute 37}	–	x		c77	
7	derivedFrom	{MKMD.mkmAttribute 10}	–	x		m	
8	behaviour	{MKMD.mkmAttribute 15}	–	x		m	
9	supports	{MKMD.mkmAttribute 52}	–	x		c78	
10	qualifiedBy	{MKMD.mkmAttribute 48}	–	x		c79	
11	roleSpecifier	{MKMD.mkmAttribute 51}	–	x		c80	

c76: if (not E.85/1b) then m else –
c77: if E.87/1a then m else –
c78: if E.87/2a then m else –
c79: if E.87/3a then m else –
c80: if E.87/4a then m else –

Table E.88 *(concluded)*

	Replace		Add		Remove		Set to default		
Index	Status	Support	Status	Support	Status	Support	Status	Support	Additional information
1	x		–		–		–		
2	x		–		–		–		
3	x		x		x		–		
4	x		x		x		–		
5	x		–		–		–		
6	x		–		–		–		
7	x		–		–		–		
8	x		–		–		–		
9	x		–		–		–		
10	x		–		–		–		
11	x		–		–		–		

Table E.89 – Action support

Index	Action type template label	Value of object identifier for action type	Constraints and values	Status	Support	Additional information
1	getTextualRepresentation	{MKMD.mkmAction 2}	–	c77		

Table E.89 *(concluded)*

Index	Subindex	Action field name label	Constraints and values	Status	Support	Additional information
1	1.1	AttributeId (INFORMATION SYNTAX)	–	c:m		
	1.2	TextualRepresentation (REPLY SYNTAX)	–	c:m		

Table E.90 – Parameter support

Index	Parameter template label	Value of object identifier for parameter	Constraints and values	Status	Support	Additional information
1	attributeNotAvailableViaThisAction	{MKMD.mkmParameter 1}		c77		

E.3.17 Relationship mapping template managed object class

Table E.91 – Relationship mapping template managed object class support

Index	Managed object class template label	Value of object identifier for class	Support of all mandatory features? (Y/N)	Is the actual class the same as the managed object class to which conformance is claimed? (Y/N)
1	relationshipMappingTemplate	{MKMD.mkmMObjectClass 17}		

If the answer to the actual class question in Table E.91 is No, the supplier of the implementation shall fill in the actual class support Table E.92.

Table E.92 – Actual class support

Index	Actual managed object class template label	Value of object identifier for actual class	Additional information

Table E.93 – Package support

Index	Action type template label	Value of object identifier for action type	Constraints and values	Status	Support	Additional information
1	templateDefinitionPackage	{MKMD.mkmPackage 3}	–	o		
2	relationshipMappingRelationshipObjectPackage	{MKMD.mkmPackage 8}	–			
3	relationshipMappingOperationsMapp ingPackage	{MKMD.mkmPackage 9}	–			