INTERNATIONAL STANDARD

ISO/IEC 14769

First edition
2001-05-15

Information technology — Open Distributed Processing — Type Repository Function

Technologies de l'information — Traitement réparti ouvert — Fonction de répertoire de types

Reference number
ISO/IEC 14769:2001(E)

© ISO/IEC 2001

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

Not for Resale

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

© ISO/IEC 2001

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 either ISO at the address below or ISO's member body in the country of the requester.

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

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
© ISO/IEC 2001 – All rights reserved

CONTENTS

		Page
1	Scope	1
2	Normative References	1
	2.1 Identical Recommendation | International Standard	1
	2.2 Additional References	2
	2.3 Specifications of the Object Management Group	2
3	Definitions	2
	3.1 Terms defined in other International Standards	2
	3.2 Terms defined in this Recommendation | International Standard	4
	3.3 Terms defined in the OMG Meta-Object Facility	5
4	Abbreviations	5
5	Overview and Motivation	5
	5.1 Type Repository	5
	5.2 Meta-Object Facility	6
6	Enterprise Specification	6
	6.1 Objective	6
	6.2 Type Repository Community	6
	6.2.1 Roles	6
	6.2.2 Behaviour	8
	6.2.3 Policies	10
	6.3 Federation	10
	6.4 Correspondences between enterprise specification concepts and the MOF	11
7	Information Specification	11
	7.1 Correspondences between information viewpoint concepts and the MOF	11
8	Computational Specification	12
	8.1 Correspondences between computational viewpoint concepts and the MOF	12
9	Conformance Statements and Reference Points	12
Annex A – ODP Type Framework		14
	A.1 ODP-RM Type System	14
	A.2 Type System for ODP Trading Function	16
	A.3 Interface Reference and Binding Type System	17
Annex B – Suggested type languages		23
Annex C – Summary of Referenced Material in OMG Meta-Object Facility		24
	C.1 Problems arising through reference to the OMG Meta-Object Facility specification	24
	C.2 Relationship with the MOF specification	24

© ISO/IEC 2001 – All rights reserved

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.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

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.

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

International Standard ISO/IEC 14769 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 7, *Software engineering*, in collaboration with ITU-T. The identical text is published as ITU-T Recommendation X.960.

Annex A forms a normative part of this International Standard. Annexes B and C are for information only.

© ISO/IEC 2001 – All rights reserved
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 prescribes the ODP Type Repository Function (subclause 14.4 of ITU-T Rec. X.903 | ISO/IEC 10746-3) to support the storage, retrieval and management of type descriptions within an identified framework for type descriptions.

ITU-T Rec. X.902 | ISO/IEC 10746-2 provides a general definition of type in subclause 9.7; this definition allows the description of types using any predicate. ITU-T Rec. X.903 | ISO/IEC 10746-3 introduces a number of target concepts specific to particular viewpoints. This Recommendation | International Standard supports the establishment of type definitions based on the concepts defined in the ODP family of Recommendations | International Standards.

This Recommendation | International Standard enables type descriptions for use by the ODP functions outlined in ISO/IEC 10746-3. Type descriptions can occur in specifications from any viewpoint, e.g. enterprise specification can introduce enterprise types. This Recommendation | International Standard specifically addresses the needs of the ODP computational and engineering viewpoint types, but is capable of supporting type descriptions coming from other viewpoint languages.

This Recommendation | International Standard permits the use of multiple type description languages. There are a number of widely used and standardized languages for type description, for example CORBA IDL, ASN.1, LOTOS, GDMO and SDL, which fulfil some of the requirements of type descriptions in ODP-RM. This Recommendation | International Standard does not define a single all-encompassing type language. Users can use either existing languages or languages defined within other ODP Recommendations | International Standards. Annex B is an informative annex outlining languages that support large sets of target concepts.

This Recommendation | International Standard supports type systems with a type Type (e.g. pass type as parameters as in the ODP computational language).

ITU-T Rec. X.903 | ISO/IEC 10746-3 defines a subtype relationship between computational operational interface signature types. This Recommendation | International Standards supports a wider variety of relationships between types, which might include the analysis of behaviour and environment contracts, but the definition of such relationships is not within the scope of this Recommendation | International Standard. Relationships between types can either be asserted or deduced. It is recognized that not all relationships (including equivalence) can always be automatically deduced. However, automatic deduction should be encouraged whenever applicable.

The type repository function supports the allocation of identifiers to types in order to allow the transmission of these "shorthand" representations across domains (i.e. between objects using different type repositories).

The type repository function addresses interworking and federation to support the distribution of the type repository function by clarifying the notion of type domains. This function supports both federation of type domains handling equivalent type systems and federation of type domains handling different type systems.

© ISO/IEC 2001 – All rights reserved

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 DISTRIBUTED PROCESSING – TYPE REPOSITORY FUNCTION

1 Scope

The concept of "type" is fundamental to ODP systems; the interaction model of ODP-RM involves strongly-typed interactions.

This Recommendation | International Standard:

- defines a framework for describing types of interest in ODP systems by determining what entities need to be typed and what needs to be said about the identified types. The primary focus of this work is the computational interface type system;
- identifies and characterizes type languages sufficient to describe the types identified above in an informative annex;
- provides enterprise, information, and computational specifications of a generic type repository function within the type description framework which can be specialized to select a specific type system or type notation. The type repository function provides:
  - storage and retrieval of type descriptions;
  - management of type descriptions;
  - management of the relationship between types including matching of types;
  - naming of types (in a manner consistent with ODP Naming Framework);
  - interworking and federation of different type repositories.

This Recommendation | International Standard provides a standard method of accessing type descriptions used within open distributed processing systems, where the type descriptions can be in various concrete syntaxes and type languages used in these open distributed processing systems. This Recommendation | International Standard also facilitates the dynamic matching of types for interactions, binding and trading purposes.

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 International 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 edition of the Recommendations and Standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards. The Telecommunication Standardization Bureau of ITU maintains a list of the currently valid ITU-T Recommendations.

2.1 Identical Recommendations | International Standards

- ITU-T Recommendations X.680-series (1997) | ISO/IEC 8824 (all parts):1998, *Information technology – Abstract Syntax Notation One (ASN.1)*.
- ITU-T Recommendation X.725 (1995) | ISO/IEC 10165-7:1996, *Information technology – Open Systems Interconnection – Structure of management information: General relationship 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

- ITU-T Recommendation X.902 (1995) | ISO/IEC 10746-2:1996, *Information technology – Open Distributed Processing – Reference Model: Foundations.*
- ITU-T Recommendation X.903 (1995) | ISO/IEC 10746-3:1996, *Information technology – Open Distributed Processing – Reference Model: Architecture.*
- ITU-T Recommendation X.910 (1998) | ISO/IEC 14771:1999, *Information technology – Open Distributed Processing – Naming framework.*
- ITU-T Recommendation X.920 (1997) | ISO/IEC 14750:1999, *Information technology – Open Distributed Processing – Interface definition language.*
- ITU-T Recommendation X.930 (1998) | ISO/IEC 14753:1999, *Information technology – Open Distributed Processing – Interface references and binding.*
- ITU-T Recommendation X.950 (1997) | ISO/IEC 13235-1:1998, *Information technology – Open Distributed Processing – Trading function: Specification.*

2.2 Additional References

- ISO/IEC 10027:1990, *Information technology – Information Resource Dictionary System (IRDS) framework.*
- ISO/IEC 13719 (all parts):1998, *Information technology – Portable Common Tool Environment (PCTE).*
- ISO/IEC 15474 (all parts):...[1], *Information technology – CDIF Framework.*

2.3 Specifications of the Object Management Group

This Recommendation | International Standard makes references to the following specifications:

- Object Management Group, ad/97-08-14 and ad/97-08-15, *Meta-Object Facility*, 1997.
- Object Management Group, ad/97-08-02 through ad/98-08-09, *Unified Modelling Language*, 1997.

Annex C identifies the clauses of this Recommendation | International Standard that reference text in the Meta-Object Facility.

3 Definitions

For the purposes of this Recommendation | International Standard, the following definitions apply, except for where the text is described as being common with the Meta-Object Facility.

3.1 Terms defined in other International Standards

This Recommendation | International Standard makes use of the following terms defined in ITU-T Rec. X.902 | ISO/IEC 10746-2 Open Distributed Processing – Reference Model: Foundations):

- action;
- behaviour;
- binding;
- client object;
- consumer object;
- creation (of an <X>);
- data type;
- deletion (of an <X>);
- domain;
- environment contract type;
- identifier;
- information;
- instance (of a type);

[1] To be published.

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

- instantiation (of an <X>);
- interface;
- interface signature;
- name;
- obligation;
- ODP standard;
- ODP system;
- object;
- policy;
- role;
- subtype;
- supertype;
- state (of an object);
- <X> template;
- trading;
- type;
- viewpoint.

This Recommendation | International Standard makes use of the following terms defined in ITU-T Rec. X.903 | ISO/IEC 10746-3 (Open Distributed Processing – Reference Model: Architecture):

- announcement signature;
- binding object;
- community;
- compound binding;
- computational interface;
- computational interface signature;
- computational interface type;
- computational object signature;
- computational specification;
- computational viewpoint;

  NOTE – Throughout this Recommendation | International Standard, the qualifier "computational" is frequently omitted.
- dynamic schema;
- engineering interface reference;
- enterprise specification;
- <X> federation;
- flow signature type;
- flow type;
- information specification;
- interrogation signature;
- invariant schema;
- invocation;
- operation;
- operation signature type;
- operational interface signature;
- primitive binding;
- primitive signal binding type;

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

- primitive stream binding type;
- service offer;
- signal interface signature type;
- signal signature type;
- static schema;
- stream interface signature type;
- termination signature type;
- trading function;
- type repository function.

This Recommendation | International Standard makes use of the following terms defined in ITU-T Rec. X.950 | ISO/IEC 13235-1 (Open Distributed Processing – Trading Function: Specification):

- service type;
- property type.

This Recommendation | International Standard makes use of the following terms defined in ITU-T Rec. X.930 | ISO/IEC 14753 (Open Distributed Processing – Interface References and Binding):

- additional information;
- behaviour;
- causality information;
- channel class;
- direct reference type;
- flow description type;
- flow type;
- group information;
- interface description type;
- interface reference;
- interface type;
- location information;
- non-interpreted reference type;
- null reference type;
- opaque information;
- operation description type;
- operation type;
- operational interface type;
- quality-of-service statement type;
- relocation information;
- security information;
- stream interface type.

3.2 Terms defined in this Recommendation | International Standard

This Recommendation | International Standard defines the following terms:

3.2.1 relationship: A predicate involving two or more roles with assigned values.

3.2.2 relationship type: A type of relationship which expresses the number and type of the roles.

3.2.3 relation: A set of relationships of the same relationship type.

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

3.3 Terms defined in the OMG Meta-Object Facility

In text described as common with the Meta-Object Facility, this Recommendation | International Standard uses the following definition.

3.3.1 Meta-Object Facility: a specification of the Object Management Group for repositories of type information for arbitrary type systems.

4 Abbreviations

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

IDL	Interface Definition Language
MOF	Meta-Object Facility
ODP	Open Distributed Processing
ODP-RM	Open Distributed Processing: Reference Model
OMG	Object Management Group
TR	ODP Type Repository

The following additional abbreviations occur in sections of the OMG Meta-Object Facility specification which are incorporated by reference in this Recommendation | International Standard:

CORBA	Common Object Request Broker Architecture
MODL	Meta-Object Definition Language
OCL	Object Constraint Language
UML	Unified Modelling Language

5 Overview and Motivation

Open distributed processing in multi-organizational environments requires that various kinds of meta-level information are available at run-time in each interoperating system.

Especially, information is needed about types and type systems, for determining:

- conformity of information presentation during compilation;
- similarity of offered and requested services in trading;
- conformity and substitutability of servers during service invocation;
- required configuration of objects for object binding;

Cooperation between autonomous systems requires knowledge about the relationships between types or type systems.

5.1 Type Repository

The type repository stores type definitions, type relations, and information about the type system itself.

A type system is structured by a set of target concepts. It should be noted that the set of target concepts may grow during the lifetime of the type system. For example, the ODP-RM computational type system includes the following target concepts: object, type, template, service, interface, operation, stream, flow, signal. Based on these target concepts, a banking application could define types applicable for banking, e.g. a BankAccount interface with operations deposit, withdraw, and balance.

No single type system or type language can be assumed. There are already a multiplicity of type systems in use (including many standardized ones) and often many type languages for each type system. For example, protocol data units can be described in ASN.1, data types in ACT-ONE, relational schemata in SQL, file formats in COBOL, interfaces in ODP IDL, pipes in ISO RPC IDL. A canonical type system or canonical type language would have to be a superset of all existing type systems and type languages. Furthermore, the set of target concepts is open-ended and so every new target concept would require the extension of the canonical type language. This is technically and politically infeasible.

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

Although no single type system or type language can be assumed, it is nonetheless possible to develop specific languages for interchange between type systems, e.g. the CDIF family of standards.

The set of relationships between types and type systems cannot be predetermined. The type repository depends on both external assertions of relationships in addition to its own ability to derive relationships through semantic analysis.

Some type descriptions can be used as templates. A template has sufficient detail to enable instantiation on a selected platform. On another platform, the same description may not be sufficient as a template.

NOTE – The architecture described conforms to the IRDS Framework.

5.2 Meta-Object Facility

This Recommendation | International Standard is technically aligned with the OMG Meta-Object Facility, a specification of a type repository system for models (types in ODP) and meta-models (type systems in ODP). The definition of a meta-model (type system) includes the definition of classes (target concepts in ODP) and associations (type relations in ODP).

The Meta-Object Facility can support multiple meta-models (type systems) and multiple models (types) within each meta-model (type system). The Meta-Object Facility unifies the handling of models and meta-models by developing a meta-meta-model (type system for describing type systems) for defining meta-models. Thus, all handling of information is performed relative to a nominated set of meta-information. A newly created Meta-Object Facility contains only the meta-meta-model, enabling the definition of meta-models (type systems), which in turn enable the definition of models (types).

The Meta-Object Facility is type-language neutral. It stores models, meta-models and its own meta-meta-model as graphs of linked CORBA objects; the mechanisms which translate to/from these graphs into particular syntaxes are outside the scope of the Meta-Object Facility specification.

The Meta-Object Facility was developed to support generic modelling needs which occur in such areas as information management, software development, and data warehousing. The overview of the Meta-Object Facility is given in Section 2: "Facility Purpose and Use" of the OMG Meta-Object Facility specification.

6 Enterprise Specification

The scope of an enterprise specification is defined in ITU-T Rec. X.903 | ISO/IEC 10746-3 and refined by ODP Enterprise Language. The enterprise specification identifies the objectives and the policy statements that govern the activities of the type repository function.

6.1 Objective

The objective of the type repository function is to manage a repository of type system descriptions, type descriptions and type relationships so that queries can be made on any stored type system description, type description or type relationship, whenever needed for the development, operation, and management of ODP systems.

6.2 Type Repository Community

A type repository community consists of objects that take on one or more roles within the community. The behaviour of each role and the behaviour of the community as a whole are governed by a set of repository policy rules. Members of the community are obliged to obey these policy rules.

6.2.1 Roles

Objects may take on the following roles within a type repository community as shown in Table 1 and as illustrated in Figure 1. A type repository is governed by a single TR type system description, but is intended to handle multiple type systems, each containing a set of types as representations for the target concepts of that type system (see examples in Annex A). This Recommendation | International Standard defines the TR type system.

NOTE 1 – Information about types and type systems includes the relationships between types and the relationships between type systems.

NOTE 2 – Examples of a type system include the Pascal programming language, SQL schemas for defining relational tables, and trader service types (see A.2.) Examples of a type include a Pascal function declaration, the definition of a payroll database in SQL, or a printer service type for the trader.

6.2.1.1 Cardinality of Roles

The purpose of this subclause is to define how many times each role can occur in the enterprise specification. It does not constrain the number of objects that can fill each role; that is the subject of 6.2.1.2.

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 type repository community contains exactly one type repository role and exactly one TR type system description that defines how the type repository is structured. There can be 0 or more TR type system users.

There can be 0 or more type system descriptions that define how type descriptions in that type system are structured.

For each type system description, there is exactly one type system author and at most one type system publisher. For each type system description, there can be 0 or more type system users.

For each type system description, there can be 0 or more type descriptions which describe a type in the structure prescribed by the type system description.

For each type description, there is exactly one type system description which prescribes the structure of the type description. For each type description, there is exactly one type author and at most one type publisher. For each type description, there can be 0 or more type users.

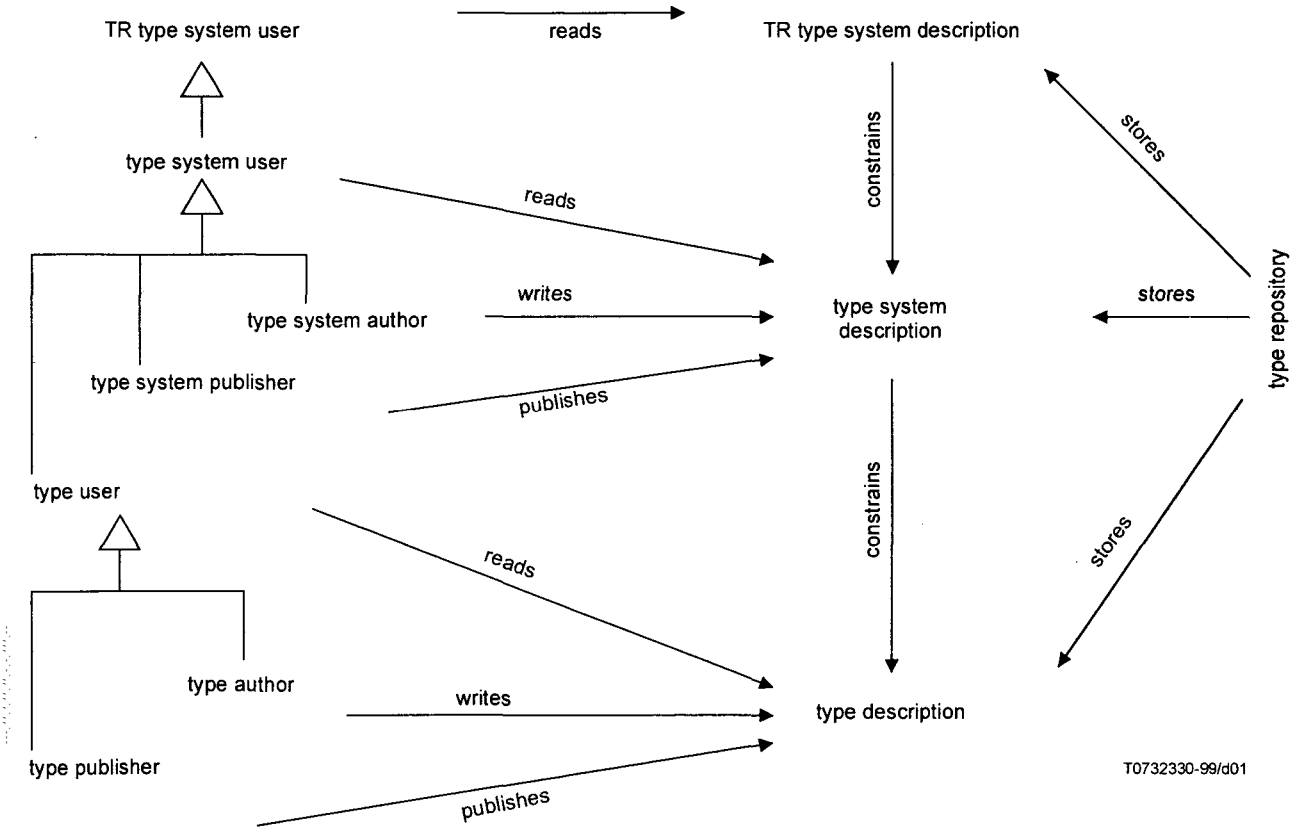


Figure 1 – Roles and Activities in the Type Repository Community

6.2.1.2 Restrictions on Filling Roles

There are no restrictions on whether any role is filled by an atomic enterprise object or by a community of enterprise objects. There are no restrictions on whether any enterprise object is involved in filling (either directly or indirectly through involvement in a community) more than one role within the community.

An enterprise object can be part of multiple type repository communities, possibly filling different roles in different communities.

NOTE – At this level of abstraction, these restrictions on cardinalities and role-filling are the minimal restrictions required for conformance. Specifications which refine this enterprise specification can introduce additional restrictions without loss of conformance to this Recommendation | International Standard.

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 1 – Roles in Type Repository Community

Role	Role behaviour
TR type system description	The TR type system description maintains information about the TR type system. It represents the target concepts of the TR type system and the type relations between them.
TR type system user	The TR type system user obtains information about the TR type system, and its target concepts and type relations. The TR type system user can use this information to develop or interpret type system descriptions.
type system description	A type system description maintains information about a type system as constrained by the TR type system. It represents the target concepts and type relations of that type system.
type system user	A type system user obtains information about a type system from the type system description. The type system user can use this information to develop or interpret types in that type system. A type system user is also a TR type system user.
type system author	A type system author creates, modifies, and deletes the information about a type system in the type system description. A type system author determines the target concepts and type relations within that type system. A type system author is also a type system user.
type system publisher	A type system publisher publishes a type system description. A type system publisher determines when a type system description is mature enough to became a stable publication. A type system publisher is also a type system user.
type description	A type description maintains information about a type as constrained by the chosen type system. The type is described in terms of the target concepts and type relationships of the chosen type system.
type user	A type user obtains information about a type from the type description. The type user can use that information to interpret or develop instances of that type. A type user is also a type system user.
type author	A type author creates, modifies, and deletes information about a type in the chosen type description.The type author determines how to describe the type in terms of the target concepts and type relationships of the chosen type system. A type author is also a type user.
type publisher	A type publisher publishes a type description. The type publisher determines when the type description is mature enough to become a stable publication. A type publisher is also a type user.
Type repository	The type repository stores the TR type system description, the type system descriptions, and the type descriptions and the type relationships between them.

6.2.2 Behaviour

Table 2 shows the interactions that occur within a type repository community:

Table 2 – Interactions in a type repository community

Interaction	Roles involved	Description
TR type system query	TR type system user TR type system description	A TR type system user obtains information about the TR type system from the TR type system description.
type system creation	type system author type system description	The type system author establishes information about a type system by creating a type system description.
type system modification	type system author type system description	The type system author updates information about a type system by modifying the type system description.
type system deletion	type system author type system description	The type system author removes information about the type system by deleting the type system description.
type system publication	type system publisher type system description	The type system publisher publishes the type system description.
type system verification	type system user type system description	A type system user verifies that a type system description conforms to the TR type system.
type system query	type system user type system description	A type system user obtains information about a type system from its type system description.
type creation	type system author type description	The type author establishes information about the type by creating a type description within a chosen type system.
type modification	type system author type description	The type system author updates information about the type by modifying the type description.
type deletion	type system author type description	The type system author removes information about type description the type by deleting a type description.
type publication	type publisher type description	The type publisher publishes a type description.
type verification	type user type description	A type user verifies that a type description conforms to its type system.
type query	type user type description	A type user obtains information about a type from a type description.

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 3 shows the internal actions that occur within a type repository community:

Table 3 – Internal actions in a type repository community

Internal action	Role involved	Description
TR type system representation	TR type system description	The TR type system description maintains information about the TR type system.
type system representation	type system description	A type system description maintains information about a type system in the manner prescribed by the TR type system description.
type representation	type description	A type description maintains information about a type in the manner prescribed by its type system description.
TR type system description storage	type repository	The type repository provides storage for a TR type system description.
type system description storage	type repository	The type repository provides storage for type system descriptions.
type description storage	type repository	The type repository provides storage for type descriptions.

6.2.2.1 Behaviour for type repository community creation

The mechanisms for forming a type repository community are beyond the scope of this Recommendation | International Standard. However, the result of any such creation must conform to the cardinality constraints given in 6.2.1.1. The minimal type repository community consists of a type repository and a TR type description.

6.2.2.2 Behaviour for usage of type system descriptions

A type system description must be created before it can be modified, deleted, queried, verified or published.

A type system description can depend on other type system descriptions.

After deletion, a type system description cannot be created, modified, deleted, queried, verified or published.

NOTE – If an identical type system description is subsequently created, it is regarded as a different type system description.

After publication, a type system description cannot be modified or deleted or published. However, it can be queried and verified.

6.2.2.3 Behaviour for usage of type descriptions

A type description must be created before it can be modified, deleted, queried, verified or published.

A type description must conform to the type system description. It is dependent on type system description and can depend on other type descriptions.

Since one type system can depend on another type systems, it follows that a type description can depend on both other type descriptions within the same type system and other type descriptions within a different type system.

After deletion, a, type description cannot be created, modified, deleted, queried, verified or published.

NOTE – If an identical type description is subsequently created, it is regarded as a different type description.

After publication, a type description cannot be modified or deleted or published. However, it can be queried and verified.

6.2.2.4 Behaviour for verification of descriptions

Verification of a description involves verification of any other description on which the original description depends directly or indirectly; this includes the verification of the type system of the original description and all dependent descriptions. A graph of dependent descriptions can be verified as a single enterprise action; this enables the verification of mutually-dependent descriptions.

A verification is valid until the modification or deletion of that description or any description upon which it directly or indirectly depends.

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

6.2.2.5 Behaviour for publication of descriptions

Publication of a description requires publication of any other description on which the original description depends directly or indirectly; this includes the publication of the type system of the original description and all dependent descriptions. A graph of dependent descriptions can be published as a single enterprise action; this enables the publication of mutually-dependent descriptions.

Since publication prevents subsequent modification and deletion, published descriptions are guaranteed to be immutable.

6.2.3 Policies

The author is responsible for the accuracy of their descriptions. Since publication prevents subsequent modification and deletion, a publisher should not publish a description until the author's work is complete.

Descriptions must be verified before they are published. That is, publication must incorporate verification to prevent malformed descriptions being published.

Authors and publishers have no obligations to maintain the stability of a description prior to publication. Users of unpublished descriptions must take responsibility for any consequences arising from changes to descriptions which were unpublished at the time of their query.

Relationships between types or between type systems can be asserted by authors or derived from descriptions stored in a type repository. The scope of relationship derivation is restricted to the descriptions stored in the type repository in which the queried relation is stored.

The type repository provides storage for descriptions; the stability of that storage is specified by Quality of Service requirements.

6.3 Federation

Since type descriptions depend on their type system description and can depend on other type descriptions (and hence their type system descriptions), the creation, modification, and query of type descriptions and type system descriptions can involve the participation of other type descriptions. In particular, verification and publication are federated operations.

There are three categories of inter-references involved in the design of type repositories:

- Type system interworking: relationships between the differing type definitions across the different type systems supported by the same type repository, thus governed by a shared TR type system definition.

  NOTE 1 – Examples of this category include IDL and ODP descriptions of an operational interface for an ODP type system.

- Type repository interworking: relationships between type definitions across similar type systems supported by different type repositories that have identical TR type system definitions.

  NOTE 2 – This is the model for CORBA type repositories where the shared MOF model is used for all CORBA type systems, and the shared CORBA specification is used for the basis of definitions of target concepts identified in the OMA.

- Type repository federation: relationships between similar type definitions across type systems supported by different type repositories that have separate TR type system definitions.

  NOTE 3 – For example, the TR type system definition may be given in a different language or the TR type system definitions may only partially overlap. Still, the type repositories may be able to cooperate, if a mapping is provided between the TR type system definitions. The mapping is not necessarily complete and the similarity of type systems is not necessarily automatically verifiable. Type repository federation enables asynchronous evolution of type repositories.

Each category of inter-referential relationships requires that the descriptions are published. In addition:

- Type system interworking requires that the published type descriptions can be compared for replacability.
- Type system interworking requires that the type repositories publish the kind of type systems they support and, at least, name the TR type system they use.
- Type repository federation requires that both the type system definitions and the TR type system definitions are published.

Federation between type repositories may require engineering solutions, e.g. interceptors, such as defined in ODP Interface References and Binding.

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

6.4 Correspondences between enterprise specification concepts and the MOF

Table 4 gives the correspondences between concepts in this enterprise specification and the MOF specification.

Table 4 – Correspondence with enterprise specification concepts

Concept in this enterprise specification	Concept in MOF specification
Type repository	MOF repository
TR type system description	meta-meta-model, MOF model
type system description	meta-model
type description	model

In the MOF specification, the relationship between the meta-meta-model and a meta-model is treated as being a specific instance of the general relationship between any meta-model and the models derived from that meta-model.

There are no explicit correspondences with the user, author, and publisher roles. It is implicit in the MOF specification that such roles are the clients of the interfaces which provide the corresponding functionality.

7 Information Specification

The information specification of this Recommendation | International Standard is given in:

- MOF Section 3: "Model and Interfaces", but excluding the IDL fragments (which form part of the computational specification);
- MOF Section 6: "MOF Semantics Details", but excluding:
  - MOF Section 6.3 "MOF Data Type Encoding and Translation Conventions";
  - MOF Section 6.6 "MOF and MetaModel Extensibility Mechanisms";
  - MOF Section 6.7 "Inter-Repository Modelling".

NOTE 1 – A text-based representation of the MOF model is given in MOF Annex B: "MODL Description of the MOF" as a machine-readable alternative to the UML notation used in MOF Section 3: MOF "Model and Interfaces".

The MOF intends to represent types, relations, and type systems in a universal, extendable way.

Types are represented as objects that have a fixed set of strongly typed properties. Relations between these type objects are represented by relation association objects. Collections of related types and relations are represented as type system objects. Type objects correspond to the type descriptions of the enterprise specification; relation association objects are not visible in the enterprise specification; type system objects correspond to the type system description in the enterprise specification.

MOF allows various type system descriptions to exist in parallel. Each type system must conform to one meta-level description that governs the structure of the type system description. The type systems can exploit the contents of each other by importing and creating associations.

NOTE 2 – By extensions of the reflective package, the type systems can obtain capabilities for learning about the descriptions of each other.

For type system federation, associations may be present also between the type system descriptions and their meta-level descriptions. In MOF, associations between type objects can only carry simple properties like association name. Together with aggregation and composition, this is adequate for type system interworking and type repository interworking.

7.1 Correspondences between information viewpoint concepts and the MOF

The most important concepts in the information viewpoint are invariant schemas, static schemas, and dynamic schemas.

In the MOF specification, invariant schemas are represented using a combination of UML diagrams, text, and constraints expressed in OCL. Generally, the UML diagrams show the state of an information object, the associated text explains the semantics informally, and the OCL defines some semantics formally.

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

In the UML diagrammatic technique, it is assumed that any information object represented is capable of being accessed and updated unless otherwise stated (e.g. by marking an attribute as being read-only); only more complex actions on the information object are explicitly represented. In the MOF specification, the dynamic schemas comprise the explicitly stated operations as well as the implicit actions to access and update individual informational objects.

There are no static schemas represented in the MOF specification.

In constructing UML models, the availability of a set of primitive types with well-known semantics must be assumed. Since the MOF specification was developed for use in CORBA environments, the primitive types assumed were those types capable of expression in CORBA IDL and hence ODP IDL. However, the use of "TypeCode" is one difference between CORBA IDL and ODP IDL. In CORBA IDL, a TypeCode is an abstract data type with specified content and operations. In ODP IDL, a TypeCode represents a data type in the specifier's environment which describes types with at least the expressive power of ODP IDL.

NOTE – MOF Section 6.3: "MOF Data Type Encoding and Translation Conventions" does not form part of this Recommendation | International Standard. However, CORBA implementers should note that this section discusses the requirements on the creation of TypeCodes in a CORBA engineering environment sufficient to ensure that IDL can be generated from TypeCodes.

8 Computational Specification

The computational specification of this Recommendation | International Standard is given in:

- IDL fragments given in MOF Section 3: "Model and Interfaces";
- MOF Section 4: "Facility Package";
- MOF Section 5: "Reflective Package Types";
- MOF Section 7: "the MOF Model to IDL Mapping";
- MOF Annex A: "Meta-Object Facility IDL Summary", which contains a consolidated collection of all IDL presented in MOF Section 3: "MOF Model and Interfaces", MOF Section 4: "Facility Package", and MOF Section 5: "Reflective Package Types".

The MOF model introduces a MofRepository object with the following interfaces or operations:

- management of a MOF model itself;
- management of type descriptions and type system descriptions stored within a MofRepository.

The meta-meta-objects, meta-objects, and objects all have a separately defined interface type.

In addition, the operations have been split into packages in order to allow easy extensibility of the model. The packages are the Facility package and the Reflective package. For the ODP computational viewpoint, the division has no meaning, it being a matter of organizational convenience.

8.1 Correspondences between computational viewpoint concepts and the MOF

The computational viewpoint defines the functional decomposition of an ODP system into objects which interact through interfaces.

The MOF specification is concerned only with the specification of operational interfaces, expressed using CORBA IDL. Apart from "TypeCode", CORBA IDL and ODP IDL are equivalent. In CORBA IDL, a TypeCode is an abstract data type with specified content and operations. In ODP IDL, a TypeCode represents a data type in the specifier's environment which describes types with at least the expressive power of ODP IDL.

NOTE – MOF Section 6.3: "MOF Data Type Encoding and Translation Conventions" does not form part of this Recommendation | International Standard. However, CORBA implementers should note that this section discusses the requirements on the creation of TypeCodes in a CORBA engineering environment sufficient to ensure that IDL can be generated from TypeCodes.

9 Conformance Statements and Reference Points

Implementations claiming conformance to the ODP Type Repository must support:

- the description of type systems according to the information viewpoint schemas given in clause 7 (MOF Section 3: "MOF Model and Interfaces");
- the provision of computational interfaces to create, query, and modify type systems (the Model module) as given in clause 8 (in particular, MOF Section 3: "MOF Model and Interfaces" and MOF Annex A: "Meta-Object Facility IDL Summary");

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

and shall state which of the following they support:

- the provision of computational interfaces to manage an ODP Type Repository (the Facility module) as defined in clause 8 (in particular, MOF Section 4: "Facility Package" and MOF Annex A: "Meta-Object Facility IDL Summary");
- the provision of computational interfaces to create, query, and modify types in any type system (the Reflective module) as defined in clause 8 (in particular, MOF Section 5: "Reflective Package Types" and MOF Annex A: "Meta-Object Facility IDL Summary";
- the provision of computational interfaces to create, query, and modify types in any type system derived from the IDL templates as defined in clause 8 (in particular, MOF Section 7: "the MOF Model to IDL Mapping");
- the provision of mechanisms to automate the generation of IDL for the computational interfaces to create, query, and modify types in any type system derived from the IDL templates given in clause 8 (in particular, MOF Section 7: "the MOF Model to IDL 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 A

ODP Type Framework

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

This annex provides the type systems described in the ODP family of Recommendations | International Standards.

The ODP type framework is described by an information model expressed in the following tables of MOF Classes (see clause 7) and MOF Associations (see clause 7) and illustrated using UML. For each MOF Class, the immediate supertypes, attributes and contained elements are given. For each MOF Association, the names, types and cardinalities of the AssociationEnds are given.

A.1 ODP-RM Type System

Table A.1 (illustrated by Figures A.1 through A.4) contains the MOF Classes for the type system defined in ITU-T Rec. X.903 | ISO/IEC 10746-3.

Table A.1 – MOF Classes for the type system in ODP-RM Part 3: Architecture

MOF Class	super classes	attributes (name and type)	contained elements (and cardinality)	See Figure
object type				Figure A.1 Figure A.3
interface type				Figure A.1 Figure A.2
interface signature type				Figure A.1 Figure A.4
operational interface signature type	interface signature type	causality: {client | server}	operation signature type (0..*)	Figure A.4
stream interface signature type	interface signature type		flow signature type (0..*)	Figure A.4
signal interface signature type	interface signature type		signal signature type (0..*)	Figure A.4
binding type			role type (0..*)	Figure A.2 Figure A.3
primitive binding type	binding type			Figure A.3
compound binding type	binding type			Figure A.3
primitive operation binding type	primitive binding type			Figure A.3
primitive signal binding type	primitive binding type			Figure A.3
primitive stream binding type	primitive binding type			Figure A.3
binding object type	compound binding type object type			Figure A.3
operation signature type				Figure A.4
announcement signature type	operation signature type		invocation signature type (1)	Figure A.4
interrogation signature type	operation signature type		invocation signature type (1) termination signature type (1:*)	Figure A.4

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 A.1 – MOF Classes for the type system in ODP-RM Part 3: Architecture *(concluded)*

MOF Class	super classes	attributes (name and type)	contained elements (and cardinality)	See Figure
invocation signature type		Name: string	parameter signature type (0..*)	Figure A.4
termination signature type		Name: string	parameter signature type (0..*)	Figure A.4
parameter signature type		Name: string		Figure A.4
data type				Figure A.4
parameter data type	data type			Figure A.4
flow data type	data type			Figure A.4
signal signature type		Causality: {initiator | responder}	parameter signature type (0..*)	Figure A.4
flow signature type		Name: string Causality: {producer | consumer}		Figure A.4
environment contract type		usage_constraint: any management_constraint: any	QoS statement type (1)	Figure A.1
behaviour type				Figure A.1
action type				Figure A.1
role type				Figure A.2

ITU-T Rec. X.903 | ISO/IEC 10746-3 defines the subtyping relationship type for ODP computational interface signature types in its subclause 7.2.4 and its Annex A. Table A.2 contains the MOF Association for that subtyping relationship, and other relationships depicted in Figures A.1 through A.4 (usually depicted without showing names of associations or association ends).

Table A.2 – ODP Relationship types in ODP-RM Part 3: Architecture

MOF Association	Association Ends: Names, Classes, Aggregations, Cardinalities	See Figure
interface is subtype of interface	supertype: interface signature type (0..*) subtype: interface signature type (0..*)	Figure A.1
object assumes environment contract	object: object type (0..*) environment contract: environment contract type (1)	Figure A.1
object offers interface	object: object type (0..*) interface: interface type (0..*)	Figure A.1
object exhibits behaviour	object: object type (0..*) behaviour: behaviour type (1)	Figure A.1
interface assumes environment contract	interface: interface type (0..*) environment contract: environment contract type (1)	Figure A.1
interface specified by interface signature	interface: interface type (0..*) interface signature: interface signature type (1)	Figure A.1
interface exhibits behaviour	interface: interface type (0..*) behaviour: behaviour type (1)	Figure A.1
environment contract has QoS statement	environment contract: environment contract type (1) QoS statement: QoS statement type (composite 1)	Figure A.1
behaviour comprises action	behaviour: behaviour type (0..*) action: action type (0..*)	Figure A.1
binding has role	binding: binding type (1) role: ole type (composite 1)	Figure A.2

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 A.2 – ODP Relationship types in ODP-RM Part 3: Architecture *(concluded)*

MOF Association	Association Ends: Names, Classes, Aggregations, Cardinalities	See Figure
role offers interface	role: role type (0..*) interface: interface type (1)	Figure A.2
stream interface signature has flow signature	stream interface: stream interface type (1) flow signature: flow signature type (composite 0..*)	Figure A.4
signal interface has signal signature	signal interface: signal interface type (1) signal signature: signal signature type (composite 0..*)	Figure A.4
operational interface has operation signature	operational interface: operational interface type (1) operation signature: operation signature type (composite 0..*)	Figure A.4
flow signature specifies flow data	flow signature: flow signature type (0..*) flow data: flow data type (1)	Figure A.4
signal signature expects parameter	signal signature: signal signature type (1) parameter: parameter type (0..*)	Figure A.4
announcement signature has invocation signature	announcement signature: announcement signature type (1) invocation signature: invocation signature type (composite 1)	Figure A.4
interrogation signature has invocation signature	interrogation signature: interrogation signature type (1) invocation signature: invocation signature type (composite 1)	Figure A.4
interrogation signature has termination signature	interrogation signature: interrogation signature type (1) termination signature: termination signature type (composite 1..*)	Figure A.4
invocation signature has parameter signature	invocation signature: invocation signature type (1) parameter signature: parameter signature type (composite 0..*)	Figure A.4
termination signature has parameter signature	termination signature: termination signature type (1) parameter signature: parameter signature type (composite 0..*)	Figure A.4
parameter signature specifies parameter data	parameter signature: parameter signature type (0..*) parameter data: parameter data type (1)	Figure A.4

A.2 Type System for ODP Trading Function

Table A.3 and Figure A.5 contain the MOF Classes for types defined in ODP Trading Function. Note that the ODP Trading Function reuses types and relationships from ITU-T Rec. X.903 | ISO/IEC 10746-3.

Table A.3 – ODP concepts in ODP Trading Function

MOF Class	super classes	attributes (name and type)	contained elements (and cardinality)	See Figure
service type			property type (0..*)	Figure A.5
property type		name: string		Figure A.5

Table A.4 contains the relationships defined in ODP Trading Function.

Table A.4 – ODP relationship types in ODP Trading Function

MOF Association	Association Ends: Names, Classes, Aggregations, Cardinalities	See Figure
service is subtype of service	supertype: service type (0..*) subtype: service type (0..*)	Figure A.5
service offers interface	service: service type (0..*) interface: interface type (1)	Figure A.5
service has property	service: service type (1) property: property type (0..*)	Figure A.5
property expressed as data	property: property type (0..*) data: data type (1)	Figure A.5

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.3 Interface Reference and Binding Type System

Table A.5 and Figures A.6 and A.7 contain the MOF Classes for types defined in ODP Interface References and Binding framework. Note that ODP Interface References and Binding reuses types and relationships defined in ITU-T Rec. X.903 | ISO/IEC 10746-3.

Table A.5 – ODP concepts in ODP Interface References and Binding framework

MOF Class	super classes	attributes (name and type)	contained elements (and cardinality)	See Figure
interface reference type				Figure A.6
null reference type	interface reference type			Figure A.6
direct reference type	interface reference type		interface description type (1)	Figure A.6
non-interpreted reference type	interface reference type	opaque_information: any		Figure A.6
interface description type		channel_class: any causality_info: any location_info: any relocation_info: any group_info: any security_info: any additional_info: any		Figure A.6
interface type				Figure A.6 Figure A.7
operational interface type	interface type	name: string	operation description type (0..*)	Figure A.7
operation description type			QoS statement type (1)	Figure A.7
operation type		name: string		Figure A.7
stream interface type	interface type	name: string	flow description type (0..*)	Figure A.7
flow description type			QoS statement type (1)	Figure A.7
flow type		name: string		Figure A.7

Table A.6 contains the relationships defined in ODP Interface References and Binding framework.

Table A.6 – ODP relationship types in ODP Interface References and Binding framework

MOF Association	Association Ends: Names, Classes, Aggregations, Cardinalities	See Figure
reference specifies interpreter	reference: non-interpreted reference type (1) interpreter: interface reference type (composite 1)	Figure A.6
referent specifies referenced	referent: direct reference type (0..*) referenced: interface type (1)	Figure A.6
direct reference has interface description	direct reference: direct reference type (1) interface description: interface description type (composite 1)	Figure A.7
operational interface has operation description	operational interface: operational interface type (1) operation description: operation description type (composite 0..*)	Figure A.7
operation description has QoS statement	operation description: operation description type (1) QoS statement: QoS statement type (composite 0..*)	Figure A.7
operation exhibits behaviour	operation: operation type (0..*) behaviour: behaviour type (1)	Figure A.7
stream interface has flow description	stream interface: stream interface type (1) flow description: flow description type (composite 0..*)	Figure A.7
flow description has QoS statement	flow description: flow description type (1) QoS statement: QoS statement type (composite 1)	Figure A.7
flow exhibits behaviour	flow: flow type (0..*) behaviour: behaviour type (1)	Figure A.7

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

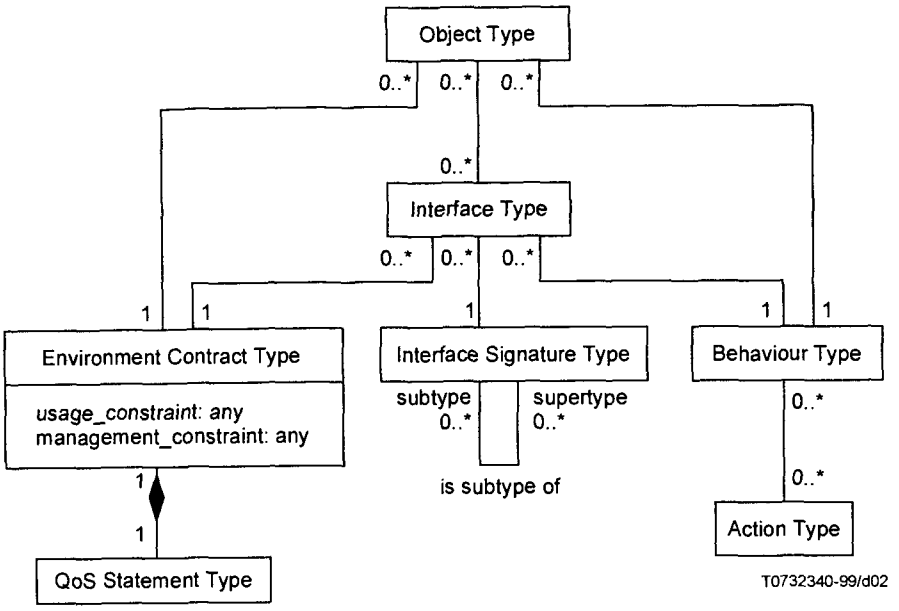


Figure A.1 – Computational Object and Interface Types

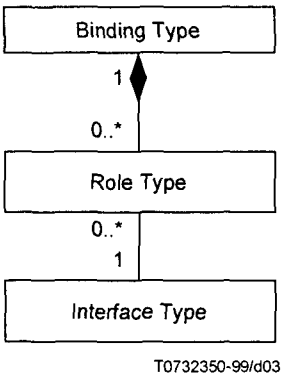


Figure A.2 – Binding Types

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

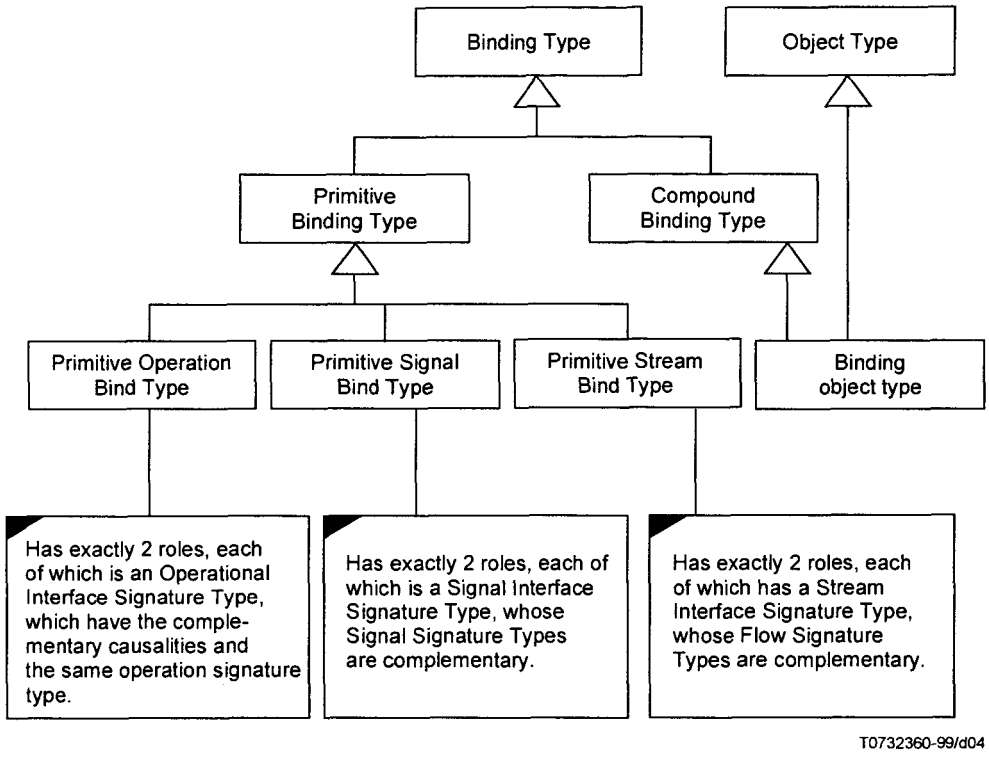


Figure A.3 – Binding Subtypes

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

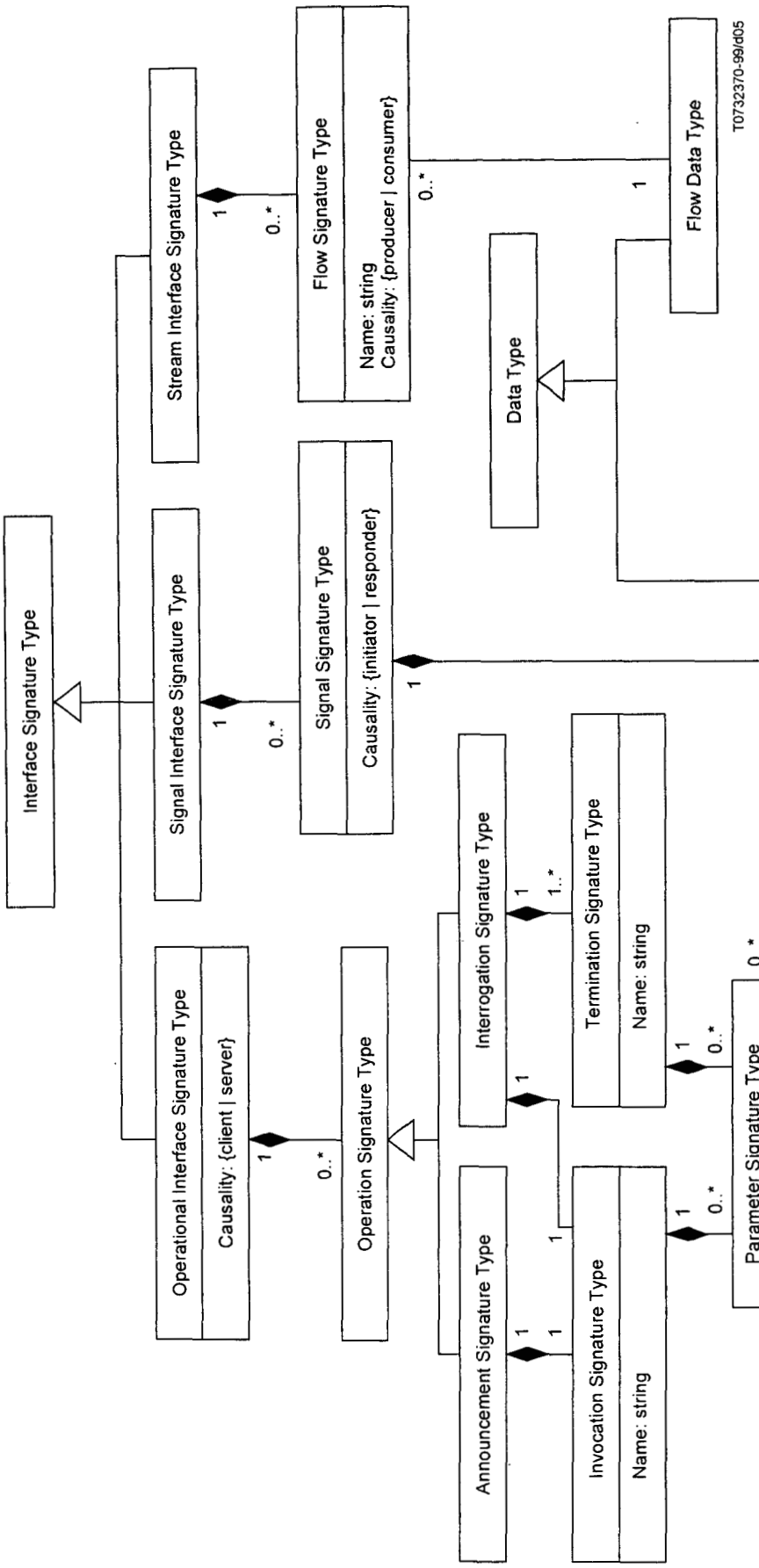


Figure A.4 – Interface Signature Types and Data Types

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

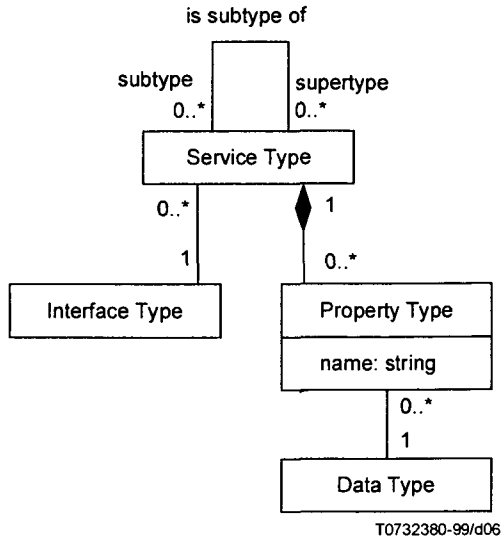


Figure A.5 – Trading Types

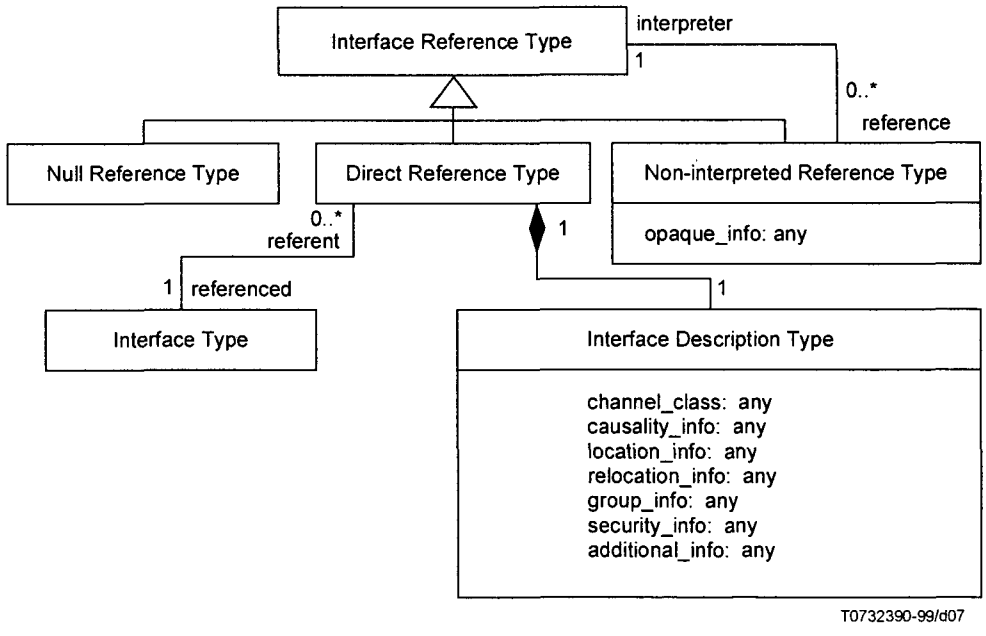


Figure A.6 – Interface References

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

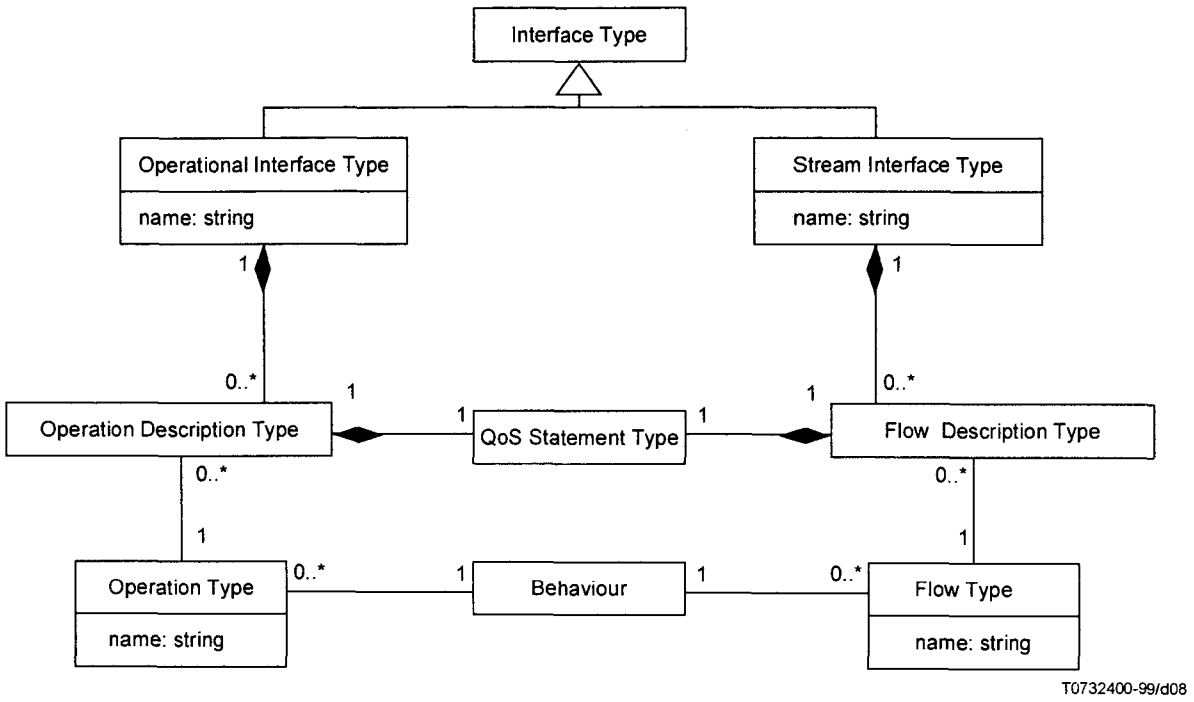


Figure A.7 – Interface Types for Binding

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

Suggested type languages

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

This is an informative annex, documenting type languages suitable for the expression of target concepts in ODP Recommendations | International Standards.

- ODP IDL for the definition of computational operational interface signatures.
- TINA ODL (Object Definition Language) for the definition of computational object signatures, computational operational interface signatures, and computational stream interface signatures.

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

instance (of a type), 2
instantiation (of an <X>), 3
interaction, 1
interface, 3, 5, 12, 13
Interface definition language, 2
Interface Definition Language, 5
interface description type, 4, 17
interface reference, 4
Interface Reference, 17
interface reference type, 17
Interface references and binding, 2
Interface References and Binding, 4
interface signature, 3
interface signature type, 14, 15
interface type, 4, 14, 15, 16, 17
interrogation signature, 3
interrogation signature type, 14, 16
invariant schema, 3, 11
invocation, 3
invocation signature type, 14, 15, 16

L

location information, 4

M

management_constraint, 15
meta-meta-model, 6, 11
meta-model, 6, 11
Meta-Object Definition Language, 5
Meta-Object Facility, 2, 5, 6, 24
model, 6, 10, 11
MOF model, 10
MOF repository, 11

N

name, 3
Naming Framework, 2
non-interpreted reference type, 4, 17
null reference type, 4, 17

O

object, iv, 3, 5, 6
Object, 6, 12
Object Constraint Language (OCL), 5
Object Management Group, 2, 5
Object Management Group (OMG), 5
object signature, 3
object type, 14, 15
objective, 6
Objective, 6
obligation, 3
OCL, 11
ODP, iv, 5, 15, 16, 17
ODP Enterprise Language, 6
ODP function, iv
ODP IDL, 5, 12, 23
ODP Interface References and Binding, 10
ODP Naming Framework, 1
ODP standard, 3
ODP system, 1, 3, 6
ODP type framework, 14
ODP Type Repository (TR), 5
ODP-RM, iv, 1, 5
ODP-RM Part 3: Architecture, 14, 15, 16
OMG, 5
opaque information, 4
Open distributed processing, 5
Open Distributed Processing: Reference Model, 5
operation, 3, 5
operation description type, 4, 17
operation signature type, 3, 14
operation type, 4, 17
operational interface signature, 3
operational interface signature type, 14
operational interface type, 4, 16, 17

P

parameter data type, 15, 16
parameter signature type, 15, 16
parameter type, 16
policy, 3, 6
primitive binding, 3
primitive binding type, 14

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

primitive operation binding type, 14
primitive signal binding type, 3, 14
primitive stream binding type, 4, 14
property type, 4, 16
publication, 9, 10
Publication, 10

Q

QoS statement type, 15, 17
quality-of-service statement type, 4

R

reflective package, 11, 12
relation, 4
relationship, iv, 1, 4, 6
relationship type, 4, 15, 16, 17
relocation information, 4
role, 3, 4, 6, 7
Role, 6, 7, 8
role type, 14, 15, 16

S

security information, 4
service, 5
service invocation, 5
service offer, 4
service type, 4, 16
signal, 5
signal interface signature type, 4
signal signature type, 4, 14, 15, 16
state (of an object), 3
static schema, 4, 11, 12
stream, 5
stream interface signature type, 4, 14
stream interface type, 4, 16, 17
subtype, 3, 15, 16
subtype relationship, iv
supertype, 3, 15, 16

T

target concept, iv, 5
template, 3, 5, 6
termination signature type, 4, 15, 16
TR type description, 9
TR type system, 6
TR type system description, 7, 8, 9, 11
TR type system description storage, 9
TR type system query, 8
TR type system representation, 9
TR type system user, 8
trading function, 2, 4, 16
type, iv, 1, 3, 4, 5, 6
type author, 8
type creation, 8
type definition, iv, 5
type deletion, 8
type description, iv, 1, 6, 8, 9, 11
type description storage, 9
type domain, iv
type language, iv, 1, 5, 23
type modification, 8
type publication, 8
type publisher, 8
type relation, 5, 6
type relationship, 6
type repository, iv, 1, 6, 9
Type repository, 5, 8, 11
type repository communities, 6, 7, 8
type repository community creation, 9
type repository function, iv, 1, 6
Type repository interworking, 10
type representation, 9
type system, iv, 1, 6
type system author, 8
type system creation, 8
type system deletion, 8
type system description, 6, 8, 9, 11
Type system interworking, 10
type system modification, 8
type system publication, 8
type system publisher, 8

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

type system query, 8
type system user, 8
type system verification, 8
type systems, 5
type user, 8
type verification, 8
TypeCode, 12

U

Unified Modelling Language (UML), 2, 5, 11, 14
usage_constraint, 15

V

verification, 9
viewpoint, iv, 3
viewpoint languages, iv

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 35.080

Price based on 28 pages

© ISO/IEC 2001 – All rights reserved

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

Not for Resale