INTERNATIONAL STANDARD

ISO/IEC 10746-1

First edition
1998-12-15

Information technology — Open Distributed Processing — Reference model: Overview

Technologies de l'information — Traitement réparti ouvert — Modèle de référence: Présentation

Contents

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

International Standard ISO/IEC 10746-1 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 Recommendation X.901.

ISO/IEC 10746 consists of the following parts, under the general title *Information technology — Open Distributed Processing — Reference Model*:

— *Part 1: Overview*

— *Part 2: Foundations*

— *Part 3: Architecture*

— *Part 4: Architectural semantics*

Annex A of this part of ISO/IEC 10746 is for information only.

Introduction

The rapid growth of distributed processing has led to a need for a coordinating framework for the standardization of Open Distributed Processing (ODP). This Reference Model provides such a framework. It creates an architecture within which support of distribution, interworking and portability can be integrated.

The Reference Model of Open Distributed Processing, ITU-T Rec. X.901 | ISO/IEC 10746-1 to ITU-T Rec. X.904 | ISO/IEC 10746-4, is based on precise concepts derived from current distributed processing developments and, as far as possible, on the use of formal description techniques for specification of the architecture.

The RM-ODP (ISO/IEC 10746) consists of:

- ITU-T Rec. X.901 | ISO/IEC 10746-1: **Overview:** contains a motivational overview of ODP giving scoping, justification and explanation of key concepts, and an outline of the ODP architecture. It contains explanatory material on how this Reference Model is to be interpreted and applied by its users, who may include standards writers and architects of ODP systems. It also contains a categorization of required areas of standardization expressed in terms of the reference points for conformance identified in ITU-T Rec. X.903 | ISO/IEC 10746-3. These common texts are not normative.
- ITU-T Rec. X.902 | ISO/IEC 10746-2: **Foundations:** contains the definition of the concepts and analytical framework for normalized description of (arbitrary) distributed processing systems. This is only to a level of detail sufficient to support ITU-T Rec. X.903 | ISO/IEC 10746-3 and to establish requirements for new specification techniques. These common texts are normative.
- ITU-T Rec. X.903 | ISO/IEC 10746-3: **Architecture:** contains the specification of the required characteristics that qualify distributed processing as open. These are the constraints to which ODP standards must conform. It uses the descriptive techniques from ITU-T Rec. X.902 | ISO/IEC 10746-2. These common texts are normative.
- ITU-T Rec. X.904 | ISO/IEC 10746-4: **Architectural semantics:** contains a normalization of the ODP modelling concepts defined in ITU-T Rec. X.902 | ISO/IEC 10746-2, clauses 8 and 9. The normalization is achieved by interpreting each concept in terms of the constructs of the different standardized formal description techniques. These common texts are normative.

This Recommendation | International Standard contains one annex.

Clause 6 explains the business benefits of open distributed systems, and how the RM-ODP and its associated ODP standards will enable corporations to realize these benefits. This clause states the “promises” of ODP – plug-and-play building blocks and system integration tools for distributed systems.

Clauses 7 to 10 explain what RM-ODP and its distributed functions are about. These clauses justify how RM-ODP supports the development of plug-and-play building blocks and system integration tools for distributed systems.

Clause 11 shows how ODP standards and specifications by other groups can be referenced in an ODP specification of a system. These relationships are key to ODP’s ability to enable integration of disparate technologies.

Clause 12 contains examples that demonstrate the use of RM-ODP and the use of underlying principles to solve business problems.

INTERNATIONAL STANDARD

ITU-T RECOMMENDATION

INFORMATION TECHNOLOGY – OPEN DISTRIBUTED PROCESSING – REFERENCE MODEL: OVERVIEW

1 Scope and field of application

This Recommendation | International Standard:

- gives an introduction and motivation for ODP;
- provides an overview of the Reference Model of Open Distributed Processing (RM-ODP) and an explanation of its key concepts;
- gives guidance on the application of the RM-ODP.

This Recommendation | International Standard covers both overview and detailed explanation, and can be consulted in various ways when reading the standards:

a) if you intend to read only this Recommendation | International Standard, to gain a general understanding of the importance of ODP to your organization, concentrate on clause 6;

b) if you intend to study the whole RM-ODP, you should also read clause 6 before moving on to ITU-T Rec. X.902 | ISO/IEC 10746-2 and ITU-T Rec. X.903 | ISO/IEC 10746-3;

c) as you read ITU-T Rec. X.902 | ISO/IEC 10746-2 and ITU-T Rec. X.903 | ISO/IEC 10746-3 you may wish to consult clauses 7 to 10, which give supporting explanation for the various concepts that these common texts define;

d) when you have completed a first reading of ITU-T Rec. X.902 | ISO/IEC 10746-2 and ITU-T Rec. X.903 | ISO/IEC 10746-3, read clauses 11 and 12 which discuss the use of standards in ODP system specifications, and provide some examples of applying the ODP concepts in the specification of systems.

2 Normative references

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

2.1 Identical Recommendations | International Standards

- ITU-T Recommendation X.200 (1994) | ISO/IEC 7498-1:1994, *Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model*.
- ITU-T Recommendation X.207 (1993) | ISO/IEC 9545:1994, *Information technology – Open Systems Interconnection – Application Layer structure*.
- ITU-T Recommendation X.720 (1993) | ISO/IEC 10165-1:1993, *Information technology – Open Systems Interconnection – Structure of management information: Management Information Model*.

- 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.904 (1997) | ISO/IEC 10746-4:1998, *Information technology – Open distributed processing – Reference Model: Architectural semantics.*

2.2 Paired Recommendations | International Standards equivalent in technical content

- ITU-T Recommendation X.290 (1995), *OSI conformance testing methodology and framework for protocol Recommendations for ITU-T applications – General concepts*.
- ISO/IEC 9646-1:1994, *Information technology – Open Systems Interconnection – Conformance testing methodology and framework – Part 1: General concepts.*

2.3 International Standards

- ISO/IEC 11578-2[1)]: *Information technology – Open Systems Interconnection – Remote Procedure Call (RPC) – Part 2: Interface Definition Notation.*
- ISO/IEC TR 10000-1:1995, *Information technology – Framework and taxonomy of International Standardized Profiles – Part 1: General principles and documentation framework*.

3 Definitions

3.1 Definitions in this Recommendation | International Standard

There are no definitions in this Recommendation | International Standard.

3.2 Definitions from other Recommendations | International Standards

This Recommendation | International Standard makes use of the following terms defined in ITU-T Rec. X.902 | ISO/IEC 10746-2:

- Abstraction;
- Action;
- Action template;
- Activity;
- Architecture;
- Atomicity;
- Base class;
- Behaviour (of an object);
- Behavioural compatibility;
- Binding;
- Binding behaviour;
- Chain (of actions);
- Class;
- Client object;
- Communication;
- Compliance;
- Composite object;
- Composition;
- Configuration (of objects);
- Conformance points;

---

1) To be published.

- Consumer object;
- Contract;
- Contractual context;
- Creation;
- Data;
- Decomposition;
- Deletion;
- Derived class;
- Distribution transparency;
- Entity;
- Environment (of an object);
- Environment contract;
- Error;
- Establishing behaviour;
- Failure;
- Fault;
- Identifier;
- Information;
- Initiating object;
- Instance;
- Instantiation;
- Interaction;
- Interface;
- Interface signature;
- Internal action;
- Interworking reference point;
- Introduction (of an <X>);
- Invariant;
- Liaison;
- Location in space;
- Management information;
- Name;
- Name resolution;
- Naming domain;
- Notification;
- Object;
- Obligation;
- ODP standards;
- ODP system;
- Perceptual reference point;
- Permission;
- Persistence;
- Policy;
- Portability;
- Producer object;
- Programmatic reference point;
- Prohibition;
- Quality of service;

- Reference point;
- Refinement;
- Responding object;
- Role;
- Server object;
- State;
- Subclass;
- Subtype;
- System;
- Template class;
- Template type;
- Terminating behaviour;
- Thread;
- Trading;
- Type;
- Unbinding;
- Viewpoint.

This Recommendation | International Standard makes use of the following terms defined in ITU-T Rec. X.903 | ISO/IEC 10746-3:

- <Viewpoint> language;
- Access control information;
- Access transparency;
- Announcement;
- Basic engineering object;
- Binder;
- Binding object;
- Capsule;
- Capsule manager;
- Channel;
- Checkpoint;
- Checkpointing;
- Cluster manager;
- Cluster template;
- Communication interface;
- Community;
- Compound binding action;
- Computational viewpoint;
- Deactivation;
- Dynamic schema;
- Engineering viewpoint;
- Enterprise viewpoint;
- Explicit binding;
- Failure transparency;
- Federation;
- Flow;
- Hide;
- Implicit binding;
- Information viewpoint;

- Interceptor;
- Interrogation;
- Invariant schema;
- Invocation;
- Location transparency;
- Migration;
- Migration transparency;
- Node;
- Nucleus;
- ODP function;
- Operation interface;
- Operation interface signature;
- Persistence transparency;
- Primitive binding actions;
- Protocol object;
- Reactivation;
- Recovery;
- Relocation transparency;
- Relocator;
- Replication schema;
- Replication transparency;
- Security authority;
- Security domain;
- Security policy;
- Signal;
- Signal interface;
- Signal interface signature;
- Static schema;
- Stream interface;
- Stream interface signature;
- Stub;
- Target;
- Technology viewpoint;
- Termination;
- Transaction transparency;
- Validate.

This Recommendation | International Standard makes use of the following terms defined in ISO/IEC 9646:

- Implementation conformance statement;
- Implementation Extra Information for Testing;
- Point of Control and Observation.

This Recommendation | International Standard makes use of the following terms defined in ITU-T Rec. X.200 | ISO/IEC 7498-1:

- Open System;
- Abstract syntax;
- Transfer syntax.

4 Abbreviations

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

A-profile	Application profile
ACID	Atomicity Consistency Isolation Durability
AE(I)	Application Entity (Invocation)
ALS	Application Layer Structure
AP(I)	Application Process (Invocation)
API	Application Program Interface
ASO	Application Service Object
BEO	Basic Engineering Object
CAD	Computer Aided Design
CD	Compact Disk
CIM	Computer Integrated Manufacturing
CMIP	Common Management Information Protocol
CMIS	Common Management Information Service
DL	Definition Language
F-profile	Format and presentation profile
FDT	Formal Description Techniques
GUI	Graphical User Interface
HCI	Human Computer Interface
HDTV	High Definition TV
ICS	Implementation Conformance Statement
IDL	Interface Definition Language
IT	Information Technology
IXIT	Implementation Extra Information for Testing
MIM	Management Information Model
MMC(S)	Multimedia Conferencing (System)
ODP	Open Distributed Processing
OMG	Object Management Group
OMT	Object Modelling Technique
OSE	Open System Environment
OSF	Open Software Foundation
OSI	Open Systems Interconnection
PCO	Point of Control and Observation
QOS	Quality of Service
RDA	Remote Database Access
RM-ODP	Reference Model of Open Distributed Processing
RPC	Remote Procedure Call
T-profile	Transfer profile
TINA	Telecommunication Information Networking Architecture
ULA	Upper Layers Architecture

5 Conventions

The following conventions are specific to this Recommendation | International Standard:

1) The first use in clauses 7 and 8 of formal terms from ITU-T Rec. X.902 | ISO/IEC 10746-2 and ITU-T Rec. X.903 | ISO/IEC 10746-3 is italicized.

2) Examples in clause 12 use OMT drawing conventions defined in [Rumbaugh 91].

3) In diagrams:

   - Objects are represented as ovals or circles.

   - The symbol "⊥" protruding from an object represents an interface.

6 ODP standardization

6.1 Objectives and motivation

The objective of ODP standardization is the development of standards that allow the benefits of distributing information processing services to be realized in an environment of heterogeneous IT resources and multiple organizational domains. These standards address constraints on system specification and the provision of a system infrastructure that accommodate difficulties inherent in the design and programming of distributed systems.

Distributed systems are important because there is a growing need to interconnect information processing systems. This need arises because of organizational trends such as downsizing, which demand the exchange of information both between groups within an organization and between cooperating organizations. Advances in technology are making it possible to respond to these trends by giving increasing importance to information service networks and personal workstations, and by permitting the construction of applications distributed across large configurations of interconnected systems.

In order both to manage system distribution and to exploit it (e.g. use the potential for availability, performance, dependability and cost optimization), organizations must deal with a number of key characteristics of system distribution:

- **Remoteness**: Components of a distributed system may be spread across space; interactions may be either local or remote.

- **Concurrency**: Any component of a distributed system can execute in parallel with any other components.

- **Lack of global state**: The global state of a distributed system cannot be precisely determined.

- **Partial failures**: Any component of a distributed system may fail independently of any other components.

- **Asynchrony**: Communication and processing activities are not driven by a single global clock. Related changes in a distributed system cannot be assumed to take place at a single instant.

- **Heterogeneity**: There is no guarantee that components of a distributed system are built using the same technology and the set of various technologies will certainly change over time. Heterogeneity appears in many places: hardware, operating systems, communication networks and protocols, programming languages, applications, etc.

- **Autonomy**: A distributed system can be spread over a number of autonomous management or control authorities, with no single point of control. The degree of autonomy specifies the extent to which processing resources and associated devices (printers, storage devices, graphical displays, audio devices, etc.) are under the control of separate organizational entities.

- **Evolution**: During its working life, a distributed system generally has to face many changes which are motivated by technical progress enabling better performance at a better price, by strategic decisions about new goals, and by new types of applications.

- **Mobility**: The sources of information, processing nodes, and users may be physically mobile. Programs and data may also be moved between nodes, e.g. in order to cope with physical mobility or to optimize performance.

Building such systems is not easy. It requires an architecture and, because a single engineering solution will not meet all requirements, it must be a flexible architecture. Moreover, since a single vendor will not have all of the answers, it is essential that the architecture, and any functions necessary to implement the architecture, be defined in a set of standards, so that multiple vendors can collaborate in the provision of distributed systems. Such standards will enable systems to be built that:

- Are **open** – Providing both portability (execution of components on different processing nodes without modification) and interworking (meaningful interactions between components, possibly residing in different systems).
- Are **integrated** – Incorporating various systems and resources into a whole without costly ad-hoc developments. This may involve systems with different architectures, and different resources with different performance. Integration helps to deal with heterogeneity.
- Are **flexible** – Capable both of evolving and of accommodating the existence and continued operation of legacy systems. An open distributed system should be capable of facing run-time changes – for example, it should be capable of being dynamically reconfigured to accommodate changing circumstances. Flexibility helps to deal with mobility.
- Are **modular** – Allowing parts of a system to be autonomous, but interrelated. Modularity is the basis for flexibility.
- Can be **federated** – Allowing a system to be combined with systems from different administrative or technical domains to achieve a single objective.
- Are **manageable** – Allowing the resources of a system to be monitored, controlled and managed in order to support configuration, QOS and accounting policies.
- Meet **quality of service** needs – Covering, for example, provision of timeliness, availability and reliability in the context of remote resources and interactions, together with provision of fault tolerance that allows the remainder of a distributed system to continue to operate in the event of failure of some part. Provision of fault tolerance (and of dependability in general) is necessary within large distributed systems where it is unlikely that all parts of the system will ever be operational simultaneously.
- Are **secure** – Ensuring that system facilities and data are protected against unauthorized access. Security requirements are made more difficult to meet by remoteness of interactions, and mobility of parts of the system and of the system users.
- Offer **transparency** – Masking from applications the details and the differences in mechanisms used to overcome problems caused by distribution. This is a central requirement arising from the need to facilitate the construction of distributed applications. Aspects of distribution which should be masked (totally or partially) include: heterogeneity of supporting software and hardware, location and mobility of components, and mechanisms to achieve the required level for QOS in the face of failures (e.g. replication, migration, checkpointing, etc.).

6.2 Realization

ODP standardization has four fundamental elements:

- an object modelling approach to system specification;
- the specification of a system in terms of separate but interrelated viewpoint specifications;
- the definition of a system infrastructure providing distribution transparencies for system applications;
- a framework for assessing system conformance.

6.2.1 Object modelling

Object modelling provides a formalization of well-established design practices of abstraction and encapsulation. Abstraction allows the description of system functionality to be separated from details of system implementation. Encapsulation allows the hiding of heterogeneity, the localization of failure, the implementation of security and the hiding of the mechanisms of service provision from the service user.

The object modelling concepts cover:

- Basic modelling concepts – Providing rigorous definitions of a minimum set of concepts (action, object. interaction and interface) that form the basis for ODP system descriptions and are applicable in all viewpoints.

- Specification concepts – Addressing notions such as type and class that are necessary for reasoning about specifications and the relations between specifications, provide general tools for design, and establish requirements on specification languages.
- Structuring concepts – Building on the basic modelling concepts and the specification concepts to address recurrent structures in distributed systems, and cover such concerns as policy, naming, behaviour, dependability and communication.

6.2.2 Viewpoint specifications

A viewpoint (on a system) is an abstraction that yields a specification of the whole system related to a particular set of concerns. Five viewpoints have been chosen to be both simple and complete, covering all the domains of architectural design. These five viewpoints are:

- the enterprise viewpoint, which is concerned with the purpose, scope and policies governing the activities of the specified system within the organization of which it is a part;
- the information viewpoint, which is concerned with the kinds of information handled by the system and constraints on the use and interpretation of that information;
- the computational viewpoint, which is concerned with the functional decomposition of the system into a set of objects that interact at interfaces – enabling system distribution;
- the engineering viewpoint, which is concerned with the infrastructure required to support system distribution;
- the technology viewpoint, which is concerned with the choice of technology to support system distribution.

For each viewpoint there is an associated viewpoint language which can be used to express a specification of the system from that viewpoint. The object modelling concepts give a common basis for the viewpoint languages and make it possible to identify relationships between the different viewpoint specifications and to assert correspondences between the representations of the system in different viewpoints.

6.2.3 Distribution transparency

Distribution transparencies enable complexities associated with system distribution to be hidden from applications where they are irrelevant to their purpose. For example:

- access transparency masks differences of data representation and invocation mechanisms for services between systems;
- location transparency masks the need for an application to have information about location in order to invoke a service;
- relocation transparency masks the relocation of a service from applications using it;
- replication transparency masks the fact that multiple copies of a service may be provided in order to provide reliability and availability.

ODP standards define functions and structures to realize distribution transparencies. However, there are performance and cost tradeoffs associated with each transparency and only selected transparencies will be relevant in many cases. Thus, a conforming ODP system must implement those transparencies that it supports in accordance with the relevant standards, but it is not required to support all transparencies.

6.2.4 Conformance

The basic characteristics of heterogeneity and evolution imply that different parts of a distributed system can be purchased separately, from different vendors. It is therefore very important that the behaviours of the different parts of a system be clearly defined, and that it be possible to assign responsibility for any failure to meet the system's specifications.

The framework defined to govern the assessment of conformance addresses these issues. It covers:

- identification of the conformance points within the set of viewpoint specifications at which observations of conformance can be made;
- definition of classes of conformance point;
- specification of the nature of conformance statements to be made in each viewpoint and the relation between them.