INTERNATIONAL STANDARD

ISO/IEC 14753

First edition
1999-07-15

Information technology — Open Distributed Processing — Interface references and binding

Technologies de l'information — Traitement distribué ouvert — Références et liaisons d'interfaces

Reference number
ISO/IEC 14753:1999(E)

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

Printed in Switzerland

CONTENTS

Page

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.

International Standard ISO/IEC 14753 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.930.

Annexes A and B form an integral part of this International Standard. Annex C 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). The Reference Model of ODP provides such a framework. It creates an architecture within which support of distribution, interworking and portability can be integrated.

One of the components of the architecture is the ODP binding function. The binding function provides the means to establish liaisons and create channels across autonomous systems in order to support interworking and communication between objects. An interface reference embodies the information needed to establish bindings and further embodies the information required to maintain bindings between computational objects in the presence of distribution.

INTERNATIONAL STANDARD

ITU-T RECOMMENDATION

INFORMATION TECHNOLOGY – OPEN DISTRIBUTED PROCESSING – INTERFACE REFERENCES AND BINDING

1 Scope and Field of application

1.1 Scope

Interface references are crucial to interworking between ODP systems and federation of groups of ODP systems. An interface reference embodies the information needed to establish bindings, including binding to objects at nodes that support several different communication protocols and binding to objects in different management domains. An interface reference further embodies the information required for the engineering mechanism to maintain bindings between computational objects in the presence of distribution transparencies such as migration transparency. They are the foundation of ODP location and relocation transparency.

This Recommendation | International Standard includes:

- a framework for binding interfaces and a generic binding protocol (for both stream and operational interfaces);
- a specification of the generic information structure of interface references (for both stream and operational interfaces);
- representation(s) for interface references when transferred using standardized protocols;
- identification of procedures for the management and transfer of interface references with respect to individual transparencies;
- identification of node management interfaces related to binding and federation which create or transform interface references;
- identification of requirements for quality of service information and for invocation of QoS or related measurement procedures.

This Recommendation | International Standard provides an engineering description of the functionality needed to support the computational binding of objects in ODP systems. Security and support for group communication are important issues, but not within the scope of this Recommendation | International Standard.

1.2 Field of Application

This Recommendation | International Standard enables interworking between ODP systems.

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

2.1 Identical Recommendations | International Standards

- ITU-T Recommendation X.901 (1997) | ISO/IEC 10746-1:1998, *Information technology – Open distributed processing – Reference Model:* Overview.
- 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 – ODP Naming framework.*
- ITU-T Recommendation X.931 (1998) | ISO/IEC 14752:1999, *Information technology – Open distributed processing – Protocol Support for Computational Interactions.*
- ITU-T Recommendation X.950 (1997) | ISO/IEC 13235-1:1998, *Information technology – Open distributed processing – Trading function: Specification.*
- ITU-T Recommendation X.960[1)] | ISO/IEC 14769[1)], *Information technology – Open distributed processing – Type repository function.*
- ISO/IEC 9075-1[1)], *Information technology – Database language SQL – Part 1: Frame.*

2.2 Specifications of the Object Management Group

- CORBA: The Common Object Request Broker: Architecture and Specification, Revision 2.1, Object Management Group, August 1997 (OMG Doc Number Formal/97-09-01).

Temporary Note: A reference explanatory report is circulated with the DIS ballot on this specification.

3 Definitions

3.1 Definitions in this Recommendation | International Standard

This Recommendation | International Standard defines the following terms.

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:

- <X> domain;
- <X> template;
- action;
- activity;
- behaviour;
- binding;
- compliance;
- configuration;
- conformance point;
- contract;

[1)] To be published.

- contractual context;
- distribution transparency;
- environment contract;
- epoch;
- failure;
- interaction point;
- interface;
- interworking reference point;
- liaison;
- location;
- notification;
- policy;
- quality of service;
- role;
- subtype;
- type (of an X).

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

- <X> federation;
- announcement;
- basic engineering object;
- binder;
- channel;
- compound binding;
- engineering interface reference;
- explicit binding;
- implicit binding;
- interceptor;
- node;
- operation interface;
- protocol object;
- signal;
- signature;
- stub;
- stream interface.

4 Abbreviations

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

QoS	Quality of Service
ODP	Open Distributed Processing
IIOP-IOR	Internet Inter-ORB Protocol – Interoperable Object Reference

5 Conventions

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

In diagrams:

- objects are represented as ovals or circles;
- the symbol '⊥' protruding from an object represents an interface;
- the symbol '<>' represents a containment of objects;
- the symbol '♦ ' represents dependent containment between objects;
- the symbol 'n .. *' denotes that the cardinality of an association must exceed n.

6 Overview of interface references and binding

6.1 Rationale

The objective of ODP standardization is to develop standards that realize the benefits of distributing information processing services in a heterogeneous environment of IT resources and multiple organizational domains. These standards address constraints on system specifications and provide a system infrastructure that addresses difficulties inherent in the design and programming of distributed systems.

Distributed systems are important because of a growing need to interconnect information processing systems. This need arises because of organizational trends (such as downsizing), which demand the exchange of information not only between groups within an organization but also between cooperating organizations. Technological advances make it possible to respond to these trends giving increased importance to information networks and personal workstations, and enabling distributed applications to be constructed across large configurations of interconnected systems.

In order to set up cooperation between organizations and their information systems, the parties must define and agree on a relationship and then maintain it. This relationship is often defined as a contract in commercial environments. To achieve cooperation between systems, after some initial contact, agreements must be made, contracts negotiated and interfaces defined, created and made available. Interworking between ODP systems requires standardized communication methods between objects that reside at autonomous systems.

This Recommendation | International Standard provides a framework for binding, including a refinement of the binding model of ITU-T Rec. X.903 | ISO/IEC 10746-3 and a generic structure of interface references. This Recommendation | International Standard is structured according to ODP viewpoints.

6.2 Overview of the binding process

6.2.1 Obtaining interface references

Whenever an interface is created (either explicitly, or during object creation), an interface reference for it is generated. This interface reference can be passed via existing communication channels from the object providing the interface. Its recipients can then pass it on, possibly via several steps, until it reaches some object which wishes to interact with the interface.

The interface reference contains sufficient information to initiate the binding process which makes interaction involving the interface possible. Often, an object will create a binding involving itself and an interface whose interface reference it has just received. However, in the general case, the creation of a binding involves a set of interfaces, not necessarily including an interface to the object performing the binding action. Such third party bindings may occur, for example, when setting up multimedia streams.

An object which is to create a binding must have information on:

a) the set of interface references for the interfaces to be bound;

b) the type of the binding needed, possibly in the form of a reference to a suitable binding template;

c) the quality of service required of the binding.

6.2.2 Binding process

This description is in terms of compound binding, in which a visible computational binding object is created. The simpler implicit or explicit primitive binding process is generally similar.

In a computational specification, an object creates a binding by performing a binding action. From an engineering point of view, it does this by invoking a binding factory, representing the mechanisms needed to allocate resources, negotiating detailed quality agreements and establishing communications paths.

In either viewpoint, the information outlined above is required to start the process. The result is the creation of a binding object and the return of an interface reference for a control interface which it provides. This control interface allows the initiator, or any other object it passes the reference to, to control the binding, to request notification of significant events, or to request destruction of the binding. The exact detail of this interface depends on the binding type.

Once the binding has been created, it can support the behaviour, in terms of operation invocations or stream flows, defined by the binding type.

6.2.3 Negotiating the properties of the binding

The way an object can interact with its environment depends on the capabilities (in terms of available protocols, stubs, etc.) of the infrastructure supporting the object, and on a set of quality of service constraints defined in the object's environment contract.

When an interface is created, the interface reference contains information about these capabilities, together with enough naming information to allow the interface to be located or relocated, and possibly some items from the object's environment contract to indicate levels of service which can be achieved. This information indicates properties of the interface which will be true for any binding it may become involved in, and therefore provides the starting point for the negotiation of binding properties.

In order to support interface reference passing and binding across federation boundaries, and to keep the size of interface references within reasonable bounds, the information to be passed may be included in shorthand form, or by reference to supporting services, rather than being explicitly encoded in the reference. The detailed format of an interface reference may be transformed as it is passed from object to object.

The binding factory combines the information in the interface references it receives with constraints in the binding type or from the initiator of the binding, to steer the negotiation of the binding's properties, and to decide on the level of resources required. This process may involve negotiation with the objects being bound to take into account their current availability of resources and aspects of their environment contracts which were not included in the interface references. A binding can only be created if a satisfactory set of properties can be identified which is consistent with the requested binding action and the environment contracts of all the objects involved in the binding.

6.2.4 Renegotiating the properties of the binding

In many situations, it may be necessary for a binding to evolve during its lifetime, either to change its properties or to modify the set of interfaces being bound. The kind of changes that can occur will depend on the type of the binding, and this will be constrained by the capabilities of the engineering infrastructure available to support it. For example, elaborate facilities for modifying bindings are likely to be required in an environment supporting mobile or nomadic computing platforms.

Changes to the binding configuration, or to the quality of service being either required or offered, will generally involve renegotiation between the participants, and may result in the addition or replacement of some supporting components. For example, the migration of an object into a different kind of environment may require modification, and thus renegotiation, of quality of service, involving the use of different network facilities, different data representations and different protocols.

6.2.5 Quality monitoring and control

In addition to the ability to modify the quality of service using the control interface of the binding object, there is, in general, a need to monitor the quality of service actually achieved. To do this, monitoring mechanisms may need to be attached to specific reference points at each of the bound interfaces.

The maintenance of agreed quality of service may involve the creation of internal feed back processes linking the observation of achieved quality at or between interfaces with modification by some quality management object of the requirements on particular parts of the binding, using the control interface to the binding object.

6.2.6 Destroying a binding

The definition of when a binding ceases to exist is part of its behaviour, and so depends on its type. A binding will generally cease to exist as a result of a request to do so, received at its control interface. It may also cease to exist as a result of an action internal to the binding, such as detection of a failure of communication or of one or more of the objects being bound.

Destruction of a binding does not, in general, imply the destruction of the interfaces being bound, or of the objects providing those interfaces.

7 Enterprise viewpoint

The purpose of the binding function is to bind together interfaces (signal, operational, stream) to enable communication between objects. The binding function selects and names the communication interfaces, checks that these interfaces conform to each other, checks that these interfaces initially satisfy the QoS requirements of each other, and forms a liaison between the interfaces. The binding liaison guarantees that the objects can interact. The binding function also provides for the management of the binding and eventual destruction of the binding.

Binding actions are of two kinds: primitive binding actions in which the objects involved are modelled as interacting directly and compound binding actions in which an intermediate object represents the mechanisms providing the binding.

Transferring operation invocations and implementing binding actions require support of the mechanisms and functions of the RM-ODP infrastructure.

7.1 Communities

The roles involved in the binding community are target interface creator, binding initiator, unbinding initiator, target interface, binding factory, binding liaison, binding controller, and channel.

The binding community has three epochs. In one epoch the initiator is bound to the binding factory. In another epoch the targets are members of a binding liaison. In the third epoch, the binding liaison has been terminated.

In addition, the binding community is supported by a channel, and therefore, the binding community may alternate between the epochs with and without an established channel.

7.2 Roles

7.2.1 Binding initiator

The binding initiator is the role of an object that initiates binding establishment between some targets by activating the binding factory.

7.2.2 Unbinding initiator

The unbinding initiator is the role of an object that initiates binding termination.

7.2.3 Binding controller

The binding controller is the role of an object that modifies the existing channel’s properties via the control interface provided by the channel. The binding controller itself may offer an interface for controlling and managing the binding liaison it supports.

7.2.4 Target interface creator

A role of an object that initiates target interface creation. There are two cases, one in which the target interface creator requests a new interface to be created by the infrastructure, and one in which the target interface creator creates a new interface on itself dynamically. In either case, a reference is associated with the interface. This interface reference is past to the binding initiator.

Target objects are those objects that have a need to interact and may assume the role of target interface creator.