INTERNATIONAL STANDARD

ISO/IEC 14752

First edition
2000-04-15

Information technology — Open Distributed Processing — Protocol support for computational interactions

Technologies de l'information — Traitement distribué ouvert — Support du protocole pour les interactions d'ordinateurs

Reference number
ISO/IEC 14752:2000(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 734 10 79
E-mail copyright@iso.ch
Web www.iso.ch

Printed in Switzerland

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.

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 14752 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.931.

Annex A forms a normative part of this International Standard. Annex B is for information only.

INTERNATIONAL STANDARD

ITU-T RECOMMENDATION

INFORMATION TECHNOLOGY – OPEN DISTRIBUTED PROCESSING – PROTOCOL SUPPORT FOR COMPUTATIONAL INTERACTIONS

1 Scope

This Recommendation | International Standard is based on the framework of abstractions and concepts developed in the Reference Model for Open Distributed Processing (ITU-T Rec. X.902 | ISO/IEC 10746-2 and ITU-T Rec. X.903 | ISO/IEC 10746-3).

This Recommendation | International Standard defines how interactions between computational objects in a computational specification of a system relate to protocol support for those interactions in an engineering specification of that system. In particular it:

- defines a General Interworking Framework (GIF);
- within the GIF, defines a set of facilities each comprising a set of functionally-related service primitives as abstract definitions of the interactions of basic engineering objects and channel objects;
- defines the parameters of the service primitives of the GIF;
- defines the permitted sequence of the service primitives by means of state tables;
- specifies, in annexes, the mapping of the GIF service primitives and their parameters to the messages and fields of particular protocols.

As specified in this Recommendation | International Standard, the GIF defines protocol support for a pragmatic subset of the possible computational interactions defined in ITU-T Rec. X.903 | ISO/IEC 10746-3. It is also restricted in the features of the protocol support and the supported transparencies.

The GIF, as specified here, defines:

- support for computational operations, but not for streams;
- support using stub, binder and protocol objects hierarchically, such that any interaction at the interworking reference point of the supporting protocol object supports liaisons of one of those objects or of the basic engineering object, and any interaction to support those liaisons is passed via that interworking reference point; and
- interactions at a single interworking reference point, from the perspective of one side; interceptors are not explicitly considered;

NOTE 1 – It is intended that the GIF could be extended, in a future amendment, to support streams and flows. The present specification is restricted to areas that are technically stable.

The GIF supports at least some forms of:

- access transparency; and
- location transparency.

The GIF as specified here also supports a limited equivalent of relocation transparency. Other transparencies are not addressed in this present specification.

NOTE 2 – It is intended that the GIF could be extended, in future amendments, to support additional transparencies.

The GIF does not explicitly model Quality of Service requirements.

The application of security-related issues to the GIF are not included in the current text and are for further study.

The set of mappings to particular protocols specified in annexes to this Recommendation | International Standard is not exhaustive. The GIF could be mapped to other protocols.

NOTE 3 – In particular, a mapping to the DCOM protocol family would be a candidate for an additional annex.

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 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 currently valid ITU-T Recommendations.

2.1 Identical Recommendation | International Standards

- ITU-T Recommendation X.210 (1993) | ISO/IEC 10731:1994, *Information technology – Open systems interconnection – Basic Reference Model – Conventions for the definition of OSI services*.
- 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.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 bindings*.

2.2 Other Specifications

The edition of [CORBA 2] indicated below was valid at the time of publication of this Recommendation | International Standard. [CORBA 2] is subject to revision, and parties to agreements based on this Recommendation | International Standard are encouraged to investigate the possibility of applying later editions of [CORBA 2] when they become available.

- [CORBA 2] – *The Common Object Request Broker: Architecture and Specification, Revision 2.3, Object Management Group, December 1998 (OMG Doc Number: Formal/98-12-01)*.
- RFC 793, "Transmission Control Protocol", 1981.

3 Definitions

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

3.1 Terms defined in the ODP Reference Model: Foundations

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

a) binding;

c) client object;

c) initiating object;

d) interface;

e) interface signature;

f) name;

g) object;

h) reference point;

i) responding object;

j) server object;

k) viewpoint.

3.2 Terms defined in the ODP Reference Model: Architecture

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

a) announcement;

b) basic engineering object;

c) binder;

d) capsule

e) channel;

f) computational object;

g) computational language;

h) computational viewpoint;

i) engineering viewpoint;

j) interrogation;

k) interceptor;

l) invocation;

m) (computational) operation;

n) operation interface;

o) protocol object;

p) signal;

q) signal interface;

r) stub;

s) termination.

3.3 Definitions for protocol support for computational interactions

This Specification makes use of the following terms.

3.3.1 access facility: A set of service primitives that allow a stub objects to negotiate the abstract and transfer syntax to be used for the operation data to be transmitted over the channel.

3.3.2 association: A relationship (binding) between protocol objects (or between a protocol object and an interceptor) that is established independently of the protocol exchanges that support a particular computational interaction.

3.3.3 association management facility: A set of service primitives which support the management of an association between protocol objects.

3.3.4 basic interworking facility: A set of service primitives which have a direct correspondence with computational signals which model computational operations.

3.3.5 client-side: A node, cluster or capsule, which:

a) contains a basic engineering object corresponding to a computational client object; and

b) contains, or is potentially capable of containing, stub, binder and protocol objects in a channel supporting operations involving the client object.

The term client-side is used prior to the establishment of a channel, during the channel's lifetime and after it has terminated.

3.3.6 deliver primitive: A service primitive for which the protocol object is the responding object of the corresponding communication.

3.3.7 invocation submit: A signal in the implicitly defined signal interface of a client computational object which has the same name and parameters as the invocation of an interrogation or announcement in the original operation interface.

3.3.8 invocation deliver: A signal in the implicitly defined signal interface of a server computational object which has the same name and parameters as the invocation of an interrogation or announcement in the original operation interface.

3.3.9 location facility: A set of service primitives that allow a client-side binder object to ask a server-side if it will accept requests carrying invocations to a particular (computational) server object. The server-side can confirm or reject the proposal or suggest an alternative server-side that is capable of handling requests.

3.3.10 server-side: A node, cluster or capsule, which:

a) contains, or is potentially capable of containing, a basic engineering object that corresponds to a computational server object and stub, binder and protocol objects in a channel supporting operations involving the server object; or

b) contains, or is a potentially capable of containing, a protocol object which (possibly via interactions with other engineering objects) can return a reply identifying another server-side.

The term server-side is used prior to the establishment of a channel, during the channel's lifetime and after it has terminated. It is also used when an appropriate basic engineering object cannot be instantiated following some received message.

3.3.11 service primitive: An abstract definition of an interaction of channel objects that causes protocol exchanges between the protocol objects in the channel.

3.3.12 submit primitive: A service primitive for which the protocol object is the initiating object of the corresponding communication.

3.3.13 termination deliver: A signal in the implicitly defined signal interface of a client computational object which has the same name and parameters as one of the terminations of an interrogation in the original operation interface.

3.3.14 termination submit: A signal in the implicitly defined signal interface of a server computational object which has the same name and parameters as one of the terminations of an interrogation in the original operation interface.

4 Abbreviations

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

GIF	General Interworking Framework
ODP	Open Distributed Processing
ODP IDL	Open Distributed Processing Interface Definition Language
OSI	Open Systems Interconnection
psci	Protocol Support for Computational Interactions
RM-ODP	Open Distributed Processing: Reference Model
TCP	Transmission Control Protocol

5 Conventions

State tables are used to specify the allowed sequence of primitives in each of the facilities of the GIF. Each state machine is initially in the "idle" state. A particular primitive is only permitted if the intersection of the current state and that primitive is non-blank. The entry in the cell defines the state subsequent to the primitive The states are defined by the top row of the state table and have self-explanatory names. If there is an inconsistency between the natural language description of state transitions and the corresponding state table, the natural language description shall take precedence.

6 Overview

6.1 General Interworking Framework

As defined in ITU-T Rec. X.903 | ISO/IEC 10746-3, operations in the computational viewpoint correspond in the engineering viewpoint to interactions between basic engineering objects. Where these engineering interactions in the engineering viewpoint are distributed, a channel connects the basic engineering objects. The establishment and use of the channel involves interactions between the various kinds of engineering object in the channel. In some form or other, these interactions result in observable events occurring at the interworking interface of the protocol objects, according to the rules of one or more protocol specifications. At least some of these observable events will have a direct correspondence to the computational interactions. Other protocol events will be concerned with the management of the engineering binding including the support of required transparencies.

The General Interworking Framework (GIF) defined in this Recommendation | International Standard defines an abstraction of the engineering interactions of the basic engineering and channel objects, including the correspondence between computational operations and the relevant engineering interactions. Mappings of the GIF to particular, implementable, protocols are specified in annexes to this Recommendation | International Standard.

The GIF comprises a set of *facilities* each consisting of a number of *service primitives*. Each facility supports the liaison between one kind of channel object (e.g. stub, binder, protocol) and the service primitives are abstract definitions of the interactions of that kind of object with its peers.

NOTE 1 – If the channel objects were themselves considered as computational objects, a facility would be seen as an interface, and the service primitives as signals.

A mapping of the GIF to a particular protocol will typically specify constraints on the sequencing of the service primitives between different facilities. Such sequencing constraints are not included in the GIF which is intended to support mappings to protocols with a variety of properties and capabilities.

NOTE 2 – For example, mappings to protocols using non-blocking connections, blocking connections and connectionless protocols each give rise to different constraints.

GIF supports both evolution and extensibility. Evolution is obtained by defining, in the abstract form of the service primitives, the architecture used and the messages exchanged for communication. Extensibility is offered by introducing optionality of some of the service primitives and by flexibility in the mapping from service primitives to particular protocols.

NOTE 3 – The GIF itself could also be extended, adding further service primitives or facilities. Such extensions could address additional transparencies.

6.2 Liaisons between channel objects

A distributed engineering binding between basic engineering objects, corresponding to a computational binding is supported by liaisons between the peer channel objects on the client and server sides. Figure 1 shows the relationships between the various objects.

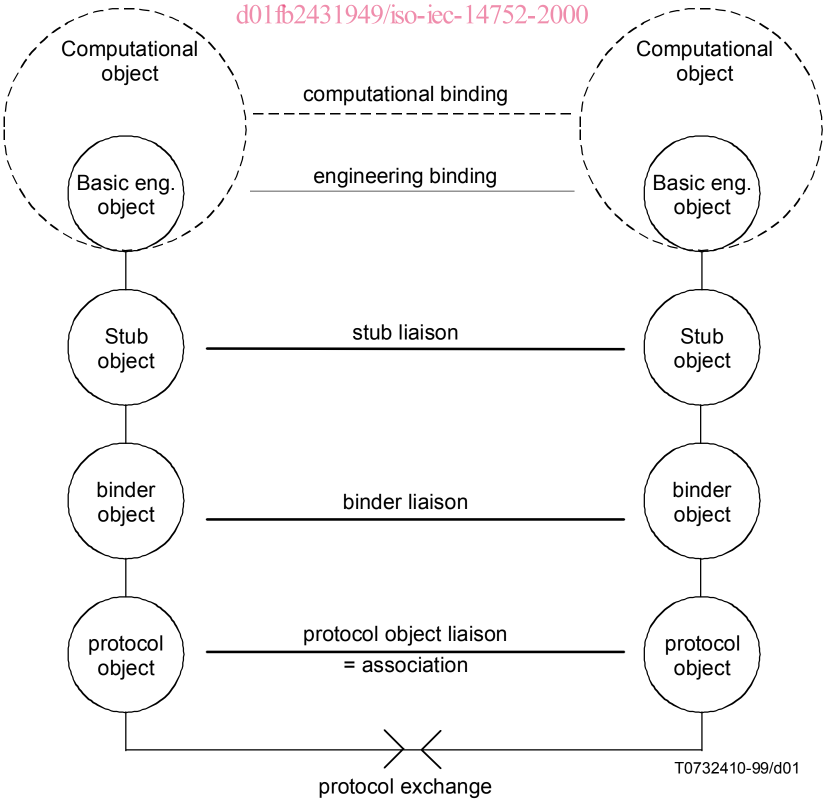

Figure 1 – Relationships of channel objects

In Figure 1, it should be noted that the computational and basic engineering objects are just corresponding views of the same thing. There is no implication that one is contained in the other although other basic engineering objects may correspond to the same computational object (see 10.2 of ITU-T Rec. X.903 | ISO/IEC 10746-3). Similarly, the computational binding and the engineering binding are different views of the same thing.

The hierarchical ordering of the channel objects and of their liaisons in Figure 1 represents the static position when the liaisons are complete and are supporting a particular instance of a computational interaction. The hierarchy should not be taken as implying restrictions on when the liaisons are established or how the channel objects interact during establishment or other liaison management. Establishment of the various liaisons may involve interaction between any of the channel objects in a given node. Other engineering objects may also be involved, in some cases causing protocol exchanges at the interworking reference point. Liaison management, including establishment, may also use other paths than the one being managed. Establishment of any of the liaisons in Figure 1 can take place at an earlier epoch, or can be overlapped with the establishment of the other liaisons. Where recovery of failed bindings is supported, the establishment of a replacement channel may involve the establishment of new liaisons (to the same or different channel objects) or the modification of the surviving liaisons.

For a particular protocol, a single event at the interworking reference point may carry semantics from several of the liaisons. This is termed "piggy-backing".

The stub, binder and protocol liaisons can be established independently of the support of a single instance of a computational interaction. These liaisons can be:

1) established prior to any specific computational interaction;

2) used for interactions between a number of different computational objects; and

3) used for an indefinite number of computational interactions, either consecutively or concurrently.

The stub, binder and protocol liaisons can also be transient, with the duration of the shared context between peer channel objects being limited to the support of a single computational interaction.

NOTE – For example, a server-side object could maintain the state that supports a liaison only from the receipt of a request to the issue of the reply.

6.3 Facilities of the GIF

As stated above, the GIF defines a set of facilities, each comprising a number of service primitives which are functionally related. Each facility is primarily the concern of one kind of engineering object.

The **basic interworking facility** supports the liaison of the basic engineering objects. It includes service primitives which have a direct correspondence with the signals which model the computational operations. This facility is supported by all protocols that support computational operations.

The **access facility** supports access transparency and is primarily the concern of the stub objects. The primitives concern the negotiation of the representation of data to be transmitted via the channel.

The **location facility** supports location transparency and is primarily the concern of the binder objects. It includes service primitives that allow a client-side protocol object to ask a server-side protocol object if it is an appropriate destination for access to a particular basic engineering object, and to allow a server-side protocol object to propose some other server-side. The use of this facility can be combined with the basic interworking facility to allow a server-side, where the client-side "expected" the target basic engineering object to be, to propose an alternative server-side. This allows a limited equivalent of relocation transparency.

The **association management facility** defines service primitives that manage associations - liaisons between protocol objects. An association has an existence independent of a particular instance of a computational interaction (see 8.2).

6.4 Computational operations and signals

The RM-ODP introduces the concept of signal to express the semantics of more abstract interactions. A signal is atomic and localised at a reference point, and represents the initiation or completion at that reference point of some more abstract distributed interaction. Thus an announcement can be delimited by two signals, representing its initiation at a reference point in the sending system and its delivery at a reference point in the receiving system. Similarly, an interrogation can be delimited by two pairs of signals.

A computational operation, as defined in the computational language description in the RM-ODP, is an interaction between a client object and a server object. It can be either an interrogation, consisting of two interactions - an invocation from the client and a replying termination, or be an announcement from the client to the server, with just an invocation. The components of the computational operations - invocation and termination - are characterised as resulting in the conveyance of information from one object to another (see ITU-T Rec. X.903 | ISO/IEC 10746-3).