

INTERNATIONAL STANDARD

ISO/IEC
10021-3

First edition
1990-12-01

Information technology — Text Communication — Message-Oriented Text Interchange Systems (MOTIS) —

Part 3: Abstract Service Definition Conventions

*Technologies de l'information — Communication de texte — Systèmes d'échange
de texte en mode message —*

Partie 3: Conventions relatives à la définition de service abstrait



Reference number
ISO/IEC 10021-3 : 1990 (E)



Documents

Document Sharing Hub

Contents

Foreword v

Introduction vi

Section one – Introduction 1

1 Scope 1

2 Normative references 1

3 Definitions 1

4 Abbreviations 3

5 Conventions 3

5.1 ASN.1 3

5.2 Terms 3

Section two – Abstract Service Definition Conventions 4

6 Overview 4

7 Abstract Models 4

7.1 Abstract Objects 4

7.2 Abstract Ports 5

7.3 Abstract Services 6

7.4 Abstract Refinements 6

8 Abstract Services 7

8.1 Abstract Procedures 7

8.2 Abstract Bind Operations 8

8.3 Abstract Unbind Operations 9

8.4 Abstract Operations 9

8.5 Abstract Errors 10

Section three – Abstract Service Realizations 11

9 Overview 11

10 OSI Realizations 11

10.1 ROS Realizations 11

10.2 Non-ROS Realizations 12

11	Proprietary Realizations	12
11.1	Distributed Realizations.....	13
11.2	Non-distributed Realizations	13

Annexes.....	14
---------------------	-----------

A	Example of Use of Abstract Service Notation.....	14
A.1	Assignment of Object Identifiers	14
A.2	Refinement of Yellow Environment	15
A.3	Definition of Yellow Abstract Service	17
A.4	Refinement of Green Environment.....	18
A.5	Definition of Green Abstract Service	20
A.6	Refinement of Yellow System.....	21
A.7	Realization of Yellow System	23
A.8	Realization of Green System.....	24
B	Reference Definition of Object Identifiers.....	27
C	Reference Definition of Notation	28
D	Differences Between ISO/IEC 10021-3 and CCITT Recommendation X.407.....	31
E	Index.....	32

List of Figures

A.1 The Yellow Environment..... 16

A.2 The Green Environment..... 19

A.3 The Yellow System..... 22

List of Tables

1 Correspondences of Abstract Services and ROS-based Protocols..... 12

2 Equivalent Abstract Service and ROS Macros 12

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 10021-3 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*.

ISO/IEC 10021-3 consists of the following parts, under the general title: *Information technology — Text Communication — Message-Oriented Text Interchange Systems (MOTIS)* —

- *Part 1: System and Service Overview*
- *Part 2: Overall Architecture*
- *Part 3: Abstract Service Definition Conventions*
- *Part 4: Message Transfer System: Abstract Service Definition and Procedures*
- *Part 5: Message Store: Abstract Service Definition*
- *Part 6: Protocol Specifications*
- *Part 7: Interpersonal Messaging System*

Annexes B and C form an integral part of this part of ISO/IEC 10021. Annexes A, D and E are for information only.

Introduction

This part of ISO/IEC 10021 is one of a number of parts of ISO/IEC 10021 (the International Standards for Message-Oriented Text Interchange Systems (MOTIS)). ISO/IEC 10021 provides a comprehensive blueprint for a Message Handling System (MHS) realized by any number of cooperating open systems.

The Message Handling System made possible by these International Standards is a complex distributed information processing task, many of whose components themselves have these characteristics.

This part of ISO/IEC 10021 specifies the conventions for defining the distributed information processing tasks of Message Handling and may also be useful for other applications.

The text of this part of ISO/IEC 10021 is the subject of joint CCITT-ISO agreement. The corresponding CCITT specification is Recommendation X.407.

Information technology - Text Communication - Message-Oriented Text Interchange Systems (MOTIS) - Part 3 : Abstract Service Definition Conventions

Section one - Introduction

1 Scope

This part of ISO/IEC 10021 specifies the conventions used to specify the distributed information processing tasks that arise in Message Handling.

This part of ISO/IEC 10021 is structured as follows. Section one is this introduction. Section two specifies the conventions for defining a distributed information processing task abstractly. Section three gives principles for realizing the communication aspects of such tasks concretely, such as by Open Systems Interconnection (OSI) protocols. Annexes provide important supplemental information.

There are no requirements for conformance to this part of ISO/IEC 10021.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 10021. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this part of ISO/IEC 10021 are encouraged to investigate the possibility of applying the most recent editions of the standards listed below. Members of ISO and IEC maintain registers of currently valid International Standards.

ISO 7498:1984,	<i>Information processing systems - Open Systems Interconnection - Basic Reference Model.</i>
ISO 8649:1988,	<i>Information processing systems - Open Systems Interconnection - Service definition for the Association Control Service Element.</i>
ISO 8824:1990,	<i>Information processing systems - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1).</i>
ISO 8825:1990,	<i>Information processing systems - Open Systems Interconnection - Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1).</i>
ISO/IEC 9072:1989,	<i>Information processing systems - Text communication - Remote operations Part 1: Model, notation and service definition.</i>

3 Definitions

For the purposes of this part of ISO/IEC 10021, the definitions given in annex E and the following definitions apply.

This part of ISO/IEC 10021 is based upon the concepts developed in ISO 7498 and uses the following terms defined in it:

- a) abstract syntax;

- b) Application Layer;
- c) application protocol data unit (APDU);
- d) application protocol;
- e) application service element (ASE);
- f) concrete transfer syntax;
- g) distributed information processing task;
- h) layer service;
- i) layer;
- j) open system;
- k) Open Systems Interconnection (OSI); and
- l) real open system.

This part of ISO/IEC 10021 makes use of the following terms defined in ISO 8824:

- a) Abstract Syntax Notation One (ASN.1);
- b) (data) type;
- c) (data) value;
- d) import;
- e) Integer;
- f) macro;
- g) module;
- h) Object Identifier; and
- i) tag.

This part of ISO/IEC 10021 makes use of the following term defined in ISO 8825:

- a) Basic Encoding Rules.

This part of ISO/IEC 10021 makes use of the following term defined in ISO 8649:

- a) application context (AC).

This part of ISO/IEC 10021 makes use of the following terms defined in ISO/IEC 9072-1:

- a) bind operation;
- b) error;
- c) linked;
- d) operation;
- e) Remote Operation Service (ROS);

- f) Remote Operations; and
- g) unbind operation.

4 Abbreviations

For the purposes of this part of ISO/IEC 10021, the abbreviations given in annex E apply.

5 Conventions

This part of ISO/IEC 10021 uses the descriptive conventions identified below.

5.1 ASN.1

This part of ISO/IEC 10021 uses for the indicated purposes the following ASN.1-based descriptive conventions:

- a) To define the OBJECT, PORT, and REFINE macros, the ASN.1 macro notation of ISO 8824.
- b) To define the ABSTRACT-BIND, -UNBIND, -OPERATION, and -ERROR macros, the BIND, UNBIND, OPERATION, and ERROR macros of ISO/IEC 9072-1.
- c) To specify the abstract syntax of information objects in the example of annex A, ASN.1 itself.
- d) To specify various *abstract models* in the example of annex A, the OBJECT, PORT, and REFINE macros of clause 7.
- e) To specify various *abstract services* in the example of annex A, the ABSTRACT-BIND, -OPERATION, and -ERROR macros of clause 8.

ASN.1 appears both in the body of this part of ISO/IEC 10021 to aid the exposition and again, largely redundantly, in annexes for reference. If differences are found between the two, a specification error is indicated.

Note that ASN.1 tags are implicit throughout the ASN.1 modules in the annexes; the modules are definitive in that respect.

5.2 Terms

Throughout this part of ISO/IEC 10021, terms are rendered in **bold** when defined, in *italic* when referenced prior to their definitions, without emphasis upon all other occasions.

Terms that are proper nouns are capitalized, generic terms are not.

Section two - Abstract Service Definition Conventions

6 Overview

When faced with the job of describing and specifying a complex distributed information processing task, one is wise to begin by specifying the task in abstract, rather than concrete terms. This approach ensures that the task's functional requirements are stated independently of its concrete realization. Such separation is important, among other reasons, because each aspect of the task may admit of *several* concrete realizations. In a Message Transfer System comprising three message transfer agents, e.g., the first and second might interact using OSI communication, the second and third by proprietary means.

This section specifies the conventions for abstractly describing a distributed information processing task both macroscopically and microscopically. The former description is called an *abstract model*, the latter an *abstract service*.

Various formal tools for specifying *abstract models* and *services* are defined in this section. A comprehensive example of their use is given in annex A. The reader may wish to refer to that annex--e.g., to its illustrations--while reading the present section.

This section covers the following topics:

- a) *Abstract models*
- b) *Abstract services*

NOTE - The formal tools mentioned above are neither a formal description language nor a substitute therefor. They are simply ASN.1 notation that supports the informal descriptive conventions this section defines.

7 Abstract Models

A macroscopic description of a distributed information processing task is called an **abstract model (model)** of that task and of the environment in which it is carried out. It is based upon the concepts of *abstract objects*, *ports*, *services*, and *refinements*. (The concept of an *abstract service* is much more fully developed in clause 8.)

7.1 Abstract Objects

An **abstract object (object)** is a functional entity, one of perhaps several which interact with one another. Objects are of different types which determine their function and behaviour. An object of one type, e.g., might represent a system, multiple objects of another type its users. Objects interact by means of *abstract ports*.

An object type is specified by means of the OBJECT macro. Such a specification lists the types of *abstract ports* that provide access to such an object. For each *asymmetric* port type, the specification indicates whether the ports of that type are *consumer* or *supplier* ports.

```

OBJECT MACRO ::=
BEGIN

TYPE NOTATION ::= "PORTS" "{" PortList "}" | empty
VALUE NOTATION ::= value (VALUE OBJECT IDENTIFIER)

PortList      ::= Port "," PortList | Port
Port          ::= value (PORT) PortType

PortType      ::= Symmetric | Asymmetric

Symmetric     ::= empty
Asymmetric    ::= Consumer | Supplier

Consumer      ::= "[C]"
Supplier      ::= "[S]"

END

```

A data value of type OBJECT is an Object Identifier that unambiguously and uniquely identifies the specified object type.

NOTE - The keyword "OBJECT" is reserved in ASN.1. Selection of a suitable replacement for use in the present context is for further study.

7.2 Abstract Ports

An **abstract port** (port) is a point at which an abstract object interacts with another abstract object. Ports are of different types which determine the kinds of interactions they enable. Ports of one type, e.g., might represent the means by which a directory system is accessed, ports of another type the means by which it is administered.

Port types are themselves of the following two varieties:

- a) **symmetric**: All instances of a symmetric port type are identical.
- b) **asymmetric**: Each instance of an asymmetric port type is of one of two kinds, **supplier** and **consumer**.

NOTE - A particular allocation of the terms "supplier" and "consumer" is often intuitive. One might naturally consider a file system, e.g., to present supplier ports to its users and administrators. Strictly speaking, however, the assignment of the two terms is arbitrary.

Two objects can interact with one another by means of a port in one and a port in the other only while those ports are in contact with one another, or **bound**. The actions by means of which this state is initiated and terminated for one or more port pairs are called **binding** and **unbinding**, respectively.

Two ports can be bound only if they **match**. Any two ports of the same, symmetric type match. Two ports of the same, asymmetric type match if and only if one is a supplier, the other a consumer.

A port type is specified by means of the PORT macro. Such a specification identifies the *abstract operations* that represent the interactions possible while two such ports are bound. If none is listed, the *abstract operations* shall be considered unspecified.

```

PORT MACRO ::=
BEGIN

TYPE NOTATION ::= Operations | empty
VALUE NOTATION ::= value (VALUE OBJECT IDENTIFIER)

Operations    ::= Symmetrical | Asymmetrical

Symmetrical   ::= "ABSTRACT" "OPERATIONS" "{" OperationList "}"
Asymmetrical  ::= OneSided | TwoSided

OneSided      ::= Consumer | Supplier
TwoSided      ::= Consumer Supplier | Supplier Consumer

Consumer      ::= "CONSUMER" "INVOKES" "{" OperationList "}"
Supplier      ::= "SUPPLIER" "INVOKES" "{" OperationList "}"

OperationList ::= Operation "," OperationList | Operation
Operation     ::= value (ABSTRACT-OPERATION) |
-- identifying the abstract operation by data value
type
-- identifying the abstract operation by data type

END

```

If the port type is symmetric, both objects offer all listed *abstract operations*. If the port type is asymmetric, the macro distinguishes between the *abstract operations* an object with the consumer port offers and those an object with the supplier port offers.

A data value of type PORT is an Object Identifier that unambiguously and uniquely identifies the specified port type.

7.3 Abstract Services

An **abstract service** is the set of capabilities that one object offers to another by means of one or more of its ports. The former object is called an **abstract service provider (provider)**, the latter an **abstract service user (user)**. Each port in question may be either symmetric or asymmetric and, if the latter, either consumer or supplier.

An abstract service may have any number of users and providers.

Whenever the abstract service ports of a provider are bound to the matching ports of a user, an **abstract association (or association)** is said to exist between the two objects.

An abstract service is specified as indicated in clause 8.

NOTE - An abstract service serves much the same purpose within the Application Layer as does one of the layer services of lower OSI layers.

7.4 Abstract Refinements

An object can be viewed in different ways at different times. On some occasions it is convenient to think of an object as atomic. This is the case, e.g., when describing how an object interacts with other objects external to it, i.e., when specifying its abstract service. On other occasions, it may be more convenient to think of an object as composite, i.e., constructed from other objects. This might be the case, e.g., when describing how an object is realized.

Like any objects, component objects have ports. Some are those visible on the "surface" of the constructed object. Others enable the component objects to interact, thus supporting the provision and use of lesser abstract services among the component objects, which cooperate to provide the overall abstract service of the constructed object.

The functional decomposition of an object into several lesser objects is called the **abstract refinement (refinement)** of that object.