

TECHNICAL REPORT

ISO/IEC TR 10023

First edition
1992-07-01

Information technology — Telecommunications and information exchange between systems — Formal description of ISO 8072 in LOTOS

*Technologies de l'information — Télécommunications et échange
d'informations entre systèmes — Description formelle de l'ISO 8072 en
LOTOS*



Reference number
ISO/IEC TR 10023:1992(E)

AVA Documents
Document Sharing Hub

Contents

Foreword iii

Introduction iv

1 Scope 1

2 Normative references 1

3 Definitions 1

4 Symbols and abbreviations 1

5 Conventions 1

6 Requirements 2

7 Introduction of the formal description 2

8 Interface data types 2

9 Global Constraints 17

10 Provision of a transport connection 18

11 Local constraints for a TCEP 19

12 End-to-end constraints for a single TC 21

13 Identification of transport connections 24

14 Acceptance of transport connections 25

15 Flow control by backpressure 26

16 Connectionless-mode transmission 26

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) together 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.

The main task of technical committees is to prepare International Standards, but in exceptional circumstances a technical committee may propose the publication of a Technical Report of one of the following types:

- type 1, when the required support cannot be obtained for the publication of an International Standard, despite repeated efforts;
- type 2, when the subject is still under technical development or where for any other reason there is the future but not immediate possibility of an agreement on an International Standard;
- type 3, when a technical committee has collected data of a different kind from that which is normally published as an International Standard ('state of the art', for example).

Technical reports of types 1 and 2 are subject to review within three years of publication, to decide whether they can be transformed into International Standards. Technical reports of type 3 do not necessarily have to be reviewed until the data they provide are considered to be no longer valid or useful.

ISO/IEC TR 10023, which is a technical report of type 2, was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*.

Introduction

In view of the complexity and widespread use of Open Systems Interconnection standards it is imperative to have precise and unambiguous definitions of these standards. Formal Description Techniques form an effective approach to providing such definitions. The use of Formal Description Techniques in this area, however, is relatively new, and their application on a wide scale requires an appropriate habituation period. Formal descriptions should be introduced gradually in standards if initially the number of National Bodies that are able to contribute to their development is too small, thus allowing time to gain experience and to develop education material.

Within ISO/IEC JTC1/SC6 an ad-hoc group for the formal description in LOTOS of the Transport Standards, viz. the Transport Service ISO 8072 and the Transport Protocol ISO 8073, was established in October 1985. The Formal Description Technique LOTOS, defined in ISO 8807, was still under development at that time within ISO/ TC 97/SC21. Following the position of a majority of National Bodies, SC6 decided in January 1988 to progress both formal descriptions as Type 2 Technical Reports. The main reason for not incorporating them into the standards was that National Bodies expressed their current lack of expertise on the subject of Formal Description Techniques. It seemed therefore appropriate to let a period of time pass, during which the formal descriptions can be read and compared with the standards, and after which the status of the formal descriptions can be re-evaluated.

The purpose of this Technical Report is to provide a complete, consistent and unambiguous description of ISO 8072. It forms therefore a companion document to ISO 8072. This Technical Report has successfully passed correctness and completeness tests with respect to syntax and static semantics, according to the requirements of clause 3 of ISO 8807. Tools are available that provide interactive simulation facilities, so that confidence can be gained about the correctness of the description with respect to the dynamic semantics (e.g. deadlock-freedom), and about its correctness and completeness with respect to representation of intended behaviour requirements. These tools are still at the stage of prototypes, however.

Significant contributions for the application of LOTOS in this work have been provided by Universities and research laboratories.

Information technology – Telecommunications and information exchange between systems – Formal description of ISO 8072 in LOTOS

1 Scope

This Technical Report describes the OSI Transport service, defined in ISO 8072, using the formal description technique LOTOS, which is defined in ISO 8807.

NOTE – The formal definitions of data types and processes presented in this Technical Report are applicable for use in formal descriptions, using LOTOS, of the OSI Transport and Session protocols.

2 Normative references

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

ISO 7498:1984, *Information Processing Systems – Open Systems Interconnection – Basic Reference Model*.

ISO 8072:1986, *Information Processing Systems – Open Systems Interconnection – Transport service definition*.

ISO 8072:1986/Addendum 1:1986, *Information Processing Systems - Open Systems Interconnection - Transport service definition - Addendum 1 - Connectionless-mode transmission*.

ISO 8072:1986/Technical Corrigendum 1:1991, *Information Processing Systems - Open Systems Interconnection - Transport service definition - Technical Corrigendum 1*.

ISO/TR 8509:1987, *Information Processing Systems – Open Systems Interconnection – Service conventions*

ISO 8807:1989, *Information Processing Systems – Open Systems Interconnection – LOTOS – A formal description technique based on the temporal ordering of observational behaviour*.

ISO/IEC TR 10024:1992, *Information technology – Telecommunications and information exchange between systems – Formal description of ISO 8073 (Classes 0,1,2,3) in LOTOS*

3 Definitions

For the purposes of this Technical Report the definitions given in ISO 8072 apply.

4 Symbols and abbreviations

This Technical Report uses the symbols defined in clause 6 (formal syntax) and Annex A (data type library) of ISO 8807.

The abbreviations contained in clause 4 of ISO 8072 are employed in this Technical Report. Usage of other symbols and abbreviations is explained upon their first occurrence.

5 Conventions

Informal explanations precede the formal definitions to which they refer, separated from them by a line identical to the first line below. Separation of formal definitions from subsequent informal explanations is indicated by a line identical to the second line below.

(*)

NOTE – This convention is consistent with the rules on delimitation of comments defined for LOTOS in ISO 8807. The formal text is set in Courier and the LOTOS keywords and operators are in boldface. Identifiers from the formal text are set in italics when used in the non-formal text.

The conventions defined in ISO/TR 8509 are adopted, but together with the following one: the term “Request” refers both to request and to response service primitives, and the term “Indication” refers both to indication and to confirm service primitives.

6 Requirements

This Technical Report complies with the requirements defined in clause 3 of ISO 8807 (see annex C for further details).

No conformity requirements are defined by this Technical Report.

7 Introduction of the formal description

The whole service boundary is formally represented by the single gate *t*. The event structure at *t* consists of a triple of values, resp. of sorts *TAddress*, *TCEI*, *TSP* (see clause 8). The first value identifies the TSAP where the interaction occurs. The second value identifies the TCEP, within that TSAP, where the interaction occurs. The third value is the Transport Service Primitive (TSP) executed in the interaction. Full account is taken of the multiplicity aspects of the service.

The behaviour of a never terminating service provider is described. The specification is not parameterized.

A constraint-oriented specification style is adopted since it is the most appropriate with respect to the definitive character of a service standard. The description is aimed at describing modalities of behaviour of the service provider in terms of the history of executed TSPs only, i.e. with no reference to nor suggestion of any internal structure of the provider itself.

The following separate constraints are found in the first decomposition of the specified behaviour (see clause 9):

- a) The service provider may offer a multiplicity of possibly concurrent connections (represented by process *TCConnections*, see clause 9), together with
- b) a means to distinguish among concurrent connections (represented by process *TCIdentification*, see clause 13), but under
- c) the possibility of internal non-determinacy in accepting to establish any additional connection, if at least one connection is being supported, (represented by process *TCAcceptance*, see clause 14), and
- d) the possibility of internal non-determinacy in accepting to transfer data (represented by process *TBackpressure*, see clause 15); by means of this, the effect of the flow control exerted by the receiving user is transferred back to the sending user.

The specification of the dynamic behaviour is preceded by that of the interface data types (see clause 8). Such definitions are to be shared with the formal descriptions that may interwork with the present one, viz. those of the transport and session protocols.

The order of presentation of the further definitions follows a desire of adherence to ISO 8072, which mainly addresses the provision of a single TC. The description of the requirements that are local to the provision of a single TC (presented in clauses 10, 11 and 12) precedes that of the global requirements respectively mentioned in b), c) and d) above.

The data types referred to in the *library* construct are imported from the LOTOS library of data types.

```
specification TransportService[t] : noexit
```

```
library Set, Element, OctetString, NatRepresentations, NaturalNumber, Boolean,  
        FBoolean, DecNatRepr  
endlib
```

```
(*
```

8 Interface data types

8.1 General

According to the representation of interactions at the TS boundary (see clause 7), the interface data types consist of three main definitions that respectively construct the sorts *TAddress* (see 8.2), *TCEI* (see 8.3), and *TSP* (see 8.4). The quality of service TSP parameter is defined in 8.5, and other parameters in 8.6. Auxiliary definitions are presented in 8.7.

8.2 Transport address

No structure of transport addresses is defined by ISO 8072. The following definition uses the definition of *GeneralIdentifier* (see 8.7) and allows to represent an infinite number of transport addresses.

```
type TransportAddress  
is GeneralIdentifier renamedby  
sortnames  
    TAddress for Identifier  
opnames  
    SomeTAddress for SomeIdentifier
```

```

    AnotherTAddress for AnotherIdentifier
endtype (* TransportAddress *)

```

8.3 Transport connection endpoint identifier

No structure of transport connection endpoint identifiers is defined by ISO 8072. The first definition below allows to represent an infinite number of them. The second definition presents TC endpoint identifiers that are global to the whole TS boundary: each of them is a pair *TAddress* x *TCEI* (see 8.7 for the generic definition *Pair* and *GeneralIdentifier*).

```

type TCEndpointIdentifier
is GeneralIdentifier renamedby
sortnames
    TCEI for Identifier
opnnames
    SomeTCEI for SomeIdentifier
    AnotherTCEI for AnotherIdentifier
endtype (* TCEndpointIdentifier *)

type TCEIdentification
is Pair actualizedby TransportAddress, TCEndpointIdentifier using
sortnames
    TAddress for Element
    TCEI for Element2
    Bool for FBool
    TId for Pair
opnnames
    TId for Pair
    TA for First
    TCEI for Second
endtype (* TCEIdentification *)

```

8.4 Transport service primitive

8.4.1 General

The TSP data type is presented starting with the basic construction of values of sort *TSP* (see below). This construction is a direct formulation of the definition given in table 3 of ISO 8072. The functions that yield TSP values are referred to as "TSP constructors". This definition imports the definitions that relate to TSP parameters (see 8.6).

NOTE – In some TSPs the *Userdata* parameter is an *OctetString* that has a fixed bound, as defined in ISO 8072. For technical convenience, this requirement is formally represented by a process constraint (See 11.1 and 11.4) instead of a type constraint.

In 8.4.2 a classification of TSPs is defined, that enables both to enrich the basic construction with further functions in a simple way (see 8.4.3) and to conservatively extend the data type in the formal description of the transport protocol.

```

type BasicTSP
is TransportAddress, TEXOption, TSQuality, OctetString, TDISReason, TsClQuality
sorts
    TSP
opns
    TCONreq, TCONind      : TAddress, TAddress, TEXOption, TQOS, OctetString -> TSP
    TCONresp, TCONconf    : TAddress, TEXOption, TQOS, OctetString -> TSP
    TDTreq, TDTind        : OctetString -> TSP
    TEXreq, TEXind        : OctetString -> TSP
    TDISreq               : OctetString -> TSP
    TDISind               : TDISReason, OctetString -> TSP
    TUDTreq, TUDTind      : TAddress, TAddress, ClQOS, OctetString -> TSP
endtype (* BasicTSP *)

```

8.4.2 Transport service primitive classification

8.4.2.1 Basic classification

A basic classification of TSPs is defined by means of *TSPSubsort*, which consists of a set of constants, each denoting a TSP name in correspondence with table 3 of ISO 8072.

The type *TSPBasicClassifiers* is a functional enrichment of the basic construction in 8.4.1, where

- a) the *Subsort* function yields the TSP name;
- b) boolean functions on TSPs, termed "TSP (subsort) recognizers", are defined according to the basic classification introduced by *TSPSubsort*.

NOTE – The auxiliary function *h* that maps TSP names to natural numbers is defined in order to simplify the specification of the boolean operations of equality on TSP names (as well as on TSPs, in 8.4.3.3). The definitions of *IsRequest*, *IsIndication* (on TSP names) and *IsTReq*, *IsTInd* (on TSPs) reflect the convention introduced in clause 5.

```

type TSPSubsort
is NaturalNumber
sorts
  TSPSubsort
opns
  TCONNECTrequest, TCONNECTindication, TCONNECTresponse, TCONNECTconfirm,
  TDATArequest, TDATAindication, TEXDATArequest, TEXDATAindication,
  TDISCONNrequest, TDISCONNindication, TUDATArequest, TUDATAindication : -> TSPSubsort
  h
    : TSPSubsort -> Nat
  Even, Odd
    : Nat -> Bool
  IsRequest, IsIndication
    : TSPSubsort -> Bool
  _eq_, _ne_
    : TSPSubsort, TSPSubsort -> Bool
eqns
forall
  s, s1 : TSPSubsort, n : Nat
ofsort Nat
  h(TCONNECTrequest) = 0;
  h(TCONNECTindication) = Succ(h(TCONNECTrequest));
  h(TCONNECTresponse) = Succ(h(TCONNECTindication));
  h(TCONNECTconfirm) = Succ(h(TCONNECTresponse));
  h(TDATArequest) = Succ(h(TCONNECTconfirm));
  h(TDATAindication) = Succ(h(TDATArequest));
  h(TEXDATArequest) = Succ(h(TDATAindication));
  h(TEXDATAindication) = Succ(h(TEXDATArequest));
  h(TDISCONNrequest) = Succ(h(TEXDATAindication));
  h(TDISCONNindication) = Succ(h(TDISCONNrequest));
ofsort Bool
  Even(0) = true;
  Even(Succ(0)) = false;
  Even(Succ(Succ(n))) = Even(n);
  Odd(n) = not(Even(n));
  IsRequest(s) = Even(h(s));
  IsIndication(s) = Odd(h(s));
  s eq s1 = h(s) eq h(s1);
  s ne s1 = not(s eq s1);
endtype (* TSPSubsort *)
type TSPBasicClassifiers
is BasicTSP, TSPSubsort
opns
  Subsort
    : TSP -> TSPSubsort
  IsTCON, IsTCON1, IsTCON2, IsTDT, IsTEX, IsTDIS, IsTCONreq, IsTCONind,
  IsTCONresp, IsTCONconf, IsTDTreq, IsTDTind, IsTEXreq, IsTEXind, IsTDISreq,
  IsTDISind, IsTUDT, IsTUDTreq, IsTUDTind, IsTReq, IsTInd : TSP -> Bool
eqns
forall
  a, a1, a2 : TAddress, x : TEXOption, q : TQOS, d : OctetString, r : TDISReason,
  t : TSP, clq : CLQOS
ofsort TSPSubsort
  Subsort(TCONreq(a1, a2, x, q, d)) = TCONNECTrequest;
  Subsort(TCONind(a1, a2, x, q, d)) = TCONNECTindication;
  Subsort(TCONresp(a, x, q, d)) = TCONNECTresponse;
  Subsort(TCONconf(a, x, q, d)) = TCONNECTconfirm;
  Subsort(TDTreq(d)) = TDATArequest;
  Subsort(TDTind(d)) = TDATAindication;
  Subsort(TEXreq(d)) = TEXDATArequest;
  Subsort(TEXind(d)) = TEXDATAindication;
  Subsort(TDISreq(d)) = TDISCONNrequest;

```

```

Subsort(TDISind(r, d)) = TDISCONNindication;
Subsort(TUDTreq(a1, a2, clq, d)) = TUDATArequest;
Subsort(TUDTind(a1, a2, clq, d)) = TUDATAindication;
ofsort Bool
  IsTCON1(t) = IsTCON1(t) or IsTCON2(t);
  IsTCON1(t) = IsTCONreq(t) or IsTCONind(t);
  IsTCON2(t) = IsTCONresp(t) or IsTCONconf(t);
  IsTDT(t) = IsTDTreq(t) or IsTDTind(t);
  IsTEX(t) = IsTEXreq(t) or IsTEXind(t);
  IsTDIS(t) = IsTDISreq(t) or IsTDISind(t);
  IsTCONreq(t) = Subsort(t) eq TCONNECTrequest;
  IsTCONind(t) = Subsort(t) eq TCONNECTindication;
  IsTCONresp(t) = Subsort(t) eq TCONNECTresponse;
  IsTCONconf(t) = Subsort(t) eq TCONNECTconfirm;
  IsTDTreq(t) = Subsort(t) eq TDATArequest;
  IsTDTind(t) = Subsort(t) eq TDATAindication;
  IsTEXreq(t) = Subsort(t) eq TEXDATArequest;
  IsTEXind(t) = Subsort(t) eq TEXDATAindication;
  IsTDISreq(t) = Subsort(t) eq TDISCONNrequest;
  IsTDISind(t) = Subsort(t) eq TDISCONNindication;
  IsTUDT(t) = IsTUDTreq(t) or IsTUDTind(t);
  IsTUDTreq(t) = Subsort(t) eq TUDATArequest;
  IsTUDTind(t) = Subsort(t) eq TUDATAindication;
  IsTReq(t) = IsRequest(Subsort(t));
  IsTInd(t) = IsIndication(Subsort(t));
endtype (* TSPBasicClassifiers *)
(*

```

8.4.2.2 Auxiliary classification

TDATAAtomSubsort introduces a further classification of the atomic constituents of T-DATA primitives, viz. user data octets and TSU terminators. However, in this specification these constituents are not specified.

NOTE – The only reason for presenting *TDATAAtomSubsort* in this description is that in this way the *TransportServicePrimitive* data type admits a conservative extension, where a more refined, non-atomic representation of T-DATA primitives can be introduced. Such an extension is needed in the formal description of the transport protocol for a correct formulation of the requirements relating to segmenting and flow control.

The boolean function *Terminates*, defined in *TSPClassifiers*, relates the atomic execution of T-DATA primitives, which characterizes the service definition, to the non-atomic one found in the formal description of the protocol.

```

*)
type TDATAAtomSubsort
is FourTuple renamedby
sortnames
  TDTASubsort for Tuple
opnames
  TDATAOCTrequest for TheOne
  TDATAOCTindication for TheOther
  TEOTrequest for TheThird
  TEOTindication for TheFourth
endtype (* TDATAAtomSubsort *)
type TSPClassifiers
is TSPBasicClassifiers, TDATAAtomSubsort
opns
  _Terminates_ : TDTASubsort, TSP -> Bool
eqns
forall
  d : OctetString, s : TDTASubsort, t : TSP
ofsort Bool
  TEOTrequest Terminates TDTreq(d) = true;
  TEOTindication Terminates TDTind(d) = true;
  TEOTrequest Terminates TDTind(d) = false;
  TEOTindication Terminates TDTreq(d) = false;
  TDATAOCTrequest Terminates t = false;
  TDATAOCTindication Terminates t = false;
  not(IsTDT(t)) => s Terminates t = false;
endtype (* TSPClassifiers *)
(*

```

8.4.3 Functions on transport service primitives

8.4.3.1 General

In 8.4.3.2 the construction presented in 8.4.2 is enriched with functions that allow to determine the value of individual parameters of TSPs. In 8.4.3.3 boolean equality is added to this construction. In 8.4.3.4 further functional enrichments are presented, that are useful to represent negotiation (see 11.3.2) and provider's non-determinacy (see 12.3.3).

8.4.3.2 Transport service primitive parameter selectors

Boolean functions are defined, to compare or select values of individual parameters of TSPs. Reason for this indirect representation is the completeness of the equational definition. The only exception is found in the direct representation of the user data parameter by the function *UserData*, as this parameter can be defined on all TSPs.

```

type TSPParameterSelectors
is TSPClassifiers
opns
  _IsCalledOf_, _IsCallingOf_, _IsRespondingOf_ : TAddress, TSP -> Bool
  _IsTEXOptionOf_ : TEXOption, TSP -> Bool
  _IsTQOSof_ : TQOS, TSP -> Bool
  _IsClQOSof_ : ClQOS, TSP -> Bool
  _IsReasonOf_ : TDISReason, TSP -> Bool
  UserData : TSP -> OctetString
eqns
forall
  a, a1, a2 : TAddress, x, x1 : TEXOption, q, q1 : TQOS, d : OctetString, r,
  r1 : TDISReason, t : TSP, clq, clq1 : ClQOS
ofsort Bool
  a IsCalledOf TCONreq(a1, a2, x, q, d) = a eq a1;
  a IsCalledOf TCONind(a1, a2, x, q, d) = a eq a1;
  a IsCalledOf TUDTreq(a1, a2, clq, d) = a eq a1;
  a IsCalledOf TUDTind(a1, a2, clq, d) = a eq a1;
  not(IsTCON1(t)) => a IsCalledOf t = false;
  a IsCallingOf TCONreq(a1, a2, x, q, d) = a eq a2;
  a IsCallingOf TCONind(a1, a2, x, q, d) = a eq a2;
  a IsCallingOf TUDTreq(a1, a2, clq, d) = a eq a2;
  a IsCallingOf TUDTind(a1, a2, clq, d) = a eq a2;
  not(IsTCON1(t)) => a IsCallingOf t = false;
  a IsRespondingOf TCONresp(a1, x, q, d) = a eq a1;
  a IsRespondingOf TCONconf(a1, x, q, d) = a eq a1;
  not(IsTCON2(t)) => a IsRespondingOf t = false;
  x IsTEXOptionOf TCONreq(a1, a2, x1, q, d) = x eq x1;
  x IsTEXOptionOf TCONind(a1, a2, x1, q, d) = x eq x1;
  x IsTEXOptionOf TCONresp(a, x1, q, d) = x eq x1;
  x IsTEXOptionOf TCONconf(a, x1, q, d) = x eq x1;
  not(IsTCON(t)) => x IsTEXOptionOf t = false;
  q IsTQOSof TCONreq(a1, a2, x, q1, d) = q eq q1;
  q IsTQOSof TCONind(a1, a2, x, q1, d) = q eq q1;
  q IsTQOSof TCONresp(a, x, q1, d) = q eq q1;
  q IsTQOSof TCONconf(a, x, q1, d) = q eq q1;
  not(IsTCON(t)) => q IsTQOSof t = false;
  r IsReasonOf TDISind(r1, d) = r eq r1;
  not(IsTDISind(t)) => r IsReasonOf t = false;
  clq IsClQOSof TUDTreq(a1, a2, clq1, d) = clq eq clq1;
  clq IsClQOSof TUDTind(a1, a2, clq1, d) = clq eq clq1;
ofsort OctetString
  UserData(TCONreq(a1, a2, x, q, d)) = d;
  UserData(TCONind(a1, a2, x, q, d)) = d;
  UserData(TCONresp(a, x, q, d)) = d;
  UserData(TCONconf(a, x, q, d)) = d;
  UserData(TDTreq(d)) = d;
  UserData(TDTind(d)) = d;
  UserData(TEXreq(d)) = d;
  UserData(TEXind(d)) = d;
  UserData(TDISreq(d)) = d;
  UserData(TDISind(r, d)) = d;
  UserData(TUDTreq(a1, a2, clq, d)) = d;

```