ETSI
TECHNICAL
REPORT

ETR 130

April 1994

Source: ETSI TC-MTS, EWOS
EWOS EGCT

Reference: DTR/MTS-02003-1
EWOS ETG 028

ICS: 33.080

Key words: Interworking, taxonomy, testing

Methods for Testing and Specification (MTS); Interoperability and conformance testing A classification scheme

ETSI

European Telecommunications Standards Institute

ETSI Secretariat

Postal address: F-06921 Sophia Antipolis CEDEX - FRANCE
Office address: 650 Route des Lucioles - Sophia Antipolis - Valbonne - FRANCE
X.400: c=fr, a=atlas, p=etsi, s=secretariat - **Internet:** secretariat@etsi.fr

Tel.: +33 92 94 42 00 - Fax: +33 93 65 47 16

Contents

Blank page

Foreword

This ETSI Technical Report (ETR) has been produced by the Methods for Testing and Specification (MTS) Technical Committee of the European Telecommunications Standards Institute (ETSI). The work has been carried out jointly by the European Telecommunications Standards Institute (ETSI) and the European Workshop for Open Systems (EWOS).

ETSI TC MTS and the EWOS Expert Group on Conformance Testing (EGCT) have agreed to issue a common text. The ETG version of this ETR is known as ETG 028 and was adopted by the EWOS TA 21 (18-19 May 1993).

ETRs are informative documents resulting from ETSI studies which are not appropriate for European Telecommunication Standard (ETS) or Interim European Telecommunication Standard (I-ETS) status. An ETR may be used to publish material which is either of an informative nature, relating to the use or the application of ETSs or I-ETSs, or which is immature and not yet suitable for formal adoption as an ETS or an I-ETS.

Introduction

This ETR complements EWOS ETG 029, the Interoperability Vocabulary. Terms which are defined in the Interoperability Vocabulary ETG are identified as bold-italic text. This ETR provides a description of the terms used and fits them into an overall reference model, and it describes different classifications and qualifications for Interoperability.

Blank page

1 Scope

This ETSI Technical Report (ETR) defines a classification of different types of Interoperability. Four different classes are defined, covering the protocol aspects, the service aspects, the application aspects and the user aspects. Qualifications on the definitions are shown in Clause 6. The relationship to corresponding standards is described, and an example of the application of the concepts to an instance of Interoperability is shown for clarification.

NOTE: The scope of the work generally refers to the functional capability of implementations to interoperate. However, Interoperability will not be achieved if the performance attributes, or robustness attributes of a Distributed System are such that a common task cannot be carried out in practice and, therefore, these aspects are, to an extent, within the scope of this ETR.

2 References

For the purposes of this ETR, the following references apply:

[1] ENV 41104: "Information systems interconnection - Packet switched data networks: Permanent access".

[2] ENV 41201: "Private message handling system - User agent and message transfer agent - Private management domain to private management domain".

[3] ITU-T Recommendation X.25: "Interface between data terminal equipment (DCE) and data circuit-terminating equipment (DCE) for terminals operating in the packet mode and connected to public data networks by dedicated circuit".

[4] CCITT Recommendation X.400: "Message handling systems: System and service overview".

3 Abbreviations

For the purposes of this ETR, the following abbreviations apply:

API	Application Programming Interface
EDI	Electronic Document Handling
FTAM	File Transfer Access and Management
IFS	Interoperability Functional Specification
ISP	International Standards Profile
MTA	Message Tranfer Agent
ODA	Office Document Architecture
ODIF	Open Document Interchange Format
PDU	Protocol Data Unit
PICS	Protocol Implementation Conformance Statement

4 Model for interoperability

In order to undertake the classification, and provide a vehicle to illustrate the vocabulary, a model is used as illustrated in figure 1.

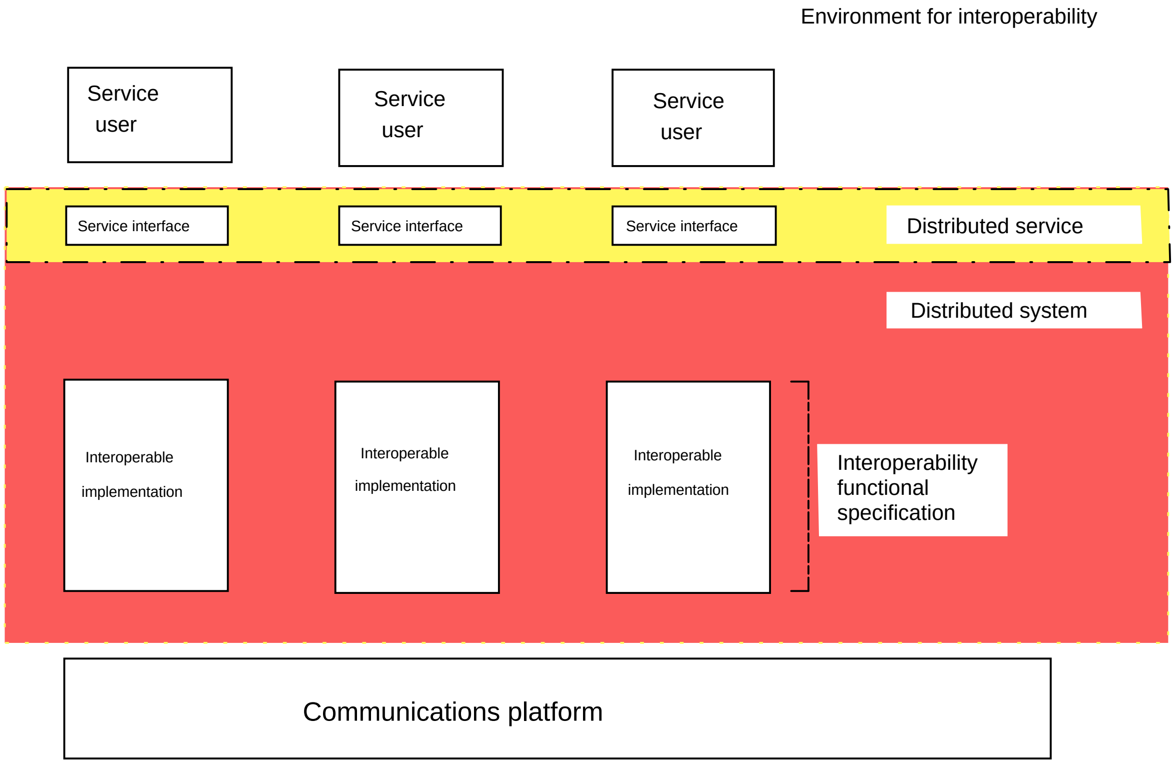

Figure 1: The domain of interoperability

The model given in figure 1 shows a set of ***Interoperable Implementations***, which are each implemented in accordance with the ***Interoperability Functional Specification*** (IFS). Collectively these are the ***Distributed System***, and offer the ***Distributed Service*** through their ***Service Interfaces***. The ***Distributed System*** exists within the ***Environment for Interoperability***.

The ***Interoperable Implementations*** make use of the ***Communications Platform*** to interchange Protocol Data Units (PDUs), and the collective term for all of the components is the ***Domain of Interoperability***.

The ***Service User*** may be an application. For example, if the ***Interoperable Implementations*** were implementations of the OSI layers 1 to 4 offering a Transport Service, then the ***Communications Platform*** is the physical network, and the ***Service User*** would be an application incorporating the Session Layer.

If the ***Interoperable Implementations*** were, for example, File Transfer Access and Management (FTAM) implementations, then the ***Service User*** may be human or application.

5 Classification

This Clause defines the Interoperability Classifications. This is illustrated in figure 2.

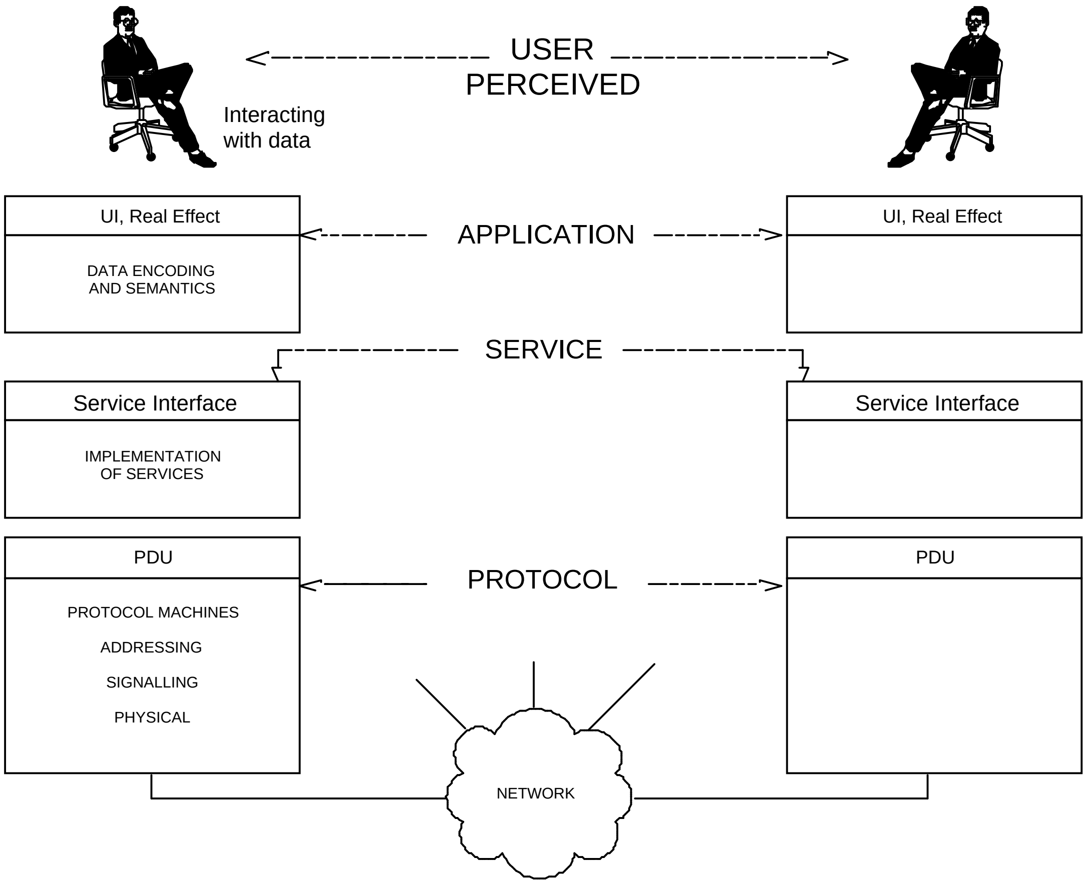

Figure 2: Illustration of interoperability classification

The interoperability classifications are summarised as follows:

Protocol Interoperability: the ability of a ***Distributed System*** to interchange PDUs via the ***Communications Platform***;

Service Interoperability: the ability of a ***Distributed System*** to support a subset of the ***Distributed Service***;

Application Interoperability: the ability of a ***Distributed System*** to provide a consistent implementation of the syntax and semantics of the data which is interchanged;

User Perceived Interoperability: the ability of the ***Service User*** (Human, Application, Machine) to exchange information via the ***Distributed System.***

5.1 Protocol interoperability

Protocol Interoperability describes the ability of ***Interoperable Implementations*** to interoperate successfully over the ***Communications Platform***, to successfully interchange Protocol Data Units (PDUs).

This implies that the following aspects, where applicable, interoperate successfully:

- physical connectors;
- electrical signalling;
- addressing and routeing;
- protocol machines;
- negotiation;
- timing.

This states nothing about the implementation of services at the boundary of the ***Interoperable Implementations***, only that the communications between them is satisfactory. The classification is equally applicable to end systems and intermediate systems.

The systems may be physically connected via the ***Communications Platform*** in a variety of ways, for example using:

- a bus;
- a local area network;
- a wide area network;
- a complex network including relays, bridges, etc.

Protocol Interoperability is a necessary condition for other classifications of ***Interoperability***. It is never sufficient to provide any useful function, as the definition excludes the provision of a ***Service***.

5.2 Service interoperability

A ***Distributed System*** may be classed as ***Service Interoperable*** if it is ***Protocol Interoperable*** and a ***Service Interface*** is implemented which offers the semantics of the services supported to an application or a human user. The set of services provided to the user of the ***Service*** needs to be compatible.

Services may be provided through, for example:

- a standardised Application Programming Interface (API);
- a proprietary API;
- a direct user interface;
- a real effect;
- a filestore.

Compatible services means that any ***Service*** which is invoked on one ***Interoperable Implementation*** results in a notification to an application on the same, or another ***Interoperable Implementation***, in accordance with the ***Interoperability Functional Specification***. This means that:

- the corresponding services need to be implemented;

- services need to be implemented in accordance with the ***same Interoperability Functional Specification***, for example the same parameter variations and length constraints need to be implemented.

Services which do not fulfil the above criteria may lead to a qualified ***Interoperability***, as shown in Clause 6.

5.3 Application interoperability

Application Interoperability is achieved when applications can communicate using ***Service Interoperable*** systems, and have a common understanding of the syntax and semantics of the data which is exchanged through the services. ***Application Interoperability*** requires:

- a common data encoding and decoding scheme;

- defined semantics for the particular application context.

Examples of ***Application Interoperability*** are the successful interworking of Electronic Document Handling (EDI) applications, reprocessing of structured word processed documents, reprocessing of graphics, etc.

5.4 User Perceived Interoperability

5.4.1 Human users

User Perceived Interoperability is achieved when ***Interoperable Implementations*** interact with human users in a way which results in a transfer of information between the humans with no loss occurring in the transfer.

Interaction with human users may be achieved at each ***Interoperable Implementation*** through display devices, keyboards and other input devices, and printers.

User Perceived Interoperability requires a common approach to:

- the use of character sets;

- the use of graphics;

- the use of fonts;

- the use of colour;

- display device characteristics (attributes, size, etc.).

5.4.2 Other users

Other users are devices such as machine tools, robots, indicator lights, acoustic signals, actuators, etc., which may be activated by the application. ***Interoperability*** is achieved if the ***Real Effect*** taking place (as observed by a human user) is that intended by the communicating applications. ***Application Interoperability*** is sufficient to pass the message "move the robot arm 90 degrees in a vertical plane" from one application to the other, but ***User Perceived Interoperability*** is achieved when the instruction has been observed to have been carried out.

5.4.3 Quality of service issues

Quality of service issues such as performance and useability are not covered, as there are in general no objective criteria for their assessment. There may be requirements on these aspects in the ***Interoperability Functional Specification***, in which case they will become criteria for ***Interoperability***.

6 Qualifications

Two qualifications to each classification of ***Interoperability*** are defined. In each case the qualifications relate to the ***Interoperability Functional Specification***. The qualifications describe the intersection of the ***Interoperability Functional Specification*** which is common to the two ***Interoperable Implementations***. This intersection is achieved through:

a) support of the mandatory features of the ***Interoperability Functional Specification***;

b) the extent of the static support of the optional features of the ***Interoperability Functional Specification;***

c) the ability to dynamically negotiate to an agreed set of features.

This is illustrated in figure 3.

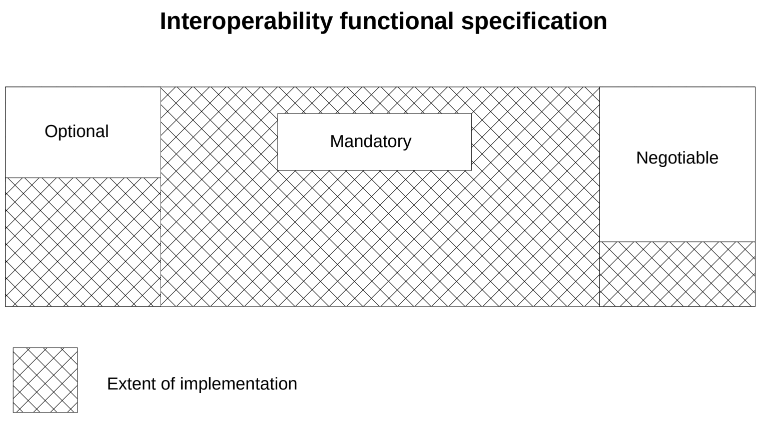

Figure 3: Composition of the interoperability functional specification

6.1 Interoperability in the full domain of the interoperability functional specification

For peer to peer systems, this means that the protocol, services, semantics and rendition or ***Real Effect*** are implemented in each end system in accordance with the ***Interoperability Functional Specification***. Therefore, the end systems are symmetrical, in that any feature available on one is also available on the other.

For client server systems, this means that although the systems are not symmetric, within the designated client/server roles, the same set of protocol options, services, semantics and real effects are supported in accordance with the ***Interoperability Functional Specification***.

6.2 Interoperability in a partial domain of the interoperability functional specification

This occurs when the ***Interoperable Implementations*** have either:

a) adopted a different implementation strategy for the same ***Interoperability Functional Specification***. This may result, for example, from selecting different option sets in the ***Interoperability Functional Specification*** or from the capabilities of the end system configuration. The intersection of the implementations and the ***Interoperability Functional Specification*** is the ***Interoperability Functional Specification Subset*** (IFS subset). This is illustrated in figure 4;

b) implemented compatible, and not identical ***Interoperability Functional Specifications***. Compatible means that two non-identical ***Interoperability Functional Specifications*** have sufficient overlap that some degree of ***Interoperability*** is possible. Clearly, the overlap needs to encompass the necessary functions to operate the basic protocol and state machines. This is illustrated in figure 5.

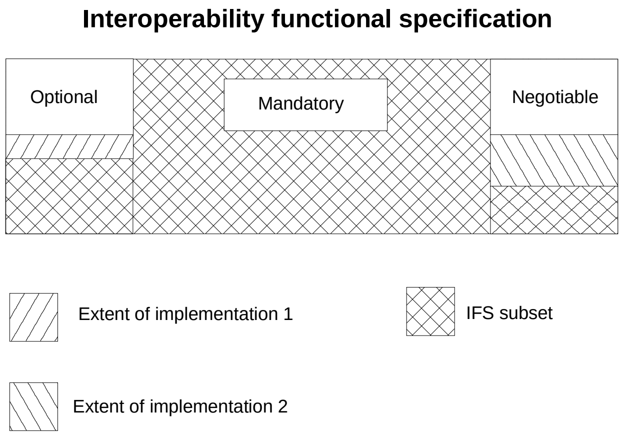

Figure 4: Different implementation strategies

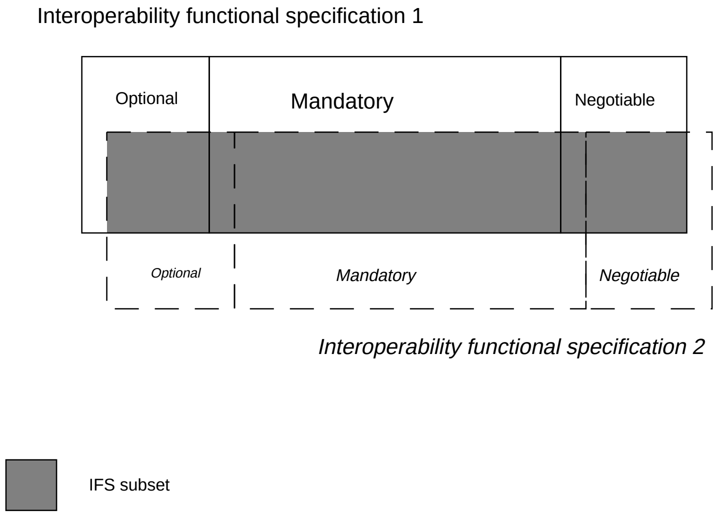

Figure 5: Non-identical interoperability functional specifications

In each case this means that the capabilities of the end systems are not aligned, and that features available on one, are not available on the other.

At one extreme, partial domain ***Interoperability*** may only mean that an infrequently used option is not supported on one system. For example, in otherwise identical implementations of ENV 41201 [2], one end system supports blind copy origination and indication, but the other does not support blind copy origination.

At the other extreme, effective ***Interoperability*** may be impossible through significantly different option sets, fall-back renditions, and display device characteristics. For example, one user creates a document which relies on colours and fonts to communicate the information. The receiving system provides a fall-back rendition using one font on a monochrome screen, and the information content is lost. An even more disastrous example is of two conformant FTAM implementations, each of which have only implemented the FTAM initiator.

Interoperability in a partial domain of the ***Interoperability Functional Specification*** spans a whole spectrum, from the useful to the useless. The extent of usefulness, however, is to a large extent a function of what a particular user requires from the ***Distributed System***, and no objective definitions can be given on what constitutes "useful" ***Interoperability***.

Interoperability in a partial domain of the ***Interoperability Functional Specification*** is not intended to suggest that this is of a lower value than ***Interoperability*** in the full domain. A user requirement may be perfectly satisfied by partial ***Interoperability***.

7 Relationship to standardization

The relevance of standards is different within each classification:

- ***Protocol Interoperability.***

  In this case the physical connectors, signalling systems and protocol machines are expected to be fully standardised, and capable of being conformance tested. The addressing and routeing aspects may be standardised in terms of their syntax and use, but require an agreement out of the domain of standardization regarding the semantics in each instance of a network;

- ***Service Interoperability.***

  The concept of ***Service Interoperability*** is only meaningful in the context of a definition of the services which are required to be supported, and the associated parameter variations and length constraints. The parameter variations and length constraints are specified in the profile. In most profiles there is no definition of the conformance to ***Service*** and, therefore, no requirement for the implementation of specific services. The required services can only be deduced by assuming that the same classification applies to the ***Service*** as applies to the associated PDU parameter in the Protocol Implementation Conformance Statement (PICS) and profile requirements list, and this may only be achieved if there is a clear mapping between the PDU Parameter and the associated ***Service***;

- ***Application Interoperability.***

  The encoding and the semantics of data interchanged is normally expected to be completely standardised, for example in Open Document Interchange Format (ODIF), and in EDI. Boundary values may be standardised in profiles, for example the Office document Architecture (ODA) "FOD" International Standards Profiles (ISPs);

- ***User Perceived Interoperability.***

  Fonts, Character Sets and graphics are subject to standardization or registration. In general the capabilities of display and input devices are not.

8 Requirements for different classes of interoperability

Not all classes are required in all instances of ***Interoperability. Protocol Interoperability*** and ***Service Interoperability*** may be regarded as fundamental to ***Interoperability*** in a ***Distributed System***.

Application Interoperability may be minimal if the ***Service*** offered to the user is a direct reflection of the ***Service Interface***, as it would be for example in the case of a VT terminal implementation, an intermediate system such as a Bridge, and some aspects of CCITT Recommendation X.400 [4] user agents.

The ***User Perceived Interoperability*** aspect may not be relevant where there is no user interaction, for example a CCITT Recommendation X.400 [4] Message Transfer Agent (MTA) using a directory server.

9 Example

The example described is the exchange of an ODA document via CCITT Recommendation X.400 [4] and ITU-T Recommendation X.25 [3]. The following table shows how the ***Interoperability*** is achieved, and what factors may prevent it being achieved.

The scenario illustrated shows two systems connected via ITU-T Recommendation X.25 [3], using profile ENV 41104 [1], supporting CCITT Recommendation X.400 [4] protocol in accordance with ENV 41201 [2], and implementing the associated services through an API. ODIF is sent between two wordprocessor applications through the API, with the expectation that the encoded document may be properly displayed to, and edited by, a human user on each system.

Classification	Achieved through	Reasons for failure
Protocol	Conformant implementations of ENV 41201 [2], and ENV 41104 [1] Common implementation of addressing scheme.	Non-conformant implementations. Incompatible addressing schemes.
Service	Implementation of Services corresponding to CCITT Recommendation X.400 [4] PDUs in an API.	Non-implementation of services corresponding to the PDUs. Different length constraints.
Application	Ability to decode and interpret ODIF.	Inability to decode ODIF, different interpretation of the semantics, non-support of certain semantics.
User	Ability to display, print and edit document, with no loss of information.	Inability to work with selected character set, inability to display or print attributes, inability to display document due to size limitations.

History

Document history	
April 1994	First Edition
February 1996	Converted into Adobe Acrobat Portable Document Format (PDF)