

I n t e r n a t i o n a l T e l e c o m m u n i c a t i o n U n i o n

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

Q.816.2

(03/2007)

SERIES Q: SWITCHING AND SIGNALLING
Q3 interface

CORBA-based TMN services: Extensions to support service-oriented interfaces

ITU-T Recommendation Q.816.2



International
Telecommunication
Union

Documents
Document Sharing Hub

SWITCHING AND SIGNALLING

SIGNALLING IN THE INTERNATIONAL MANUAL SERVICE	Q.1–Q.3
INTERNATIONAL AUTOMATIC AND SEMI-AUTOMATIC WORKING	Q.4–Q.59
FUNCTIONS AND INFORMATION FLOWS FOR SERVICES IN THE ISDN	Q.60–Q.99
CLAUSES APPLICABLE TO ITU-T STANDARD SYSTEMS	Q.100–Q.119
SPECIFICATIONS OF SIGNALLING SYSTEMS No. 4, 5, 6, R1 AND R2	Q.120–Q.499
DIGITAL EXCHANGES	Q.500–Q.599
INTERWORKING OF SIGNALLING SYSTEMS	Q.600–Q.699
SPECIFICATIONS OF SIGNALLING SYSTEM No. 7	Q.700–Q.799
Q3 INTERFACE	Q.800–Q.849
DIGITAL SUBSCRIBER SIGNALLING SYSTEM No. 1	Q.850–Q.999
PUBLIC LAND MOBILE NETWORK	Q.1000–Q.1099
INTERWORKING WITH SATELLITE MOBILE SYSTEMS	Q.1100–Q.1199
INTELLIGENT NETWORK	Q.1200–Q.1699
SIGNALLING REQUIREMENTS AND PROTOCOLS FOR IMT-2000	Q.1700–Q.1799
SPECIFICATIONS OF SIGNALLING RELATED TO BEARER INDEPENDENT CALL CONTROL (BICC)	Q.1900–Q.1999
BROADBAND ISDN	Q.2000–Q.2999
SIGNALLING REQUIREMENTS AND PROTOCOLS FOR THE NGN	Q.3000–Q.3999

For further details, please refer to the list of ITU-T Recommendations.

ITU-T Recommendation Q.816.2

CORBA-based TMN services: Extensions to support service-oriented interfaces

Summary

ITU-T Recommendation Q.816.2 defines a set of TMN CORBA services required to support service-oriented interfaces. It specifies how the ORB and common object services should be used in a lightweight fashion for supporting service-oriented interfaces, and defines extensions to the TMN-specific support services defined in ITU-T Recommendations Q.816 and Q.816.1. A CORBA IDL module defining the interfaces to the new TMN-specific support services is provided. The new services and the lightweight use of other CORBA services, along with ITU-T Recommendation X.780.2, compose a framework for CORBA-based service-oriented TMN interfaces with a wide range of applications.

Source

ITU-T Recommendation Q.816.2 was approved on 16 March 2007 by ITU-T Study Group 4 (2005-2008) under the ITU-T Recommendation A.8 procedure.

Keywords

Common object request broker architecture (CORBA), CORBA services, distributed processing, façade, interface definition language (IDL), managed object, managed system, managing system, service orientation, service-oriented façade object, TMN interface.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure e.g. interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at <http://www.itu.int/ITU-T/ipr/>.

© ITU 2008

All rights reserved. No part of this publication may be reproduced, by any means whatsoever, without the prior written permission of ITU.

CONTENTS

Page

1	Scope	1
1.1	Purpose	2
1.2	Application	2
2	References.....	3
3	Definitions	5
4	Abbreviations.....	7
5	Conventions	10
6	Service-oriented interface design considerations	10
6.1	Flexible use of façade design pattern	11
6.2	Lightweight use of ORB.....	12
6.3	Use of session service by managed and managing systems	15
6.4	Lightweight use of naming service.....	15
6.5	Lightweight use of notification service	18
6.6	Lightweight use of telecom log service.....	20
6.7	Relationship to coarse-grained interface design considerations.....	22
7	Service-oriented framework and requirements overview	22
7.1	Framework overview	22
7.2	Framework constituents overview	24
7.3	Relationships between service-oriented, coarse-grained and fine-grained frameworks	30
8	Framework ORB and common object services usage requirements for supporting service-oriented interfaces	32
8.1	ORB usage requirements	33
8.2	Naming service	33
8.3	Notification service	34
8.4	Telecom log service.....	40
8.5	Concurrency control and object transaction services	41
8.6	CORBAsecurity.....	42
9	Framework support services requirements for supporting service-oriented interfaces.....	44
9.1	Session service.....	44
9.2	Other ITU-T support services.....	49
10	Service-oriented compliance and conformance.....	49
10.1	Standards document compliance	49
10.2	System conformance	50
10.3	Conformance statement guidelines.....	53

	Page
Annex A – Service-oriented framework support services IDL.....	54
A.1 Module <code>idlVersion</code>	55
A.2 Module <code>session</code>	55
A.3 Module <code>nmsSession</code>	56
A.4 Module <code>emsSession</code>	58
A.5 Module <code>emsSessionFactory</code>	59
Bibliography.....	61

iv

CORBA-based TMN services: Extensions to support service-oriented interfaces

1 Scope

The TMN architecture defined in ITU-T Rec. M.3010 (2000) introduces concepts from distributed processing and includes the use of multiple management protocols. ITU-T Recs Q.816 and X.780 subsequently define within this architecture a framework for applying the common object request broker architecture (CORBA) as one of the TMN management protocols. In this approach, manageable network resources are modelled as software objects accessible using CORBA. Information models written in the CORBA interface definition language (IDL) describe the object interfaces. In the original CORBA TMN framework, each managed object class (MOC) is a CORBA IDL interface and so each manageable resource is an independent CORBA object identified and accessed by an interoperable object reference (IOR) that allows for location-transparency. CORBA-based TMN interfaces using the approach where each manageable resource is addressable with a unique IOR have become known as "fine-grained" interfaces. This approach flexibly allows each managed object to reside anywhere – but at the expense that managing systems have on hand an IOR for each managed object they wish to access.

CORBA-based TMN interfaces where not an IOR need be assigned to each manageable resource have become known as "coarse-grained" interfaces. A coarse-grained TMN interface exposes a coarser level of abstraction between a managing system and a managed system for accessing manageable resources but without loss of any detail. At a coarse-grained TMN interface a CORBA object (with an IOR), which is used to access manageable resources having no IOR and invoke operations on them, is referred to as a façade, while an object that is accessed through a façade is referred to as a lightweight object. ITU-T Recs Q.816.1 and X.780.1 extend the framework to support coarse-grained interfaces where one or more façades are defined for each MOC.

This Recommendation, along with ITU-T Rec. X.780.2, add specifications to the framework to enable it to support a service-oriented style of interaction between managing systems and managed systems in addition to the fine-grained and coarse-grained styles specified in the other framework documents. This style of interaction has certain benefits. For example, it can relieve a managing system from having to separately retrieve an identifier or a location information for each type or even each instance of manageable resource it wishes to access, and it can provide a more efficient and very flexible separation of behaviour and state of managed objects. By introducing the so-called "service-oriented façades" it also changes somewhat the way software may be structured on the managed systems, a flexibility which some managed system suppliers may prefer.

The service-oriented framework adopts a lightweight specific use of CORBA to maximize interoperability and interface performance. As a consequence, the use of the naming service can be minimalistic and the use of the ORB and all common object services is lightweight by nature.

A service-oriented architecture (SOA) is an architectural style that aims at maximizing service sharing, reuse and interoperability in distributed environments through loose coupling among interacting components that expose their behaviour through interfaces. In anticipation of the frequent discovery of new business opportunities or threats, an SOA aims at providing open and agile business solutions that can rapidly extend or change on demand, and so SOA-based solutions are composed of reusable services with published and standards-compliant interfaces. The service-oriented approach to CORBA interface design is intended to support efficient SOA interfaces and allow for growth and change as technologies and services are evolving.

The scope of this Recommendation is the same as the fine-grained and coarse-grained TMN CORBA frameworks. These frameworks and the service-oriented extensions cover all interfaces in the TMN where CORBA may be used. These interfaces are OS-OS interfaces according to ITU-T Rec. M.3010, where one OS takes a client/manager role (e.g., an NMS) and the other OS takes a server/agent role (e.g., an EMS). To be concrete, the service-oriented framework support services IDL of Annex A refers to an NML-EML interface but it can be applied to any managing system and managed system. It is expected, however, that not all capabilities and services defined here are required in all TMN interfaces. This implies that the framework can be used for interfaces between management systems at all levels of abstractions (inter- and intra-administration at various logical layers) as well as between management systems and network elements.

1.1 Purpose

The purpose of this Recommendation and the companion ITU-T Rec. X.780.2 is to extend the TMN CORBA framework to enable it to be used in a wider range of applications. The extensions enable a lightweight mode of interaction between the managing and managed systems which may be preferred in many situations. They also enable the endorsement of *de facto* CORBA services and information models usage which are recognized in the telecommunications industry. Thus, this Recommendation is intended for use by various groups specifying network management interfaces.

1.2 Application

ITU-T Rec. X.780.2 accompanies this Recommendation and extends the object modelling guidelines, the superclasses, and the standard sets of data types, exceptions, notifications and constants defined in ITU-T Recs X.780 and X.780.1. Collectively, ITU-T Rec. X.780.2 and this Recommendation define a *framework* for CORBA-based service-oriented TMN interfaces.

ITU-T Rec. M.3010 provides, in its Amendment 1 (2003), conformance definitions for TMN interfaces between physical blocks. When CORBA is used as the TMN management protocol, the conformance criteria refer to the CORBA framework which provides more than one paradigm choice for ORB and CORBA services usage (Q.816-series ITU-T Recommendations) and information modelling in IDL (X.780-series ITU-T Recommendations). Currently these choices are the fine-grained and coarse-grained approaches. According to ITU-T Rec. M.3010 an operations system (OS) interface may make a claim, by level, of *TMN interface information conformance* for each management capability that the interface supports. The supported management capability sets shall be specified by an information model document. Level A, Level B and Level C are defined and distinguished only by the source of the information models that are applied to specify the managed object classes the OS interface supports. The applicable information models need to be specified and well-documented in:

- ITU-T Recommendations, in case of Level A conformance;
- standards of other *de jure* or *de facto* standards bodies, in case of Level B conformance;
- a non-standard way, in case of Level C conformance.

In all cases, implementation conformance statements proformas following the X.781-series of ITU-T Recommendations, as appropriate, shall be provided.

Because ITU-T Recs X.780, X.780.1 and X.780.2 define slightly different approaches to modelling manageable resources on fine-grained, coarse-grained and service-oriented interfaces, interface model specifications will be slightly different for the fine-grained, coarse-grained and service-oriented framework paradigms. While information modelling according to ITU-T Recs X.780 and X.780.1 will always lead to Level A TMN interface information conformance, the lightweight information modelling according to ITU-T Rec. X.780.2 may lead to any conformance level.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

- [1] ITU-T Recommendation M.3010 (2000), *Principles for a telecommunications management network (including Amendment 1 (2003); Amendment 2 (2005))*.
- [2] ITU-T Recommendations M.3050-series (2004), *Enhanced Telecom Operations Map (eTOM)*.
NOTE – This series of Recommendations has the following structure:
M.3050.0 – eTOM – Introduction.
M.3050.1 – eTOM – The business process framework.
M.3050.2 – eTOM – Process decompositions and descriptions.
M.3050.3 – eTOM – Representative process flows.
M.3050.4 – eTOM – B2B integration: Using B2B inter-enterprise integration with the eTOM.
M.3050/Suppl.1 – eTOM – ITIL application note.
M.3050/Suppl.2 – eTOM – Public B2B Business Operations Map (BOM).
M.3050/Suppl.3 – eTOM to M.3400 mapping.
- [3] ITU-T Recommendation Q.816 (2001), *CORBA-based TMN services (including Corrigendum 1 (2001); Corrigendum 2 (2002); Amendment 1 (2001); Amendment 2 (2002))*.
- [4] ITU-T Recommendation Q.816.1 (2001), *CORBA-based TMN services: Extensions to support coarse-grained interfaces*.
- [5] ITU-T Recommendation Q.821.1 (2001), *CORBA-based TMN alarm surveillance service*.
- [6] ITU-T Recommendation X.734 (1992) | ISO/IEC 10164-5:1993, *Information technology – Open Systems Interconnection – Systems management – Event report management function (including Corrigendum 1 (1994); Corrigendum 2 (1999); Amendment 1 (1995) and its Corrigendum 1 (1996))*.
- [7] ITU-T Recommendation X.735 (1992) | ISO/IEC 10164-6:1993, *Information technology – Open Systems Interconnection – Systems management: Log control function (including Corrigendum 1 (2001); Amendment 1 (1995) and its Corrigendum 1 (1996))*.
- [8] ITU-T Recommendation X.780 (2001), *TMN guidelines for defining CORBA managed objects (including Corrigendum 1 (2001); Corrigendum 2 (2002); Amendment 1 (2002))*.
- [9] ITU-T Recommendation X.780.1 (2001), *TMN guidelines for defining coarse-grained CORBA managed object interfaces (including Corrigendum 1 (2002); Amendment 1 (2002))*.
- [10] ITU-T Recommendation X.780.2 (2007), *TMN guidelines for defining service-oriented CORBA managed objects and façade objects*.
- [11] ITU-T Recommendation X.920 (1997) | ISO/IEC 14750:1999, *Information technology – Open distributed processing – Interface definition language*.
- [12] OMG Document formal/98-07-01, *The Common Object Request Broker: Architecture and Specification, Revision 2.2*.

- [13] OMG Document formal/99-10-07, *The Common Object Request Broker: Architecture and Specification*, Revision 2.3.1.
- [14] OMG Document formal/01-02-33, *The Common Object Request Broker: Architecture and Specification*, Revision 2.4.2.
- [15] OMG Document formal/00-06-22, *Property Service Specification*, Version 1.0. See also Chapter 13 of [34], July 1996.
- [16] OMG Document formal/04-10-01, *Lightweight Services Specification*, Version 1.0.
- [17] OMG Document formal/04-10-03, *Naming Service Specification*, Version 1.3. See also Chapter 3 of [34], March 1995, and Chapter 5 of [16], October 2004.
- [18] OMG Document formal/04-10-02, *Event Service Specification*, Version 1.2. See also Chapter 4 of [34], March 1995, and Chapter 6 of [16], October 2004.
- [19] OMG Document formal/04-10-13, *Notification Service Specification*, Version 1.1. See also OMG TC Document telecom/98-11-01, *Notification Service Joint Revised Submission*, November 1998.
- [20] OMG Document formal/03-07-01, *Telecom Log Service Specification*, Version 1.1.2.
- [21] OMG Document formal/02-05-15, *The Common Object Request Broker Architecture and Specification*, Revision 2.6.1.
- [22] OMG Document formal/04-03-12, *Common Object Request Broker Architecture: Core Specification*, Revision 3.0.3. See also OMG TC Document ptc/02-01-14, *Draft CORBA Core 3.0 consisting of CORBA Core 2.6 + Core and Interop RTF 12/2000 Changes + Components FTF Changes*, January 2002.
- [23] ETSI TS 132 150 V6.5.0 (2006), *Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); Telecommunication management; Integration Reference Point (IRP) Concept and definitions* (3GPP TS 32.150 version 6.5.0 Release 6).
- [24] ETSI TS 132 303 V6.6.0 (2005), *Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); Telecommunication management; Notification Integration Reference Point (IRP): Common Object Request Broker Architecture (CORBA) Solution Set (SS)* (3GPP TS 32.303 version 6.6.0 Release 6).
- [25] ETSI TS 132 333 V6.1.0 (2006), *Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); Telecommunication management; Notification Log (NL) Integration Reference Point (IRP): Common Object Request Broker Architecture (CORBA) Solution Set (SS)* (3GPP TS 32.333 version 6.1.0 Release 6).
- [26] ETSI TS 132 111-3 V6.7.0 (2007), *Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); Telecommunication management; Fault Management; Part 3: Alarm Integration Reference Point (IRP): Common Object Request Broker Architecture (CORBA) Solution Set (SS)* (3GPP TS 32.111-3 version 6.7.0 Release 6).
- [27] TM Forum TMF814 Version 3.0, Multi-Technology Network Management (MTNM) NML-EML Interface: CORBA IDL Solution Set, *Supporting Document "Programmatic Versioning"*, file "versioning.pdf", March 2004.
- [28] TM Forum TMF814 Version 3.0, Multi-Technology Network Management (MTNM) NML-EML Interface: CORBA IDL Solution Set, *Supporting Document "Guidelines for Using the OMG Notification and Telecom Log Service"*, file "OMGservicesUsage.pdf", March 2004.
- [29] OASIS Document soa-rm (2006), *Reference Model for Service Oriented Architecture 1.0*.

3 Definitions

The following terms used in this Recommendation are defined in ITU-T Rec. M.3010:

- **interface;**
- **operations system (OS).**

The following term used in this Recommendation is adapted from ITU-T Rec. X.700:

- **managed object (MO):** The management view of a resource, as defined by ITU-T Rec. M.3050.1, that may be managed through the use of a management protocol (such as CORBA).

NOTE 1 – ITU-T Rec. X.700 restricts the semantics of "resource" to resources which "provide interconnection capabilities" and "allow communications to take place"; it explicitly excludes resources which "provide data storage or processing capabilities". ITU-T Rec. M.3050.1 draws resources from the eTOM's application, computing and network domains.

NOTE 2 – A managed object is characterized in terms of attributes it possesses, operations that may be performed upon it, notifications that it may issue, and its relationships with other managed objects.

The following terms used in this Recommendation are adapted from ITU-T Rec. X.701:

- **agent:** An OS which has taken an agent role.
- **agent role:** A role taken by an OS in which it is capable of performing management operations on managed objects and of emitting notifications on behalf of managed objects.
- **managed object class (MOC):** A named set of managed objects sharing the same (possibly named) set of attributes (the MOC's state) and the same (possibly named) set of operations and notifications (the MOC's behaviour).
- **managed system:** A synonym for agent.
- **manager:** An OS which has taken a manager role.

NOTE 3 – Some products that implement the service-oriented framework use this term, or the term "object manager", as a synonym for service-oriented façade.

- **manager role:** A role taken by an OS in which it is capable of issuing management operations and of receiving notifications.
- **managing system:** A synonym for manager.

NOTE 4 – ITU-T Rec. X.701 defines the term "(management) interaction" and uses it to state more sophisticated definitions of agent and manager.

The following term used in this Recommendation is defined in ITU-T Rec. X.703:

- **notification.**

NOTE 5 – ITU-T Rec. X.703: "*Information technology – Open Distributed Management Architecture*" provides the ODMA definition, or ODP-RM definition, of notification. ITU-T Rec. X.703/Amd.1: "*Support using Common Object Request Broker Architecture (CORBA)*" relates this definition to the OMG event definition [18] (generic or typed event message), and hence to the OMG notification definition [19] (generic or typed or structured event message).

The following term used in this Recommendation is defined in ITU-T Rec. Q.816 and in [18]:

- **event channel.**

The following term used in this Recommendation is defined in [19]:

- **notification channel.**

The following terms used in this Recommendation are defined in ITU-T Rec. X.734:

- **event forwarding discriminator (EFD);**
- **event report.**

The following terms used in this Recommendation are defined in ITU-T Rec. X.735 and in [20]:

- **log**;
- **log record**.

The following term used in this Recommendation is defined in [20]:

- **notification log**.

The following term used in this Recommendation is defined in ITU-T Rec. X.780.1:

- **façade**.

The following terms are defined in this Recommendation in harmony with [29]:

- **service broker (SB)**: An entity, also known as service registry, which allows SPs to register (and maintain) SDs and allows SCs to find SDs in a location-transparent way.
- **service consumer (SC)**: An entity which seeks to satisfy a particular need through the use of capabilities offered by means of a service.
- **service description (SD)**: The information needed in order to use, or consider using, a service. See section 3.3.1 of [29].

NOTE 6 – The most important part of a service description is the association of one or more service interfaces that enable access to the capabilities provided by the service.

NOTE 7 – When a service description includes constraints and policies, it is called a service contract.

- **service interface (SI)**: The means by which all or part of the underlying capabilities of a service are accessed. It is a means for interacting with a service (see section 3.3.1.4 of [29]).

NOTE 8 – A service interface delineates and exposes an external view of functionality of a service; it defines all or part of that service's action boundary.

- **service-oriented approach/architecture (SOA)**: An ICT/IT&T architecture of services, policies, best practices and frameworks in which components can be reused and repurposed rapidly in order to achieve shared and new functionality.

NOTE 9 – An SOA provides a uniform means to offer, discover, interact with and use capabilities to produce desired effects consistent with measurable preconditions and expectations. It is a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. It is an architectural style that aims at maximizing service sharing, reuse and interoperability in distributed environments through loose coupling among interacting components that expose their behaviour through interfaces. In anticipation of the frequent discovery of new business opportunities or threats, SOAs aim at providing open and agile business solutions that can rapidly be extended or changed on demand, and so SOA-based solutions are composed of reusable services with published and standards-compliant interfaces. SOA entities are accessible only through interfaces and connected by service descriptions. The SOA paradigm enables fast adaptation to changing business needs as well as cost reduction in the deployment of new services and maintenance of deployed services.

- **service-oriented façade (SO façade)**: A synonym for service-oriented façade interface or for service-oriented façade object, depending on the context.
- **service-oriented façade interface (SO façade interface)**: An object interface defined to provide access to a set of service-oriented managed objects, and optionally also to other objects.
- **service-oriented façade object (SO façade object)**: An object which exists in the managed system and provides, in a steward role, access to allocated service-oriented managed objects, and optionally also to other objects, of the managed system. It is an instance of an SO façade.

- **service-oriented managed object (SO MO):** A managed object which is only accessible via an assigned service-oriented façade and provides behaviour (operations and notifications) to managing systems not directly but only through behaviour of its assigned façade.

NOTE 10 – A managed object represents a manageable resource in terms of state (attributes) and behaviour (operations and notifications). Functional components of the managed object such as state and behaviour can be distributed either in an implementation or in its interface definition or in both. A service-oriented managed object is a managed object whose state and behaviour are distributed in the interface definition; its state is left with the object while its behaviour is delegated to a steward object, namely its assigned façade. Whether this also implies a separation of the managed object implementation is not visible at the interface; it is an implementation-defined issue that is left to the managed system's discretion.

- **service provider (SP):** An entity that offers the use of capabilities by means of a service.
- **service resource (SR):** A physical or logical source or sink of data, or a state-aware either synchronous or asynchronous ICT/IT&T data/signal processing capability.

NOTE 11 – Service resources are used by services to deliver and maintain (persistent or temporary) data. When the data is specified by an information model, service resources can be used to construct service interfaces and processable service descriptions (i.e., metadata). Examples of service resources are network elements, network infrastructure items (e.g., connectivity or profile information), software/firmware, database systems, and COTS technology components.

- **SOA service:** A means by which the needs of a consumer are brought together with the capabilities of a provider, and which is realized as a set of behaviours that are accessible through, or are using, one or more service interfaces. Refer to section 3.1 of [29] for details.

NOTE 12 – An SOA incarnates any service as one or more entities that expose service interfaces. A service in a provider role constitutes an action boundary regarding access and control between one or more service resources and one or more consumers of the resources.

4 Abbreviations

This Recommendation uses the following abbreviations:

3GPP	3rd Generation Partnership Project
AMI	Asynchronous Messaging Invocation
ATM	Asynchronous Transfer Mode
BPON	Broadband Passive Optical Network
CBTS	CORBA-Based TMN Services (ITU-T Rec. Q.816)
CCBTS	Coarse-grained CORBA-Based TMN Services (ITU-T Rec. Q.816.1)
CCS	Concurrency Control Service
CM	Configuration Management
CMIP	Common Management Information Protocol
CORBA	Common ORB Architecture
COS	Common Object Service(s)
COTS	Commercial Off-The-Shelf technology
CSI	Common Secure Interoperability
DIOA	Distributed Interface-Oriented Architecture
DN	Distinguished Name

DSL	Digital Subscriber Line
EFD	Event Forwarding Discriminator
EML	Element Management Layer
EMS	Element Management System
eTOM	enhanced Telecom Operations Map
fd	filterable data
FIFO	First In, First Out
GDCCMO	Guidelines for Defining Coarse-grained CORBA Managed Object Interfaces (ITU-T Rec. X.780.1)
GDCMO	Guidelines for Defining CORBA Managed Objects (ITU-T Rec. X.780)
GDMO	Guidelines for the Definition of Managed Objects (ITU-T Rec. X.721/X.722)
GDSOCMO	Guidelines for Defining Service-Oriented CORBA Managed Objects (ITU-T Rec. X.780.2)
GIOP	General Inter-ORB Protocol
ICT	Information and Communications Technology
ID	Identifier
IDL	Interface Definition Language
IIOP	Internet Inter-ORB Protocol
IIOP/SSL	IIOP over SSL
IIOP/TLS	IIOP over TLS
IKE	Internet Key Exchange
IOP	Inter-ORB Protocol
IOR	Interoperable Object Reference
IPSec	IP Security
IRP	Integration Reference Point
IT	Information Technology
IT&T	Information Technology & Telecommunication(s)
LDAP	Lightweight Directory Access Protocol
MO	Managed Object
MOC	Managed Object Class
MOO	Multiple Object Operation
MTNM	Multi-Technology Network Management
NE	Network Element
NGOSS	New Generation Operations Systems and Software
NL	Notification Log
NML	Network Management Layer
NMS	Network Management System

NV pair	name/value pair with name being a string and with a "name=value" semantics
OAM&P	Operations, Administration, Maintenance & Provisioning
OASIS	Organization for the Advancement of Structured Information Standards
OCN	Object Creation Notification
ohf	optional header field
OMG	Object Management Group
OO	Object-Oriented
OOAD	Object-Oriented Analysis and Design
ORB	Object Request Broker
OS	Operations System
OTS	Object Transaction Service
OTS	OMG Transaction Service
PIM	Platform Independent Modelling
PKI	Public Key Infrastructure
PM	Performance Management
POA	Portable Object Adapter
PSM	Platform Specific Modelling
QoS	Quality of Service
RFP	Request For Proposal(s)
SB	Service Broker
SC	Service Consumer
SD	Service Description
SECIOP	Secure Inter-ORB Protocol
SECP	Security Protocol
SI	Service Interface
SO	Service-Oriented
SOA	Service-Oriented Approach/Architecture
SOCBTS	Service-Oriented CORBA-Based TMN Services (ITU-T Rec. Q.816.2)
SOF	Service-Oriented Façade
SP	Service Provider
SR	Service Resource
SS	Solution Set
SSL	Secure Sockets Layer
SSL/TLS	SSL or TLS
TC	Technical Committee
TCA	Threshold Crossing Alert
TCP	Transmission Control Protocol

TII	Time-Independent Invocation
TLS	Transport Layer Security
TMN	Telecommunications Management Network
TS	Technical Specification
UML	Unified Modelling Language
UMTS	Universal Mobile Telecommunications System
UTRAD	Unified TMN Requirements, Analysis and Design

5 Conventions

A few conventions are followed in this Recommendation to make the reader aware of the purpose of the text. While most of the Recommendation is normative, paragraphs succinctly stating mandatory requirements to be met by a management system (managing and/or managed) are preceded by a boldface "R" enclosed in parentheses, followed by a short name indicating the subject of the requirement, and a number. For example:

(R) EXAMPLE-1 An example mandatory requirement.

Requirements that may be optionally implemented by a management system are preceded by an "O" instead of an "R". For example:

(O) EXAMPLE-2 An example optional requirement.

The requirement statements are used to create compliance and conformance profiles.

Examples of CORBA IDL are included in this Recommendation, and normative IDL specifying the TMN-specific framework support services, and associated data types, is included in Annex A. The IDL is written in a 9-point courier typeface:

```
// Example IDL
void getManager(
    in string managerName,
    out common::Common_I managerInterface)
    raises(globaldefs::ProcessingFailureException);
```

Annex A contains source code that implementers will want to extract and compile. It is normative and should be used by developers implementing systems that conform to this Recommendation. Refer to clause 5.2/Q.816.1 for recommendations with regard to cutting and pasting the IDL of this Recommendation into plain text files that may then be compiled. For example, the entire IDL file "itut_q816_2.idl" can be generated by cutting Annex A from the Microsoft® Word® version of this Recommendation and saving as "Text Only with Line Breaks". This IDL file contains some ordinary comments and a large number of formatted comments to be parsed by compilers used to convert IDL to HTML for easier reading. A formatted comment begins with /** and ends with */ and is associated with the next IDL construct. HTML formatting tags are allowed with these comments as are certain keywords (preceded by a '@' symbol) that are converted by the IDL-to-HTML compilers into additional formatting. Users of the IDL may want to generate HTML as the resulting HTML files have links that make for quick navigation through the files.

6 Service-oriented interface design considerations

This clause identifies several design considerations that should be addressed by the framework as support for lightweight use of CORBA through service-oriented interfaces is added.

CORBA realizes a distributed interface-oriented architecture (DIOA) where location-transparent objects expose one or more interfaces for access to their encapsulated state and behaviour. The concepts of coarse-grained object modelling and service-oriented architecture (SOA) resolve issues

with object-oriented analysis and design (OOAD) best practices. Classical OOAD focuses on the class level, i.e., encapsulates behaviour and related data in the same object. This has become known as "fine-grained" approach. A general "coarse-grained" approach uses the *façade* design pattern¹ to separate state and behaviour (and share the state) of some classes, i.e., the behaviour of some (shared) objects is no longer controlled by themselves but by façade objects. In a coarse-grained object-oriented approach, data is still encapsulated but data access and control are organized at a separate upper level which results in coarser "grains" being accessed and controlled.

ITU-T Recs Q.816.1 and X.780.1 use a specific form of the façade design pattern (e.g., an X.780.1 façade can only provide access to a single MOC) and other design considerations to define support for coarse-grained interfaces. This Recommendation, along with ITU-T Rec. X.780.2, define a lightweight generic use of the façade design pattern which, together with other lightweight design considerations, generalize the coarse-grained approach (and therefore indirectly also the fine-grained approach). This novel approach to interface design is called "service-oriented" since it paves the way for introducing SOA principles to the TMN interface specification methodology. CORBA-based service orientation requires the flexibility of application-specific access granularity where well-defined sets of TMN entity types are accessed through assigned façades with IORs (see clause 4.2.1/X.780). Refer to clause 7.3 below and clause 12/X.780.2 for a thorough description of the relationships between the three approaches. Additional information is available from [53].

The design considerations related to CORBA services concern the lightweight and service-oriented use of the ORB, the use of session objects to control the communication between managing and managed systems, and the lightweight use of naming, notification and telecom log services.

6.1 Flexible use of façade design pattern

The coarse-grained framework introduces façades but also requires certain coupling conditions between a façade and the managed objects that are accessible through the façade. For example, a managed system shall provide at least one façade interface for each class of managed objects that may be instantiated on it, even if these objects also support direct CORBA interfaces, and may provide multiple façade interfaces for a given class of managed objects. These requirements effectively reduce the coarse-grained approach to a sort of class-grained approach. Refer to Appendix I/Q.816 for the concepts of class-grained approach and grain-neutral approach.

A coarse-grained interface is created by first defining a fine-grained interface according to the guidelines defined in ITU-T Rec. X.780 which cover both creating GDMO-like IDL interfaces from scratch as well as translating a GDMO interface to IDL. Once a fine-grained interface is defined, façade interfaces for each of the managed object interfaces are developed according to rules defined in ITU-T Rec. X.780.1. The fine-grained managed object interface specifications shall be retained. All of the other constructs defined for the fine-grained interface, including data types, value types, exceptions, notifications, and factories, are reused without modification on the coarse-grained interface. As a major example, ITU-T Rec. M.3120 applies the X.780 and X.780.1 guidelines to ITU-T Recs M.3100 and G.855.1 and translates manually all of their GDMO interfaces to first fine-grained and then coarse-grained CORBA IDL interfaces.

A service-oriented interface *can* depend on predefined coarse-grained or fine-grained interfaces but usually does not. The service-oriented approach requires the introduction of *service-oriented façades*, which are CORBA objects, and *managed objects*, which are accessed and controlled

¹ The façade design pattern [52] is a structural pattern with the intent to provide a unified interface to a set of interfaces in a subsystem. A façade defines a higher-level interface that makes the subsystem easier to use. A façade promotes loose coupling between the subsystem and its clients. Loose coupling makes it possible to vary the components and interfaces of the subsystem without affecting its clients.

(preferably exclusively) through service-oriented façades². Both types of objects are instantiated at the managed system. Only loose coupling between managed objects and service-oriented façades is required from the outset (see clause 7.2.1.1) leaving detailed relationships to the discretion of the managed system, which may also support dynamic binding of managing systems to one or more service-oriented façades (i.e., systems interactions are established and broken dynamically at run time and not yet defined statically at design time).

While transition from a fine-grained model to a coarse-grained model is defined *bottom-up* through deterministic rules for coarsening of a fine-grained interface, transition from a service-oriented model to a coarse-grained or fine-grained model would be defined (if required someday) *top-down* through rules for stepwise refining of a service-oriented interface according to ITU-T Recs X.780.1 and X.780. However, the service-oriented approach mainly promotes coexistence of fine-grained, coarse-grained and (information model-dependent) service-oriented interfaces, and initially does not aim at transition of one TMN interface kind to another.

6.2 Lightweight use of ORB

The OMG CORBA specifications 2.2 [12], 2.3 [13] and 2.4 [14] consist of several parts for several aspects of CORBA that shall or should be offered by compliant ORB vendors. See also ITU-T Rec. X.920. Most important for use by the TMN frameworks are the following *two aspects of CORBA*:

- **IDL modelling repertoire of ORB products:** these are mainly the chapters 3 "OMG IDL Syntax and Semantics" and 5 "Value Type Semantics" of the CORBA specifications;
- **built-in basic CORBA system services of ORB products:** these are mainly the chapters 4 "ORB Interface" and 11 "The Portable Object Adapter" of the CORBA specifications.

The lightweight use of the first aspect, *the IDL repertoire and modelling*, including IDL style considerations (in compliance with [51] and similar to Annex D of [23]), is specified in the companion ITU-T Rec. X.780.2 whilst the lightweight use of the second aspect, *the basic CORBA system services*, is specified in this Recommendation.

Revisions 2.2 and 2.3 of CORBA introduced the *portable object adapter (POA)* and *value types*, respectively. Both capabilities are required by the fine-grained and coarse-grained frameworks. These requirements are implied in the explicit requirement that the supported version of CORBA shall be 2.3.1 or later. The fine-grained and coarse-grained frameworks do specify neither separate POA features nor individual value type features required for ORB compliance.

CORBA Revision 2.4 introduced among other things *Minimum CORBA* and *CORBA Messaging*, in particular asynchronous messaging invocation (AMI) including time-independent invocation (TII). Minimum CORBA aims for ORB size reduction and overall performance improvements. It omits the features from the CORBA and PortableServer modules of the CORBA specification that support the run-time aspects of CORBA (e.g., dynamic interfaces at client side or server side, interceptors, dynamic mode of POA operation) and indicates conformant ways for vendor-specific optimizations. CORBA messaging standardizes and quality assures asynchronous requests, which previously could be issued only with the "ORB best effort" semantics of one-way operations using

² Some products that implement the service-oriented framework use the term "object manager" or just "manager", as a synonym for service-oriented façade. Another synonym sometimes used is "managing object" in contrast with managed object. Evolving managed objects to "service-oriented managed objects" means separating their state and behaviour, and outsourcing the behaviour to assigned "managing objects" that take a steward role regarding the operations and notifications of their allocated managed objects. This fundamental SOA principle can also be described as separating the *definition and storage of data* from the *delivery of data*. Since the terms "managing object" and "manager" may give rise to misunderstandings with regard to the managing role of systems and subsystems, this Recommendation mainly uses the term "service-oriented façade" but may sometimes use the synonyms for explanation purposes.

application-specific callback objects. It also provides support for "persistent" requests. It is based on an IDL-to-"implied-IDL" mapping that is carried out by messaging-enabled IDL code generators when generating language-specific client stubs. Every (synchronous) IDL operation is copied to an asynchronous operation whose signature is changed by adding either a callback interface (reply handler) as input parameter or a (queriable) poller value type (reply wrapper) as return value.

6.2.1 Use of dynamic and asynchronous capabilities

This clause discusses *the basic CORBA system services aspect* of CORBA with regard to the service-oriented framework.

Though being CORBA-specific, the service-oriented framework supports OMG's platform independent modelling (PIM) in UML and aims at the introduction of SOA principles to the TMN interface specification methodology UTRAD (see M.3020-series ITU-T Recommendations). SOA is interface-oriented and defines two views on components: the inward view of a service consumer, who requires and consumes one or more interfaces, and the outward view of a service provider, who implements and provides interfaces. A core SOA principle is a loose coupling of components that interact according to the publish-find-negotiate-bind-execute paradigm, which implies the possibility of dynamic binding, unbinding and rebinding between a required interface and a located provided interface. Refer to clause 7.2.1.2 for a summary of the basic SOA artefacts.

As a consequence, the dynamic and asynchronous aspects of CORBA are very important for the specification of service-oriented interfaces even though some of them are a bit sophisticated. But another SOA principle is ease of component development and deployment which is enabled by lightweight features. This framework therefore adopts a stepwise approach to run-time and asynchronous CORBA features, which extends the fine-grained and coarse-grained interface design approaches to ORB usage. The following steps are defined:

- mandatory use of the POA as defined by Minimum CORBA (Note 1);
- optional use of dynamic mode of POA operation;
- optional use of other dynamic aspects of CORBA omitted by Minimum CORBA;
- use of one-way operations with application-specific callback objects only in exceptional and provisional cases and then only in conjunction with a synchronization policy as specified by CORBA Messaging; as a rule TMN IDL interfaces are defined to be synchronous;
- optional use of transient asynchronous requests as defined by the AMI specification and further detailed in clause 6.4/Q.816 (Note 2);
- optional use of persistent asynchronous requests as defined by the TII specification.

The service-oriented framework therefore concurs with the fine-grained and coarse-grained frameworks regarding the use of POA and AMI except that it allows quality assured one-way operations in exceptional and provisional cases. It adds a prioritization of optional capabilities whose details and impacts on CORBA services usage (e.g., transactional asynchronous requests) are for further study. As a consequence, version 2.2 or later of CORBA shall be supported. From an IDL modelling point of view, CORBA 2.3 is only needed when value types shall be used (see clause 6.2.2/X.780.2). From a run-time performance point of view CORBA 2.3 may be beneficial for the processing of unbounded sequences, which are heavily used in COS and telecom IDL specifications (and even with complex members such as nested strings), since ORB products that implement CORBA 2.3 are said to treat all kinds of variable-length structures more efficiently. CORBA 2.3 also offers improved type code techniques for introspection and manipulation of the IDL type `any`. Furthermore ORB interoperability strongly depends on the ORB version (see clause 8.6).

NOTE 1 – Support of basic POA capabilities and exclusion of preceding object adapters ("BOA begone") is important to all frameworks because it enables implementations based on a framework to scale up to millions upon millions of instantiated objects, a magnitude required for telecommunications network management applications, and ensures portability between ORB products. The object adapter is the built-in CORBA mechanism that connects a request with the right code to service that request. A programming language entity such as a C++ object that implements requests on one or more CORBA objects, i.e., operations of one or more CORBA interfaces, is called a servant. The POA achieves maximum portability between ORBs, minimizes for a request the implementation code route covered to connect to the incarnating servant and process the request, and allows to persistently store object states between operation invocations. Using the POA is a *conditio sine qua non* when implementing fine-grained interfaces. For coarse-grained and service-oriented interfaces the POA provides, besides ORB portability, also considerable performance advantages when few servants are called with exceedingly high frequency. These considerations do show that the object adapter will no longer be a potential bottleneck for scalability and overall performance but they need to be supplemented by considerations for the lightweight use of the Naming Service (see clause 6.4).

NOTE 2 – Asynchronous requests are a client side (i.e., managing system) programming language mapping issue. The server side is not affected by invocation policies of clients but sticks to always expecting to be invoked synchronously. To allow nevertheless synchronous clients to make non-blocking requests on a CORBA object, which may be required to avoid poor performance due to network latency, the service-oriented framework recommends that CORBA servers (i.e., managed systems) implement multithreading (e.g., by using a thread pool policy with appropriate locking granularity). The fine-grained and coarse-grained frameworks require the client side to implement multi-threading in case the AMI is not used. Within the service-oriented framework, multi-threaded managing systems are optional.

6.2.2 IDL repertoire and ways of MOC specification

This clause addresses *the IDL repertoire and modelling aspect* of CORBA with regard to the service-oriented framework.

The CORBA IDL-related constituents of the service-oriented framework are described in the companion ITU-T Rec. X.780.2, including the minimum IDL repertoire that should be used, based on the service-oriented object model IDL and the service-oriented CORBA modelling guidelines. A particularly important part is the modelling of managed objects and service-oriented façades, which inherit from a set of common attributes called *Common_T* (which is a virtual struct) respectively from a root service-oriented façade interface called *Common_I* (which is virtual). The *Common_I* façade interface specifies among other things the setter/mutator functions for the settable common attributes. All three frameworks group the attributes of managed object types into separate entities, the managed object classes (MOCs), which can be interfaces or be instantiated (by the managed system or the managing system or both, as specified) to non-CORBA managed objects that are identifiable by distinguished names. The fine-grained and coarse-grained frameworks require value types as the grouping mechanism, while the service-oriented framework allows for more flexibility in TMN interface design and defines *four alternative ways of MOC specification*:

- an MOC may be an interface that has the attributes as CORBA attributes or explicitly defined accessor (and mutator) functions;
- an MOC may be a struct that has the attributes as record members;
- an MOC may be a valuetype that has the attributes as public state members;
- an MOC may be a CORBA sequence of name/value pairs with string names and any values, i.e., an MOC may be of type `sequence<struct {string name; any value;};>`, that has the attributes as name/value pairs with a "name=value" semantics.

Refer to ITU-T Rec. X.780.2 for details and recommendations for the use of the respective way of MOC specification. As a rule, OMG specifications define properties of entities as name/value pairs with string names and any values [12], [13], [14], [15] and [19] but also introduce such name/value pairs with string values [17]. To support and enable extensibility, the service-oriented framework makes vigorous use of free format name/value pairs where the name and the value are both a simple string. Companion ITU-T Rec. X.780.2 also specifies rules and guidelines for the

lightweight definition of operations, exceptions and notifications. Since the three TMN CORBA frameworks are meant to be used for the management of telecommunications networks, ITU-T Rec. X.780.2 moreover specifies rules for the IDL modelling of multiple telecom transmission technologies according to the principles of ITU-T Recs G.805 and G.809. This is *the multi-technology part* of the service-oriented framework, which can also be used together with the coarse-grained and fine-grained frameworks.

6.3 Use of session service by managed and managing systems

The session service provides capabilities to manage a client/server connection between a managing system and a managed system, which is called a session. It enables either a client or a server to detect the loss of communication with the associated party. For a single session between a managing system and a managed system, there are two session objects. One is maintained on the managing system (client); the other one is maintained on the managed system (server).

A session object has a read-only attribute that contains a reference to the session object on the other side (managing/managed system) to which the object is associated. This attribute allows for callback invocation of the associated session object and to check in case of communication failure if the association is still valid. Loss of communication may be detected through a `ping` operation, and a one-way `endSession` operation allows for a controlled disconnect between parties that releases all resources allocated for this session object and destroys the object.

A client session provides operations for callback by the server in case of notification losses and termination of a loss period. A server session provides operations to query the names of the supported managing objects (manager interfaces), to retrieve for each supported manager its object reference to gain access to it without using the naming service, and to retrieve the object reference of the unique event channel to be used by the managing system in this session.

To enable the session service, the managed system (server) registers one or more instances of the EMS session factory interface with the naming service, which then constitutes the unique entry point(s) to the managed system. The managing system instantiates a client session object and calls the `getEmsSession` operation of an EMS session factory, together with authentication data, to request instantiation and association of a server session object and return of its object reference. Before establishing a session with the managed system, the managing system may use the version check facility of the EMS session factory to avoid connection with an unpopular IDL version.

6.4 Lightweight use of naming service

A CORBA *name component* is a pair of strings that is called an id/kind pair since it is of type `struct NameComponent {string id; string kind;};`. A CORBA *name* is a list of name components, that is a CORBA name is of type `sequence<NameComponent>`. An OMG naming service [17] maps CORBA names, or name components, to IORs. A name-to-IOR, or name component-to-IOR, association is briefly called a (name, or name component,) *binding*. A *naming context*, or just a *context* for short, is a CORBA system object, namely an instance of the `CosNaming::NamingContext` interface, which stores bindings by implementing a table that maps unique CORBA names, or name components, to IORs. Different names, or no name, can be bound to any individual IOR in the same or different naming contexts. A binding can refer to an application object or another naming context of the same or a federated naming service.

The bindings of a naming service are usually depicted as a CORBA *naming graph* that shows context IORs with hollow nodes, application IORs as solid/filled nodes, and arrows between nodes representing the bindings. Each binding arrow is labelled with the binding's name part and points to its IOR part, and so collectively the arrows originating from an individual context node represent its binding table. In a given context, any CORBA name is represented by a sequence of nodes that originates from this context and traverses the naming graph until a leaf is reached. In particular, each node of a naming graph represents a CORBA name component. The ORB can be configured