

ETSI TS 101 878 V4.1.1 (2003-11)

Technical Specification

Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON) Release 4; Service Capability Definition; Service Capabilities for TIPHON Release 4



Reference

RTS/TIPHON-01009R4

Keywords

IP, service, telephony, VoIP

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

Individual copies of the present document can be downloaded from:

<http://www.etsi.org>

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF). In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at

<http://portal.etsi.org/tb/status/status.asp>

If you find errors in the present document, send your comment to:

editor@etsi.org

Copyright Notification

No part may be reproduced except as authorized by written permission.
The copyright and the foregoing restriction extend to reproduction in all media.

© European Telecommunications Standards Institute 2003.
All rights reserved.

DECTTM, **PLUGTESTS**TM and **UMTS**TM are Trade Marks of ETSI registered for the benefit of its Members.
TIPHONTM and the **TIPHON logo** are Trade Marks currently being registered by ETSI for the benefit of its Members.
3GPPTM is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.

Contents

Intellectual Property Rights	5
Foreword.....	5
Introduction	5
1 Scope	6
2 References	6
3 Definitions and abbreviations.....	6
3.1 Definitions	6
3.2 Abbreviations	7
4 Services and service capabilities	7
5 Naming	7
6 Overall UML class model	7
6.1 Other entities in the TIPHON SC Model.....	9
6.1.1 TIPHON User	9
6.2 Scope of service capabilities with respect to user and network.....	9
7 Bearer class service capabilities	10
7.1 Bearer attribute (data element) definitions	11
7.2 Service capabilities	12
7.2.1 Create	12
7.2.2 Modify	12
7.2.3 Delete.....	12
7.2.4 Join.....	12
7.2.5 setCondition.....	12
7.2.6 clearCondition.....	13
8 Call class service capabilities	13
8.1 Call attribute (data element) definitions	14
8.2 Service capabilities	14
8.2.1 Setup	14
8.2.2 identityDelivery	14
8.2.3 Redirect.....	15
8.2.4 setPriority.....	15
8.2.5 cleardown.....	15
8.2.6 Join.....	15
8.2.7 Interrogate.....	15
8.2.8 setCondition.....	15
8.2.9 clearCondition.....	16
8.2.10 locationDelivery	16
8.2.11 Route.....	16
9 Profile class service capabilities	17
9.1 Service capabilities	17
9.1.1 Register	17
9.1.2 Attach.....	18
9.1.3 Authenticate	18
9.1.4 getStatus.....	18
9.1.5 Deregister.....	18
9.1.6 Transfer.....	18
9.1.7 Authorize	18
9.1.8 setStatus	19
9.1.9 setCondition.....	19
9.1.10 clearCondition.....	19
10 Message class service capabilities.....	20

10.1	Message attributes	21
10.2	Service capabilities	21
10.2.1	Create	21
10.2.2	Retrieve	21
10.2.3	setStatus	21
10.2.4	getStatus	21
10.2.5	Delete	22
11	Media class service capabilities	22
11.1	Data attributes	22
11.2	Service capabilities	22
11.2.1	setMediaEncode	22
11.2.2	clearMediaEncode	23
11.2.3	createTransport	23
11.2.4	clearTransport	23
Annex A: (informative) Method of developing meta-protocols from service capabilities		24
A.1	Overview	24
A.2	Service capability model	24
A.3	From SC to Meta-protocol	25
Annex B: (informative) Interconnection agreements		26
History		27

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: "*Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards*", which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (<http://webapp.etsi.org/IPR/home.asp>).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Specification (TS) has been produced by ETSI Project Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON).

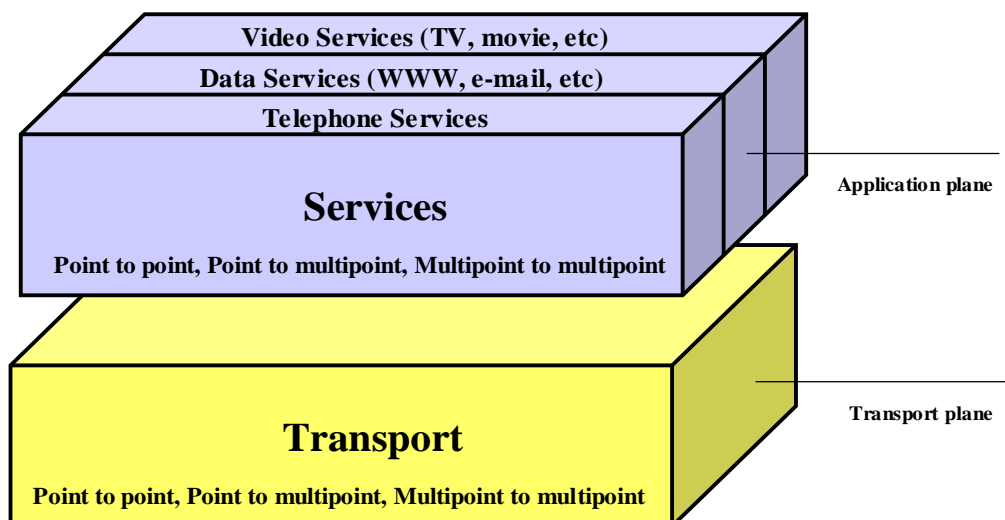
Introduction

The present document uses the Universal Modelling Language (UML) to model the TIPHON Release 4 service capabilities in an object oriented approach. The advantages of such an approach include:

- the ability to express service capabilities in a semi-formal modelling language;
- inheritance of the Object Oriented Design paradigm and with it concepts of reuse, inheritance, polymorphism, and extensibility.

NOTE: The approach to the definition of service capabilities in TIPHON Release 3 (v1.1.1 of the present document) has been modified in this edition. The approach in v1.1.1 did not specify object-based design methods to develop the service capability model and instead used text to define the service capabilities where those service capabilities were derived from and abstracted from the capabilities and supplementary services of ISDN.

The service capabilities described in the present document exist in the control plane of TIPHON, where this application plane may be vertically apportioned as shown in the figure below:



NOTE: Some of the application plane services will only be available in TIPHON R5 or later.

Figure 0: TIPHON application and transport planes

1 Scope

The present document specifies those service capabilities sufficient to implement the services in the scope of TIPHON Release 4 defined in ETSI TR 101 301 [1] and examples of which are given in ETSI TS 101 315 [4].

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication and/or edition number or version number) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies.

Referenced documents which are not found to be publicly available in the expected location might be found at <http://docbox.etsi.org/Reference>.

- [1] ETSI TR 101 301: "Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON) Release 4; Release Definition".
- [2] Void.
- [3] ETSI TS 101 329-2: "Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON) Release 3; End-to-end Quality of Service in TIPHON Systems; Part 2: Definition of Speech Quality of Service (QoS) Classes".
- [4] ETSI TS 101 315: "Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON) Release 4; Flow and Reference Point Definitions; Guidelines for application of TIPHON functional architecture to inter-domain services".
- [5] ITU-T Recommendation E.164: "The international public telecommunication numbering plan".
- [6] IETF RFC 2543: "SIP: Session Initiation Protocol".
- [7] ITU-T Recommendation H.323: "Packet-based multimedia communications systems".
- [8] ETSI TS 101 882 (all parts): "Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON) Release 4; Protocol Framework Definition".
- [9] ITU-T Recommendation Z.120: "Message sequence chart (MSC)".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

bearer: logical association of the entities in an IP streamed media application and corresponding transport network which, together, create an appropriate end-to-end media flow for no longer than the duration of a particular call

service: set of telecommunication-related tasks performed for a customer by a Service Provider and supplied in a business context

user identifier: information that enables an end user or access to be uniquely known

user profile: service specific information about a user of a service application

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

HMSC	High-level Message Sequence Chart
ICF	Inter-Connect Function
ISDN	Integrated Services Digital Network
ITU	International Telecommunications Union
MSC	Message Sequence Chart
NGN	Next Generation Network
SC	Service Capability
SDL	Specification and Description Language
SIP	Session Initiation Protocol
TIPHON	Telecommunications and Internet Protocol Harmonization Over Networks
UML	Unified Modelling Language

4 Services and service capabilities

TIPHON standardizes modules of technical functionality that can be implemented using different technologies and terms these as Service Capabilities.

A service application is the technical description which, when combined with a commercial arrangement, specifies a service.

A service capability is a specified set of functionalities that is used to provide a component part of a service application. A single service capability may be used in more than one service application.

A service capability is further defined (for the purposes of the present document) as an operation within one of these root classes or any derived class.

5 Naming

TIPHON Release 4 for the support of Public Telephony shall use naming based on the use of numbering plans derived from the application of ITU-T Recommendation E.164 [5].

EXCEPTION: Numbering schemes used in private networks.

EXCEPTION: Service invocation codes, e.g. emergency number.

6 Overall UML class model

Analysis of the suite of services required in TIPHON Release 4 and any current provision of those services in the ISDN or in the Internet (for SIP [6] and ITU-T Recommendation H.323 [7]), suggest that the following set of UML classes can be used to define TIPHON.

- Profile:
 - The profile class specifies the attributes and operations required to support operations on the user profile or service profile (e.g. registration, authentication).
- Call:
 - The call class specifies the attributes and operations required for the establishment, modification and clearing of a multi-media connection between two users.

- Bearer:
 - The bearer class specifies the attributes and operations required for control of connection based service domain operations.
- Media:
 - The media class specifies the attributes and operations required to enable communications payload encoding and characterization.
- Message:
 - The message class specifies those attributes and operations required for control of message handling including storage, retrieval and maintenance, in the service domain.

In addition the following object types are defined for TIPHON:

- Transport:
 - The transport object is manipulated by the media and bearer classes.

Two stereotypes are defined in TIPHON:

- <<sc>> = Service capability;
- <<return>> = Notification or return message.

It is noted that the <<sc>> stereotypes are typically invoked and run asynchronously and many <<return>> operations may be invoked prior to completion of a single <<sc>> operation.

The identified classes are shown in figure 1.

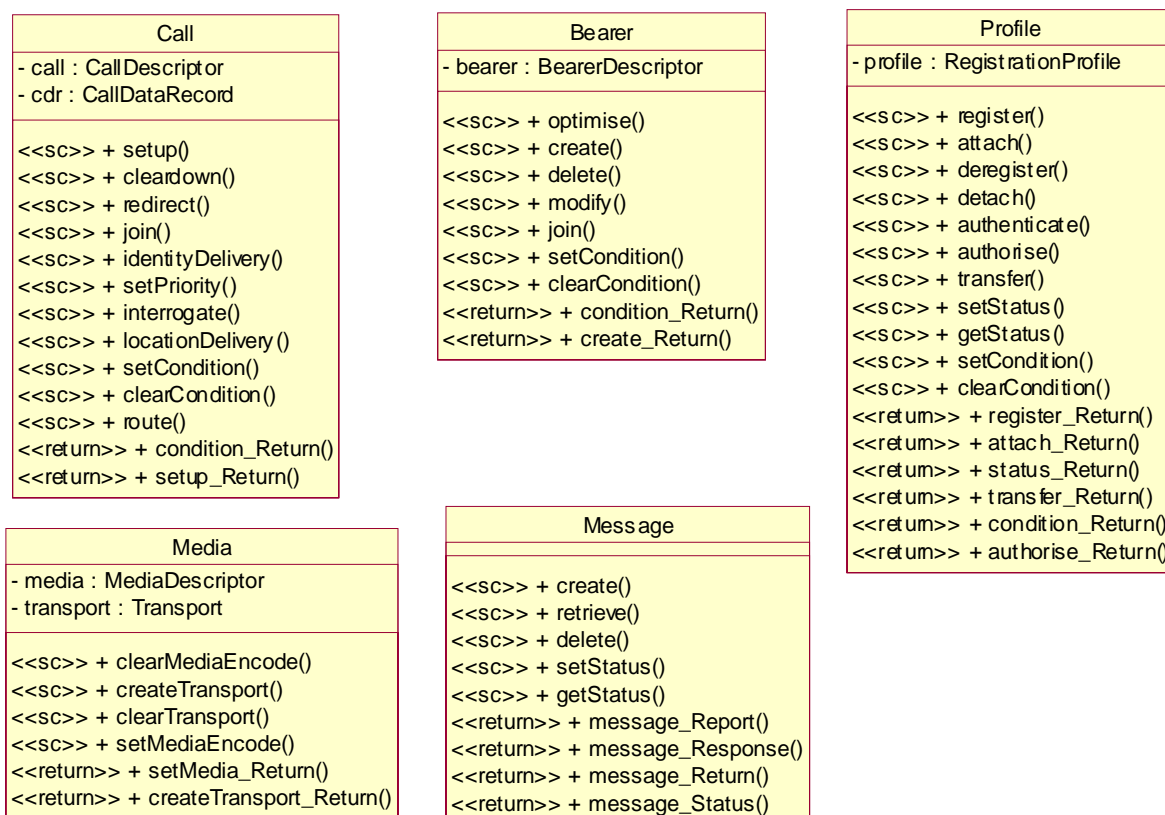


Figure 1: Overall UML class model for TIPHON (NGN) Service Capabilities

6.1 Other entities in the TIPHON SC Model

6.1.1 TIPHON User

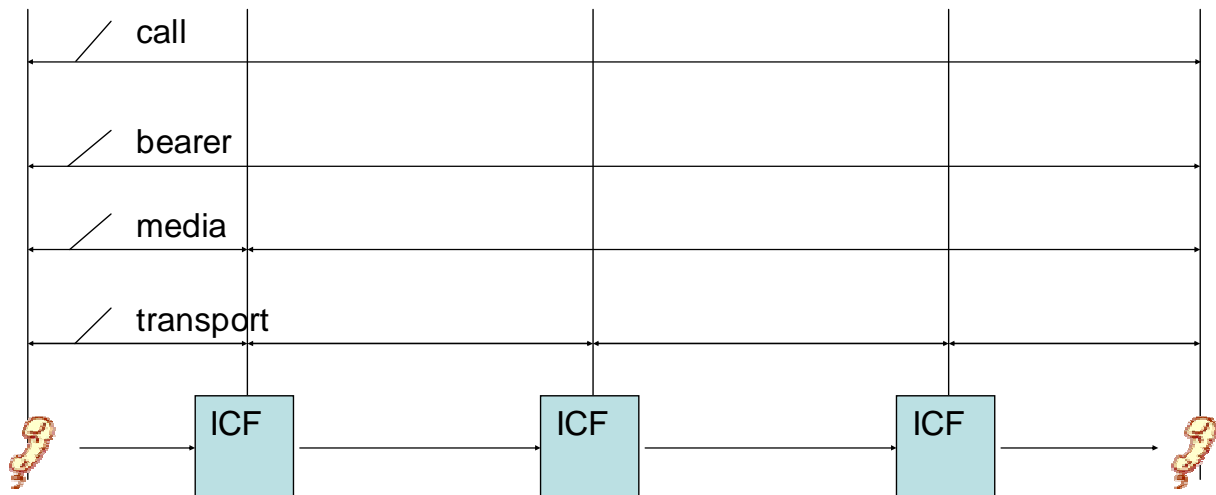
The TIPHON user is that entity, generally outside of the TIPHON model and whose state is not recorded in the model, that initializes and terminates transactions such as telephone calls.

NOTE: Although there are signals required to be sent to the user these do not form capabilities as defined in other parts of the present document. Therefore other than recognizing the existence of the TIPHON user as an element in the system that uses service capabilities no specification of the TIPHON user is given.

A stereotype <<userNotice>> may be required to allow notification of the user of the state of operations in the other parts of the service capability model.

6.2 Scope of service capabilities with respect to user and network

The figures that follow show, for a network connection between two users, the scope of each of the classes defined in the present document.



NOTE 1: Call is defined end-to-end.

NOTE 2: Bearer is the abstraction of all transport and media encoding seen by the call.

NOTE 3: Media objects are carried nominally end-to-end but a single call may have many media objects serially concatenated across transcoding points.

NOTE 4: Transport objects exist within a single transport domain between terminal points (either user-terminals or Inter-Connect Functions (ICFs)).

Figure 2: Scope of classes with respect to operations

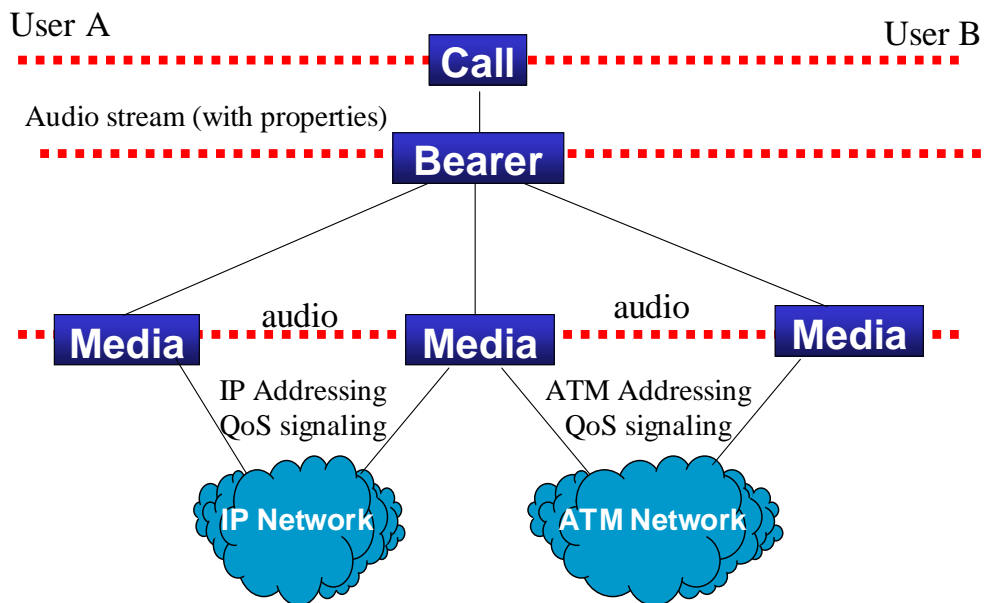
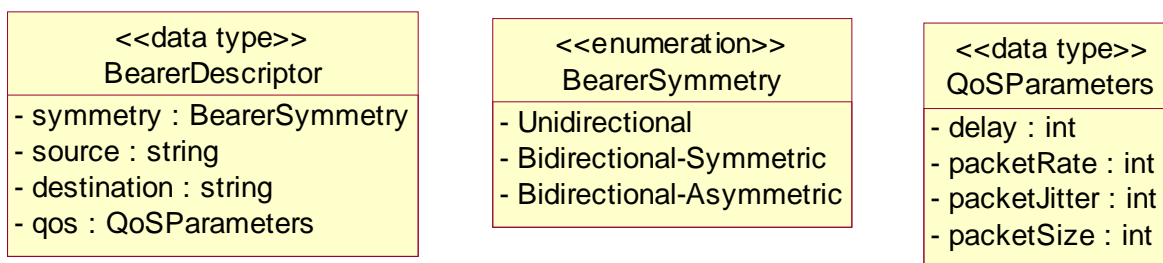
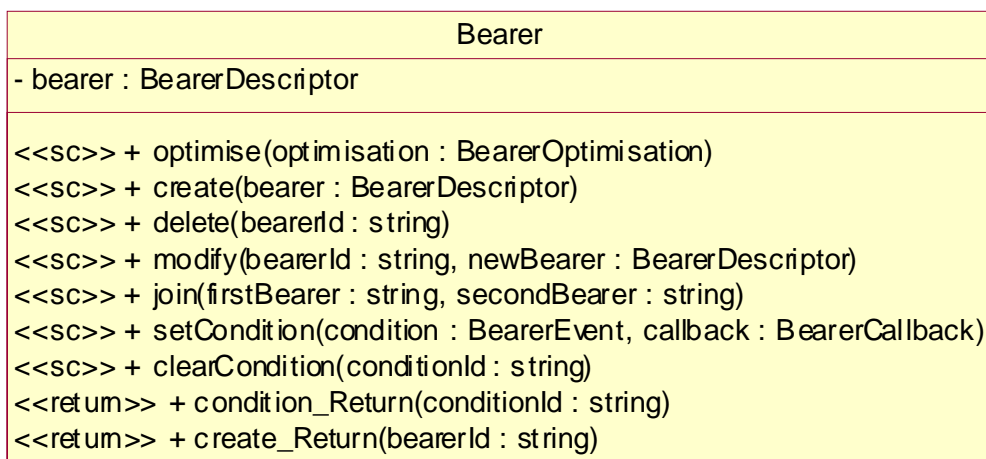


Figure 3: Scope of classes with respect to operations, alternative view

7 Bearer class service capabilities

The bearer service capability suite contains those capabilities required to manage a transport plane connection, or set of transport plane connections (in the case of asymmetric and multi-point calls).

The bearer class specifies the attributes and operations required for control of connection based service domain operations.



NOTE 1: Service capabilities are shown in the class diagram as stereotypes of type <<sc>>.

NOTE 2: Some of the elements in the enumerated types may not be available in TIPHON-R4 but are provided for information.

Figure 4: Class diagram for bearer

7.1 Bearer attribute (data element) definitions

The bearer class contains a single attribute:

BearerDescriptor: Identifies the QoS parameters, bearer symmetry and the source and destination end-points for the bearer object.

This attribute is a complex data structure that identifies in addition the following sub-attributes:

BearerType: Identifies the format of the bearer. This may be one of the following:

- Unidirectional;
- Bidirectional symmetric;
- Bidirectional asymmetric.

NOTE: In TIPHON R4 where a simple call is created the bearerType is by default Bidirectional symmetric.

bearerIdentity: Assigned during the create() operation and used as a unique identifier in the delete() and join() operations.

qosParameters: as defined in TS 101 329-2 [3], details the QoS in terms of delay, packet/bit rate, packet/bit loss rate, packet jitter, and the integrity requirement.

In order to allow the generation of complex functions one additional data type is defined:

BearerEvent: An enumerated type that indicates the condition of the bearer.

7.2 Service capabilities

7.2.1 Create

Description: Operation that establishes a bearer (transport entity (*or object*)) between two points for the purpose of carrying encoded user data. The creation of the bearer may take into account one or more optimization parameters.

Preconditions: None.

Post-conditions: A handle returned (bearerIdentity) which can be used to uniquely identify the selected bearer in subsequent operations.

Parameters: Bearer characteristics.

7.2.2 Modify

Description: Operation that modifies an established bearer.

Preconditions: An established bearer.

Post-conditions: None.

Parameters: Bearer characteristics (modified)

7.2.3 Delete

Description: Operation that clears a bearer previously established and identified by its bearer identity.

Preconditions: Bearer-id indicating pre-existing bearer exists.

Post-conditions: No bearer exists.

Parameters: Bearer-id.

7.2.4 Join

Description: Capability that allows two or more bearers sharing a common end-point to be joined at that end-point.

Preconditions: 2 or more pre-existing bearers.

Post-conditions: 1 bearer exists as combination of the pre-existing bearers.

Parameters: Bearer-ids.

7.2.5 setCondition

Description: Sets a trigger based upon a condition related to a monitored bearer class. A specified service capability is invoked when the condition is met.

Preconditions: None.

Post-conditions: Condition is set and action to be taken recorded, Event-handle returned.

Parameters: Trigger, Callback (Service Capability to be invoked and service capability specific parameters, Operation that triggered the event).

7.2.6 clearCondition

Description: Clears a previously set condition.

Preconditions: Condition is set.

Post-conditions: Condition is cleared.

Parameters: Event-handle.

8 Call class service capabilities

This class contains all of the operations and attributes that belong to the class of "calls" which are end-to-end operations.

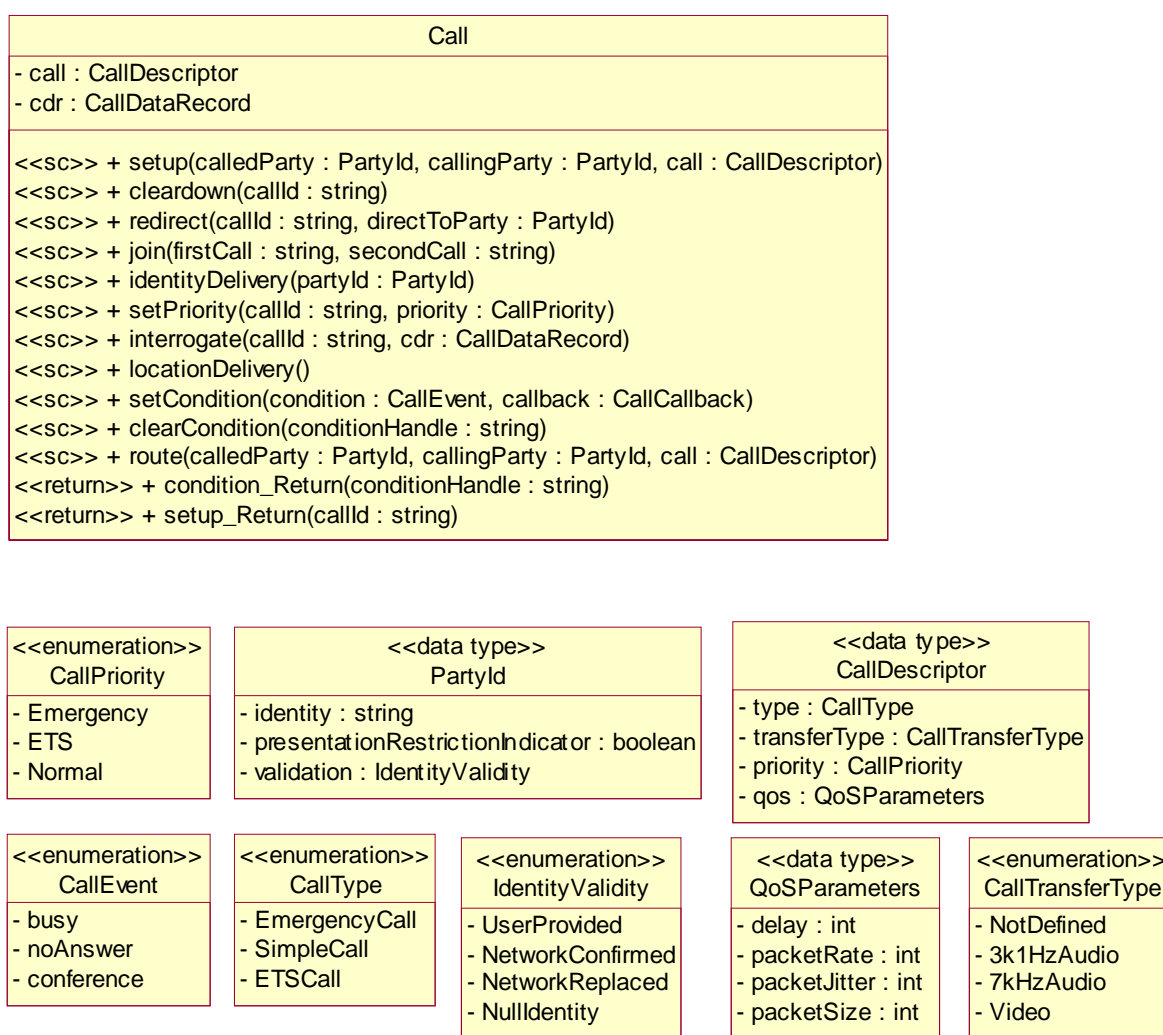


Figure 5: Class diagram for call

8.1 Call attribute (data element) definitions

callChargeRecord: Contains data relating to the current cost of a call (for use during or after a call) or to the balance available (on a pre-paid account for example).

callDataRecord: Audit or accounting record of a call (may contain identity, start/stop time).

calledUserIdentifier: Used to identify the called party. May be of numeric, alpha or alphanumeric format.

callIdentity: Returned on successful invocation of setup to act as a handle on the call.

callingUserIdentifier: Used to identify the calling party. May be of numeric, alpha or alphanumeric format.

callType: Used to identify variations of call type that may have extended or modified behaviour.

failureReason: Used to indicate in detail the reason for failure of a call.

identityPresentationRestrictionIndication: To indicate if the calling party identity should be offered to the called party.

identityAttributes: data used to qualify the identity.

priority: Used to establish priority for call processing/retention. Used primarily for emergency calls (e112) or EMTEL calls.

qosServiceClass: as defined in TS 101 329-2 [3], offered as a shorthand for user to network signalling.

redirectToAddress: If the call is being redirected using the redirect capability this attribute indicates where the call is redirected to.

returnPartyUserIdentifier: Used to replace the calling user's identity with a preferred identity for return calls (note requires identity to be presented to called party).

serviceProviderPreference: Used to indicate that a preference for service provider. May be used to join a VPN too.

8.2 Service capabilities

8.2.1 Setup

Description: Capability to establish a call (end-to-end). Returns callIdentity on success.

Preconditions: None.

Post-conditions: An end to end connection between the calling and called parties exists.

Parameters: calledUserIdentifier, callingUserIdentifier, callIdentity, callType, Service Provider Preference, QoS Service Class.

8.2.2 identityDelivery

Description: The ability to deliver to an authorized user the identity of a party involved in an establishing or established call.

Preconditions: None.

Post-conditions: None.

Parameters: User Identity, Identity Attributes, Delivery Address.

8.2.3 Redirect

Description: Capability to change one of the end-points of a call based upon an event (for example to change called party when called party is busy, to perform park and retrieve operations).

Preconditions: Call setup is in progress or call is established.

Post-conditions: Call setup is completed to the Revised Called User Address

Parameters: Call Identity, Revised Called User Address.

8.2.4 setPriority

Description: Capability to modify the priority assigned to a call (may be used to set Emergency priority on a dialled call).

Preconditions: None.

Post-conditions: The call is treated at the new priority level.

Parameters: Call Identity, Priority.

8.2.5 cleardown

Description: Capability to close the call by removing the end-to-end connection.

Preconditions: Call setup is in progress or call is established.

Post-conditions: No call exists.

Parameters: Call Identity.

8.2.6 Join

Description: Capability to join two or more calls sharing a common end-point.

Preconditions: Calls to be joined are established.

Post-conditions: One call exists as a combination of the pre-existing calls.

Parameters: Call Identity List.

8.2.7 Interrogate

Description: The ability to query the value of a user-specific attribute such as the contents of the callChargeRecord.

NOTE: The interrogate SC does not alter the data being interrogated.

Preconditions: None.

Post-conditions: None.

Parameters: Call Identity, identification of the attribute to be queried.

8.2.8 setCondition

Description: Sets a trigger based upon a condition related to a monitored call class. A specified service capability is invoked when the condition is met.

Preconditions: None.

Post-conditions: Condition is set and action to be taken recorded, Event-handle returned.

Parameters: Trigger, Callback (Service Capability to be invoked and service capability specific parameters).

EXAMPLE: A condition may be set to trigger on entering a "busy" state with the callback parameter invoking the redirect service capability as a means of simulating the ISDN supplementary service Call Forward on Busy.

8.2.9 clearCondition

Description: Clears a previously set condition.

Preconditions: Condition is set.

Post-conditions: Condition is cleared.

Parameters: Event-handle.

8.2.10 locationDelivery

Description: Capability to provide details of a registered user's geographical position.

Preconditions: None.

Post-conditions: None.

Parameters: User Identity.

8.2.11 Route

Description: used within any instance of a call to allow determination of the next call instance.

Preconditions: None.

Post-conditions: None.

Parameters: calledUserIdentifier, callingUserIdentifier, Service Provider Preference, QoS Service Class.

9 Profile class service capabilities

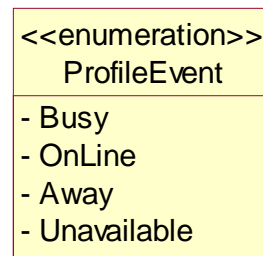
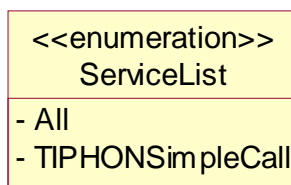
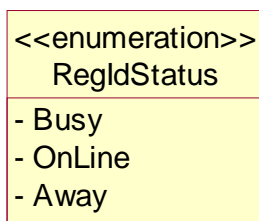
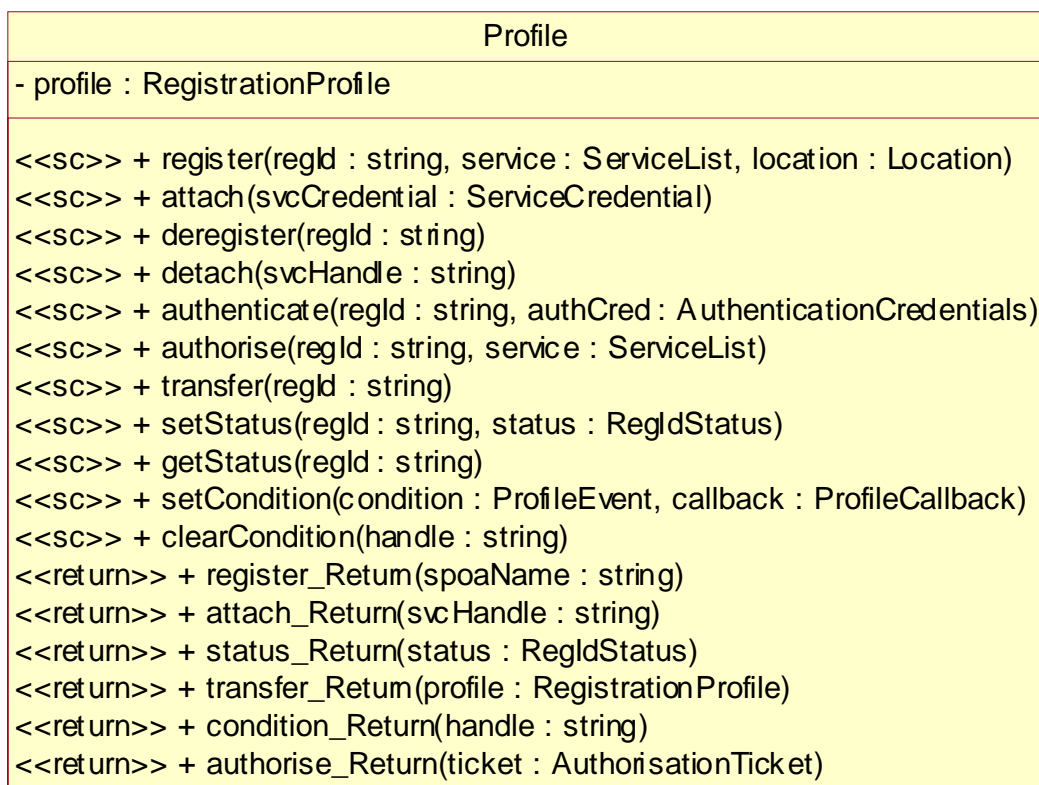


Figure 6: Class diagram for profile

9.1 Service capabilities

9.1.1 Register

Description: Capability to allow a user to request the provision of a specific service at a specific location.

Preconditions: User is provisioned within the registrar and has a profile.

Post-conditions: The identity of the service provider is returned to the user and the user is marked active in the profile.

Parameters: Registration identity, service to be registered for, location at which service is to be provided.

9.1.2 Attach

Description: Allows explicit attachment to a service provider using credentials received during registration.

Preconditions: None.

Post-conditions: User is offered service at the identified service provider.

Parameters: Service credentials.

9.1.3 Authenticate

Description: The ability to formally validate identity of a user or service provider.

Preconditions: None.

Post-conditions: The identity of the user or service provider is confirmed or, if authentication fails, not confirmed.

Parameters: Identity of entity to be authenticated, Authentication credentials.

9.1.4 getStatus

Description: Allows an authorized user to query the current registration status of a user (the requesting user or another).

Preconditions: Requesting user is authorized to request the status of the specified user.

Post-conditions: The registration status of the specified user is returned to the requesting user.

Parameters: User identity.

9.1.5 Deregister

Description: Terminates a specific registration.

Preconditions: A registration is in progress or has been completed.

Post-conditions: The user is marked as not-available in the profile.

Parameters: Registration identity.

9.1.6 Transfer

Description: The ability to request the transfer of a user's profile from one location to another.

Preconditions: None.

Post-conditions: The user's profile is established at the specified location.

Parameters: User identity, destination location.

9.1.7 Authorize

Description: The ability to establish a user's permission to use a specified service capability.

Preconditions: None.

Post-conditions: The user is allowed to invoke the requested service capability if authorization is successful or, if authorization fails, not allowed to invoke the service capability.

Parameters: User identity, service capability to be authorized.

9.1.8 setStatus

Description: Capability to change the current status of a user profile (may take a number of values including Available, Do-not-disturb, Not available).

Preconditions: None.

Post-conditions: None.

Parameters: User identity, value to be set.

9.1.9 setCondition

Description: Sets a trigger based upon a condition related to a monitored profile class. A specified service capability is invoked when the condition is met.

Preconditions: None.

Post-conditions: Condition is set and action to be taken recorded, Event-handle returned.

Parameters: Trigger, Callback (Service Capability to be invoked and service capability specific parameters).

EXAMPLE: A condition may be set to trigger on change of the user status with the callback parameter sending notification of the new state to any registered parties, this may be a means of describing the Instant Messaging watcher service.

9.1.10 clearCondition

Description: Clears a previously set condition.

Preconditions: Condition is set.

Post-conditions: Condition is cleared.

Parameters: Event-handle.

10 Message class service capabilities

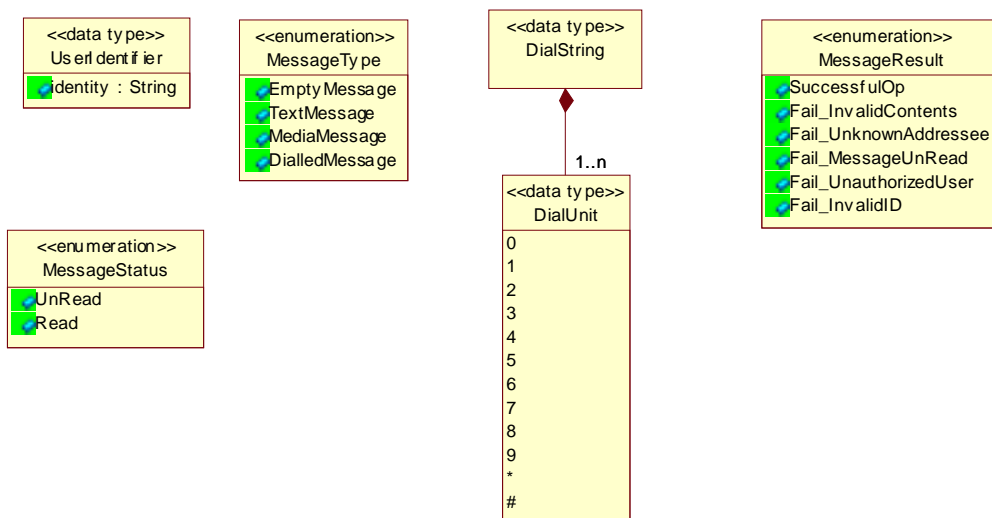
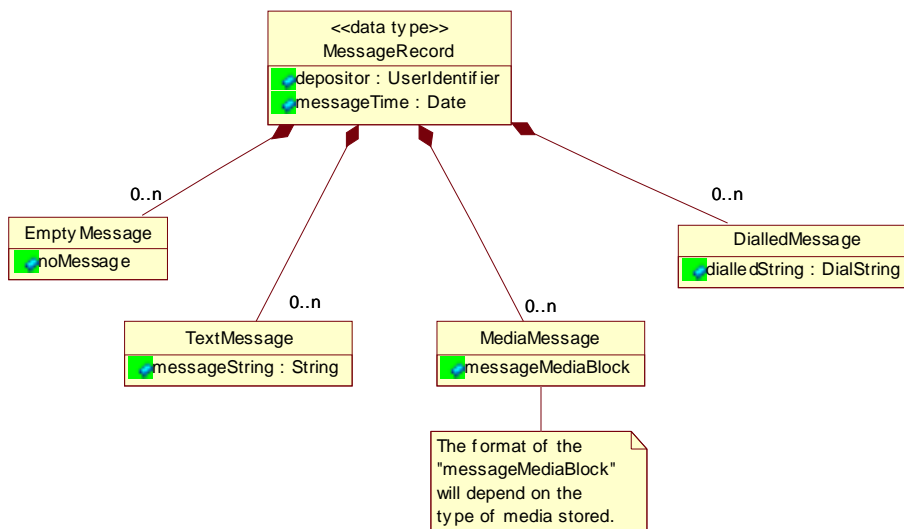
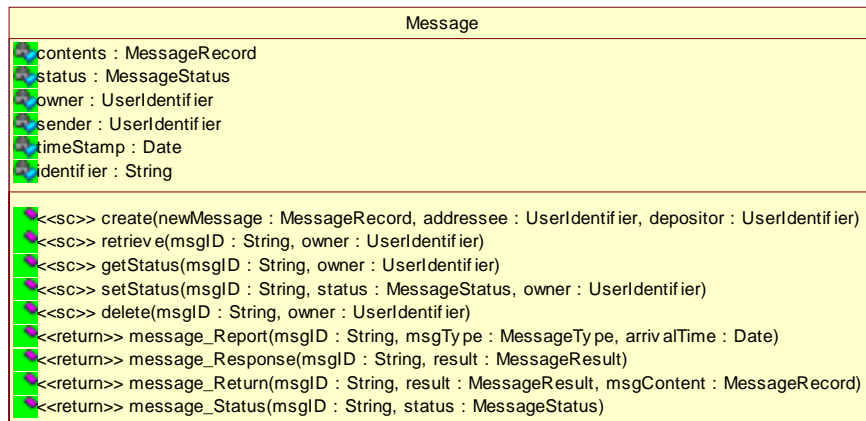


Figure 7: Message class and associated types

10.1 Message attributes

identifier: Unique string identifying each message.

contentType: Indicates the form of the message. The message may be a signal (i.e. no additional content) or may contain some form of content (e.g. text, video).

contents: Contains the data supplied as text and/or recorded media (e.g., voice) when the message is created.

owner: Identifies the user for whom the message was intended at creation (addressee).

sender: Identifies the user who creates the message.

timeStamp: Holds the date and time that the message was created.

status: Identifies whether the message has been read or not.

10.2 Service capabilities

10.2.1 Create

Description: Capability to create a new message and, if appropriate, insert the supplied text and/or media as its contents.

Preconditions: An end to end signalling association between the calling and called parties exists.

Post-conditions: A new message is created with the specified content. Details of the message (sender, content type and identifier) have been sent to the called party; the identifier has been sent to the caller.

Parameters: Message contents, message addressee (owner), message sender.

10.2.2 Retrieve

Description: Capability to deliver the message contents to the calling user.

Preconditions: An existing message.

Post-conditions: The contents of the message have been sent to the requesting user.

Parameters: Message identifier, message owner identity.

10.2.3 setStatus

Description: Capability to modify the current status of an existing message.

Preconditions: An existing message.

Post-conditions: The status is set to the value specified.

Parameters: Message identifier, message owner identity, new message status (read or unread).

10.2.4 getStatus

Description: Capability to inform the message owner of the current status of an existing message.

Preconditions: An existing message.

Post-conditions: Details of the current status of the message have been returned to the user.

Parameters: Message identifier, message owner identity.

10.2.5 Delete

Description: Capability to remove an existing message.

Preconditions: An existing message.

Post-conditions: The message is removed.

Parameters: Message identifier, message owner identity.

11 Media class service capabilities

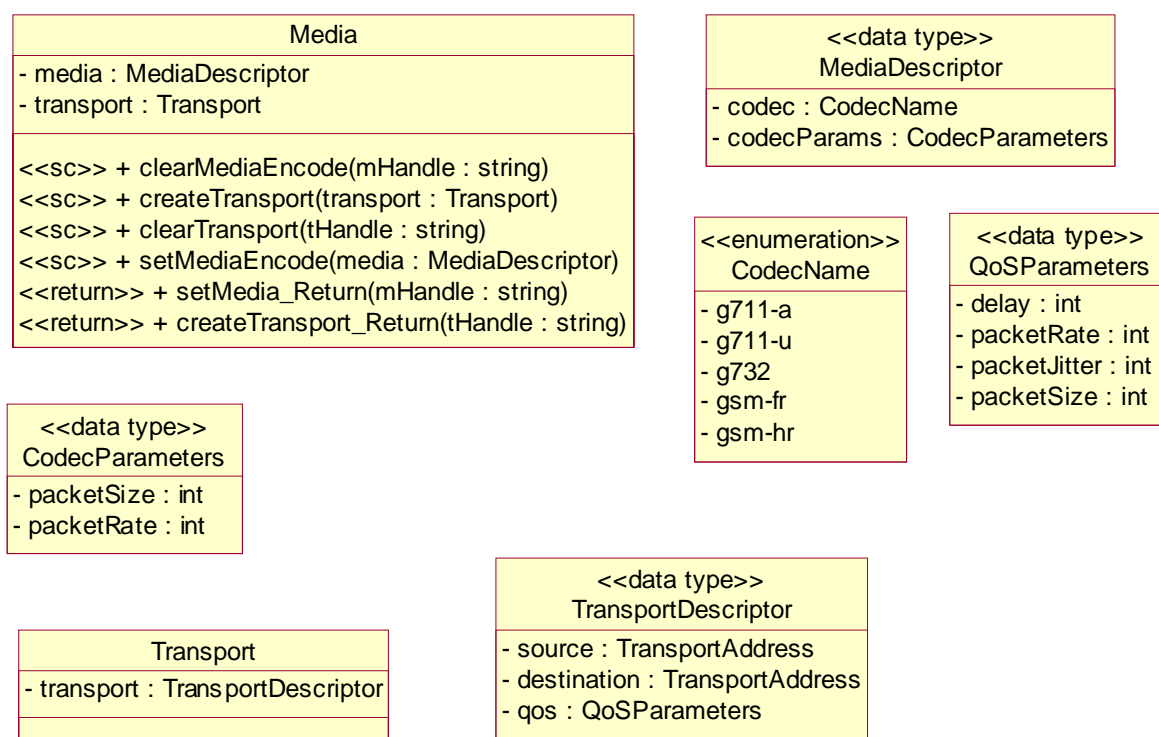


Figure 8: Extended class diagram for media service capabilities

11.1 Data attributes

mediaReport: contains the QoS parameters required to support a particular codec.

mediaType: Identifies the form of presenting information to a user, e.g. voice, fax, video.

11.2 Service capabilities

11.2.1 setMediaEncode

Description: Sets the media encoding/decoding for a particular media type. On successful operation returns the characteristics required to transport the particular media encoding.

Preconditions: None.

Post-conditions: An appropriate media-encoder for the media type is allocated and a media-handle returned.

Parameters: media type, media attributes, transport characteristics.

11.2.2 clearMediaEncode

Description: Clears a previously set media encoding/decoding.

Preconditions: Media-handle exists.

Post-conditions: Media encoding/decoding resources are cleared.

Parameters: Media-handle.

11.2.3 createTransport

Description: Establishes a transport object.

Preconditions: None.

Post-conditions: An appropriate transport object for the media type is allocated and a transport-handle returned.

Parameters: Transport descriptor.

11.2.4 clearTransport

Description: Clears a previously established transport object.

Preconditions: Transport handle exists.

Post-conditions: Transport resources are cleared.

Parameters: Transport-handle.

Annex A: (informative)

Method of developing meta-protocols from service capabilities

A.1 Overview

TIPHON builds upon the developments seen over the past 30 years or so in digital telecommunication. One of the development aims in TIPHON has been to be able to support the existing services of the PSTN/ISDN but to ensure their support over a packet based network (IP), and to also support new services and applications, with the ability to deliver new services quickly. TIPHON standardizes the capabilities required to define services but does not standardize the services themselves. A 3-thread approach to the implementation of TIPHON is shown below:

- Convergence:
 - Bringing IP and SCN together for voice services.
- Replacement:
 - Allow replacement of SCN by IP.
- Improvement:
 - Provide services not available before on the SCN with improvements in QoS and Security in particular.

In order to achieve these goals a number of design options can be considered. The method selected in TIPHON is an object oriented building block approach, where the rules for joining the blocks together are fully defined. This approach builds on the original ISDN model to some extent, and to more recent developments in computer programming and internet application design. An object engineering approach allows the adoption of a number of significant tools, the most used one being polymorphism. Polymorphism allows for inheritance of properties from a parent and the ability to modify the inherited properties. Polymorphism and overloading are often considered equivalent in that a single command can be implemented for a number of different application environments but in each case to exhibit the same abstract behaviour. A common example of this is a *print()* command which is overloaded for different printers and for different document types. In telecommunications a similar example would be a *setup()* that is overloaded for different protocols and network technologies but exhibits consistent behaviour for all. A printer from Canon might differ in detail from a printer from Epson but each inherits the properties that belong to the class of *printer*. If we assume that the *print()* command is defined in the parent *printer* class then each type of printer inherits this command and may extend it. In the telecommunications model defined in the present document we have a set of root classes (call, bearer, profile, message). An implementation of a service may inherit from one of more of these parent classes to define a service.

A.2 Service capability model

The model defined and presented in the core of the present document proposes the development of services from service capabilities, where the service capabilities are the objects referred to in clause A.1. The service capabilities are broadly defined as **operations** acting on **attributes** within a **class**. The polymorphism property that allows a single capability (e.g. *Call:Setup()*) to work for many types of data is defined in TS 101 882 [8] in the form of a meta-protocol to which real protocols are mapped for implementation.

4 classes of service capability are defined:

- **Call:** Those capabilities required to establish, clear down and modify end-to-end user sessions;
- **Bearer:** Those capabilities required to establish, clear down and modify transport resources within the network;
- **Profile:** Those capabilities required to register, authorize and authenticate a user to his service profile;
- **Message:** Those capabilities required to send messages between users (store and forward or direct to user).

A.3 From SC to Meta-protocol

Each service capability is further defined using an UML activity diagram that forms the skeleton of a High Level Message Sequence Chart (HMSC) as defined in ITU-T Recommendation Z.120 [9]. There are well defined rules for combining HMSCs that will allow service capabilities to be combined.

A meta-protocol shall be developed for each service capability as a block-type and within it a set of process-types. The use of packages as containers will map to the capability classes defined in the main body of this TS.

NOTE: The development of the SC to HMSC to SDL packages is for further study.

Annex B: (informative)

Interconnection agreements

Services are in general private offerings of a single service provider, however, the service capabilities are generic. The support of service capabilities, and the boundary conditions used in invoking those service capabilities, may be used in interconnection agreements allowing service support across service provider domains without disclosure of service logic.

History

Document history		
V1.1.1	February 2002	Publication
V4.1.1	November 2003	Publication