The drawaing contained in this Recommendation have been done in Autocad

Recommendation T.330

TELEMATIC ACCESS TO INTERPERSONAL MESSAGE SYSTEM

(Melbourne, 1988)

The establishment in various countries of telematic services and computer—based store—and—forward message service in association with public data networks creates a need to produce standards to facilitate international message exchange between subscribers to such services.

The CCITT,

considering

(a) the need for interpersonal messaging and message transfer services;

(b) the need to transfer messages of different types having a large variety of formats;

(c) that within the X Series of Recommendations services and optional user facilities for public data networks are defined;

(d) that the F Series of Recommendations defines telematic services and that the T Series of Recommendations defines terminal equipment and control procedures for telematic services;

(e) that a set of Recommendation describes various aspects of message handling systems: X.400 Series;

(f) that Recommendation T.300 describes general principles of telematic interworking,

unanimously declares

that this Recommendation describes the access protocol to be used by telematic terminals when making additional use of the interpersonal messaging system.

CONTENTS

0 *Introduction*

1 *Scope and field of application*

2 *References*

3 *Definitions*

4 *Abbreviations*

5 *Conventions*

6 *Overview of telematic access to IPMS*

6.1 Abstract model

6.2 Functional model

6.3 Access for registered and non—registered users

7 *IPMS in the context of telematic interworking*

7.1 Objects and ports description

7.2 Origination, reception and management ports, services and operations

7.3 Miscellanea port services and operations

8 *Refinement of the TLMA object*

8.1 Object and ports description

8.2 The mhs—doc—xfer port operations

9 *Abstract errors*

10 *Realization of abstract operations*

10.1 Description of TAPDU

10.2 Operation of the TLMAU

11 *Formats and coding of TAPDU*

11.1 Principles

11.2 Structure and format of TAPDU

11.3 Coding of TAPDU

11.4 Format of TAPDU

11.5 Reference between TAPDU components and its coding format

12 *Error recovery*

13 *Control procedures*

13.1 Session control procedures

13.2 Document control procedures

13.3 Log—on procedures

Annex A — Formal definition of TLMA abstract service

Annex B — Format of TAPDU components

Annex C — Element ID list

Annex D — Element of service for TTX/IPM service intercommunications

0 Introduction

Recommendation T.330 is one of a series of Recommendations dealing with telematic interworking. Telematic interworking is the generic name for a set of applications provided to telematic users. Each of those applications is called a telematic interworking application (TIA).

Access to and participating in interpersonal messaging system (IPMS) are one of the telematic interworking applications. This Recommendation aims at specifying this application.

1 Scope and field of application

This Recommendation defines the abstract service provided by the telematic agent (TLMA) which is defined as an object of IPMS. It specifies not only abstract operations provided by TLMAU but also access protocol (P5) to be used between a TLMAU and a telematic (TLM) terminal, when participating in and accessing the IPMS. The P5 access protocol is a generalized access protocol; it is applicable to other applications such as network based storage for the teletex service. The TLM terminals being considered in this Recommendation are teletex, G4 facsimile and mixed mode terminals. The use of other types of TLM terminals are for further study.

Other Recommendations in the series contain description on telematic interworking model, the functions of the telematic access unit (TLMAU), and telematic access protocol to specific services, such as telematic, telex, directory, etc. Recommendation T.300 outlines the principles of telematic interworking procedures.

Section 6 of this Recommendation defines overview of telematic access to IPMS provided by TLMA object. Section 7 defines the IPMS in the context of telematic interworking. Section 8 refines the TLMA object and defines abstract operations at a specific port of TLMAU and TLM terminal. Section 9 defines abstract errors used in telematic interworking. Section 10 specifies an access protocol (P5). Section 11 specifies formatting and coding rule of protocol. Section 12 specifies an error recovery mechanism. Section 13 specifies control procedures.

The purpose of a TLMAU is to aid the user of a TLM terminal in gaining access to the features of the IPMS. The TLMAU, which is associated with a message transfer system (MTS), provides the TLM terminal with access to the IPMS.

The TLMAU may also provide a *document storage (DS)* facility to accept delivery of messages from the MTS for TLM users. Document storage is basically defined as a TLM terminal storage extension facility located in the TLMAU allowing reservation of a specific amount of storage for an individual user. Users of TLM terminals may also be registered as users of DS.

2 References

This Recommendation cites the documents listed below.

2.1 *Telematic interworking*

— Rec. T.300: General principles of telematic interworking.

2.2 *Message handling systems*

— Rec. X.400: Message handling systems: System and service overview

— Rec. X.402: Message handling systems: Overall architecture

— Rec. X.407: Message handling systems: Abstract service definition conventions

— Rec. X.411: Message handling systems: Message transfer system: Abstract service definition and procedures

— Rec. X.413: Message handling systems: Message store: Abstract service definition

— Rec. X.419: Message handling systems: Protocol specifications

— Rec. X.420: Message handling systems: Interpersonal messaging system

2.3 *Control procedures*

— Rec. T.62: Control procedures for Teletex and Group 4 facsimile services

2.4 *ASN.1 coding*

— Rec. X.208: Specification of abstract syntax notation one (ASN.1)

— Rec. X.219: Remote operation

2.5 *Address*

— Rec. X.121: International numbering plan for public data networks

2.6 *Character repertoires*

— Rec. T.61: Character repertoire and coded character sets for the international Teletex service

2.7 *Intercommunication*

— Rec. F.422: Intercommunication between Teletex service and IPM service.

— Rec. F.203: Network based storage for the Teletex service.

3 Definitions

This Recommendation uses the terms many of those used in Recommendations X.402, X.411 and X.420.

In addition to the above terms, this Recommendation uses as terms the names of abstract objects, ports, operations and errors; the names of ASN.1 data types; the names of the information item types and values this Recommendation specifies.

4 Abbreviations

ASN.1	Abstract syntax notation one
AU	Access unit
C	Conditional/consumer
CDC	Command document continue
CF	Conversion facility
CSCC	Command session change control

CSS	Command session start
DN	Delivery status notification
DS	Document storage
G3	Group 3 facsimile
G4	Group 4 facsimile
ID	Identity
IP	Interpersonal
IPM	Interpersonal messaging
IPMAS	Interpersonal messaging abstract service
IPME	Interpersonal messaging environment
IPMS	Interpersonal messaging system
IPM—UA	Interpersonal messaging user agent
IPN	Interpersonal notification
M	Mandatory
MS	Message store
MT	Message transfer
MTA	Message transfer agent
MTAS	Message transfer abstract service
MTS	Message transfer system
NDN	Non—delivery status notification
NL	New line
NRN	Non—receipt notification
O/R	Originator/receipt
PDAU	Physical delivery access unit
PTTXAU	Public Teletex access unit
P5	Telematic access protocol
RN	Receipt status notification
S	Supplier
TAPDU	Telematic access protocol data unit
TIA	Telematic interworking application
TID	Terminal identification
TLM	Telematic
TLMA	Telematic agent
TLMAU	Telematic access unit
TLM—TER	Telematic terminal
TLXAU	Telex access unit
TTX	Teletex
UA	User agent

5 Conventions

This Recommendation uses the descriptive conventions identified below.

5.1 *ASN.1*

This Recommendation uses the following ASN.1—based descriptive conventions for the indicated purposes:

a) to specify the functional objects, the OBJECT and REFINE macros and associated conventions of Recommendation X.407;

b) to specify the information objects (and other data types and values of all kinds), ASN.1 itself;

c) to specify the abstract service, the PORT and ABSTRACT—BIND, —UNBIND, —OPERATION, and —ERROR macros and associated conventions of Recommendation X.407.

5.2 *Grade*

Whenever this Recommendation describes a class of data structure (e.g. Headings) having components (e.g. fields), each component is categorized as one of the following grades:

a) Mandatory (M): A mandatory component shall be present in every member of the class.

b) Conditional (C): A conditional component shall be present in a member of the class as dictated by this Recommendation.

6 Overview of telematic access to IPMS

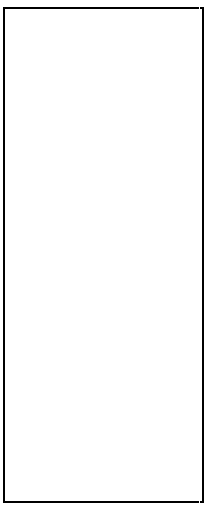
6.1 *Abstract model*

This Recommendation makes use of the message handling abstract service definitions conventions defined in Recommendation X.407. These conventions provide a descriptive tool for the specification of information processing tasks in abstract terms. This ensures that a tasks functional requirements are stated independently of its realization.

6.2 *Functional model*

This section provides a functional model of telematic access to IPMS. The purpose of this model is to provide a general description of the functional entities, which are then explicitly defined using the definitions and conventions found in Recommendation X.407, and further refined as necessary, in following sections (see Figure 1/T.330).

Fig. 1/T.330/T0803980-89 = 7 cm



The functional model comprises the following functional entities:

— Telematic agent (TLMA): Logical entity only which comprises the TLMAU and the telematic terminal. The TLMA is useful as an object in the refinement of the IPMS.

— Telematic access unit (TLMAU): Functional entity which provides all of the interworking functions between telematic codes and protocols and IPMS codes and protocols. The TLMAU also supports the DS functionality.

— Telematic terminal (TLM—TER): The telematic terminal.

— Access unit (AU): Functional entity which provides access to message handling applications for indirect users of the MTS.

— Document storage (DS): Extension of the telematic terminal storage capabilities. The TLMAU may optionally, on a subscription basis, deliver messages to a DS. The terminal may then retrieve the message for the document storage when convenient.

— Message store (MS): Functional entity which provides single direct user of message handling with capabilities for message storage. Although the MS and DS provide a similar functionality, there is no relationship between the two.

— Message transfer system (MTS): Functional entity which conveys information objects between individual users and members of distribution lists.

— User agent (UA): Functional entity by means of which a direct user engages in message handling.

6.3 *Access for registered and non—registered users*

Two types of access to the IPM service are defined within this Recommendation. Registered users of the IPM service who wish to use telematic terminal equipment to access the IPM service are provided with complete IPM service functionality with any full implementation of this Recommendation.

Telematic terminal equipment users who are not registered IPM service subscribers but who wish to direct a message to an IPM service user are provided with a subset of the functionality defined within this Recommendation, in accordance with Recommendation F.422 and Annex D of this Recommendation. This functionality is referred to as a public teletex access unit (PTTXAU).

7 IPMS in the context of telematic interworking

7.1 *Objects and ports description*

The refinement of the IPMS is found in Recommendation X.420 (interpersonal messaging system). The IPMS refinement describes secondary objects, one of which is the telematic agent (TLMA) which is paired to the MTS by the import and export ports.

The TLMA is visible to the telematic user through four ports, namely: origination, reception, management and miscellanea. The origination, reception and management port services and operations are described fully in Recommendation X.420. The miscellanea port services and operations are described in this Recommendation. The import and export port services and operations are described in Recommendation X.411.

```
tlma                OBJECT
        PORTS { origination     [S],
                reception       [S],
                management      [S],
                miscellanea     [S],
                import          [C],
                export          [C] }
        ::= id—ot—tlma
```

The IPMS comprises any number of TLMA.

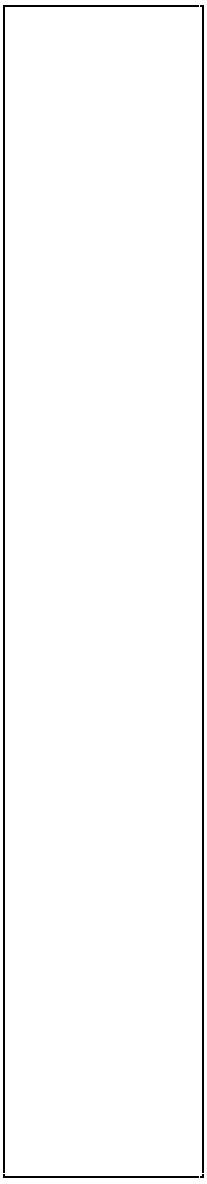
TLM users are communicants in telematic interworking. A TLM user originates or receives information objects whose types are specified in Recommendation X.420 and this Recommendation.

```
tlma-user               OBJECT
          PORTS { origination       [C],
                  reception         [C],
                  management        [C],
                  miscellanea       [C] }
          ::= id—ot—tlm—user
```

A telematic user is associated with the TLMA by means of the origination, reception, management and miscellanea ports. A telematic user is a supplier [S] of no ports and a consumer [C] of all TLMA ports. The TLMA is a supplier of all TLMA ports and consumer of no ports.

The general access to IPMS is illustrated in Figure 2/T.330.

Fig. 2/T.330/T0803990-89 = 18



An interpersonal messaging user agent (IPM—UA) is a secondary object that provides the interpersonal messaging abstract service (IPMAS) to a single IPM user. An IPM—UA is a specialized instance of the more general object, UA. An IPM—UA performs its function with help from the MTS.

A telematic agent (TLMA) is an object that provides the abstract service which comprises IPMAS and telematic specific abstract service, to a single TLM user. A TLMA is an instance of the more general object UA. A TLMA performs its function with help from the MTS.

A message transfer system (MTS), upon which all other IPMS components relay, is the provider of the message transfer abstract service (MTAS). It performs its function without assistance.

An interpersonal messaging system (IPMS) is the object by means of which all users communicate in interpersonal messaging.

The access unit (AU) could be a physical delivery access unit (PDAU), or telex access unit (TLXAU). The descriptions of these objects found in relevant Recommendations.

7.2 *Origination, reception and management ports, services and operations*

The abstract operations available at these ports, as described in X.420, are:

```
origination PORT
        CONSUMER INVOKERS { OriginateProbe,
                            OriginateIPM,
                            OriginateRN,
                            CancelIPM }
                ::= id—pt—origination

reception PORT

        CONSUMER INVOKERS { ReceiveReport,
                            ReceiveIPM,
                            ReceiveRN,
                            ReceiveNRN }
                ::= id—pt—reception

management PORT

        CONSUMER INVOKERS { ChangeAutoDiscard,
                            ChangeAutoAcknowledgment,
                            ChangeAutoForwarding }
                ::= id—pt—management
```

The abstract operations are fully described in Recommendation X.420.

7.3 *Miscellanea port services and operations*

Besides IPM abstract services, the following abstract services are available at the miscellanea port. They are provided by the TLMA object as the miscellanea abstract services.

```
miscellanea PORT

        SUPPLIER PERFORMS { ChangeSubscriptionProfile,
                            DSList,
                            DSDelete,
                            DSFetch,
                            MessageStatus }
                ::= id—pt—miscellanea
```

7.3.1 *ChangeSubscriptionProfile*

The ChangeSubscriptionProfile abstract operation enables a user to change the registered subscription profile which specifies relationship with the TLMAU, such as DS mode, error recovery mode and message delete mode.

```
ChangeSubscriptionProfile ::= ABSTRACT—OPERATION

        ARGUMENT SET { ds—mode              [0] DSMode OPTIONAL,
                       error—recovery—mode  [1] ErrorRecoveryMode OPTIONAL,
                       message—delete—mode  [2] MessageDeleteMode OPTIONAL }

        RESULT { }
        ERRORS { name—error,
                 ds—error,
                 subscription—profile—error }
```

7.3.1.1 *Arguments of ChangeSubscriptionProfile*

This abstract operation has the following arguments:

a) DS—mode (C): The document storage mode to be applied. One of the following values:

1) retrieval: In the mode, the TLMAU holds the messages in the DS until they are explicitly deleted by the user;

2) auto output: In this mode, the TLMAU tries to output messages under user subscribed conditions after they are delivered to the DS.

b) Error—recovery—mode (C): This mode, whose recovery mechanism is defined in § 12 of this Recommendation has to be applied. (Recovery—1, 2 or 3.)

c) Message—delete—mode (C): Mode to be applied. One of the following values:

1) auto delete: In this mode, the messages in the DS are deleted as soon as they are output to the user by the performance of the DS fetch abstract operation with no delete—after—output argument (in case of retrieval mode), or by the automatically output (in case of auto—output mode);

2) manual delete: In this mode, the messages in the DS are held until the DS delete abstract operation or DS fetch abstract operation whose delete—after—output argument is "delete after output", will be carried out.

7.3.1.2 *Results of ChangeSubscriptionProfile*

This abstract operation has no results.

7.3.1.3 *Error of ChangeSubscriptionProfile*

This abstract operation has name—error, ds—error and subscription—profile error. These abstract errors are commonly described in § 9.

7.3.2 *DSList*

The DSList abstract operation enables a user to get a list of messages (IPMs, IPNs or reports) currently held in the document storage (DS).

```
DSList          ::= ABSTRACT—OPERATION

                ARGUMENT { }
                RESULT SET { [0] SET OF ListReport OPTIONAL }
                ERRORS { subscription—error,
                         name—error,
                         ds—error }

ListReport      ::= SET { retrieval—id          [0] RetrievalIdentifier,
                          message—type          [1] MessageType,
                          priority              [2] Priority OPTIONAL,
                          message—length        [3] MessageLength OPTIONAL,
                          originator—name             [4] OrName OPTIONAL }
```

7.3.2.1 *Argument of DSList*

This abstract operation has no argument.

7.3.2.2 *Results of DSList*

This abstract—operation has the following results:

a) List—report: The characteristics of message held in DS.

1) Retrieval—id (M): The retrieval—id assigned to the message in DS.

2) Message—type (M): The type of message (IPM, RN, NRN or report).

3) Priority (C): The priority of the message (normal, non—urgent or urgent).

4) Message—length (C): The length of the message in octet.

5) Originator—name (C): The originator name of the message.

7.3.2.3 *Errors of DSList*

This abstract operation has subscription—error, name—error and ds—error. These abstract errors are described in § 9.

7.3.3 *DSDelete*

The DSDelete abstract operation enable a user to delete one or more specified messages in DS.

```
DSDelete      ::= ABSTRACT—OPERATION

        ARGUMENT SET { selector [0] SET OF RetrievalIdentifier }
        RESULT { }
        ERRORS { subscription—error,
                 name—error,
                 ds—error }
```

7.3.3.1 *Arguments of DSDelete*

This abstract operation has the following arguments:

a) Selector (M): The selector is the list of the retrieval—id of messages that have to be deleted.

7.3.3.2 *Results of DSDelete*

This abstract operation has no results.

7.3.3.3 *Errors of DSDelete*

This abstract operation has subscription—error, name—error and ds—error. These abstract errors are described in § 9.

7.3.4 *DSFetch*

The DSFetch abstract operation enables a user to get one or more specified messages (IPMs, IPNs or reports) from DS.

```
DSFetch          ::= ABSTRACT—OPERATION

        ARGUMENT SET OF { retrieval—id                  [0] RetrievalIdentifier,
                          delete—after—output           [1] DeleteAfterOutput OPTIONAL }
        RESULT SET { retrieval—id                       [0] RetrievalIdentifier,
                     message—report                [1] MessageReport }
        ERRORS { subscription—error,
                 name—error,
                 ds—error }
```

7.3.4.1 *Arguments of DSFetch*

This abstract operation has the following arguments:

a) Retrieval—id (M): The retrieval—id assigned to the message in DS.

b) Delete—after—output (C): This value indicates whether or not the message is deleted after retrieval. If this argument does not exist, registered mode, message—delete—mode, is applied.

7.3.4.2 *Results of DSFetch*

This abstract—operation has the following results:

a) Retrieval—id (M): The retrieval—id assigned to the message that was reported.

b) Message report (M): Envelope and content of reported message IPM, RN, NRN or report), assigned by retrieval—id.

7.3.4.3 *Errors of DSFetch*

This abstract operation has subscription—error, name—error and ds—error. These abstract errors are described in § 9.

7.3.5 *MessageStatus*

The MessageStatus abstract operation enables a user to get an information on the actual status of the previously submitted IPM.

```
MessageStatus        ::= ABSTRACT—OPERATION

                         ARGUMENT SET { [0] QueryIdentifier OPTIONAL }
                         RESULT SET { report—time                    [0] DateandTime,
                                      reported—message—id            [1] MessageIdentifier,
                                                                     [2] SEQUENCE OF StatusInfo }

                         ERRORS { subscription—error,
                                  name—error,
                                  message—status—error }

QueryIdentifier      ::= CHOICE { submission—id                      [0] MessageIdentifier,
                                  correlation—info                   [1] CallIdentification }

StatusInfo           ::= SET { status                    [0] Status,
                               per—recipient—info        [1] PerRecipientReportDeliveryFields OPTIONAL }
```

7.3.5.1 *Arguments of MessageStatus*

This abstract operation has the following arguments:

a) Query—identifier (C): This identifier enables the TLMAU to identify the message whose status is being reported. Two types of query—identifiers are available:

   1) submission—id (C): The message—id of the originated message whose status wants to query, returned as a result of the OriginateIPM abstract operation;

   2) correlation—info (C): The call—identification of the originated message whose status wants to query.

7.3.5.2 *Results of MessageStatus*

This abstract operation has the following results:

a) Report—time (M): The date and time the report is made.

b) Message—id (M): The message—identifier of the originated message whose status is being reported, returned as a result of the OriginateIPM abstract operation.

c) Status—info (M): The status information of previously submitted messages.

   1) Status: The status of the previously submitted IPM (in—process, delivered or non—delivered).

   2) Per—recipient—info: Information about subject—message's status with respect to particular intended—recipients. A sequence of MTS per—recipient—field items, one for each recipient. This component does not exist until status component become delivered or non—delivered.

7.3.5.3 *Errors of MessageStatus*

This abstract operation has subscription—error, name—error and message—status—error. These abstract errors are described in § 9.

8 Refinement of the TLMA object

8.1 *Object and ports description*

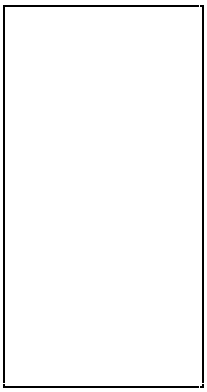
In this Recommendation, the TLMA is refined further into secondary objects namely: the TLMA and the TLM—TER object.

```
tlma—refinement REFINE tlma AS
 tlmau      mhs—doc—xfer          [S] PAIRED with { tlm—ter }
 tlm—ter    origination           [S] VISIBLE
            reception             [S] VISIBLE
            management            [S] VISIBLE
            miscellanea           [S] VISIBLE
            ::= id—ref—secondary
```

The mhs—doc—xfer is a port that enables the interaction of the TLM—TER and the TLMAU.

Figure 3/T.330 illustrates refinement of TLMA.

Fig. 3/T.330/T0804000-89 = 5 cm



A telematic access unit (TLMAU) is a secondary object to the TLMA object. It provides a TLM—TER with access to any TLM user within the interpersonal messaging environment. (IPME: see Recommendation X.420.)

The TLM—TER is a secondary object to the TLMA object.

TLM—TERs are communicants in telematic interworking. A TLM—TER sends or receives documents, embodying information objects whose types are specified in Recommendation X.420 and this Recommendation.

TLM—TER shall be addressable by at least a Network address (see Recommendation X.402), and may also be addressed by one or more other forms of ORName.

```
tlm—ter          OBJECT
          PORTS { origination        [S],
                  reception          [S],
                  management         [S],
                  miscellanea        [S],
                  mhs—doc—xfer       [C] }
          ::= id—ot—tlm—ter

tlmau            OBJECT
          PORTS { mhs—doc—xfer       [S],
                  import             [C],
                  export             [C] }
          ::= id—ot—tlm—user
```

The TLMA comprises one TLM terminal and one TLMAU.

8.2 *The mhs—doc—xfer port operations*

The following abstract operations are available at the mhs—doc—xfer port. The correspondence between mhs—doc—xfer port abstract operations and IPMS ports plus telematic specific port abstract operations are described in Table 1/T.330.

In this Recommendation TLM terminals implicitly bind a certain port at the time that the session is established and implicitly unbind a certain port at the time the session is released because Recommendation T.62 session procedure does not have association control.

```
mhs—doc—xfer PORT
          SUPPLIER PERFORMS { MessageSend,
                              MessageProbe,
                              ExplicitReceive,
                              MessageCancel,
                              Register,
                              DSList,
                              DSDelete,
                              DSFetch,
```

MessageStatus }
CONSUMER PERFORMS { MessageDeliver,
ReceiptStatusNotice,
DeliveryStatusNotice }
::= id—pt—mhs—doc—xfer

TABLE 1/T.330

Operations of mhs—doc—xfer port

IPMS ports and telematic specific port				mhs—doc—xfer port		
Port	Abstract operation	Invoker	Performer	Abstract operation	Invoker	Performer
Origination	(1) OriginateIPM (2) OriginateProbe (3) OriginateRN (4) CancelIPM	TLM—User	TLM—TER	(1) MessageSend (2) MessageProbe (3) ExplicitReceive (4) MessageCancel	TLM—TER	TLMAU
Reception	(1) ReceiveIPM (2) ReceiveRN (3) ReceiveNRN (4) ReceiveReport	TLM—TER	User	(1) MessageDeliver (2) ReceiptStatus-Notice (3) ReceiptStatus-Notice (4) DeliveryStatus-Notice	TLMAU	TLM—TER
Management	(1) ChangeAutoDis-card (2) ChangeAutoAc-know ledgment (3) ChangeAutoFor-warding	TLM—User	TLM—TER	(1) Register (2) Register (3) Register	TLM—TER	TLMAU
Miscellanea	(1) ChangeSubscrip-tionProfile (2) DSList (3) DSDelete (4) DSFetch (5) MessageStatus	TLM—User	TLM—TER	(1) Register (2) DSList (3) DSDelete (4) DSFetch (5) MessageStatus	TLM—TER	TLMAU

8.2.1 *MessageSend*

MessageSend is the abstract operation at mhs—doc—xfer port that is invoked by TLM terminal to perform OriginateIPM abstract operation at TLM terminal. This abstract operation is used to submit the IPM from TLM terminal to TLMAU.

The description of OriginateIPM abstract operation is in Recommendation X.420.

8.2.2 *MessageProbe*

MessageProbe is the abstract operation at mhs—doc—xfer port that is invoked by TLM terminal to perform OriginateProbe abstract operation at TLM terminal. This abstract operation is used to determine whether or not this IPM could be delivered to one or more recipients.

The description of OriginateProbe abstract operation is in Recommendation X.420.

8.2.3 *ExplicitReceive*

ExplicitReceive is the abstract operation at mhs—doc—xfer port that is invoked by TLM terminal perform OriginateRN abstract operation at TLM terminal. This abstract operation is used to be originated by the actual—recipient of the subject IPM of whom RN is requested by means of notification—requests component of the subject IPM's recipient—specification.

The description of OriginateRN abstract operation is in Recommendation X.420.

8.2.4 *MessageCancel*

MessageCancel is the abstract operation at mhs—doc—xfer port that is invoked by TLM terminal to perform CancelIPM abstract operation at TLM terminal. This abstract operation is used to cancel if it can the delivery of previously originated message whose content is an IPM and for which deferred delivery was requested. There is no result in MessageCancel abstract operation.

The description of CancelIPM abstract operation is in Recommendation X.420.

8.2.5 *MessageDeliver*

MessageDeliver is the abstract operation at mhs—doc—xfer port that is invoked by TLMAU to perform ReceiveIPM at TLM terminal. This abstract operation is used to deliver the IPM from TLMAU to TLM terminal. There is no result or error in MessageDeliver abstract operation.

The description of ReceiveIPM abstract operation is in Recommendation X.420.

8.2.6 *ReceiptStatusNotice*

ReceiptStatusNotice is the abstract operation at mhs—doc—xfer port that is invoked by TLMAU to perform ReceiveRN or ReceiveNRN abstract operation at TLM terminal. This abstract operation is used to report the IPN that was invoked by an IPM originated by means of the MessageSend abstract operation. There is no result or error in ReceiptStatusNotice abstract operation.

The description of ReceiveRN or ReceiveNRN abstract operation is in Recommendation X.420.

8.2.7 *DeliveryStatusNotice*

DeliveryStatusNotice is the abstract operation at mhs—doc—xfer port that invoked by TLMAU to perform ReceiveReport abstract operation at TLM terminal. This abstract operation is used to deliver the DN that was invoked by a IPM originated by means of the MessageSend abstract operation. There is no result or error in DeliveryStatusNotice abstract operation.

The description of ReceiveReport abstract operation is in Recommendation X.420.

8.2.8 *Register*

Register is the abstract operation at mhs—doc—xfer port that is invoked by TLM terminal to perform all management port's abstract operations and ChangeSubscriptionProfile mode abstract operation. This abstract operation is used to register or change the parameters that will be kept on the parameter list of TLMAU.

The description of all management port's abstract operations is in Recommendation X.420 and ChangeSubscriptionProfile abstract operation found in § 7.3.1 of this Recommendation.

8.2.9 *DSList*

DSList is the abstract operation at mhs—doc—xfer port that is invoked by TLM terminal to perform DSList abstract operation at TLM terminal. This abstract operation is used to request the status list of a previously delivered IPMs, RNs, NRNs or reports.

The description of DSList abstract operation is in § 7.3.2 of this Recommendation.

8.2.10 *DSDelete*

DSDelete is the abstract operation at mhs—doc—xfer port that is invoked by TLM terminal to perform DSDelete abstract operation at TLM terminal, and is used to delete one or more messages from the DS. There is no result in DSDelete abstract operation.

The description of DSDelete abstract operation is in § 7.3.3 of this Recommendation.

8.2.11 *DSFetch*

DSFetch is the abstract operation at mhs—doc—xfer port that is invoked by TLM terminal to perform DSFetch abstract operation, and is used to fetch one specified message (IPM, RN, NRN or report), from the DS.

The description of DSFetch abstract operation is in § 7.3.4 of this Recommendation.

8.2.12 *MessageStatus*

MessageStatus is the abstract operation at mhs—doc—xfer port that invoked by TLM terminal to perform MessageStatus abstract operation. This abstract operation is used to know the status of previously submitted IPM by means of MessageSend abstract operation.

The description of MessageStatus abstract operation is in § 7.3.5 of this Recommendation.

9 Abstract errors

The abstract errors that may be reported in response to the invocation of abstract operations at the IPM's origination, reception and management ports are subscription error, name error and cancellation error, and in miscellanea port, subscription profile error, DS error and message status error. They are defined and described in the present section.

a) Subscription error

The subscription error abstract error reports that the user has not subscribed to one or more of the element of service implicit in his invocation of the abstract operation when performance is aborted.

The description of abstract error macro and abstract errors of subscription error is in Recommendation X.420.

b) Name error

The name error abstract error reports that one or more of the O/R names supplied as argument of the abstract operation whose performance is aborted, or as components of its arguments, are invalid.

The description of abstract error macro and abstract errors of name error is in Recommendation X.420.

c) Cancellation error

The cancellation error abstract error reports that the user's request to cancel the delivery of a message cannot be performed.

The description of abstract error macro and abstract errors of cancellation error is in Recommendation X.420.

d) Subscription profile error

The user's request to change his subscription—prpfile cannot be performed, because one or more arguments proposed are inacceptable.

```
subscription—profile—error        ABSTRACT—ERROR
        PARAMETER SET { problem [0] SubscriptionProfileProblem }
        ::= 0
```

This abstract error has the following parameters:

1) Problem (M): The specific subscription profile related problem encountered.

```
SubscriptionProfileProblem ::= CHOICE { [0] not—changed }
```

This parameter may assume any one of the following values:

— not—changed: One or more subscription—profile arguments proposed are unacceptable, this abstract—operation is not performed.

e) DS error

The argument related DS cannot be performed because one or more arguments are improperly specified.

```
ds—error                ABSTRACT—ERROR
        PARAMETER SET { problem [0] DSProblem }
        ::= 1
```

This abstract error has the following parameter:

1) Problem (M): The specific DS related problem encountered.

```
DSProblem      ::= CHOICE { [0] no—message—in—ds,
                            [1] ds—not—supported,
                            [2] ds—not—subscribed,
                            [3] retrieval—identifier—invalid,
                            [4] parameter—invalid }
```

This parameter may assume any one of the following values:

— no—message—in—ds: User requests to perform DS related abstract operation when there is no message in DS.

— ds—not—supported: User requests to perform DS related abstract—operation when TLMAU does not provide DS.

— ds—not—subscribed: User requests to perform DS related abstract—operation when he does not subscribe to DS.

— retrieval—identifier—invalid: The retrieval—id proposed is invalid.

— parameter—invalid: One or more arguments proposed are invalid.

f) MessageStatusError

No such message can be assigned by the query—identifier for message status abstract operation.

```
message—status—error            ABSTRACT—ERROR

    PARAMETER SET { problem [0] MessageStatusProblem }
    ::= 2
```

This abstract—error has the following parameter:

1) Problem (M): The specific message status related problem encountered.

```
MessageStatusProblem ::= CHOICE { [0] query—identifier—invalid }
```

This parameter may assume any one of the following values:

— query—identifier—invalid: The query—identifier proposal is unacceptable.

10 Realization of abstract operations

How a TLMAU realizes the mhs—doc—xfer port by means of which it interacts with a TLM terminal is specified in this section. But how a TLMA realizes the ports by means of which it interacts with a TLM user and MTS is outside the scope of this Recommendation.

Telematic access protocol for accessing to IPMS, called P5 protocol, is provided to realize the interaction, which means abstract operations performed at the mhs—doc—xfer port, between a TLMAU and a TLM terminal. The concrete interactions, which correspond to abstract operations, are realized as telematic access protocol data units (TAPDUs).

It should be noted that the TLMAU may not support all the conditional TAPDUs and all the optional elements or parameters of a TAPDU. The actual support of the TAPDUs and parameters depends on the application and the version of the colocated MTA.

The relationship between abstract operations at the mhs—doc—xfer port and associated TAPDUs are summarized in Table 2/T.330.

10.1 *Description of TAPDU*

10.1.1 *MessageSend*

A TLM terminal sends a Send—TAPDU to invoke the MessageSend abstract operation. The TLMAU returns a SendAck—TAPDU to report the result of that operation, or may return an Exception—TAPDU (§ 10.1.1.3) to report an abstract error.