



INTERNATIONAL TELECOMMUNICATION UNION

CCITT

THE INTERNATIONAL
TELEGRAPH AND TELEPHONE
CONSULTATIVE COMMITTEE

X.420

(11/1988)

SERIES X: DATA COMMUNICATION NETWORKS:
MESSAGE HANDLING SYSTEMS

**MESSAGE HANDLING SYSTEMS:
INTERPERSONAL MESSAGING SYSTEM**

Reedition of CCITT Recommendation X.420 published in
the Blue Book, Fascicle VIII.7 (1988)

NOTES

1 CCITT Recommendation X.420 was published in Fascicle VIII.7 of the *Blue Book*. This file is an extract from the *Blue Book*. While the presentation and layout of the text might be slightly different from the *Blue Book* version, the contents of the file are identical to the *Blue Book* version and copyright conditions remain unchanged (see below).

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

Recommendation X.420

MESSAGE HANDLING SYSTEMS: INTERPERSONAL MESSAGING SYSTEM¹⁾

(Malaga-Torremolinos, 1984; amended at Melbourne, 1988)

The establishment in various countries of telematic services and computer-based store-and-forward message services 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 message handling systems;
- (b) the need to transfer and store messages of different types;
- (c) that Recommendation X.200 defines the reference model of open systems interconnection for CCITT applications;
- (d) that Recommendations X.208, X.217, X.218 and X.219 provide the foundation for CCITT applications;
- (e) that the X.500-series Recommendations define directory systems;
- (f) that message handling systems are defined in a series of Recommendations X.400, X.402, X.403, X.407, X.408, X.411, X.413, and X.419;
- (g) that interpersonal messaging is defined in Recommendations X.420 and T.330,

unanimously declares

- (1) that the abstract information objects users exchange in interpersonal messaging are defined in Section 2;
- (2) that the abstract service offered to users in interpersonal messaging is defined in Section 3;
- (3) that how that abstract service is provided is specified in Section 4.

TABLE OF CONTENTS

SECTION 1 – Introduction

0	<i>Introduction</i>
1	<i>Scope</i>
2	<i>References</i>
3	<i>Definitions</i>
4	<i>Abbreviations</i>
5	<i>Conventions</i>
5.1	ASN.1
5.2	Grade
5.3	Terms

¹⁾ Recommendation X.420 and ISO 10021—7 [Information Processing Systems — Text Communication — MOTIS — Interpersonal Messaging System] were developed in close collaboration and are technically aligned, except for the differences.

SECTION 2 – *Abstract information objects*

- 6 *Overview*
- 7 *Interpersonal messages*
 - 7.1 Heading fields component types
 - 7.2 Heading fields
 - 7.3 Body part types
- 8 *Interpersonal notifications*
 - 8.1 Common fields
 - 8.2 Non-receipt fields
 - 8.3 Receipt fields

SECTION 3 – *Abstract service definition*

- 9 *Overview*
- 10 *Primary object types*
 - 10.1 Interpersonal messaging system user
 - 10.2 Interpersonal messaging system
- 11 *Primary port types*
 - 11.1 Origination
 - 11.2 Reception
 - 11.3 Management
- 12 *Abstract operations*
 - 12.1 Origination abstract operations
 - 12.2 Reception abstract operations
 - 12.3 Management abstract operations
- 13 *Abstract errors*
 - 13.1 Subscription error
 - 13.2 Recipient improperly specified
- 14 *Other capabilities*

SECTION 4 – *Abstract service provision*

- 15 *Overview*
- 16 *Secondary object types*
 - 16.1 Interpersonal messaging system user agent
 - 16.2 Interpersonal messaging system message store
 - 16.3 Telematic agent
 - 16.4 Telex access unit
 - 16.5 Physical delivery access unit
 - 16.6 Message transfer system
- 17 *Secondary port types*
 - 17.1 Submission
 - 17.2 Delivery

- 17.3 Retrieval
- 17.4 Administration
- 17.5 Import
- 17.6 Export
- 18 *User agent operation*
 - 18.1 State variables
 - 18.2 Performance of origination operations
 - 18.3 Performance of management operations
 - 18.4 Invocation of reception operations
 - 18.5 Internal procedures
- 19 *Message store operation*
 - 19.1 Creation of information objects
 - 19.2 Maintenance of attributes
 - 19.3 Notification of non-receipt
 - 19.4 Auto-forwarding
- 20 *Message contents*
 - 20.1 Content
 - 20.2 Content type
 - 20.3 Content length
 - 20.4 Encoded information types
- 21 *Port realization*
- 22 *Conformance*
 - 22.1 Originating versus reception
 - 22.2 Statement requirements
 - 22.3 Static requirements
 - 22.4 Dynamic requirements
- Annex A* – Heading extensions
- Annex B* – Extended body part types
- Annex C* – Message store attributes
- Annex D* – Reference definition of object identifiers
- Annex E* – Reference definition of abstract information objects
- Annex F* – Reference definition of functional objects
- Annex G* – Reference definition of abstract service
- Annex H* – Reference definition of heading extensions
- Annex I* – Reference definition of extended body part types
- Annex J* – Reference definition of message store attributes
- Annex K* – Reference definition of upper bounds
- Annex L* – Support of the interpersonal messaging service

Annex M – Differences between CCITT Recommendation and ISO Standard

Annex N – Summary of changes to 1984 specification

SECTION 1 – INTRODUCTION

0 Introduction

This Recommendation is one of a set of Recommendations for message handling. The entire set provides a comprehensive blueprint for a message handling system (MHS) realized by any number of cooperation open systems.

The purpose of an MHS is to enable users to exchange messages on a store-and-forward basis. A message submitted on behalf of one user, the originator, is conveyed by the message transfer system (MTS) and subsequently delivered to the agents of one or more additional users, the recipients. Access units (AUs) link the MTS to communication systems of other kinds (e.g., postal systems). A user is assisted in the preparation, storage, and display of messages by a user agent (UA). Optionally, it is assisted in the storage of messages by a message store (MS). The MTS comprises a number of message transfer agents (MTAs) which collectively perform the store-and-forward message transfer function.

This Recommendation defines the message handling application called *interpersonal messaging*, specifying in the process the message content type and associated procedures known as *P2*.

The text of this Recommendation is the subject of joint CCITT-ISO agreement. The corresponding ISO/IEC specification is ISO 10021-7.

1 Scope

This Recommendation defines **interpersonal messaging**, a form of message handling tailored for ordinary interpersonal business or private correspondence.

This Recommendation is one of a series on message handling. Recommendation X.402 constitutes the introduction to the series and identifies the other documents in it.

The architectural basis and foundation for message handling are defined in still other Recommendations. Recommendation X.402 identifies those documents as well.

This Recommendation is structured as follows. Section 1 is this introduction. Section 2 defines the kinds of information objects exchanged in interpersonal messaging. Section 3 defines the associated abstract service. Section 4 specifies how it is provided. Annexes provide important supplemental information.

The requirements for conformance to this Recommendation are given in § 22.

2 References

This Recommendation cites Recommendation X.402, many of the documents it cites, and those below.

ISO Standard 639.2	Code for the representation of names of languages.
Recommendation T.4	Standardization of Group 3 facsimile apparatus for document transmission.
Recommendation T.30	Procedures for document facsimile transmission in the general switched telephone network.
Recommendation T.100	International information exchange for interactive videotex.
Recommendation T.101	International interworking for videotex services.
Recommendation T.330	Telematic access to IPMS.
Recommendation X.420 (1984)	Message handling systems: Interpersonal messaging user agent layer. X.400-series implementor's guide, Version 6, 6 November 1987.

3 Definitions

For the purposes of this Recommendation, the definitions of Recommendation X.402 apply.

4 Abbreviations

For the purposes of this Recommendation, the abbreviations of Recommendation X.402 apply.

5 Conventions

This Recommendation uses the descriptive conventions identified below.

5.1 ASN.1

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

- a) to define the information objects of interpersonal messaging, and other data types and values of all kinds, ASN.1 itself;
- b) to define the functional objects of interpersonal messaging, the OBJECT and REFINE macros of Recommendation X.407;
- c) to define the abstract service of interpersonal messaging, the PORT and ABSTRACT-OPERATION and -ERROR macros of Recommendation X.407;
- d) to define the *heading extensions*, the HEADING-EXTENSION macro of § 7.2.17;
- e) to define *extended body part types*, the EXTENDED-BODY-PART-TYPE macro of § 7.3.12;
- f) to define MS attributes, the ATTRIBUTE macro of Recommendation X.500.

The various uses of the ASN.1 notation are summarized in Table 1/X.420. With the two exceptions evident from the table, whenever ASN.1 is employed, it appears both in the body of the Recommendation to aid the exposition, and again, largely redundantly, in an Annex for reference.

TABLE 1/X.420

Uses of the ASN.1 notation

Subject matter	Exposition	Reference
Object identifiers	–	Annex D
Abstract information objects	Section 2	Annex E
Functional objects	§§ 10, 11, 16	Annex F
Abstract service	§§ 12-13	Annex G
Heading extensions	Annex A	Annex H
Extended body part types	Annex B	Annex I
Message store attributes	Annex C	Annex J
Upper bounds	–	Annex K

If differences are found between the ASN.1 used in the exposition and that supplied for reference, a specification error is indicated.

Note that ASN.1 tags are implicit throughout the ASN.1 module the Annex defines; the module is definitive in that respect.

Note 1 – The use of ASN.1 to describe a class or piece of information does not in itself imply that that information is transported between open systems. The fact that the information, by virtue of its description in ASN.1 and of ASN.1's basic encoding rules, has a concrete transfer syntax may be immaterial. Information actually conveyed between systems is designated as such by its inclusion in an application protocol.

Note 2 – The use of the ABSTRACT-OPERATION and -ERROR macros, derived from the correspondingly named macros of remote operations, does not imply that the abstract operations and errors are invoked and reported across the boundary between open systems. The fact that the abstract operations and errors, by virtue of their description using these macros and with minimal additional specification, actually could be invoked via ROS is immaterial in the present context.

5.2 *Grade*

This Recommendation uses the concept of grade as developed in Recommendation X.402.

5.3 *Terms*

Throughout this Recommendation, terms are rendered in **bold** when defined, in *italic* when referenced prior to their definitions, without emphasis upon all other occasions.

Terms that are proper nouns are capitalized, generic terms are not.

SECTION 2 – ABSTRACT INFORMATION OBJECTS

6 **Overview**

This section abstractly describes the information objects that users exchange in interpersonal messaging. They are of two kinds, *interpersonal messages (IPMs)* and *interpersonal notifications (IPNs)*. One of the latter acknowledges a user's receipt of one of the former.

```
InformationObject ::= CHOICE {  
    ipm [0] IPM,  
    ipn [1] IPN }
```

This section covers the following topics:

- a) interpersonal messages;
- b) interpersonal notifications.

Note 1 – The use, throughout this section, of words such as “originator” and “recipient” anticipates the fact that *IPMs* and *IPNs* are conveyed between users as the contents of messages (see § 20). These words, therefore, refer to the roles users and DLs play in such transmittals.

Note 2 – An *IPM* may appear (see § 7.3.8) in the *body* of another *IPM* which itself is conveyed as the content of a message. The words “originator” and “recipient” shall be understood in the context of an *IPM's* conveyance as the (entire) content of a message, not as a component of the *body* of another *IPM* so conveyed.

Note 3 – An *IPM* or *IPN* makes various assertions about its own transmittal (e.g., who originates the message containing it). Furthermore, an *IPN* makes assertions about the transmittal of the *IPM* to which it responds. All of these assertions are unverified.

7 **Interpersonal messages**

An **interpersonal message (IPM)** is a member of the primary class of information object conveyed between users in interpersonal messaging.

```
IPM ::= SEQUENCE {  
    heading      Heading,  
    body         Body }
```

It has the following components:

- a) **heading**: A set of **heading fields** (or **fields**), each an information item that gives a characteristic of the IPM (e.g., its importance);
- b) **body**: A sequence of **body parts**, each an information object that the IPM is intended to convey between users (e.g., a document).

```
Body ::= SEQUENCE OF BodyPart
```

The structure of an IPM is depicted in Figure 1/X.420.

This paragraph defines and describes the most prominent heading field component types and the defined heading fields and body part types.

Note – An IPM may be likened to a business memo. In fact, the terms “heading” and “body” appeal to that analogy.

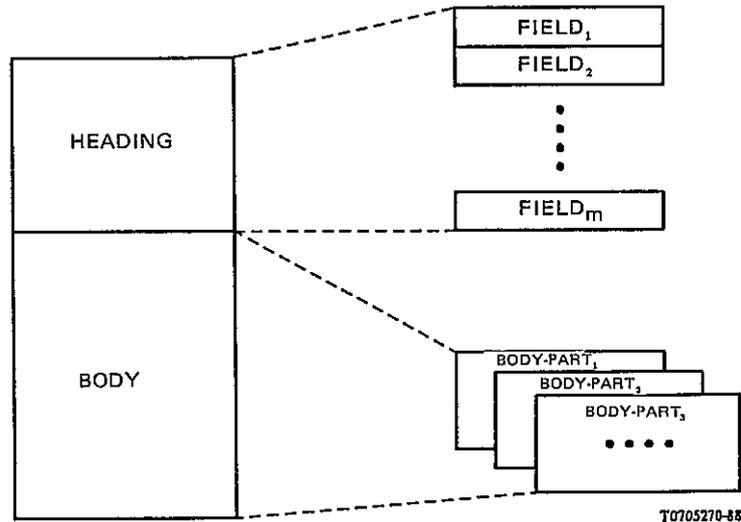


FIGURE 1/X.420
An interpersonal message

7.1 *Heading field component types*

Information items of several kinds appear throughout the heading. These heading field component types --*IPM identifier*, *recipient specifier*, and *O/R descriptor*-- are defined and described below.

7.1.1 *IPM identifier*

An **IPM identifier** is an information item that unambiguously and uniquely identifies an IPM, distinguishing it from all other IPMs ever conveyed by any user.

```
IPMIdentifier ::= [APPLICATION 11] SET {
    user ORAddress OPTIONAL,
    user-relative-identifier LocalIPMIdentifier }
```

An IPM identifier has the following components:

- a) **user** (O): Identifies the user who originates the IPM. One of the user's O/R addresses. This component's omission is discouraged;
- b) **user-relative-identifier** (M): Uniquely and unambiguously identifies the **IPM**, distinguishing it from all other IPMs that the user who is identified by the user component originates. A printable string of from zero to a prescribed number of characters (see Annex K). A length of zero is discouraged.

```
LocalIPMIdentifier ::= PrintableString
    (SIZE (0..ub-local-ipm-identifier))
```

Note – The “11” in IPMIdentifier is the only ASN.1 application-wide tag this Recommendation assigns.

7.1.2 Recipient specifier

A **recipient specifier** is an information item that identifies a (preferred) recipient of an IPM and that may make certain requests of him.

```
RecipientSpecifier ::= SET {
    recipient           [0]    ORDescriptor,
    notification-requests [1]    NotificationRequests DEFAULT { },
    reply-requested     [2]    BOOLEAN DEFAULT FALSE }
```

A recipient specifier has the following components:

- a) **recipient** (M): Identifies the preferred recipient in question. An *O/R descriptor*.

If the *notification-requests* or *reply-requested* component makes a request of the preferred recipient, the *formal-name* component of the *O/R descriptor* above shall be present.

- b) **notification-requests** (D no values): May make certain requests of the preferred recipient denoted by the recipient component.

```
NotificationRequests ::= BIT STRING {
    rn           (0),
    nrn         (1),
    ipm-return  (2) }
```

This component may assume any of the following values simultaneously, except that the value *rn* shall not be selected unless the value *nrn* is selected:

- i) *rn*: A *receipt notification* is requested in the circumstances prescribed in § 8.
 - ii) *nrn*: A *non-receipt notification* is requested in the circumstances prescribed in § 8.
 - iii) *ipm-return*: It is requested that the IPM be returned in any *non-receipt notification*.
- c) **reply-requested** (D *false*): Indicates whether a reply is requested of the preferred recipient denoted by the recipient component. A Boolean.

A **reply** is one IPM sent in response to another. A user may reply to an IPM even though no reply is requested of him and, indeed, even if he is not among the IPM's preferred recipients. Furthermore, a user of whom a reply is requested may refrain from replying.

7.1.3 O/R descriptor

An **O/R descriptor** is an information item that identifies a user or DL.

```
ORDescriptor ::= SET {
    formal-name           ORName OPTIONAL,
    free-form-name       [0]    FreeFormName OPTIONAL,
    telephone-number     [1]    TelephoneNumber OPTIONAL }
```

An O/R descriptor has the following components:

- a) **formal-name** (C): Identifies the user or DL in question. One of its O/R names.

This conditional component shall be present if (but not only if) one or more of the following criteria are satisfied:

- i) The *free-form-name* component is absent.
 - ii) The O/R descriptor appears in the *reply recipients* heading field.
 - iii) The O/R descriptor is the recipient component of a recipient specifier and the conditions stated in item 2 of § 7.1.2 are satisfied.
- b) **free-form-name** (O): Identifies the user or DL in question. A telex string of from zero to a prescribed number of characters (see Annex K), chosen from the graphic subset of the teletex string character set. A length of zero is discouraged.

FreeFormName ::= TeletexString (SIZE (0..ub-free-form-name))

- c) **telephone-number** (O): Provides the telephone number of the user or DL in question. A printable string of from zero to a prescribed number of characters (see Annex K), chosen from the graphical subset of the printable string character set. A length of zero is discouraged.

PhoneNumber ::= PrintableString (SIZE (0..ub-telephone-number))

Note – One or more O/R descriptors may appear in each of the following heading fields: originator, authorizing users, primary recipients, copy recipients, blind copy recipients, and reply recipients. In addition, an O/R descriptor may appear in the following notification fields (see § 8): IPN originator and IPM preferred recipient.

7.2 Heading fields

The fields that appear in the heading of an IPM are defined and described below.

Heading ::= SET {

this-IPM		ThisIPMField,
originator	[0]	OriginatorField OPTIONAL,
authorizing-users	[1]	AuthorizingUsersField OPTIONAL,
primary-recipients	[2]	PrimaryRecipientsField DEFAULT { },
copy-recipients	[3]	CopyRecipientsField DEFAULT { },
blind-copy-recipients	[4]	BlindCopyRecipientsField OPTIONAL,
replied-to-IPM	[5]	RepliedToIPMField OPTIONAL,
obsoleted-IPMs	[6]	ObsoletedIPMsField DEFAULT { },
related-IPMs	[7]	RelatedIPMsField DEFAULT { },
subject	[8]	EXPLICIT SubjectField OPTIONAL,
expiry-time	[9]	ExpiryTimeField OPTIONAL,
reply-time	[10]	ReplyTimeField OPTIONAL,
reply-recipients	[11]	ReplyRecipientsField OPTIONAL,
importance	[12]	ImportanceField DEFAULT normal,
sensitivity	[13]	SensitivityField OPTIONAL,
auto-forwarded	[14]	AutoForwardedField DEFAULT FALSE,
extensions	[15]	ExtensionsField DEFAULT { }

Some fields have components and thus are composite, rather than indivisible. A field component is called a **sub-field**.

7.2.1 This IPM

The **this IPM** heading field (M) identifies the IPM. It comprises an IPM identifier.

ThisIPMField ::= IPMIdentifier

7.2.2 Originator

The **originator** heading field (O) identifies the IPM's originator. It comprises an O/R descriptor.

OriginatorField ::= ORDescriptor

7.2.3 Authorizing users

The **authorizing users** heading field (C) identifies the zero or more users who are the IPM's *authorizing users*. It comprises a sequence of sub-fields, each an O/R descriptor, one for each such user.

AuthorizingUsersField ::= SEQUENCE OF AuthorizingUsersSubfield

AuthorizingUsersSubfield ::= ORDescriptor

An **authorizing user** is a user who, either individually or in concert with others, authorizes the origination of an IPM. The word “authorizes” above is not precisely defined by this Recommendation; it is given meaning by users.

This conditional field shall be present if, and only if, the authorizing users are other than the IPM's originator alone.

Note – Suppose, for example, that a manager instructs his secretary to originate an IPM on his behalf. In this case, the secretary, the IPM's originator, might consider the manager the authorizing user.

7.2.4 Primary recipients

The **primary recipients** heading field (D not subfields i.e., elements) identifies the zero or more users and DLs who are the “primary recipients” of the IPM. It also identifies the responses the authorizing users ask of each of those users and of each member of those DLs. It comprises a Sequence of sub-fields, each a recipient specifier, one for each primary recipient.

PrimaryRecipientsField ::= SEQUENCE OF PrimaryRecipientsSubfield

PrimaryRecipientsSubfield ::= RecipientSpecifier

The phrase “primary recipients” above is not precisely defined by this Recommendation; it is given meaning by users.

Note – The primary recipients, e.g., might be those users and those DLs whose members are expected to act upon the IPM.

7.2.5 Copy recipients

The **copy recipients** heading field (D no subfields i.e., elements) identifies the zero or more users and DLs who are the “copy recipients” of the IPM. It also identifies the responses the authorizing users ask of each of those users and of each member of those DLs. It comprises a sequence of sub-fields, each a recipient specifier, one for each copy recipient.

CopyRecipientsField ::= SEQUENCE OF CopyRecipientsSubfield

CopyRecipientsSubfield ::= RecipientSpecifier

The phrase “copy recipients” above is not precisely defined by this Recommendation; it is given meaning by users.

Note – The copy recipients, for example, might be those users to whom, and those DLs to whose members the IPM is conveyed for information.

7.2.6 Blind copy recipients

The **blind copy recipients** heading field (C) identifies zero or more users and DLs who are intended *blind* copy “recipients” of the IPM. It also identifies the responses the authorizing users ask of each of those users and of each member of those DLs. It comprises a sequence of sub-fields, each a recipient specifier, one for each *blind* copy recipient.

BlindCopyRecipientsField ::= SEQUENCE OF BlindCopyRecipientsSubfield

BlindCopyRecipientsSubfield ::= RecipientSpecifier

The phrase “copy recipients” above has the same meaning as in § 7.2.5. A **blind** copy recipient is one whose role as such is disclosed to neither primary nor copy recipients.

In the instance of an IPM intended for a blind copy recipient, this conditional field shall be present and identify that user or DL. Whether it shall also identify the other blind copy recipients is a local matter. In the instance of the IPM intended for a primary or copy recipient, the field shall be absent or identify no users or DLs.

7.2.7 Replied-to IPM

The **replied-to IPM** heading field (C) identifies the **IPM** to which the present IPM is a reply. It comprises an IPM identifier.

RepliedToIPMField ::= IPMIdentifier

This conditional field shall be present if, and only if, the IPM is a reply.

Note – In the context of *forwarding*, care should be taken to distinguish between the *forwarding IPM* and the *forwarded IPM*. This field should identify whichever of these two IPMs to which the reply responds.

7.2.8 Obsolete IPMs

The **obsolete IPMs** heading field (D no subfields i.e., elements) identifies zero or more IPMs that the authorizing users of the present IPM consider it to obsolete. It comprises a sequence of sub-fields, each an IPM identifier, one for each IPM.

ObsoleteIPMsField ::= SEQUENCE OF ObsoleteIPMsSubfield

ObsoleteIPMsSubfield ::= IPMIdentifier

Note – In the context of *forwarding*, care should be taken to distinguish between the *forwarding IPM* and the *forwarded IPM*. This field should identify whichever of these two IPMs the present IPM obsoletes.

7.2.9 *Related IPMs*

The **related IPMs** heading field (D no subfields i.e., elements) identifies zero or more IPMs that the authorizing users of the present IPM consider related to it. It comprises a sequence of sub-fields, each an IPM identifier, one for each IPM.

RelatedIPMsField ::= SEQUENCE OF RelatedIPMsSubfield

RelatedIPMsSubfield ::= IPMIdentifier

The word “related” above is not precisely defined by this Recommendation; it is given meaning by users.

Note 1 – A related IPM, e.g., might be one discussed in the body of the present IPM.

Note 2 – In the context of *forwarding*, care should be taken to distinguish between the *forwarding IPM* and the *forwarded IPM*. This field should identify whichever of these two IPMs is related to the present IPM.

7.2.10 *Subject*

The **subject** heading field (O) identifies the subject of the **IPM**. It comprises a teletex string of from zero to a prescribed number of characters (see Annex K), chosen from the graphic subset of the teletex string character set. A length of zero is discouraged.

SubjectField ::= TeletexString (SIZE (0..ub-subject-field))

7.2.11 *Expiry time*

The **expiry time** heading field (O) identifies when the authorizing users consider the IPM to lose its validity. It comprises a date and time.

ExpiryTimeField ::= Time

7.2.12 *Reply time*

The **reply time** heading field (O) identifies by when the authorizing users request (but do not demand) that any replies to the present IPM be originated. It comprises a date and time.

ReplyTimeField ::= Time

7.2.13 *Reply recipients*

The **reply recipients** heading field (C) identifies zero or more users and DLs whom the authorizing users request (but do not demand) be among the preferred recipients of any replies to the present IPM. It comprises a sequence of sub-fields, each an O/R descriptor, one for each user or DL.

ReplyRecipientsField ::= SEQUENCE OF ReplyRecipientsSubfield

ReplyRecipientsSubfield ::= ORDescriptor

This conditional field shall be present if, and only if, the desired reply recipients are other than the originator of the present IPM alone.

Note – If this field is present and identifies several users and DLs, the originator may include himself among them. If he elects not to do so, he will not be considered among the desired reply recipients.

7.2.14 *Importance*

The **importance** heading field (D *normal*) identifies the importance that the authorizing users attach to the IPM. It may assume any one of the following values: *low*, *normal*, or *high*.

ImportanceField ::= ENUMERATED {
 low (0),
 normal (1),
 high (2) }

The values above are not defined by this Recommendation; they are given meaning by users.

7.2.15 Sensitivity

The **sensitivity** heading field (C) identifies the sensitivity that the authorizing users attribute to the IPM.

```
SensitivityField ::= ENUMERATED {  
    personal          (1),  
    private           (2),  
    company-confidential (3) }
```

This field may assume any one of the following values:

- a) *personal*: The IPM is conveyed to its preferred recipients as individuals, rather than in their professional capacities.
- b) *private*: The IPM should be conveyed to no one other than its preferred recipients.
- c) *company-confidential*: The IPM contains information that should be handled according to company-specific procedures.

The conditional field shall be present if, and only, if the IPM is sensitive.

7.2.16 Auto-forwarded

The **auto-forwarded** heading field (D *false*) indicates whether the IPM is the result of *auto-forwarding*. It is a Boolean.

```
AutoForwardedField ::= BOOLEAN
```

7.2.17 Extensions

The **extensions** heading field (D no *extensions*, i.e., members) conveys information accommodated by no other heading field. It comprises a Set of zero or more **heading extensions** (or **extensions**), each conveying one such information item.

```
ExtensionsField ::= SET OF HeadingExtension
```

```
HeadingExtension ::= SEQUENCE {  
    type          OBJECT IDENTIFIER,  
    value         ANY DEFINED BY type DEFAULT NULL NULL }
```

Each extension has the following components:

- a) **type** (M): Identifies the semantics and restricts the abstract syntax of the *value* component. An object identifier.
- b) **value** (D null): An information item whose abstract syntax is restricted only by the type component. An Any.

The type components of all the extensions in the extensions field shall differ. Not every defined extension need appear in the field.

All extensions are defined in Annex A. Thus, each extension's type component shall have one of the values given in that Annex. An extension whose type component has another value shall be ignored.

Every extension is defined by means of the following macro.

```
HEADING-EXTENSION MACRO ::=  
BEGIN  
    TYPE    NOTATION ::= "VALUE" type | empty  
    VALUE  NOTATION ::= value (VALUE OBJECT IDENTIFIER)  
END
```

An instance of the macro's type notation identifies the data type to which the extension's value component shall be restricted. If no type is identified explicitly, null is implied.

An instance of the macro's value notation identifies the object identifier that shall appear as the extension's type component.

Note – Future versions of this Recommendation may define additional extensions. Furthermore, future versions are likely to add to the heading only by means of this field.

7.3 *Body part types*

The types of body parts that may appear in the body of an IPM are defined and described below:

```
BodyPart ::= CHOICE {  
    ia5-text           [0]    IA5TextBodyPart,  
    voice              [2]    VoiceBodyPart,  
    g3-facsimile       [3]    G3FacsimileBodyPart,  
    g4-class1          [4]    G4Class1BodyPart,  
    teletex            [5]    TeletexBodyPart,  
    videotex           [6]    VideotexBodyPart,  
    encrypted          [8]    EncryptedBodyPart,  
    message             [9]    MessageBodyPart,  
    mixed-mode         [11]   MixedModeBodyPart,  
    bilaterally-defined [14]   BilaterallyDefinedBodyPart,  
    nationally-defined  [7]    NationallyDefinedBodyPart,  
    externally-defined  [15]   ExternallyDefinedBodyPart }
```

Body parts of some of the types defined below have two components, *parameters* and *data*. The **parameters** component (M) comprises a sequence of information items that describe the information object the body part represents and that typically are format and control parameters. The **data** component (M) is the information object itself.

Note 1 – In Recommendation X.420 (1984), context-specific tags 1 and 10 denote telex and simple formattable document body parts, respectively, which are no longer defined. These tags, therefore, are avoided in BodyPart.

Note 2 – Under some circumstances, an IPM may be subjected to conversion while in transit between users. Such a transmittal event may alter a body part's type.

7.3.1 *IA5 text*

An **IA5 text** body part represents IA5 text. It has parameters and data components.

```
IA5TextBodyPart ::= SEQUENCE {  
    parameters          IA5TextParameters,  
    data                 IA5TextData }  
  
IA5TextParameters ::= SET {  
    repertoire          [0] Repertoire DEFAULT ia5 }  
  
IA5TextData ::= IA5String
```

The parameters component comprises the following parameters:

- a) **repertoire** (D IA5): Identifies the character set to which the data component is constrained.

```
Repertoire ::= ENUMERATED {  
    ita2 (2),  
    ia5 (5) }
```

This parameter may assume one of the following values:

- i) **ITA2**: The data component shall be limited to ITA2 (i.e., telex) character set.
ii) **IA5**: The data component may draw upon the full IA5 character set.

The data component is the text, an IA5 string. It may contain lines of any length. Whenever the component is rendered (e.g., displayed to or printed for a user), all (rather than only a part) of the text must be communicated (e.g., lines may be folded but shall not be truncated).

Note – Many terminals have a maximum line length of 80 characters. Therefore, lines that do not exceed that length are most likely to be satisfactorily rendered (e.g., are most likely to avoid being folded).

7.3.2 *Voice*

A **voice** body part represents speech. It has parameter and data components.

```
VoiceBodyPart ::= SEQUENCE {  
    parameters          VoiceParameters,  
    data                 VoiceData }
```

VoiceParameters ::= SET -- for further study

VoiceData ::= BIT STRING -- *for further study*

The parameters of such a body part, and the digitized speech encoding technique that those parameters might identify and parameterize, are for further study.

The data component is the speech, a bit string.

7.3.3 G3 facsimile

A **G3 facsimile** body part represents Group 3 facsimile images. It has parameters and data components.

G3FacsimileBodyPart ::= SEQUENCE {
 parameters G3FacsimileParameters,
 data G3FacsimileData }

G3FacsimileParameters ::= SET {
 number-of-pages [0] INTEGER OPTIONAL,
 non-basic-parameters [1] G3FacsimileNonBasicParameters OPTIONAL }

G3FacsimileData ::= SEQUENCE OF BIT STRING

The parameters component comprises the following parameters:

- a) **number-of-pages** (O): Identifies the number of pages of Group 3 facsimile data present in the data component. A non-negative integer.
- b) **non-basic-parameters** (C): Identifies the non-basic parameters (NBPs) for Group 3 facsimile that characterize the data component. A G3 NBPs descriptor.

This conditional parameter shall be present if (but not only if) the body contains two or more G3 facsimile body parts.

The data component is the facsimile images, a sequence of bit strings, each encoding a single page of Group 3 facsimile data as dictated by Recommendations T.4 and T.30.

Note 1 – The number-of-pages component identifies the number of elements in the sequence that constitutes the data component and is thus redundant.

Note 2 – If the body comprises a single such body part, its NBPs may (but need not) be conveyed by means of the envelope of the message that contains the IPM.

7.3.4 G4 Class 1

A **G4 Class 1** body part represents a final-form of document of the sort that is processable by Group 4 Class 1 facsimile terminals. It comprises a sequence of protocol elements which describe the document's layout structure.

G4Class1BodyPart ::= SEQUENCE OF ProtocolElement

7.3.5 Teletex

A **teletex** body part represents a teletex document. It has parameters and data components.

TeletexBodyPart ::= SEQUENCE {
 parameters TeletexParameters,
 data TeletexData }

TeletexParameters ::= SET {
 number-of-pages [0] INTEGER OPTIONAL,
 telex-compatible [1] BOOLEAN DEFAULT FALSE,
 non-basic-parameters [2] TeletexNonBasicParameters OPTIONAL }

TeletexData ::= SEQUENCE OF TeletexString

The parameters component comprises the following parameters:

- a) **number-of-pages** (O): Identifies the number of pages of teletex text present in the data component. A non-negative integer.
- b) **telex-compatible** (D *false*): Indicates whether the document in the data component is telex-compatible. A Boolean.

If this parameter has the value *true*, every teletex string in the data component shall be restricted to the ITA2 character set. No line shall exceed 69 characters in length.

- c) **non-basic parameters** (C): Identifies the NBPs for teletex that characterize the data component. A teletex NBPs descriptor.

This conditional parameter shall be present if (but not only if) the body contains two or more teletex body parts.

The data component is the document, a sequence of teletex strings, each of which encodes one of its pages.

Note 1 – The number-of-pages component identifies the number of elements in the sequence that constitutes the data components and is thus redundant.

Note 2 – If the body comprises a single such body part, its NBPs may (but need not) be conveyed by means of the envelope of the message that contains the IPM.

7.3.6 Videotex

A **videotex** body part represents videotex data. It has parameters and data components.

```
VideotexBodyPart ::= SEQUENCE {
    parameters          VideotexParameters,
    data                VideotexData }

VideotexParameters ::= SET {
    syntax              [0] VideotexSyntax OPTIONAL }

VideotexData ::= VideotexString
```

The parameters component comprises the following parameters:

- a) **syntax** (O): Identifies the syntax of the data component. In the parameter's absence, the syntax shall be considered unspecified.

```
VideotexSyntax ::= INTEGER {
    ids                  (0),
    data-syntax1        (1),
    data-syntax2        (2),
    data-syntax3        (3) }
```

This parameter may assume any one of the following values, each of which denotes as follows one of the videotex syntaxes defined in Recommendations T.100 and T.101:

- i) *ids*: The IDS syntax.
- ii) *data-syntax1*: Data syntax 1.
- iii) *data-syntax2*: Data syntax 2.
- iv) *data-syntax3*: Data syntax 3.

The data component is the videotex data, a videotex string. It shall conform to the videotex syntax denoted by the syntax parameter.

7.3.7 Encrypted

An **encrypted** body part represents the result of encrypting a body part of a type defined by this Recommendation. It has parameters and data components.

```
EncryptedBodyPart ::= SEQUENCE {
    parameters          EncryptedParameters,
    data                EncryptedData }

EncryptedParameters ::= SET -- for further study

EncryptedData ::= BIT STRING -- for further study
```

The parameters of such a body part, and the encryption technique that those parameters might identify and parameterize, are for further study.

The data component is the encrypted body part, a bit string. The bits of the bit string shall encrypt a data value of (ASN.1) type BodyPart encoded in accordance with the basic encoding rules of Recommendation X.209.

7.3.8 *Message*

A **message** body part represents an IPM and, optionally, its delivery envelope. It has parameters and data components.

```
MessageBodyPart ::= SEQUENCE {  
    parameters      MessageParameters,  
    data            MessageData }  
  
MessageParameters ::= SET {  
    delivery-time      [0] MessageDeliveryTime OPTIONAL,  
    delivery-envelope  [1] OtherMessageDeliveryFields OPTIONAL }  
  
MessageData ::= IPM
```

The parameters component comprises the following parameters:

- a) **delivery-time** (O): The date and time the IPM was delivered. The presence of this component in the absence of the delivery-envelope component is discouraged.
- b) **delivery-envelope** (O): The IPM's other message delivery fields. The presence of this component in the absence of the delivery-time component is discouraged.

The data component is the IPM.

Including one IPM in another as described in the present clause is called **forwarding** that IPM. The enclosing IPM is called the **forwarding IPM**, the enclosed IPM the **forwarded IPM**.

Note 1 – The possible future inclusion of a message identifier in the parameters component is for further study. Its present omission provides compatibility with Recommendation X.420 (1984).

Note 2 – That the IPM and purported delivery envelope of a message body part are, in any sense, genuine is unverified.

7.3.9 *Mixed-mode*

A **mixed-mode** body part represents a final-form document of the sort that is processable by mixed-mode Teletex terminals and Group 4 Classes 2 and 3 facsimile terminals. It comprises a sequence of protocol elements which describe the document's layout structure.

```
MixedModeBodyPart ::= SEQUENCE OF ProtocolElement
```

7.3.10 *Bilaterally defined*

A **bilaterally defined** body part represents an information object whose semantics and abstract syntax are bilaterally agreed by the IPM's originator and all of its potential recipients. It comprises an OctetString.

```
BilaterallyDefinedBodyPart ::= OCTET STRING
```

Note – The use of this body parts is discouraged. It predates the externally defined body part type and is retained for backward compatibility with Recommendation X.420 (1984). The externally defined body part type provides the same capabilities and more, its use is preferred, e.g., because such use clearly distinguishes between the body parts defined by one community of users and those defined by another.

7.3.11 *Nationally defined*

A **nationally defined** body part represents an information object whose semantics and abstract syntax are nationally defined by a country whose identity is bilaterally agreed by the IPM's originator and all of its potential recipients. It comprises an Any.

```
NationallyDefinedBodyPart ::= ANY
```

Note 1 – This body part is intended for use in domestic communication where the country in question is implicitly that of the originator and all of the potential recipients.

Note 2 – The use of this body part type is discouraged. It predates the externally defined body part type and is retained for backward compatibility with Recommendation X.420 (1984). The externally defined body part type provides the same capabilities and more, and its use is preferred, e.g., because such use clearly distinguishes between the body parts defined by one country and those defined by another.

7.3.12 Externally defined

An **externally defined** body part represents an information object whose semantics and abstract syntax are denoted by an object identifier which the body part carries. It has parameters and data components.

```
ExternalDefinedBodyPart ::= SEQUENCE {  
    parameters          [0] ExternallyDefinedParameters OPTIONAL  
    data                ExternallyDefinedData }
```

```
ExternallyDefinedParameters ::= EXTERNAL
```

```
ExternallyDefinedData ::= EXTERNAL
```

The parameters and data components are externals (see § 32 of Recommendation X.208). Their direct-reference components shall be present, their indirect-reference and data-value-descriptor components absent.

On the basis of the externally defined body part type, all body part types are divided into two important classes as follows:

- a) **basic**: Said of any body part type except externally defined. Denoted by an integer (an ASN.1 context-specific tag).

All basic body part types are defined in this Recommendation.

- b) **extended**: Some of the externally defined body part type restricted to any one value of the direct-reference component of the data component of such a body part. Denoted by an object identifier.

Some (but not necessarily all) extended body part types are defined in Annex B of this Recommendation.

Every extended body part type this Recommendation defines is defined by means of the following macro. Every extended body part type defined elsewhere shall be so defined as well.

```
EXTENDED-BODY-PART-TYPE MACRO ::=  
BEGIN  
  
    TYPE      NOTATION ::= Parameters Data  
    VALUE     NOTATION ::= value (VALUE OBJECT IDENTIFIER)  
  
    Paramaters      ::= "PARAMETERS" type "IDENTIFIED" "BY" value (OBJECT  
                        IDENTIFIER) | empty  
  
    Data            ::= "DATA" type  
  
END
```

An instance of the macro's type notation defines, by means of its PARAMETERS clause, the type of the data value that is represented by the parameters component of such an (externally defined) body part (an external), and the object identifier that appears in the direct-reference component of this parameters component. The omission of the parameters component is implied by the omission of this clause. An instance of the type notation also defines, by means of its DATA clause, the type of the data value that is represented by the data component of such a body part (an external).

An instance of the macro's value notation defines the object identifier that appears as the direct-reference component of the data component of such an (externally defined) body part. The object identifier identifies the encoding rules for the body part. Those body parts whose types this Recommendation defines shall be encoded using ASN.1 basic encoding rules.

Note 1 – This body part enables the exchange of information objects of all kinds, each unambiguously and uniquely identified. This identification relies upon the direct—reference component mentioned above, which is an object identifier. Object identifiers are easily obtained, e.g., by national bodies and private organizations.

Note 2 – If an externally defined body part has a parameters component, the object identifier in its direct-reference component is allocated at the same time and by the same naming authority as that in the direct-reference component of the data component.

Note 3 – Like body parts of other types, an externally defined body part may be subjected to conversion. However, specification of the conversion algorithms may be outside the scope of Recommendation X.408.

Note 4 – The basic body part exists for purely historical reasons, predating the externally defined body part type.

8 Interpersonal notifications

An **interpersonal notification (IPN)** is a member of a secondary class of information object conveyed between users in interpersonal messaging.

```
IPN ::= SET {
  -- common-fields -- COMPONENTS OF CommonFields,
  choice [0] CHOICE {
    non-receipt-fields          [0] NonReceiptFields,
    receipt-fields              [1] ReceiptFields } }
```

An IPN may take either of the following forms:

- a) **non-receipt notification (NRN)**: An IPN that reports its originator's failure to receive, to accept, or his delay in receiving, an IPM.

```
NRN ::= IPN -- with non-receipt-fields chosen
```

- b) **receipt notification (RN)**: An IPN that reports its originator's receipt, or his expected and arranged future receipt, of an IPM.

```
RN ::= IPN -- with receipt-fields chosen
```

The IPM to which an IPN refers is called the **subject IPM**. Only a UA to which the subject IPM is actually delivered shall originate an IPN relating to it, and it shall originate at most one such IPN which shall be conveyed to the subject IPM's originator alone.

An actual recipient shall originate an IPN only in accordance with the notification-requests component of the *subject recipient specifier*. The **subject recipient specifier** is that recipient specifier in the subject IPM's heading as a result of which the subject IPM is delivered to that user.

The subject recipient specifier is determined by examining the sequence of recipient specifiers that constitute the subject IPM's primary, copy and blind copy recipients heading fields. The fields are examined in the order in which they are mentioned in the preceding sentence. Within each field, the specifiers are examined in the order in which they appear there. The subject recipient specifier is the first one found whose recipient component has as its value an O/R descriptor whose formal-name component is present and has as its value either an O/R name of the preferred recipient as a result of which the subject IPM was delivered to the user on whose behalf the examination is performed or, if the IPM reached the user because of his membership in a DL, an O/R name appearing in the message's DL expansion history (see § 8.3.1.1.1.7 of Recommendation X.411).

An IPN comprises a set of information items called **notification fields (or fields)**, each of which is of one of the following classes:

- a) **common field**: A notification field applicable to both NRNs and RNs.
b) **non-receipt field**: A notification field applicable to NRNs alone.
c) **receipt field**: A notification field applicable to RNs alone.

The structure of an IPN is depicted in Figure 2/X.420.

The fields, in each of the above classes, that may appear in an IPN are defined and described below.

8.1 Common fields

The common fields are defined and described below.

```
CommonFields ::= SET {
  Subject-ipm          SubjectIPMField,
  Ipn-originator       [1] IPNOriginatorField OPTIONAL,
  Ipm-preferred-recipient [2] IPMPreferredRecipientField OPTIONAL,
  Conversion-eits      ConversionEITsField OPTIONAL }
```

8.1.1 Subject IPM

The **subject IPM** common field (M) identifies the subject IPM. It comprises an IPM identifier.

```
SubjectIPMField ::= IPMIdentifier
```

8.1.2 *IPN originator*

The IPN originator common field (O) identifies the IPN's originator. It comprises an O/R descriptor.

IPNOriginatorField ::= ORDescriptor

If the IPN's originator is a preferred recipient of the subject IPM, the O/R descriptor above shall be precisely that which is the value of the recipient component of the subject recipient specifier.

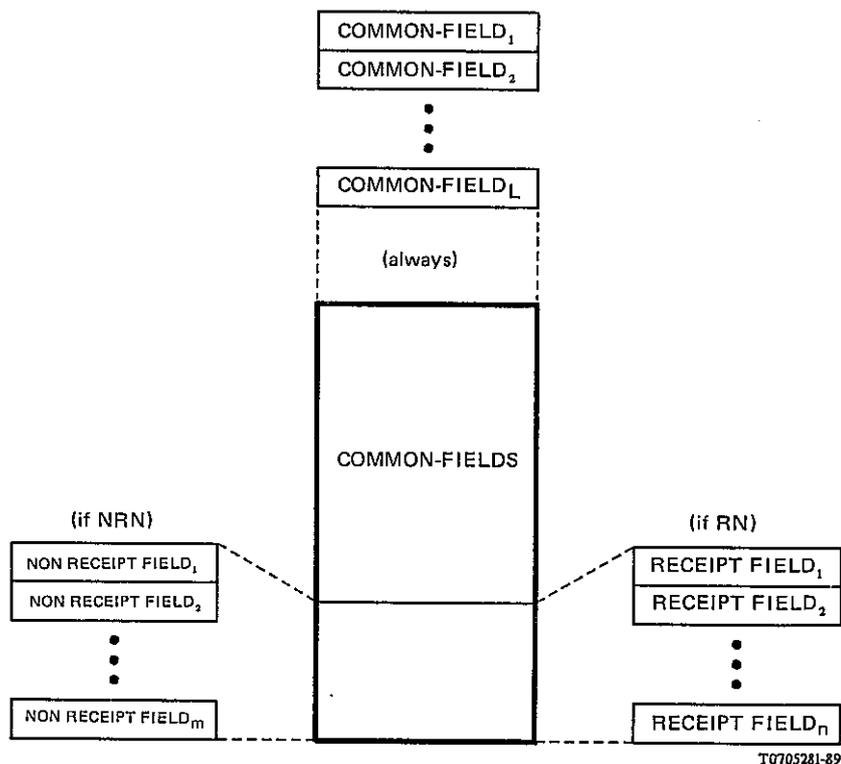


FIGURE 2/X.420
An interpersonal notification

8.1.3 IPM preferred recipient

the **IPM preferred recipient** common field (C) identifies the preferred recipient of the subject IPM who gives rise to its delivery to the IPN's originator (an alternate, (DL) member, or substitute recipient). It comprises an O/R descriptor.

IPMPreferredRecipientField ::= ORDescriptor

The O/R descriptor above shall be precisely that which is the value of the recipient component of the subject recipient specifier.

This conditional field shall be present if, and only if, it would identify a user other than the IPN's originator or a DL.

8.1.4 Conversion EITs

The **conversion EITs** common field (C) identifies the EITs of the subject IPM upon delivery to the IPN's originator. It comprises an EITs descriptor.

ConversionEITsField ::= EncodedInformationTypes

This conditional field shall be present if, and only if, the IPM was subjected to conversion for delivery to the IPN's originator.

8.2 Non-receipt fields

The non-receipt fields are defined and described below.

NonReceiptFields ::= SET {
 non-receipt-reason [0] NonReceiptReasonField,
 discard-reason [1] DiscardReasonField OPTIONAL,
 auto-forward-comment [2] AutoForwardCommentField OPTIONAL,
 returned-ipm [3] ReturnedIPMField OPTIONAL }

8.2.1 Non-receipt reason

The **non-receipt reason** non-receipt field (M) indicates why the NRN's originator has not received the subject IPM (even though it was delivered to him).

NonReceiptReasonField ::= ENUMERATED {
 ipm-discarded (0),
 ipm-auto-forwarded (1) }

This field may assume one of the following values:

- a) *ipm-discarded*: The IPM was discarded. This case is further illuminated by the *discard reason* field.
- b) *ipm-auto-forwarded*: The IPM was auto-forwarded. This case is further illuminated by the *auto-forward comment* field.

8.2.2 Discard reason

The **discard reason** non-receipt field (C) indicates why the subject IPM was discarded (subsequent to its delivery to the NRN's originator and prior to its receipt).

DiscardReasonField ::= ENUMERATED {
 ipm-expired (0),
 ipm-obsolete (1),
 user-subscription-terminated (2) }

This field may assume any one of the following values:

- a) *ipm-expired*: *auto-discard* was in effect, expired IPMs were being discarded, and the time identified by the subject IPM's expiry time heading field had arrived.
- b) *ipm-obsolete*: *auto-discard* was in effect, obsolete IPMs were being discarded, and the obsolete IPMs heading field of another IPM, delivered to the NRN's originator, identified the subject IPM.
- c) *user-subscription-terminated*: The interpersonal messaging subscription of the NRN's originator was terminated.

This conditional field shall be present only if the non-receipt reason field has the value *ipm-discarded*.

8.2.3 *Auto-forward comment*

The **auto-forward comment** non-receipt field (C) is information pre-supplied for this purpose by the NRN's originator. It comprises a printable string of from zero to a prescribed number of characters (see Annex K), chosen from the printable string character set. A length of zero is discouraged.

AutoForwardCommentField ::= AutoForwardComment

AutoForwardComment ::= PrintableString
(SIZE (0..ub-auto-forward-comment))

The value of this field shall be precisely the auto-forward-comment argument of the *change auto-forwarding* abstract operation as a result of which the subject IPM was auto-forwarded.

This conditional field shall be present if, and only if, the non-receipt reason field has the value *ipm-auto-forwarded* and the auto-forward-comment argument above was supplied.

8.2.4 *Returned IPM*

The **returned IPM** non-receipt field (C) is precisely the subject IPM.

ReturnedIPMField ::= IPM

This conditional field shall be present if, and only if, *ipm-return* is among the values of the notification-requests component of the subject recipient specifier and the subject IPM was not subjected to conversion for delivery to the NRN's originator.

8.3 *Receipt fields*

The receipt fields are defined and described below.

ReceiptFields ::= SET {
 receipt-time [0] ReceiptTimeField,
 acknowledgment-mode [1] AcknowledgmentModeField DEFAULT manual,
 suppl-receipt-info [2] SupplReceiptInfoField DEFAULT "" }

8.3.1 *Receipt time*

The **receipt time** receipt field (M) identifies when the RN's originator received the subject IPM. It comprises a date and time.

ReceiptTimeField ::= Time

8.3.2 *Acknowledgment mode*

The **acknowledgment mode** receipt field (D *manual*) identifies the manner in which the RN was originated.

AcknowledgmentModeField ::= ENUMERATED {
 manual (0),
 automatic (1) }

This field may assume any one of the following values:

- a) *manual*: The RN was originated by means of the *originate RN* abstract operation.
- b) *automatic*: The RN was originated as a result of *auto-acknowledgment*.

8.3.3 *Suppl receipt info*

The **suppl receipt info** receipt field (O) gives supplementary information about the receipt of the subject IPM by the RN's originator. It comprises a printable string of from zero to a prescribed number of characters (see Recommendation X.411), chosen from the printable string character set.

SupplReceiptInfoField ::= SupplementaryInformation

SECTION 3 – ABSTRACT SERVICE DEFINITION

9 Overview

This section defines the abstract service that characterizes interpersonal messaging, and describes the environment in which service is supplied and consumed. It does both using the abstract service definition conventions of Recommendation X.407.

This section covers the following topics:

- a) primary object types,
- b) primary port types,
- c) abstract operations,
- d) abstract errors,
- e) other capabilities.

10 Primary object types

The environment in which interpersonal messaging takes place can be modelled as an abstract object which is hereafter referred to as the **interpersonal messaging environment (IPME)**.

ipme OBJECT
 ::= id-ot-ipme

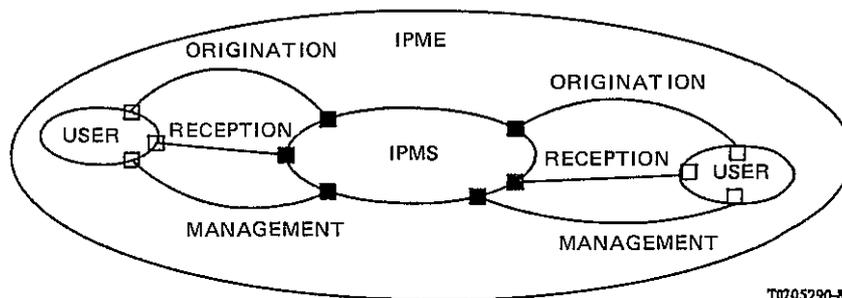
When refined (i.e., functionally decomposed), the IPME can be seen to comprise lesser objects which interact by means of ports.

ipme-refinement REFINE ipme AS
 ipms
 origination [S] PAIRED WITH ipms-user
 reception [S] PAIRED WITH ipms-user
 management [S] PAIRED WITH ipms-user
 ipms-user-RECURRING
 ::= id-ref-primary

The lesser objects are referred to as the **primary objects of interpersonal messaging**. They include a single, central object, the *interpersonal messaging system (IPMS)*, and numerous peripheral objects called *interpersonal messaging system users (IPMS users)*.

The structure of the IPME is depicted in Figure 3/X.420.

The primary object types are defined and described below. The types of ports by means of which they interact are discussed in § 11.



T0705290-88

FIGURE 3/X.420

The interpersonal messaging environment

10.1 *Interpersonal messaging system user*

An **interpersonal messaging system user (IPMS user)** is a user that engaged in interpersonal messaging. An IPMS user originates, receives, or both originates and receives information objects of the types defined in § 2.

```
ipms-user OBJECT
  PORTS {
    origination      [C],
    reception        [C],
    management       [C] }
  ::= id-ot-ipms-user
```

The IPME comprises any number of of IPMS users.

Note 1 – As its name suggests, interpersonal messaging is typically an activity of people. Often, therefore, this Recommendation uses personal pronouns (e.g., “he”) to refer to IPMS users. This practice, however, is not intended to preclude other, atypical uses of interpersonal messaging in which IPMS users are not people.

Note 2 – For brevity, the term “user” is used throughout the rest of this Recommendation with the meaning of “IPMS user”.

10.2 *Interpersonal messaging system*

The **interpersonal messaging system (IPMS)** is the object by means of which all users communicate with one another in interpersonal messaging.

```
ipms OBJECT
  PORTS {
    origination      [S],
    reception        [S],
    management       [S] }
  ::= id-ot-ipms
```

The IPME comprises exactly one IPMS.

11 **Primary port types**

The primary objects of interpersonal messaging are joined to and interact with one another by means of ports. These ports, which the IPMS supplies, are referred to as the **primary ports** of interpersonal messaging. They are of the three types defined below.

Note – In § 16 to follow, the IPMS is decomposed into still lesser objects, among which is the MTS. This fact is anticipated in the present paragraph by the inclusion of certain MTS capabilities in the IPMS abstract service.

11.1 *Origination*

An **origination port** is the means by which a single user conveys to the IPMS messages containing information objects of the types defined in section two. Through such a port the user originates *interpersonal messages* and *receipt notifications*. In addition, the user may originate probes through such a port.

The IPMS supplies one origination port to each user (with the exception of indirect users served by PDAUs; see § 16.5).

11.2 *Reception*

A **reception port** is the means by which the IPMS conveys to a single user messages containing information objects of the types defined in section two. Through such a port the user receives *interpersonal messages* and *interpersonal notifications*. In addition, the user may receive reports through such a port.

The IPMS supplies one reception port to each user.

11.3 *Management*

A **management port** is the means by which a single user changes information about himself on file with the IPMS. By means of such a port the user enables and disables *auto-discard*, *-acknowledgment*, and *-forwarding*.

The IPMS supplies one management to each user (with the exception of indirect users served by PDAUs; see § 16.5).

12 Abstract operations

The IPMS abstract service is the set of capabilities that the IPMS provides to each user by means of one origination, one reception, and one management port. Those capabilities are modelled as abstract operations, which may encounter abstract errors when invoked.

The abstract operations available at origination, reception, and management ports, respectively, are defined and described below. The abstract errors they may provoke are the subject of § 13.

Note 1 – The IPMS abstract service involves neither abstract bind nor abstract unbind operations.

Note 2 – The IPMS authenticates (i.e., establishes the identity of) the typical user before offering the IPMS abstract service to him. By this means it can verify, for example, that the user is an IPMS subscriber. Authentication, where required, is implicit (rather than explicit) in the definition of the IPMS abstract service.

Note 3 – The purpose of the IPMS abstract service definition is not to prescribe the user interfaces of implementations of portions of the IPMS, but rather to clarify the meaning and intended use of the information objects of section two. A user interface need not provide commands in one-to-one correspondence to the service's abstract operations, nor indeed even divide the labor between the user and the IPMS as the service does.

Note 4 – In § 16 to follow, the IPMS is decomposed into objects among which is the MTS. The present paragraph reflects this fact by its inclusion of various MTS-defined information items in the IPMS abstract service.

12.1 Origination abstract operations

The abstract operations available at an origination port are invoked by the user and performed by the IPMS.

```
origination PORT
  CONSUMER INVOKES {
    OriginateProbe,
    OriginateIPM,
    OriginateRN }
  ::= id-pt-origination
```

12.1.1 Originate probe

The **originate probe** abstract operation originates a probe concerning (a class of) messages whose contents are IPMs.

```
OriginateProbe ::= ABSTRACT-OPERATION
  ARGUMENT SET {
    envelope      [0] ProbeSubmissionEnvelope,
    content       [1] IPM }
  RESULT SET {
    submission-identifier [0] ProbeSubmissionIdentifier,
    submission-time      [1] ProbeSubmissionTime }
  ERRORS {
    SubscriptionError,
    RecipientImproperlySpecified }
```

This abstract operation has the following arguments:

- a) **envelope** (M): A probe submission envelope, whose makeup the MTS abstract service defines. The UA supplies all but the following envelope components, which the user provides:
 - i) The desired per-message options (i.e., per-message indicators and extensions).
 - ii) The O/R names of the preferred recipients and the per-recipient options (i.e., originator report request, explicit conversion, and extensions) desired for each.
- b) **content** (M): An instance of the class of IPM whose deliverability is to be probed.

This abstract operation has the following results:

- 1) **submission-identifier** (M): The probe submission identifier the MTS assigns to probe.
- 2) **submission-time** (M): The date and time the probe was directly submitted.

12.1.2 Originate IPM

The **originate IPM** abstract operation originates a message whose content is an IPM.

```
OriginateIPM ::= ABSTRACT-OPERATION
  ARGUMENT SET {
    envelope      [0] MessageSubmissionEnvelope,
    content       [1] IPM }
  RESULT SET {
    submission-identifier [0] MessageSubmissionIdentifier,
    submission-time      [1] MessageSubmissionTime }
  ERRORS {
    SubscriptionError,
    RecipientImproperlySpecified }
```

This abstract operation has the following arguments:

- a) **envelope** (M): A message submission envelope, whose makeup the MTS abstract service defines. The UA supplies all but the following envelope components, which the user provides:
 - i) The desired per-message options (i.e., priority, per-message indicators, deferred delivery time, and extensions).
 - ii) The O/R names of the preferred recipients and the per-recipient options (i.e., originator report request, explicit conversion, and extensions) desired for each.
- b) **content** (M): The IPM being originated. Its auto-forwarded heading field shall be absent or have the value *false*.

This abstract operation has the following results:

- 1) **submission-identifier** (M): The message submission identifier the MTS assigns to the submission.
- 2) **submission-time** (M): The date and time the message was directly submitted.

12.1.3 Originate RN

The **originate RN** abstract operation originates a message whose content is RN.

```
OriginateRN ::= ABSTRACT-OPERATION
  ARGUMENT SET {
    envelope      [0] MessageSubmissionEnvelope,
    content       [1] RN }
  RESULT SET {
    submission-identifier [0] MessageSubmissionIdentifier,
    submission-time      [1] MessageSubmissionTime }
  ERRORS {
    SubscriptionError,
    RecipientImproperlySpecified }
```

An RN shall be originated only by an actual recipient of the subject IPM of whom an RN is requested by means of the notification-requests component of the subject IPM's subject recipient specifier.

The user shall not have previously originated an RN in response to the subject IPM, by means of either the present abstract operation or auto-acknowledgment.

This abstract operation has the following arguments:

- a) **envelope** (M): A message submission envelope, whose makeup the MTS abstract service defines. The UA supplies all but the following envelope components, which the user provides:
 - i) The desired per-message options (i.e., priority, per-message indicators, and extensions). Implicit conversion shall be prohibited, priority that of the subject IPM.
 - ii) The O/R names of the preferred recipients and the per-recipient options (i.e., explicit conversion and extensions) desired for each. Reports shall not be requested.
- b) **content** (M): The RN being originated.

This abstract operation has the following results:

- 1) **submission-identifier** (M): The message submission identifier the MTS assigns to the submission.
- 2) **submission-time** (M): The date and time the message was directly submitted.

12.2 Reception abstract operations

The abstract operations available at a reception port are invoked by the IPMS and performed by the user.

```
reception PORT
  SUPPLIER INVOKES {
    ReceiveReport,
    ReceiveIPM,
    ReceiveRN,
    ReceiveNRN }
  ::= id-pt-reception
```

Note 1 – As abstractly defined, the IPMS provides no storage for received messages because whether or not it does so for a particular user has no impact upon that user's ability to communicate with other users. Thus the provision of storage is a local matter.

Note 2 – Elaborating upon the above, the *receive IPM* abstract operation, e.g. expels an IPM from the IPMS because its purpose is to clarify the meaning of the receipt transmittal step. In contrast, the capabilities of a user to whom storage for received messages is provided might include a “Display IPM” command that enables the user to view the delivered (and perhaps already received) IPM whose IPM identifier he specifies, and that allows him to do so any number of times by repeatedly invoking the command. The first, but not subsequent uses of the command to view a particular IPM represents the concrete realization of the receive IPM abstract operation in such an implementation.

12.2.1 Receive report

The **receive report** abstract operation receives a report.

```
ReceiveReport ::= ABSTRACT-OPERATION
  ARGUMENT SET {
    envelope           [0] ReportDeliveryEnvelope,
    undelivered-object [1] InformationObject OPTIONAL }
  RESULT
  ERRORS { }
```

The report received may concern any of the following previously originated by the report's recipients:

- a) A probe concerning a message whose content was an IPM that was originated with the originate probe abstract operation.
- b) A message whose content was an NRN originated as a result of auto-discard or auto-forward.
- c) A message whose content was an RN that was originated with the originate RN abstract operations by *auto-acknowledgment*.
- d) A message whose content was an IMP that was originated with the originate IPM abstract operation or by *auto-forwarding*.

This abstract operation has the following arguments:

- 1) **envelope** (M): A report delivery envelope, whose make up the MTS abstract service defines.
- 2) **undelivered-object** (C): The content of the message whose status is being reported. An IPM or IPN.

If the report was provoked by a previous originate probe abstract operation invocation, this conditional agreement shall be absent. If the report was provoked by a previous originate IPM abstract operation invocation, the argument shall be present if, and only if, content return was requested. Otherwise (i.e., if the report was provoked by an IPN), the argument shall be absent.

This abstract operation has no results.

12.2.2 Receive IPM

The **receive IPM** abstract operation receives a message whose content is an IPM.

```
ReceiveIPM ::= ABSTRACT-OPERATION
  ARGUMENT SET {
    envelope [0] MessageDeliveryEnvelope,
    content  [1] IPM }
  RESULT
  ERRORS { }
```

This abstract operation has the following arguments:

- a) **envelope** (M): The message's delivery envelope.
- b) **content** (M): The IPM that is the message's content.

This abstract operation has no results.

12.2.3 *Receive RN*

The **receive RN** abstract operation receives a message whose content is an RN. The RN is provoked by an IPM originated with the originate IPM abstract operation.

```
ReceiveRN ::= ABSTRACT-OPERATION
  ARGUMENT SET {
    envelope      [0] MessageDeliveryEnvelope,
    content       [1] RN }
  RESULT
  ERRORS { }
```

This abstract operation has the following arguments:

- a) **envelope** (M): The message's delivery envelope.
- b) **content** (M): The RN that is the message's content.

This abstract operation has no results.

12.2.4 *Receive NRN*

The **receive NRN** abstract operation receives a message whose content is an NRN. The NRN is provoked by an IPM originated with the originate IPM abstract operation.

```
ReceiveNRN ::= ABSTRACT-OPERATION
  ARGUMENT SET {
    envelope      [0] MessageDeliveryEnvelope,
    content       [1] NRN }
  RESULT
  ERRORS { }
```

This abstract operation has the following arguments:

- a) **envelope** (M): The message's delivery envelope.
- b) **content** (M): The NRN that is the message's content.

This abstract operation has no results.

12.3 *Management abstract operations*

The abstract operations available at a management port are invoked by the user and performed by the IPMS.

```
management PORT
  CONSUMER INVOKES {
    ChangeAutoDiscard,
    ChangeAutoAcknowledgment,
    ChangeAutoForwarding }
  ::= id-pt-management
```

12.3.1 *Change auto-discard*

The **change auto-discard** abstract operation enables or disables **auto-discard**, the automatic discard by the IPMS of expired or obsolete IPMs delivered to, but not yet received by the user.

```
ChangeAutoDiscard; ::= ABSTRACT-OPERATION
  ARGUMENT SET {
    auto-discard-expired-IPMs  [0] BOOLEAN,
    auto-discard-obsolete-IPMs [1] BOOLEAN }
  RESULT
  ERRORS { }
```

When it auto-discards an IPM, the IPMS originates an NRN on the user's behalf if, and only if, one was requested of him by means of the notification-requests component of the subject recipient specifier.

This abstract operation has the following arguments:

- a) **auto-discard-expired-IPMs** (M): Whether or not expired IPMs are to be auto-discarded. A Boolean.
- b) **auto-discard-obsolete-IPMs** (M): Whether or not obsolete IPMs are to be auto-discarded. A Boolean.

This abstract operation has no results.

12.3.2 *Change auto-acknowledgment*

The **change auto-acknowledgment** abstract operation enables or disables **auto-acknowledgment**, the automatic origination of RNs by the IPMS on the user's behalf. Such origination occurs upon delivery of IPMs that request RNs of the user by means of the notification-requests components of their subject recipient specifiers.

```
ChangeAutoAcknowledgment ::= ABSTRACT-OPERATION
  ARGUMENT SET {
    auto-acknowledge-IPMs           [0] BOOLEAN,
    auto-acknowledge-suppl-receipt-info [1]
    SupplementaryInformation OPTIONAL }
  RESULT
  ERRORS {
    SubscriptionError }
```

This abstract operation has the following arguments:

- a) **auto-acknowledge-IPMs** (M): Whether or not IPMs are to be auto-acknowledged. A Boolean.
- b) **auto-acknowledge-suppl-receipt-info** (C): The suppl receipt info receipt field of each RN provoked by auto-acknowledgment.

This conditional argument shall be present if, and only if, the auto-acknowledge-IPMs argument has the value *true*.

This abstract operation has no results.

12.3.3 *Change auto-forwarding*

The **change auto-forwarding** abstract operation enables or disables **auto-forwarding**, the automatic forwarding of IPMs by the IPMS to pre-specified users or DLs. Such forwarding occurs upon delivery of the IPMs.

```
ChangeAutoForwarding ::= ABSTRACT-OPERATION
  ARGUMENT SET {
    auto-forward-IPMs           [0] BOOLEAN,
    auto-forward-recipients     [1] SEQUENCE OF ORName OPTIONAL,
    auto-forward-heading        [2] Heading OPTIONAL,
    auto-forward-comment        [3] AutoForwardComment OPTIONAL }
  RESULT
  ERRORS {
    SubscriptionError,
    RecipientImproperlySpecified }
```

The body of each IPM the IPMS originates as a result of auto-forwarding comprises a single body part of type message. The content of the message represented by that body part is the forwarded IPM.

When it auto-forwards an IPMS, the IPMS originates an NRN on the user's behalf if, and only if, one was requested of him by means of the notification-requests component of the subject recipient specifier.

This abstract operation has the following arguments:

- a) **auto-forward-IPMs** (M): Whether or not IPMs are to be auto-forwarded. A Boolean.
- b) **auto-forward-recipients** (C): The users or DLs to which IPMs are to be auto-forwarded. A sequence of O/R names.

This conditional argument shall be present if, and only if, the auto-forward-IPMs argument has the value *true*.

- c) **auto-forward-heading** (C): The heading that is to be used for each forwarding IPM. Its auto-forwarded heading field shall have the value *true*.

This conditional argument shall be present if, and only if, the auto-forward-IPMs argument has the value *true*.

- d) **auto-forward-comment** (C): The value that is to be supplied as the auto-forward comment non-receipt field of each NRN conveyed to the originator of an auto-forwarded IPM.

The conditional argument shall be present if, and only if, the auto-forward-IPMs argument has the value *true*.

This abstract operation has no results.

Note - This abstract operation is intended to define the essence of auto-forwarding, sophisticated auto-forwarding capabilities, e.g., like those of an MS.

13 Abstract errors

The abstract errors that may be reported in response to the invocation of the abstract operations available at origination, reception, and management ports are defined and described below or as part of the MTS abstract service definition.

Note - The set of abstract errors represented below is intended to be illustrative, rather than exhaustive.

13.1 Subscription error

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

```
SubscriptionError ::= ABSTRACT-ERROR
    PARAMETER SET {
        problem [0] SubscriptionProblem }
```

This abstract error has the following parameters:

- a) **problem** (M): The subscription-related problem encountered.

```
SubscriptionProblem ::= ENUMERATED {
    ipms-eos-not-subscribed(0),
    mts-eos-not-subscribed (1) }
```

This parameter may assume any one of the following values:

- i) *IPMS-eos-not-subscribed*: An IPMS element of service is not subscribed.
- ii) *MTS-eos-not-subscribed*: An MTS element of service is not subscribed.

13.2 Recipient improperly specified

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

This abstract error is defined by the MTS abstract service.

14 Other capabilities

In addition to the capabilities embodied in the IPMS abstract service, defined above, the IPMS shall transparently extend to each use the other MS and MTS capabilities identified below. (The enumeration of these capabilities necessarily anticipates the fact, stated in § 16, that MSs and the MTS are among the IPMS' component parts.)

The following additional capabilities shall be provided:

- a) *Submission*: Capabilities of the MS' or MTS' submission port not embodied in the IPMS abstract service e.g., the ability to cancel delivery of a previously originated message whose content is an IPM (but not an RN), if deferred delivery was selected.
- b) *Delivery*: Capabilities of the MTS' delivery port not embodied in the IPMS abstract service, e.g., the ability to temporarily control the kinds of information objects the MTS conveys to the user's *UA*.
- c) *Administration*: The capabilities of the MS's or MTS's administration port.
- d) *Retrieval*: The capabilities of the MS' retrieval port.

In addition to the above and as a local matter, the IPMS may provide to users additional capabilities neither defined nor limited by this Recommendation. Among such capabilities are those of the directory.

Note – The required capabilities of this clause are excluded from the formal definition of the IPMS abstract service for purely pragmatic reasons, in particular, because their inclusion would largely and needlessly reproduce the definitions of the MS and MTS abstract operations upon which the capabilities are based.

SECTION 4 – ABSTRACT SERVICE PROVISION

15 Overview

This section specifies how the IPMS provides the IPMS abstract service to users.

This section covers the following topics:

- a) secondary object types,
- b) secondary port types,
- c) user agent operation,
- d) message store operation,
- e) message contents,
- f) port realization,
- g) conformance.

16 Secondary object types

The IPMS can be modeled as comprising lesser objects which interact with one another by means of (additional) ports.

```

ipms-refinement REFINE ipms AS
  mTS
    submission [S] PAIRED WITH ipms-ua, ipms-ms
    delivery [S] PAIRED WITH ipms-ua, ipms-ms
    administration [S] PAIRED WITH ipms-ua, ipms-ms
  ipms-ua RECURRING
    origination [S] VISIBLE
    reception [S] VISIBLE
    management [S] VISIBLE
  ipms-ms RECURRING
    submission [S] PAIRED WITH ipms-ua
    retrieval [S] PAIRED WITH ipms-ua
    administration [S] PAIRED WITH ipms-ua
  tlma RECURRING
    origination [S] VISIBLE
    reception [S] VISIBLE
    management [S] VISIBLE
  tlxau RECURRING
    origination [S] VISIBLE
    reception [S] VISIBLE
    management [S] VISIBLE
  pdau RECURRING
    reception [S] VISIBLE
  ::= id-ref-secondary
  
```

These lesser objects are referred to as the **secondary objects** of interpersonal messaging. They include a single, central object, the MTS, and numerous peripheral objects: *interpersonal messaging system user agents (IPMS UAs)*, *interpersonal messaging system message stores (IPMS MSs)*, *telematic agents (TLMAs)*, *telex access units (TLXAU)s*, and PDAUs.

The structure of the IPMS is depicted in Figure 4/X.420. As shown by Figure 4/X.420, *IPMS UAs*, *TLMAs*, *TLXAU)s*, and PDAUs are the instruments by means of which the IPMS provides the IPMS abstract service to users.

The secondary object types are defined and described below. The types of ports by means of which they interact are discussed in § 17.

Note 1 – The refinement above encompasses all possible interconnection of all possible objects. It ignores the possible absence of objects of a particular type (e.g., PDAU), and specific logical configurations of the *IPMS MS*. The latter are identified in Recommendation X.402.

Note 2 – Recommendation T.330 effectively extends the abstract service of Interpersonal Messaging by its definition of a *miscellanea* port, which is not shown in the figure. See the note in § 16.3.

Note 3 – The MTS supplies import and export ports. However, since those ports are not formally defined (in Recommendation X.411), they are not included in the formal refinement above.

16.1 *Interpersonal messaging system user agent*

An **interpersonal messaging system user agent (IPMS UA)** is a UA tailored so as to better assist a single user to engage in interpersonal messaging. It helps him originate, receive, or both originate and receive messages containing information objects of the types defined in section two.

```

ipms-us OBJECT
  PORTS {
    origination[S],
    reception      [S],
    management     [S],
    submission     [C],
    delivery       [C],
    retrieval      [C],
    administration [C] }
 ::= id-ot-ipms-ua
  
```

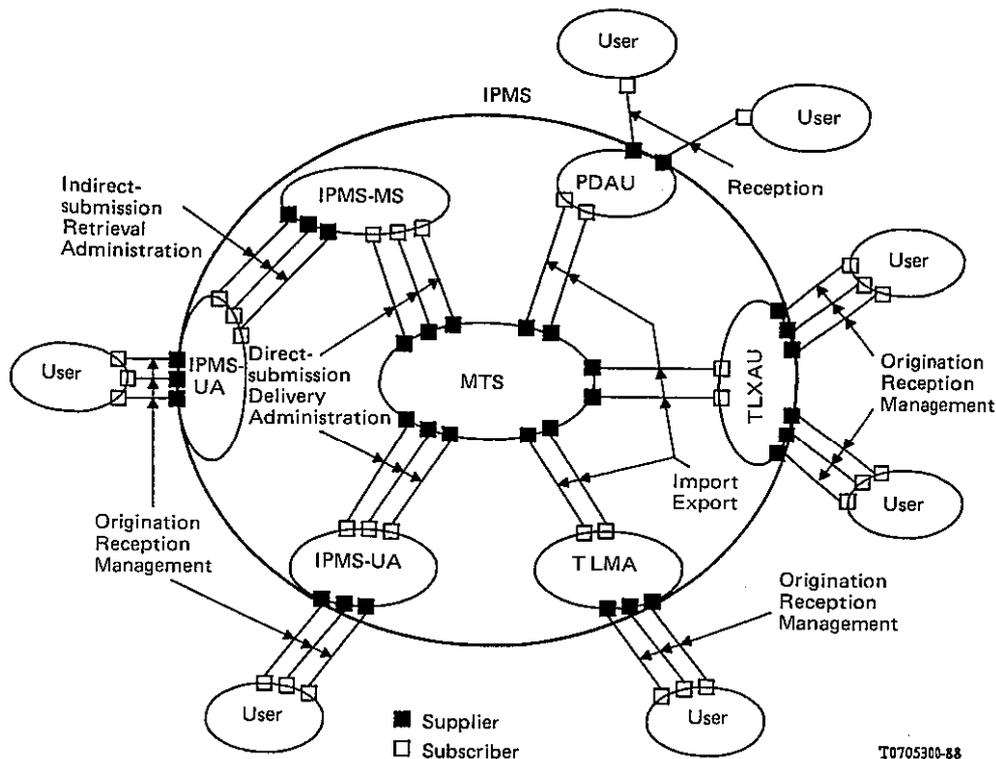


FIGURE 4/X.420
The interpersonal messaging system

The IPMS comprises any number of IPMS UAs.

Note – For brevity, the term “UA” is used throughout the rest of this Recommendation with the meaning of “IPMS UA”.

16.2 *Interpersonal messaging system message store*

An **interpersonal messaging system message store (IPMS MS)** is an MS tailored so as to better assist a single UA engage in interpersonal messaging. It helps it submit, take delivery of, or both submit and take delivery of messages containing information objects of the types defined in Section 2.

```
ipms-ms OBJECT
  PORTS {
    submission      [S],
    retrieval       [S],
    administration  [S],
    submission      [C],
    delivery        [C],
    administration  [C] }
  ::= id-ot-ipms-ms
```

The IPMS comprises any number of IPMS MSs.

Note – For brevity, the term “MS” is used throughout the rest of this Recommendation with the meaning of “IPMS MS”.

16.3 *Telematic agent*

A **telematic agent (TLMA)** is an AU that helps a single indirect user engage in interpersonal messaging from a telematic terminal, along with that terminal and the network that joins the two. A TLMA helps the user originate, receive, or both originate and receive messages containing information objects of the types defined in Section 2.

```
tlma OBJECT
  PORTS {
    origination[S],
    reception   [S],
    management  [S],
    miscellanea [S] }
  ::= id-ot-tlma
```

The IPMS comprises any number of TLMAAs.

Note 1 – A TLMA consumes import and export ports. However, since those ports are not formally defined (in Recommendation X.411), they are not included in the formal definition of TLMA above.

Note 2 – A TLMA's *miscellanea* port is defined in Recommendation T.330. It is not part of the IPMS abstract service in its most general form, which is the subject of this Recommendation. Rather it embodies capabilities available only to a TLMA user. For this reason, it is not considered further here and is not included in the formal refinement of the IPMS found in § 16.

16.4 *Telex access unit*

A **telex access unit (TLXAU)** is an AU that helps any number of indirect users engage in interpersonal messaging from telex terminals. It helps them originate, receive, or both originate and receive messages containing information objects of the types defined in section two.

```
tlxau OBJECT
  PORTS {
    origination[S],
    reception   [S],
    management  [S] }
  ::= id-ot-tlxau
```

The IPMS comprises any number of TLXAUs.

Note – A TLXAU consumes import and export defined (in Recommendation X.411) they are not included in the formal definition of TLXAU above.

16.5 *Physical delivery access unit*

In the present context, a PDAU helps any number of indirect users engage in interpersonal messaging by means of a PDS. It helps them receive (but not originate) messages containing information objects of the types defined in section two.

```
pdau OBJECT
  PORTS {
    reception          [S] }
  ::= id-ot-pdau
```

The IPMS comprises any number of PDAUs.

Note - A PDAU consumes import and export ports. However, since those ports are not formally defined (in Recommendation X.411), they are not included in the formal definition of PDAU above.

16.6 *Message transfer system*

In the present context, the MTS conveys information objects of the types defined in Section 2 between UAs, MSs, TLMAs, and AUs.

The IPMS comprises a single MTS.

17 **Secondary port types**

The secondary objects of interpersonal messaging are joined to and interact with one another by means of ports. These ports, which MSs and the MTS supply, are referred to as the **secondary ports** of interpersonal messaging. They are of the types identified below.

The capabilities embodied in one submission, one retrieval, and one administration port constitute the MS abstract service. They are defined in Recommendation X.413.

The capabilities embodied in one submission, one delivery, and one administration port constitute the MTS abstract service. They are defined in Recommendation X.411.

Note – By means of the abstract bind operation which guards its ports, an MS or the MTS typically authenticates another secondary object before offering its abstract service to that object.

17.1 *Submission*

In the present context, a submission port is the mean by which a UA (directly or indirectly) or an MS (directly) submits probes concerning and messages containing information objects of the types defined in section two.

An MS supplies one submission port to its UA.

The MTS supplies one submission port to each UA configured without an MS and to each MS.

17.2 *Delivery*

In the present context, a delivery port is the means by which a UA or MS takes delivery of reports concerning and messages containing information objects of the types defined in Section 2.

The MTS supplies one delivery port each UA configured without an MS and to each MS.

17.3 *Retrieval*

In the present context, a retrieval port is the means by which a UA retrieves reports concerning and messages containing information objects of the types defined in Section 2.

An MS supplies one retrieval port to its UA.

17.4 *Administration*

In the present context, an administration port is the means by which a UA changes information about itself or its user on file with its MS, or a UA or MS changes such information on file with the MTS.

An MS supplies one administration port to its UA.

The MTS supplies one administration port to each UA configured without an MS and to each MS.

17.5 *Import*

In the present context, an import port is the means by which the MTS imports reports concerning and messages containing information objects of the types defined in Section 2.

The MTS supplies one import port to each AU (or TLMA).

17.6 *Export*

In the present context, an export port is the means by which the MTS exports probes concerning and messages containing information objects of the types defined in Section 2.

The MTS supplies one export port to each AU (or TLMA).

18 **User agent operation**

A UA must employ the MTS in a particular way in order to (correctly) provide the IPMS abstract service to its user. If the user is equipped with an MS, the latter contributes to the provision of the abstract service and, therefore, is subject to the same rules.

The rules that govern the operation of a UA (and MS) are the subject of the present clause. The operation of a TLMA or AU is beyond the scope of this Recommendation.

Note 1 – It is for historical reasons that the Recommendation that defines the IPMS abstract service also specifies how a UA (and MS), but not a TLMA or AU, provides it.

Note 2 – The purpose of this clause is not to dictate or constrain the implementation of a real UA unnecessarily, but rather to clarify the meaning and intended effect of the IPMS abstract service.

18.1 *State variables*

The operation of a UA is described below with the aid of *state variables*. A **state variable** is an information item whose value records the results of the UA's past interactions with its user and influences future interactions. State variables are common to (i.e., shared by) the UA's origination, reception, and management ports.

The UA maintains each state variable continuously, i.e., throughout its user's IPMS subscription. Each Boolean state variable is assigned the value *false* when the subscription commences. The initial values of other state variables are immaterial and therefore unspecified.

The UA alters its state variables when performing or invoking abstract operations. It consults them in determining how to perform, or whether or how to invoke abstract operations. Their values (if any) transcend the binding and unbinding of ports.

Note – State variables are pedagogic devices not intended to constrain the implementation of a real UA unnecessarily. In particular, a UA need not maintain run-time data structures corresponding to the state variables if the behaviour required of it can be assured in another way.

18.2 *Performance of origination operations*

A UA shall perform the abstract operations it makes available at its origination port as prescribed below. The UA alters none of its state variables in the performance of these particular operations.

In the performance of these operations, the UA invokes the following abstract operations of the MTS abstract service (which, for the remainder of this paragraph, are unqualified as to their source):

- a) probe submission,
- b) message submission.

Note – In response to the invocation of these abstract operations, a UA reports abstract errors as appropriate. Specification of the precise circumstances under which each abstract error should be reported is beyond the scope of this Recommendation.

18.2.1 *Originate probe*

A UA shall perform the originate probe abstract operation by invoking probe submission with the arguments indicated below, and by returning to its user the results indicated below.

The arguments of probe submission shall be as follows:

- a) *Envelope*: The components of this argument that constitute per-probe fields shall be as follows;
 - i) *Originator-name*: The O/R name of the UA's user.
 - ii) *Content-type*, *content-length*, and *original-encoded-information-types*: Determined from originate probe's content argument as specified in §§ 20.2 to 20.4.
 - iii) *Content-identifier*: Specified or omitted as a local matter.

The components of this argument that constitute per-recipient fields shall be as specified by originate probe's envelope argument.

The result of originate probe shall be as follows:

- 1) *Submission-identifier*: Probe submission's probe-submission-identifier result.
- 2) *Submission-time*: Probe submission's probe-submission-time result.

Note 1 – The UA shall ignore all properties of originate probe's content argument other than those mentioned above.

Note 2 – How the UA employs probe submission's content-identifier result is a local matter.

18.2.2 *Originate IPM*

A UA shall perform the originate IPM abstract operation by invoking message submission with the arguments indicated below, and by returning to its user the results indicated below.

The arguments of message submission shall be as follows:

- a) *Envelope*: The components of this argument that constitute per-message fields shall be as follows; those not explicitly mentioned below shall be as specified by originate IPM's envelope argument:
 - i) *Originator-name*: The O/R name of the UA's user.
 - ii) *Content-type* and *original-encoded-information-types*: Determined from originate IPM's content argument as specified in §§ 20.4 and 20.4, respectively.
 - iii) *Content-identifier*: Specified or omitted as a local matter.

The components of this argument that constitute per-recipient fields shall be as specified by originate IPM's envelope argument.

- b) *Content*: Determined from originate IPM's content argument (identified as an IPM) as specified in § 20.1.

If the blind copy recipients heading field of the IPM identifies one or more users and DLS, the UA shall invoke message submission multiple times, upon each occasion varying the heading field so as to comply with the information hiding requirements of § 7.2.6.

The results of originate IPMS shall be as follows:

- 1) *Submission-identifier*: Message submission's message-submission-identifier result.
- 2) *Submission-time*: Message submission's message-submission-time result.

Note 1 – How the UA employs message submission's content-identifier result is a local matter.

Note 2 – The inclusion of message submission's extensions result among originate IPM's results in proper and for further study.

18.2.3 *Originate RN*

A UA shall perform the originate RN abstract operation by invoking message submission with the arguments indicated below, and by returning to its user the results indicated below.

The arguments of message submission shall be as follows:

- a) *Envelope*: The components of this argument that constitute per-message fields shall be as follows; those not explicitly mentioned below shall be as specified by originate RN's envelope argument:

- i) *Originator-name*: The O/R name of the UA's user.
- ii) *Content-type* and *original-encoded-information-types*: Determined from the RN as specified in §§ 20.2 and 20.4, respectively.
- iii) *Content-identifier*: Specified or omitted as a local matter.
- iv) *Deferred-delivery-time*: Omitted.

The components of this argument that constitute per-recipient fields shall be as specified by originate RN's envelope argument.

- b) *Content*: Determined from originate RN's content argument (identified as an RN) as specified in § 20.1.

The result of originate RN shall be as follows:

- 1) *Submission-identifier*: Message submission's message-submission-identifier result.
- 2) *Submission-time*: Message submission's message-submission-time result.

Note 1 – How the UA employs message submission's content-identifier result is a local matter.

Note 2 – The inclusion of message submission's extensions result among originate RN's results is proper and for further study.

18.3 *Performance of management operations*

A UA shall perform the abstract operations it makes available at its management port as specified below. The UA alters one or more of its state variables (see below) in the performance of each operation.

Note – In response to the invocation of these abstract operations, a UA reports abstract errors as appropriate. Specification of the precise circumstances under which each abstract error should be repeated is beyond the scope of this Recommendation.

18.3.1 *Change auto-discard*

To assist it in providing this abstract operation, a UA maintains the following state variables:

- a) **Auto-discard-expired-IPMs**: A Boolean that indicates whether or not auto-discard is in effect for expired IPMs.
- b) **Auto-discard-obsolete-IPMs**: A Boolean that indicates whether or not auto-discard is in effect for obsolete IPMs.

A UA shall perform the change auto-discard abstract operation by recording the values of the auto-discard-expired-IPMs and auto-discard-obsolete-IPMs arguments in the correspondingly named state variables.

18.3.2 *Change auto-acknowledgment*

To assist it in providing this abstract operation, a UA maintains the following state variables:

- a) **auto-acknowledge-IPMs**: A Boolean that indicates whether or not auto-acknowledgment is in effect.
- b) **auto-acknowledge-suppl-receipt-info**: The suppl receipt info field of each RN provoked by auto-acknowledgment.

A UA shall perform the change auto-acknowledgment abstract operation by recording the value of the auto-acknowledge-IPMs argument in the correspondingly named state variable. If the value is *true*, it also shall record the value of the Auto-acknowledge-suppl-receipt-info argument in the correspondingly named state variable.

18.3.3 *Change auto-forwarding*

To assist it in providing this abstract operation, a UA maintains the following state variables:

- a) **auto-forward-IPMs**: A Boolean that indicates whether or not auto-forwarding is in effect.
- b) **auto-forward-recipients**: A sequence of O/R names that identify the users and DLs to which IPMs are being auto-forwarded.
- c) **auto-forward-heading**: The heading of each forwarding IPM provoked by auto-forwarding. Its auto-forwarded field has the value *true*.
- d) **auto-forward-comment**: The auto-forward comment non-receipt field of each NRN conveyed to the originator of an auto-forwarded IPM.

A UA shall perform the change auto-forwarding abstract operation by recording the value of the auto-forward-IPMs argument in the correspondingly named state variable. If the value is *true*, it also shall record the values of the auto-forward-recipients, auto-forward-heading, and auto-forward-comment arguments in the correspondingly named state variables.

18.4 *Invocation of reception operations*

A UA shall invoke that abstract operations available at its reception port as specified below. The UA alters none of its state variables in connection with its invocation of these operations.

The UA invokes these operations in response to the MTS' invocation of the following abstract operations of the MTS abstract service (which, for the remainder of this paragraph, are unqualified as to their source):

- a) report delivery,
- b) message delivery.

Note – The abstract operation of a reception port report no errors.

18.4.1 *Receive report*

Whenever the MTS invokes report delivery at a UA's delivery port, the UA shall invoke the receive report abstract operation with the following arguments:

- a) *Envelope*: Report delivery's envelope argument.
- b) *Undelivered-object*: Determined from report delivery's returned-content argument as specified in § 20.1.

Note – How the UA employs the content-identifier component of report delivery's envelope argument is a local matter.

18.4.2 *Receive IPM*

Whenever the MTS invokes message delivery at a UA's delivery port, and its content argument encodes an IPM as specified in § 20.1, the UA shall invoke the receive IPM abstract operation with the following arguments, provided that the message is subject to neither auto-forwarding nor auto-discard (see § 18.5):

- a) *Envelope*: Message delivery's envelope argument.
- b) *Content*: Determined from message delivery's content argument as specified in § 20.1 (but no longer marked as an IPM).

18.4.3 *Receive RN*

Whenever the MTS invokes message delivery at a UA's delivery port, and its content argument encodes in RN as specified in § 20.1, the UA shall invoke the receive RN abstract operation with the following arguments:

- a) *Envelope*: Message delivery's envelope argument.
- b) *Content*: Determined from message delivery's content argument as specified in § 20.1 (but no longer marked as an RN).

18.4.4 *Receive NRN*

Whenever the MTS invokes message delivery at a UA's delivery port, and its content argument encodes an NRN as specified in § 20.1, the UA shall invoke the receive NRN abstract operation with the following arguments:

- a) *Envelope*: Message delivery's envelope argument.
- b) *Content*: Determined from message delivery's content argument as specified in § 20.1 (but no longer marked as an NRN).

18.5 *Internal procedures*

A UA shall perform as specified below the internal procedures of auto-discard, -acknowledgment, and -forwarding in ultimate fulfilment of the abstract operations available at its management port.

The procedures involve the following abstract operations of the MTS abstract service (which, for the remainder of this paragraph, are unqualified as to their source):

- a) message submission,
- b) message delivery.

As implied by the above, in the course of the procedures, the UA has occasion to invoke message submission. What it does with the results of this abstract operation is a local matter.

The UA shall consider as a candidate for each procedure individually every message for which all of the following conditions hold:

- a) The MTS has conveyed the message to the UA by invoking message delivery at the UA's delivery port.
- b) The UA has not conveyed the message to the user by invoking receive IPM at the user's reception port.
- c) The message contains an IPM (rather than an IPN).

Note – With reference to item b) above, the message might be detained in the UA, e.g., as might be typical, because of the user's unavailability.

18.5.1 *Auto-discard*

The UA shall subject to auto-discard each candidate message with respect to whose content either of the following conditions holds:

- a) The auto-discard-expires-IPMs state variable has the value *true* and the date and time denoted by the IPM's expiry time field have past.
- b) The auto-discard-obsolete-IPMs state variable has the value *true* and another candidate IPM identifies the present candidate IPM by means of its obsoleted IPMs heading field.

The UA shall auto-discard each such message as follows.

18.5.1.1 *Discard of IPM*

The UA shall discard the IPM, so as to never convey it to the user.

18.5.1.2 *Construction of NRN*

The UA shall construct an NRN if, and only if, one is requested by means of the notification-requests component of the IPM's subject recipient specifier.

The NRN shall have the common fields prescribed for auto-acknowledgment (see § 18.5.2.1).

The NRN shall have the following receipt fields:

- a) Non-receipt reason: The value *ipm-discarded*.
- b) Discard reason: The value *ipm-expired* or *ipm-obsolete*, whichever applies. If both apply, either value may be specified.
- c) Auto-forward comment: Omitted.
- d) Returned IPM: If the IPM's return is requested by means of the notification-requests component of its subject recipient specifier, and the converted-encoded-information-types component of message delivery's envelope argument is absent, IPM. Omitted otherwise.

18.5.1.3 *Submission of NRN*

The UA shall submit the NRN (if any) above by invoking message submission. Its envelope argument shall be as prescribed for auto-acknowledgment (see § 18.5.2.2), its content argument determined from the NRN as specified in § 20.1.

18.5.2 *Auto-acknowledgment*

The UA shall subject to auto-acknowledgment each candidate message with respect to whose content the following condition holds:

- a) The auto-acknowledgment state variable has the value *true* and the IPM requests an RN of the UA's user by means of the notification-requests component of the IPM's subject recipient specifier.

The UA shall auto-acknowledge each such message as follows.

18.5.2.1 *Construction of RN*

The UA shall construct an RN.

The RN shall have the following common fields:

- a) *Subject IPM*: The IPM's This IPM heading field.
- b) *IPN originator*: Specified or omitted as a local matter (but, of course, in accordance with § 8.1.2).
- c) *IPM preferred recipient*: The recipient component of the IPM's subject recipient specifier, unless its formal-name component is the O/R name of the UA's user, in which case the field shall be omitted.
- d) *Conversion EITs*: Converted-encoded-information-types component of message delivery's envelope argument.

The RN shall have the following receipt field:

- a) *Receipt time*: The current date and time.
- b) *Acknowledgment mode*: The value automatic.
- c) *Suppl receipt info*: The auto-acknowledge-suppl-receipt-info state variable.

18.5.2.2 *Submission of RN*

The UA shall submit the RN above by invoking message submission with the following arguments:

- a) *Envelope*: The components of this argument shall be as prescribed for performance of the originate RN abstract operation with the following exceptions:
 - i) *Priority*: As specified by message delivery's envelope argument.
 - ii) *Per-message-indicators*: A local matter, except that *conversion-prohibited* shall be among the values specified.
 - iii) *Per-recipient-fields*: A single field whose recipient-name component shall be the originator-name component of message delivery's envelope argument. Reports shall not be requested.
- b) *Content*: Determined from the RN as specified in § 20.1.

18.5.3 *Auto-forwarding*

The UA shall subject to auto-forwarding every candidate message, provided that the auto-forward-IPMs state variable has the value *true*.

The UA shall auto-forward each such message as follows.

18.5.3.1 *Prevention of loops*

The UA shall suppress auto-forwarding if, and only if, the IPM to be forwarded itself contains a forwarding IPM that the UA previously created. Auto-forwarding shall be suppressed whether the forwarding IPM appears (directly) in a message bodypart of the IPM to be forwarded, or (nested) in a message body part of the IPM that appears in such a body part.

The UA shall consider itself to have created the forwarding IPM above (whose auto-forwarded heading fields has the value *true*) if, and only if, the originator-name component of the IPM's parameters component matches the O/R name of the UA's user.

Note – Auto-forwarding an IPM of the kind described above would constitute an auto-forwarding “loop”.

18.5.3.2 *Construction of IPM*

The UA shall construct a forwarding IPM whose heading is the auto-forward-heading state variable (its auto-forwarded field having the value *true*) and whose body comprises a single body part of type message.

The message body part shall have the following components:

- a) *Parameters*: The envelope argument of message delivery.
- b) *Data*: The IPM to be forwarded.

18.5.3.3 *Submission of IPM*

The UA shall submit the IPM it constructed above by invoking message submission with the following arguments:

- a) *Envelope*: The components of this argument shall be as follows:
 - i) *Originator-name*: The O/R name of the UA's user.
 - ii) *Content-type* and *original-encoded-information-types*: Determined from the IPM as specified in §§ 20.2 and 20.4.
 - iii) *Content-identifier*: Specified or omitted as a local matter.
 - iv) *Priority*: As specified by message delivery's envelope argument.
 - v) *Per-message-indicators* and *extensions*: A local matter.
 - vi) *Deferred-delivery-time*: Omitted.
 - vii) *Per-recipient-fields*: Their recipient-name components shall be the O/R names that make up the auto-forward-recipients state variable. Their other components are a local matter.
- b) *Content*: Determined from the IPM as specified in § 20.1.

18.5.3.4 *Construction of NRN*

The UA shall construct an NRN if, and only if, one is requested by means of the notification-requests component of the forwarded IPM's subject recipient specifier.

The NRN shall have the common fields prescribed for the performance of auto-acknowledgment.

The NRN shall have the following receipt fields:

- a) *Non-receipt reason*: The value ipm-auto-forwarded.
- b) *Discard reason*: Omitted.
- c) *Auto-forward comment*: The auto-forward-comment state variable.
- d) *Returned IPM*: If the IPM's return is requested by means of the notification-requests component of its subject recipient specifier, and the converted-encoded-information types component of message delivery's envelope argument is absent, the IPM. Omitted otherwise.

18.5.3.5 *Submission of NRN*

The UA shall submit the NRN (if any) above by invoking message submission. Message submission's envelope argument shall be as prescribed for auto-acknowledgment, its content argument determined from the NRN as specified in § 20.1.

19 **Message store operation**

An MS must perform certain interpersonal messaging-specific functions to qualify as an IPMS MS and thus distinguish itself from a generic MS. These functions are the subject of the present paragraph.

19.1 *Creation of information objects*

An IPMS shall satisfy the following requirements related to the information objects it maintains:

- a) The MS shall maintain a separate information object for each (message containing an) IPM or IPN that is delivered to it.
- b) The MS shall maintain as a separate information object not only each (message containing a) forwarding IPM (pursuant to Item a)) but also each (message containing a) forwarded IPM (recursively).
- c) The MS shall assign sequence numbers depth-first to the messages in the hierarchy formed by a forwarding IPM and its forwarded IPMs.

Example – If IPM A contains IPMs B and C among its body parts, and if IPM B contains IPMS D and E among its body parts, sequence numbers will be assigned to the IPMs in the following order: A, B, D, E, and C.

19.2 *Maintenance of attributes*

An IPMS MS shall satisfy the following requirements related to MS attributes:

- a) For each IPM or IPN it holds, the MS shall support the attributes of Annex C as specified therein.
- b) For each IPM it holds, the MS shall give the following meanings to the defined values of the MS-status attribute:
 - i) *new*: No attribute values have been conveyed to the UA.
 - ii) *listed*: At least one attribute value has been conveyed to the UA, and at least one body part has not been conveyed.
 - iii) *processed*: All body parts have been conveyed to the UA.
- c) For each IPN it holds, the MS shall give the following meanings to the defined values of the MS-status attribute:
 - i) *new*: No attribute values have been conveyed to the UA.
 - ii) *listed*: At least one attribute value has been conveyed to the UA, and at least one attribute other than Returned IPM has not been conveyed.
 - iii) *processed*: All attributes, with the possible exception of returned IPM, have been conveyed to the UA.
- d) The MS-status attribute shall reflect the state of affairs prior to an abstract operation invocation that alters its value.
- e) The content-type attribute of each (message containing an) IPM or IPN that is delivered to the MS shall have the value id-mct-p2-1984 or id-mct-p2-1988 (see Annex D), as appropriate, depending upon the content type of the delivered message (see § 20.2).

19.3 *Notification of non-receipt*

When it discards an IPM while performing the delete abstract operation of the MS abstract service, the MS shall submit a NRN if one is requested and the IPM's MS-status attribute has the value *listed*.

19.4 *Auto-forwarding*

An IPMS MS shall perform the auto-forward action of Recommendation X.413 as specified in § 18.5.3. It makes use of the other-parameters component of the auto-forward-registration argument of the register MS abstract operation of the MS abstract service. The data type of the other-parameters component is defined as follows:

```
Forwarded Info ::= SET {  
    auto-forwarding-comment [0] AutoForwardComment OPTIONAL,  
    cover-note [1] IA5TextBodyPart OPTIONAL,  
    this-ipm-prefix [2] PrintableString (SIZE  
    (1..ub-ipm-identifier-suffix)) OPTIONAL }
```

In addition, the MS shall satisfy the following requirements:

- a) Submit an NRN even if it retains a copy of the forwarded IPM.
- b) Draw the NRN's auto-forward comment field, if any, from the other-parameters component.
- c) Draw the cover-note, if any, to be included with the forwarded IPM, from the other-parameters component.
- d) Prefix the user-relative-identifier component of this IPM field of the forwarding IPM's heading with, if present, this-ipm-prefix.

Note – An (IPMS) MS performs neither auto-discard nor auto-acknowledgment, except possibly as a local matter.

19.5 *Manual forwarding*

An IPMS shall support the manual forwarding of a message using the forwarding-request extension of Recommendation X.413, as specified in § 6.6. The IPMS MS user may submit an IPM, including heading and body, using the message submission operation, and identify by using the forwarding-request extension a message that is already in the MS, and to be combined with the submitted message body for forwarding to the message's recipient.

The submitted message body and the forwarded message are then combined by inserting the forwarded message as a message body part into the submitted message body.

20 Message contents

As has already been seen, various secondary objects (e.g., UAs) have occasion to convey the information objects of section two as the contents of messages, as well as to convey probes concerning such messages. This paragraph specifies precisely how they shall do this.

The rules governing the transmittal of such messages and probes, and the semantics and abstract and transfer syntaxes of their contents, are called the **interpersonal messaging protocol (P2)**.

Note – The name “P2” reflects the historical fact that this was the second message handling protocol to be developed.

20.1 Content

A secondary object that submits a message containing an IPM or IPN shall supply as the octets of the octet string that constitutes the content of the message the result of encoding the InformationObject of section two in accordance with the basic encoding rules of Recommendation X.209.

20.2 Content type

A secondary object that submits a message containing an IPM or IPN shall select its content type as follows.

If the IPM or IPN satisfies all of the following constraints, the Integer 2 shall be specified:

- i) The heading (of an IPM) lacks the extensions field.
- ii) The body (of an IPM) lacks externally defined body parts.
- iii) The parameters element of any videotex body part: . . . of (an IPM) lacks the syntax member.
- iv) Every component of the IPM or IPN that is a value of a data type defined as part of the MTS abstract service meets the constraints of Recommendation X.411 (1984).

The types in question are those listed in the IMPORTS clause of the ASN.1 module defined in Annex E. The constraints in question are detailed in an annex of Recommendation X.419.

- v) The data element of any message body part (of an IPM) satisfies these same constraints (recursively).

Otherwise, the Integer 22 shall be specified.

Note 1 – The message content protocol (here) denoted by the Integer 2 is identical to that specified by Recommendation X.420 (1984) (as clarified by Version 6 of the *X.400-series Implementor's Guide*), except that the simple formattable document body part type, defined in the latter, is omitted from the former.

Note 2 – The Integer 2 is favoured, above, over the Integer 22 to foster interworking between systems conforming to this Recommendation and systems conforming (only) to Recommendation X.420 (1984).

Note 3 – The MTS does not convert between message content protocols. Thus it does not convert between P2 as defined by this Recommendation alone (and denoted by the Integer 22) and P2 as defined by both this Recommendation and Recommendation X.420 (1984) (and denoted by the Integer 2).

20.3 Content length

A secondary object that submits a probe concerning a message containing an IPM or IPN shall specify as the length of the message's content the size in octets of the encoding of the instance in question of the InformationObject of Section 2 (a choice of an IPM or an IPN) when the basic encoding rules of Recommendation X.209 are followed. If those rules permit several (e.g., both primitive and constructed) encodings of that InformationObject, the content length may reflect any one of them.

20.4 *Encoded information types*

A secondary object that submits a message containing an IPM or IPN shall specify the basic encoded information type (EITs) and non-basic parameters (NBPs) of the message as follows.

In the case of an IPN, the basic EITs shall be *unspecified*.

In the case of an IPM, the basic EITs and NBPs shall be specified in accordance with the following rules:

- a) *Multiple body parts*: The basic EITs (if any) and NBPs (if any) of the message shall comprise the logical union of the basic EITs and NBPs of the IPM's individual body parts, respectively.
- b) *(Forwarded) message body part*: The basic EITs (if any) and NBPs (if any) of a message body part shall be those of the forwarded message.
- c) *Externally defined body part*: An externally defined body part whose extended type corresponds to a basic type (see Annex B) shall be treated in the manner prescribed for the basic type.
Any other extended body part shall be handled as follows. If there corresponds to the type one or more externally defined EITs, they shall be specified. Otherwise, the *undefined* EIT shall be indicated. In either case, no NBPs shall be specified.
- d) *Basic body part*: The basic EITs (if any) and NBPs (if any) of an individual body part of type other than message and externally defined shall depend upon that body part type as specified in Table 2/X.420. A body part type for which the table specifies no basic EITs shall result in the setting of no bits in the basic EITs bit string.
- e) *Encrypted body part* : The effect of an encrypted body part upon the basic EITs and NBPs to be specified is for further study.

TABLE 2/X.420

Interpersonal messaging basic EITs and NBPs

Body part type	Basic EIT	NBPs
IA5 text	IA5 text	–
Voice	Voice	–
G3 facsimile	G3 facsimile	G3 facsimile
G4 class 1	G4 class 1	G4 class 1/mixed-mode
Teletex	Teletex	Teletex
Videotex	Videotex	–
Encrypted		
Message	(see text)	(see text)
Mixed-mode	Mixed-mode	G4 class 1/mixed-mode
Bilaterally defined	Undefined	–
Nationally defined	Undefined	–
Externally defined	(see text)	(see text)

21 Port realization

How an MS or the MTS concretely realizes the secondary ports it supplies is specified in Recommendation X.419.

How a UA, TLMA, or AU concretely realizes the primary ports it supplies is beyond the scope of this Recommendation.

Note 1 – A UA's user interface is a local matter. A wide variety of interfaces involving, e.g., a wide variety of input/output devices are possible.

Note 2 – A TLMA's realization of its primary ports is specified by Recommendation T.330.

Note 3 – An AU provides its primary ports by means of the particular communication system to which that AU provides access.

22 Conformance

The requirements a secondary object (excluding the MTS) and its implementor shall meet when the latter claims the former's conformance to this Recommendation are identified below. A number of the conformance requirements distinguish between *support upon origination* and *support upon reception*.

22.1 *Origination versus reception*

A UA, TLMA or AU shall be said to **support upon origination** a particular heading field, heading extension, basic body part type, or extended body part type if, and only if, it accepts, preserves, and emits, in full accord with this Recommendation, that particular heading field or extension, or body parts of that particular basic or extended type, whenever a user calls upon it to convey an IPM containing them to the MTS or the user's MS (the latter only in the case of a UA).

A UA, TLMA, or AU shall be said to **support upon reception** a particular heading field, heading extension, basic body part type, or extended body part type if, and only if, it accepts, preserves, and emits, in full accord with this Recommendation, that particular heading field or extension, or body parts of that particular basic or extended type, whenever the MTS or a user's MS (the latter only in the case of a UA) calls upon it to convey to the user an IPM containing them.

Note – In point of fact, a PDAU supports nothing upon origination because it is not a supplier of the origination port.

22.2 *Statement requirements*

The implementor of an IPMS, UA, IPMS MS, TLMA, or AU shall state the following. For each item below he shall make separate statements concerning conformance upon origination and conformance upon reception:

- a) The heading fields and heading extensions for which he claims conformance.
- b) The basic and extended body part types for which he claims conformance.
- c) In the case of an IPMS UA or IPMS MS, the interpersonal messaging-specific MS attributes for which it claims conformance.

22.3 *Static requirements*

An IPMS UA, IPMS MS, TLMA or AU shall satisfy the following static requirements:

- a) An IPMS UA, IPMS MS, TLMA, or AU shall implement the heading fields and heading extensions, and the basic and extended body part types for which conformance is claimed.
- b) An IPMS UA or IPMS MS shall support the interpersonal messaging-specific MS attributes for which conformance is claimed, but including as a minimum those designated mandatory in Annex C.
- c) An IPMS UA, IPMS MS, TLMA, or AU shall concretely realize its abstract ports as specified in § 21.
- d) An IPMS UA or IPMS MS shall be able to both submit and accept delivery of messages of both of the content types of § 20.2. A TLMA or AU shall be able to both import and export such messages.

22.4 *Dynamic requirements*

An IPMS UA, IPMS MS, TLMA, or AU shall satisfy the following dynamic requirements:

- a) An IPMS UA or IPMS MS shall follow the rules of operation specified in § 18 or § 19, respectively.
- b) An IPMS UA, IPMS MS, TLMA, or AU shall submit and accept delivery of messages whose contents are as specified in § 20.
- c) An IPMS UA, IPMS MS, TLMA, or AU shall register with the MTS its ability to accept delivery of messages of both of the content types of § 20.2.

ANNEX A
(to Recommendation X.420)

Heading extensions

This Annex is an integral part of this Recommendation.

This Annex defines all (presently defined) heading extensions.

A.1 *Incomplete copy*

The **incomplete copy** heading extension, by its presence, indicates that one or more body parts or heading fields are absent from the body of (the present instance of) the IPM. The extension comprises a null (by default).

Incomplete-copy HEADING-EXTENSION
 ::= id-hex-incomplete copy

If this extension is absent from the extension heading field, all body parts shall be considered present.

A.2 *Languages*

The **languages** heading extension identifies the languages used in the composition of the IPM's subject heading field and body. The extension comprises a set of zero or more printable strings, each one of the two-character language codes identified by ISO 639.2.

languages HEADING-EXTENSION
 VALUE SET OF Language
 ::= id-hex-languages
 Language ::= PrintableString (SIZE (2..2))

If this extension is absent from the extensions heading field or no languages are indicated, the languages shall be considered unspecified.

ANNEX B
(to Recommendation X.420)

Extended body part types

This Annex is an integral part of this Recommendation.

For each basic body part type, this Recommendation defines as follows an equivalent extended body part type.

ia5-text-body-part EXTENDED-BODY-PART-TYPE
 PARAMETERS IA5TextParameters IDENTIFIED BY id-ep-ia5-text
 DATA IA5TextData
 ::= id-et-ia5-text

voice-body-part EXTENDED-BODY-PART-TYPE
 PARAMETERS VoiceParameters IDENTIFIED BY id-ep-voice
 DATA VoiceData
 ::= id-et-voice

g3-facsimile-body-part EXTENDED-BODY-PART-TYPE
 PARAMETERS G3FacsimileParameters IDENTIFIED BY id-ep-g3-facsimile
 DATA G3FacsimileData
 ::= id-et-g3-facsimile

g4-class1-body-part EXTENDED-BODY-PART-TYPE
 DATA G4ClassBodyPart
 ::= id-et-g4-class1

teletex-body-part EXTENDED-BODY-PART-TYPE
 PARAMETERS TeletexParameters IDENTIFIED BY id-ep-teletex
 DATA TeletexData
 ::= id-et-teletex

```

videotex-body-part EXTENDED-BODY-PART-TYPE
    PARAMETERS      VideotexParameters IDENTIFIED BY id-ep-videotex
    DATA            VideotexData
    ::= id-et-videotex

encrypted-body-part EXTENDED-BODY-PART-TYPE
    PARAMETERS      EncryptedParameters IDENTIFIED BY id-ep-encrypted
    DATA            EncryptedData
    ::= id-et-encrypted

message-body-part EXTENDED-BODY-PART-TYPE
    PARAMETERS      MessageParameters IDENTIFIED BY id-ep-message
    DATA            MessageData
    ::= id-et-message

mixed-mode-body-part EXTENDED-BODY-PART-TYPE
    DATA            MixedModeBodyPart
    ::= id-et-mixed-mode

bilaterally-defined-body-part EXTENDED-BODY-PART-TYPE
    DATA            BilaterallyDefinedBodyPart
    ::= id-et-bilaterally-defined

nationally-defined-body-part EXTENDED-BODY-PART-TYPE
    DATA            NationallyDefinedBodyPart
    ::= id-et-nationally-defined

```

ANNEX C

(to Recommendation X.420)

Message store attributes

This Annex is an integral part of this Recommendation.

As described in Recommendation X.413, an MS maintains and provides access to certain attributes (e.g., the importance) of each information object it holds. An attribute comprises a type and, depending upon the type, one or more values. Attributes that may assume several values simultaneously (all pertaining to one object) are termed multi-valued; those that may assume just one value, single-valued. Some attributes pertain to information objects of all kinds, others only to those of certain kinds (e.g., those of Section 2).

This Annex defines the MS attributes specific to interpersonal messaging.

All of the attributes defined in this Annex, except those corresponding to extended body part types (which cannot be enumerated; see § C.3.6), are listed alphabetically, for reference, in the first column of Table C-1/X.420. This Table records their presence in a delivered message entry. None of them appears in either a delivered report entry or a returned content entry. See Recommendation X.413 for an elaboration of the Table's legend.

C.1 *Summary attributes*

Some attributes summarize an interpersonal messaging information object. These attributes are defined and described below.

C.1.1 *IPM entry type*

The **IPM entry type** attribute identifies an information object's type.

```

ipm-entry-type ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX IPMEntryType
    MATCHES FOR EQUALITY
    SINGLE VALUE
    ::= id-sat-ipm-entry-type

```

```

IPMEntryType ::= ENUMERATED {
    ipm(0),
    rn(1),
    nrn(2) }

```

This attribute may assume any one of the following values:

- a) *ipm*: The information object is an IPN.
- b) *rn*: The information object is an RN.
- c) *nrn*: The information object is an NRN.

An MS that supports this attribute shall maintain it for an information object that it holds if, and only if, that object is a message whose content is an IPM or IPN.

TABLE C-1/X.420

Summary of MS attributes

Attribute	V	L	P			L	S
			IPM	NRN	RN		
Acknowledgement mode	S	O	-	-	M	Y	Y
Authorizing users	M	O	C	-	-	Y	N
Auto-forward comment	S	O	-	C	-	Y	N
Auto-forwarded	S	O	C	-	-	Y	Y
Bilaterally defined body parts	M	O	C	-	-	N	N
Blind copy recipients	M	O	C	-	-	Y	N
Body	S	M	M	-	-	N	N
Conversion EITs	M	O	-	C	C	Y	N
Copy recipients	M	O	C	-	-	Y	N
Discard reason	S	O	-	C	-	Y	Y
Encrypted body parts	M	O	C	-	-	N	N
Encrypted data	M	O	C	-	-	N	N
Encrypted parameters	M	O	C	-	-	N	N
Expiry time	S	O	C	-	-	Y	N
Extended body part types	M	O	C	-	-	Y	Y
G3 facsimile body parts	M	O	C	-	-	N	N
G3 facsimile data	M	O	C	-	-	N	N
G3 facsimile parameters	M	O	C	-	-	N	N
G4 class 1 body parts	M	O	C	-	-	N	N
Heading	S	M	M	-	-	N	N
IA5 text body parts	M	O	C	-	-	N	N
IA5 text data	M	O	C	-	-	N	N
IA5 text parameters	M	O	C	-	-	N	N
Importance	S	O	C	-	-	Y	Y
Incomplete copy	S	O	C	-	-	Y	N
IPM entry type	S	M	M	M	M	Y	Y
IPM preferred recipient	S	O	-	C	C	Y	N
IPM synopsis	S	O	M	-	-	N	N
IPM originator	S	O	-	C	C	Y	N
Languages	M	O	C	-	-	Y	N
Message body parts	M	O	C	-	-	N	N
Message data	M	O	C	-	-	N	N
Message parameters	M	O	C	-	-	N	N
Mixed-mode body parts	M	O	C	-	-	N	N
Nationally defined body parts	M	O	C	-	-	N	N
Non-receipt reason	S	O	-	M	-	Y	Y
NRN requestors	M	O	C	-	-	Y	N
Obsoleted IPMs	M	O	C	-	-	Y	N
Originator	S	O	C	-	-	Y	N

TABLE C-1/X.420 (cont.)

Attribute	V	L	IPM	P NRN	RN	L	S
Primary recipients	M	O	C	-	-	Y	N
Receipt time	S	O	-	-	M	Y	N
Related IPMs	M	O	C	-	-	Y	N
Replied-to IPM	S	O	C	-	-	Y	N
Reply recipients	M	O	C	-	-	Y	N
Reply requestors	M	O	C	-	-	Y	N
Reply time	S	O	C	-	-	Y	N
Returned IPM	S	O	-	C	-	Y	N
RN requestors	M	O	C	-	-	Y	N
Sensitivity	S	O	C	-	-	Y	Y
Subject	S	O	C	-	-	Y	N
Subject IPM	S	M	-	M	M	Y	N
Suppl. receipt info	S	O	-	-	C	Y	N
Teletex body parts	M	O	C	-	-	N	N
Teletex data	M	O	C	-	-	N	N
Teletex parameters	M	O	C	-	-	N	N
This IPM	S	M	M	-	-	Y	N
Videotex body parts	M	O	C	-	-	N	N
Videotex data	M	O	C	-	-	N	N
Videotex parameters	M	O	C	-	-	N	N
Voice body parts	M	O	C	-	-	N	N
Voice data	M	O	C	-	-	N	N
Voice parameters	M	O	C	-	-	N	N

V Single/multi valued

L Support level by MS and access UA

P Presence in delivered message entry

L Available for List, alert

S Available for summarize

C.1.2 IPM synopsis

The **IPM synopsis** attribute gives the structure, characteristics, size and processing status of an IPM at the granularity of individual body parts.

```
ipm-synopsis ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX IPMSynopsis
  SINGLE VALUE
  ::= id-sat-ipm-synopsis
```

The synopsis of an IPM comprises a synopsis of each of its body parts. The synopses appear in the order in which the body parts appear.

```
IPMSynopsis ::= SEQUENCE OF BodyPartSynopsis
```

The synopsis of a body part takes either of two forms depending upon whether the body part is of type message. This enables the synopsis of a forwarding IPM to encompass the body parts of each forwarded IPM (recursively), as well as those of the forwarding IPM itself.

```
BodyPartSynopsis ::= CHOICE {
  message      [0]  MessageBodyPartSynopsis
  non-message  [1]  NonMessageBodyPartSynopsis }

MessageBodyPartSynopsis ::= SEQUENCE {
  number      [0]  SequenceNumber,
  synopsis    [1]  IPMSynopsis }
```

```

NonMessageBodyPartSynopsis ::= SEQUENCE {
    type          [0]    OBJECT IDENTIFIER,
    parameters    [1]    ExternallyDefinedParameters,
    size          [2]    INTEGER,
    processed     [3]    BOOLEAN DEFAULT FALSE }

```

The synopsis of a message body part has the following components:

- a) **Number** (M): The sequence number that the MS assigns to the entry that the message body part represents.
- b) **Synopsis** (M): The synopsis of the IPM that forms the content of the message that the body part represents.

The synopsis of a body part of type other than message has the following components. For purposes of this synopsis, the body part is considered to be of type externally defined, whether or not (see Annex B) it was so conveyed to the MS:

- a) **Type** (M): The body part's extended type, i.e., the direct-reference component of the body part's data component. An object identifier.
- b) **Parameters** (M): The body part's format and control parameters, i.e., the body part's parameters component. An Any.
- c) **Size** (M): The size in octets of the encoding of the Encoding component of the body part's Data component when the basic encoding rules of Recommendation X.209 are followed. If those rules permit several (e.g., both primitive and constructed) encodings of the component, the size may reflect any one of them. An Integer.
- d) **Processed** (D *false*): An indication of whether or not the body part has been conveyed to the UA by means of the MS' list or fetch abstract operation. A Boolean.

An MS that supports this attribute shall maintain it for an information object that it holds if, and only if, that object is a message whose content is an IPM.

Note – As a consequence of its variability, the value of the size component should be considered only an estimate of the body part's size.

C.2 *Heading attributes*

Some attributes are derived from the heading of an IPM. These attributes are defined and described below.

C.2.1 *Heading*

The **heading** attribute is the (entire) heading of an IPM.

```

heading ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX Heading
    SINGLE VALUE
    ::= id-hat-heading

```

An MS that supports this attribute shall maintain it for an information object that it holds if, and only if, that object is a message whose content is an IPM.

C.2.2 *Heading analyses*

Some attributes have as their values O/R descriptors selected after analysis of the heading. They identify the “primary”, “copy”, and blind “copy” recipients of an IPM of whom an RN, NRN, or reply is requested.

```

rn-requestors ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX ORDescriptor
    MATCHES FOR EQUALITY
    MULTI VALUE
    ::= id-hat-rn-requestors

```

```

nrn-requestors ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX ORDescriptor
    MATCHES FOR EQUALITY
    MULTI VALUE
    ::= id-hat-nrn-requestors

```

reply-requestors ATTRIBUTE
WITH ATTRIBUTE-SYNTAX ORDescriptor
MATCHES FOR EQUALITY
MULTI VALUE
::= id-hat-reply-requestors

An MS that supports one of these attributes shall maintain it for an information object that it holds if, and only if, that object is a message whose content is an IPM whose heading requests, of at least one user, or DL, an RN, NRN, or reply, respectively. It shall maintain one attribute value for every recipient specifier in the IPM's primary, copy, or blind copy recipients field whose notification-requests component includes the value rn (in the case of the first attribute) or nrn (in the case of the second, or whose Reply-requested component signifies, by either its presence or its absence, that a reply is requested (in the case of the third). The value shall be the recipient's specifier's recipient component.

C.2.3 *Heading fields*

Some attributes bear the names of heading fields and have those fields as their values. The ordering for the expiry time and reply time attributes is increasing chronological order.

this-ipm ATTRIBUTE
WITH ATTRIBUTE-SYNTAX This IPMField
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-hat-this-ipm

originator ATTRIBUTE
WITH ATTRIBUTE-SYNTAX OriginatorField
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-hat-originator

replied-to-IPM ATTRIBUTE
WITH ATTRIBUTE-SYNTAX RepliedToIPMField
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-hat-replied-to-IPM

subject ATTRIBUTE
WITH ATTRIBUTE-SYNTAX SubjectField
MATCHES FOR EQUALITY SUBSTRINGS
SINGLE VALUE
::= id-hat-subject

expiry-time ATTRIBUTE
WITH ATTRIBUTE-SYNTAX ExpiryTimeField
MATCHES FOR EQUALITY ORDERING
SINGLE VALUE
::= id-hat-expiry-time

reply-time ATTRIBUTE
WITH ATTRIBUTE-SYNTAX ReplyTimeField
MATCHES FOR EQUALITY ORDERING
SINGLE VALUE
::= id-hat-reply-time

importance ATTRIBUTE
WITH ATTRIBUTE-SYNTAX ImportanceField
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-hat-importance

sensitivity ATTRIBUTE
WITH ATTRIBUTE-SYNTAX SensitivityField
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-hat-sensitivity

auto-forwarded ATTRIBUTE
WITH ATTRIBUTE-SYNTAX AutoForwardedField
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-hat-auto-forwarded

An MS that supports one of these attributes shall maintain it for an information object that it holds if, and only if, that object is a message whose content is an IPM whose heading contains the field whose name the attribute bears.

C.2.4 *Heading sub-fields*

Some attributes bear the names of heading fields and have sub-fields of those fields as their values.

authorizing-users ATTRIBUTE
WITH ATTRIBUTE-SYNTAX AuthorizingUsersSubfield
MATCHES FOR EQUALITY
MULTI VALUE
::= id-hat-authorizing-users

primary-recipients; ATTRIBUTE
WITH ATTRIBUTE-SYNTAX PrimaryRecipientsSubField
MATCHES FOR EQUALITY
MULTI VALUE
::= id-hat-primary-recipients

copy-recipients ATTRIBUTE
WITH ATTRIBUTE-SYNTAX CopyRecipientsSubfield
MATCHES FOR EQUALITY
MULTI VALUE
::= id-hat-copy-recipients

blind-copy-recipients ATTRIBUTE
WITH ATTRIBUTE-SYNTAX BlindCopyRecipientsSubfield
MATCHES FOR EQUALITY
MULTI VALUE
::= id-hat-blind-copy-recipients

obsoleted-IPMs ATTRIBUTE
WITH ATTRIBUTE-SYNTAX ObsoletedIPMsSubfield
MATCHES FOR EQUALITY
MULTI VALUE
::= id-hat-obsoleted-IPMs

related-IPMs ATTRIBUTE
WITH ATTRIBUTE-SYNTAX RelatedIPMsSubfield
MATCHES FOR EQUALITY
MULTI VALUE
::= id-hat-related-IPMs

reply-recipients ATTRIBUTE
WITH ATTRIBUTE-SYNTAX ReplyRecipientsSubfield
MATCHES FOR EQUALITY
MULTI VALUE
::= id-hat-reply-recipients

An MS that supports one of these attributes shall maintain it for an information object that it holds if, and only if, that object is a message whose content is an IPM whose heading contains the field whose name the attribute bears. It shall maintain one attribute value for each sub-field.

C.2.5 *Heading extensions*

Some attributes bear the names of heading extensions and have as their values the values of those extensions or a part thereof.

incomplete-copy ATTRIBUTE
WITH ATTRIBUTE-SYNTAX incompleteCopyExtensionValue
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-hat-incomplete-copy

languages ATTRIBUTE
WITH ATTRIBUTE-SYNTAX Language
MATCHES FOR EQUALITY
MULTI VALUE
::= id-hat-languages

An MS that supports one of these attributes shall maintain it for an information object that it holds if, and only if, that object is a message whose content is an IPM whose heading contains the extension whose name the attribute bears. In the case of the language attribute, the MS shall maintain one attribute value for each language the extension identifies.

C.3 *Body attributes*

Some attributes are derived from the body of an IPM. These attributes are defined and described below.

C.3.1 *Body*

The **body** attribute is the (entire) body of an IPM.

body ATTRIBUTE
WITH ATTRIBUTE-SYNTAX Body
SINGLE VALUE
::= id-bat-body

An MS that supports this attribute shall maintain it for an information object that it holds if, and only if, that object is a message whose content is an IPM.

C.3.2 *Basic body parts*

Some attributes bear the names of basic body part types and have, with one exception, such body parts as their values.

An MS holds each forwarded IPM (i.e., each message body part) as an information object in its own right, separate from the forwarding IPM. That information object, of course, is a message whose content is an IPM. The message body parts attribute below, therefore, has as its values the sequence numbers the MS assigns to these messages.

ia5-text-body-parts ATTRIBUTE
WITH ATTRIBUTE-SYNTAX IA5TextBodyPart
MULTI VALUE
::= id-bat-ia5-text-body-parts

voice-body-parts ATTRIBUTE
WITH ATTRIBUTE-SYNTAX VoiceBodyPart
MULTI VALUE
::= id-bat-voice-body-parts

g3-facsimile-body-parts ATTRIBUTE
WITH ATTRIBUTE-SYNTAX G3FacsimileBodyPart
MULTI VALUE
::= id-bat-g3-facsimile-body-parts

g4-class1-body-parts ATTRIBUTE
WITH ATTRIBUTE-SYNTAX G4Class1BodyPart
MULTI VALUE
::= id-bat-g4-class1-body-parts

teletex-body-parts ATTRIBUTE
WITH ATTRIBUTE-SYNTAX TeletexBodyPart
MULTI VALUE
::= id-bat-teletex-body-parts

videotex-body-parts ATTRIBUTE
WITH ATTRIBUTE-SYNTAX VideotexBodyPart
MULTI VALUE
::= id-bat-videotex-body-parts

encrypted-body-parts ATTRIBUTE
WITH ATTRIBUTE-SYNTAX EncryptedBodyPart
MULTI VALUE
::= id-bat-encrypted-body-parts

message-body-parts ATTRIBUTE
 WITH ATTRIBUTE-SYNTAX SequenceNumber
 MULTI VALUE
 ::= id-bat-message-body-parts

mixed-mode-parts ATTRIBUTE
 WITH ATTRIBUTE-SYNTAX MixedModeBodyPart
 MULTI VALUE
 ::= id-bat-mixed-mode-body-parts

bilaterally-defined-body-parts ATTRIBUTE
 WITH ATTRIBUTE-SYNTAX BilaterallyDefinedBodyPart
 MULTI VALUE
 ::= id-bat-bilaterally-defined-body-parts

nationally-defined-body-parts ATTRIBUTE
 WITH ATTRIBUTE-SYNTAX NationallyDefinedBodyPart
 MULTI VALUE
 ::= id-bat-nationally-defined-body-parts

An MS that supports one of these attributes shall maintain it for an information object that it holds if, and only if, that object is a message whose content is an IPM whose body contains one or more body parts of the type whose name the attribute bears. It shall maintain one attribute value for each such body part.

C.3.3 *Basic body part parameters components*

Some attributes bear the names of basic body part types and have the parameters components of such body parts as their values.

ia5-text-parameters ATTRIBUTE
 WITH ATTRIBUTE-SYNTAX IA5TextParameters
 MULTI VALUE
 ::= id-bat-ia5-text-parameters

voice-text-parameters ATTRIBUTE
 WITH ATTRIBUTE-SYNTAX IA5VoiceParameters
 MULTI VALUE
 ::= id-bat-voice-parameters

g3-facsimile-parameters ATTRIBUTE
 WITH ATTRIBUTE-SYNTAX G3FacsimileParameters
 MULTI VALUE
 ::= id-bat-g3-facsimile-parameters

teletex-parameters ATTRIBUTE
 WITH ATTRIBUTE-SYNTAX TeletexParameters
 MULTI VALUE
 ::= id-bat-teletex-parameters

videotex-parameters ATTRIBUTE
 WITH ATTRIBUTE-SYNTAX VideotexParameters
 MULTI VALUE
 ::= id-bat-videotex-parameters

encrypted-parameters ATTRIBUTE
 WITH ATTRIBUTE-SYNTAX EncryptedParameters
 MULTI VALUE
 ::= id-bat-encrypted-parameters

message-parameters ATTRIBUTE
 WITH ATTRIBUTE-SYNTAX MessageParameters
 MULTI VALUE
 ::= id-bat-message-parameters

An MS that supports one of these attributes shall maintain it for an information object that it holds if, and only if, that object is a message whose content is an IPM whose body contains one or more body parts of the type whose name the attribute bears. It shall maintain one attribute value for each such body part.

C.3.4 *Basic body part data components*

Some attributes bear the names of basic body part types and have the data components of such body parts as their values.

```
ia5-text-data ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX IA5TextData
    MULTI VALUE
    ::= id-bat-ia5-text-data

voice-data ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX VoiceData
    MULTI VALUE
    ::= id-bat-voice-data

g3-facsimile-data ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX G3FacsimileData
    MULTI VALUE
    ::= id-bat-g3-facsimile-data

teletex-data ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX TeletexData
    MULTI VALUE
    ::= id-bat-teletex-data

videotex-data ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX VideotexData
    MULTI VALUE
    ::= id-bat-videotex-data

encrypted-data ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX EncryptedData
    MULTI VALUE
    ::= id-bat-encrypted-data

message-data ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX MessageData
    MULTI VALUE
    ::= id-bat-message-data
```

An MS that supports one of these attributes shall maintain it for an information object that it holds if, and only if, that object is a message whose content is an IPM whose body contains one or more body parts of the type whose name the attribute bears. It shall maintain one attribute for each such body part.

C.3.5 *Extended body part types*

The **extended body part types** attribute identifies the extended body part types represented in an IPM.

```
extended-body-parts ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER
    MATCHES FOR EQUALITY
    MULTI VALUE
    ::= id-bat-extended-body-parts
```

An MS that supports this attribute shall maintain it for an information object that it holds if, and only if, that object is a message whose content is an IPM whose body contains one or more externally defined body parts. It shall maintain one attribute value for every such type present. The value shall denote the type as specified in § 7.3.12.

Note – Each value of this attribute corresponds to one of the attributes described in § C.3.6 below.

C.3.6 *Extended body parts*

Some attributes, unnamed, have as their values the encoding components (see § 7.3.12) of the ASN.1 externals that constitute the data components of externally defined body parts.

To each extended body part type there correspond two attributes. The first attribute is denoted by the object identifier that is the direct-reference component (again, see § 7.3.12) of the external that constitutes the data component of a body part of that type. The content of this first attribute is that data component. The second attribute is denoted by the object identifier that is the direct-reference component of the external that constitutes the parameters component of a body part of that type. The content of this second attribute is that parameters component.

An MS that supports one of these body parts shall maintain both attributes for an information object that it holds if, and only if, that object is a message whose content is an IPM whose body contains one or more body parts of the type that corresponds to that attribute. It shall maintain one value of each attribute for each such body part.

Note 1 – The extended body part attributes cannot be enumerated in practice because the extended body part types cannot be so enumerated.

Note 2 – The extended body part types attribute (see § C.3.5) determines the extended body part attributes for a particular IPM.

C.4 *Notification attributes*

Some attributes are derived from an IPN. These attributes are defined and described below.

C.4.1 *Common fields*

Some attributes that bear the names of common fields and have those fields as their values.

subject-ipm ATTRIBUTE

WITH ATTRIBUTE-SYNTAX SubjectIPMField
MATCHES FOR EQUALITY SUBSTRINGS
SINGLE VALUE
::= id-nat-subject-ipm

ipn-originator ATTRIBUTE

WITH ATTRIBUTE-SYNTAX IPNOriginatorField
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-nat-ipn-originator

ipn-preferred-recipient ATTRIBUTE

WITH ATTRIBUTE-SYNTAX IPMPreferredRecipientField
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-nat-ipm-preferred-recipient

conversion-eits ATTRIBUTE

WITH ATTRIBUTE-SYNTAX MS-EITs
MATCHES FOR EQUALITY
MULTI VALUE
::= id-nat-conversion-eits

An MS that supports one of these attributes shall maintain it for an information object that it holds if, and only if, that object is a message whose content is an IPN that contains the field whose name the attribute bears.

C.4.2 *Non-receipt fields*

Some attributes bear the names of non-receipt fields and have those fields as their values.

non-receipt-reason ATTRIBUTE

WITH ATTRIBUTE-SYNTAX NonReceiptReasonField
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-nat-non-receipt-reason

discard-reason ATTRIBUTE

WITH ATTRIBUTE-SYNTAX DiscardReasonField
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-nat-discard-reason

auto-forward-comment ATTRIBUTE

WITH ATTRIBUTE-SYNTAX AutoForwardCommentField
MATCHES FOR EQUALITY SUBSTRINGS
SINGLE VALUE
::= id-nat-auto-forward-comment

```

returned-IPM ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX ReturnedIPMField
  SINGLE VALUE
  ::= id-nat-returned-ipm

```

An MS that supports one of these attributes shall maintain it for an information that it holds if, and only if, that object is a message whose content is an NRN that contains the field whose name the attribute bears.

C.4.3 *Receipt fields*

Some attributes bear the names of receipt fields and have those fields as their values. The ordering for the receipt time attribute is increasing chronological order.

```

receipt-time ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX ReceiptTimeField
  MATCHES FOR EQUALITY ORDERING
  SINGLE VALUE
  ::= id-nat-receipt-time

acknowledgment ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX AcknowledgmentModeField
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-nat-acknowledgment-mode

suppl-receipt-info ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX SupplReceiptInfoField
  MATCHES FOR EQUALITY SUBSTRINGS
  SINGLE VALUE
  ::= id-nat-suppl-receipt-info

```

An MS that supports one of these attributes shall maintain it for an information object that it holds if, and only if, that object is a message whose content is an RN that contains the field whose name the attribute bears.

ANNEX D

(to Recommendation X.420)

Reference definition of object identifiers

This Annex is an integral part of this Recommendation.

This Annex defines for reference purposes various object identifiers cited in the ASN.1 modules of subsequent Annexes. It uses ASN.1.

All object identifiers this Recommendation assigns are assigned in this Annex. The Annex is definitive for all but those for ASN.1 modules and the IPMS application itself. The definitive assignments for the former occur in the modules themselves; other references to them appear in IMPORT clauses. The latter is fixed.

```

IMPSObjectIdentifiers {joint-iso-ccitt
  mhs-motis(6) ipms(1) modules(0) object-identifiers(0)}
DEFINITIONS IMPLICIT TAGS ::=
BEGIN

-- Prologue
-- Exports everything.

IMPORTS -- nothing --;

ID ::= OBJECT IDENTIFIER

-- Interpersonal Messaging (not definitive)
id-ipms ID ::= {joint-iso-ccitt mhs-motis(6) ipms(1)} - -not definitive

```

-- *Categories*

id-mod ID ::= { id-ipms 0 } -- *modules; not definitive*
id-ot ID ::= { id-ipms 1 } -- *object types*
id-pt ID ::= { id-ipms 2 } -- *port types*
id-ref ID ::= { id-ipms 3 } -- *refinements*
id-et ID ::= { id-ipms 4 } -- *extended body part types*
id-hex ID ::= { id-ipms 5 } -- *heading extensions*
id-sat ID ::= { id-ipms 6 } -- *summary attributes*
id-hat ID ::= { id-ipms 7 } -- *heading attributes*
id-bat ID ::= { id-ipms 8 } -- *body attributes*
id-nat ID ::= { id-ipms 9 } -- *notification attributes*
id-mct ID ::= { id-ipms 10 } -- *message content types*
id-ep ID ::= { id-ipms 11 } -- *extended body part parameters*

-- *Modules*

id-mod-object-identifiers ID ::= { id-mod 0 } -- *not definitive*
id-mod-functional-objects ID ::= { id-mod 1 } -- *not definitive*
id-mod-information-objects ID ::= { id-mod 2 } -- *not definitive*
id-mod-abstract-service ID ::= { id-mod 3 } -- *not definitive*
id-mod-heading-extensions ID ::= { id-mod 6 } -- *not definitive*
id-mod-extended-body-part-types ID ::= { id-mod 7 } -- *not definitive*
id-mod-message-store-attributes ID ::= { id-mod 8 } -- *not definitive*
id-mod-upper-bounds ID ::= { id-mod 10 } -- *not definitive*

-- *Object types*

id-ot-ipme ID ::= { id-ot 0 }
id-ot-ipms-user ID ::= { id-ot 1 }
id-ot-ipms ID ::= { id-ot 2 }
id-ot-ipms-ua ID ::= { id-ot 3 }
id-ot-ipms-ms ID ::= { id-ot 4 }
id-ot-tlma ID ::= { id-ot 5 }
id-ot-tlxau ID ::= { id-ot 6 }
id-ot-pdau ID ::= { id-ot 7 }

-- *Port types*

id-pt-origination ID ::= { id-pt 0 }
id-pt-reception ID ::= { id-pt 1 }
id-pt-management ID ::= { id-pt 2 }

-- *Refinements*

id-ref-primary ID ::= { id-ref 0 }
id-ref-secondary ID ::= { id-ref 1 }

-- *Extended body part types*

id-et-ia5-text ID ::= { id-et 0 }
id-et-voice ID ::= { id-et 1 }
id-et-g3-facsimile ID ::= { id-et 2 }
id-et-g4-class1 ID ::= { id-et 3 }
id-et-teletex ID ::= { id-et 4 }
id-et-videotex ID ::= { id-et 5 }
id-et-encrypted ID ::= { id-et 6 }
id-et-message ID ::= { id-et 7 }
id-et-mixed-mode ID ::= { id-et 8 }
id-et-bilaterally-defined ID ::= { id-et 9 }
id-et-nationally-defined ID ::= { id-et 10 }

-- *Heading extensions*

id-hex-incomplete-copy ID ::= { id-hex 0 }
id-hex-languages ID ::= { id-hex 1 }

-- Summary attributes

id-sat-ipm-entry-type ID ::= { id-sat 0 }
id-sat-ipm-synopsis ID ::= { id-sat 1 }

-- Heading attributes

id-hat-heading ID ::= { id-hat 0 }
id-hat-this-ipm ID ::= { id-hat 1 }
id-hat-originator ID ::= { id-hat 2 }
id-hat-replied-to-IPM ID ::= { id-hat 3 }
id-hat-subject ID ::= { id-hat 4 }
id-hat-expiry-time ID ::= { id-hat 5 }
id-hat-reply-time ID ::= { id-hat 6 }
id-hat-importance ID ::= { id-hat 7 }
id-hat-sensitivity ID ::= { id-hat 8 }
id-hat-auto-forwarded ID ::= { id-hat 9 }
id-hat-authorizing-users ID ::= { id-hat 10 }
id-hat-primary-recipients ID ::= { id-hat 11 }
id-hat-copy-recipients ID ::= { id-hat 12 }
id-hat-blind-copy-recipients ID ::= { id-hat 13 }
id-hat-obsolete-IPMs ID ::= { id-hat 14 }
id-hat-related-IPMs ID ::= { id-hat 15 }
id-hat-reply-recipients ID ::= { id-hat 16 }
id-hat-incomplete-copy ID ::= { id-hat 17 }
id-hat-languages ID ::= { id-hat 18 }
id-hat-rn-requestors ID ::= { id-hat 19 }
id-hat-nrn-requestors ID ::= { id-hat 20 }
id-hat-reply-requestors ID ::= { id-hat 21 }

-- Body attributes

id-bat-body ID ::= { id-bat 0 }
id-bat-ia5-text-body-parts ID ::= { id-bat 1 }
id-bat-voice-body-parts ID ::= { id-bat 2 }
id-bat-g3-facsimile-body-parts ID ::= { id-bat 3 }
id-bat-g4-class1-body-parts ID ::= { id-bat 4 }
id-bat-teletex-body-parts ID ::= { id-bat 5 }
id-bat-videotex-body-parts ID ::= { id-bat 6 }
id-bat-encrypted-body-parts ID ::= { id-bat 7 }
id-bat-message-body-parts ID ::= { id-bat 8 }
id-bat-mixed-mode-body-parts ID ::= { id-bat 9 }
id-bat-bilaterally-defined-body-parts ID ::= { id-bat 10 }
id-bat-nationally-defined-body-parts ID ::= { id-bat 11 }
id-bat-extended-body-part-types ID ::= { id-bat 12 }
id-bat-ia5-text-parameters ID ::= { id-bat 13 }
id-bat-voice-parameters ID ::= { id-bat 14 }
id-bat-g3-facsimile-parameters ID ::= { id-bat 15 }
id-bat-teletex-parameters ID ::= { id-bat 16 }
id-bat-videotex-parameters ID ::= { id-bat 17 }
id-bat-encrypted-parameters ID ::= { id-bat 18 }
id-bat-message-parameters ID ::= { id-bat 19 }
id-bat-ia5-text-data ID ::= { id-bat 20 }
id-bat-voice-data ID ::= { id-bat 21 }
id-bat-g3-facsimile-data ID ::= { id-bat 22 }
id-bat-teletex-data ID ::= { id-bat 23 }
id-bat-videotex-data ID ::= { id-bat 24 }
id-bat-encrypted-data ID ::= { id-bat 25 }
id-bat-message-data ID ::= { id-bat 26 }

-- Notification attributes

id-nat-subject-ipm ID ::= { id-nat 0 }
id-nat-ipn-originator ID ::= { id-nat 1 }
id-nat-ipm-preferred-recipient ID ::= { id-nat 2 }

```

id-nat-conversion-eits          ID ::= { id-nat 3 }
id-nat-non-receipt-reason      ID ::= { id-nat 4 }
id-nat-discard-reason         ID ::= { id-nat 5 }
id-nat-auto-forward-comment    ID ::= { id-nat 6 }
id-nat-returned-ipm           ID ::= { id-nat 7 }
id-nat-receipt-time            ID ::= { id-nat 8 }
id-nat-acknowledgment-mode     ID ::= { id-nat 9 }
id-nat-suppl-receipt-info      ID ::= { id-nat 10 }

```

-- *Message content types (for use by MS only)*

```

id-mct-p2-1984 ID ::= { id-mct 0 } -- P2 1984
id-mct-p2-1988 ID ::= { id-mct 1 } -- P2 1988

```

-- *Extended body part parameters*

```

id-ep-ia5-text          ID ::= { id-ep 0 }
id-ep-voice              ID ::= { id-ep 1 }
id-ep-g3-facsimile      ID ::= { id-ep 2 }
id-ep-teletex           ID ::= { id-ep 4 }
id-ep-videotex          ID ::= { id-ep 5 }
id-ep-encrypted         ID ::= { id-ep 6 }
id-ep-message           ID ::= { id-ep 7 }

```

END -- *of IPMSObjectIdentifiers*

ANNEX E

(to Recommendation X.420)

Reference definition of abstract information objects

This Annex is an integral part of this Recommendation.

This Annex, a supplement to Section 2, defines for reference purposes the abstract information objects of interpersonal messaging.

```

IPMSInformationObjects {joint-iso-ccitt
  mhs-motis(6) ipms(1) modules(0) information-objects(2) }

```

DEFINITIONS IMPLICIT TAGS ::=

BEGIN

-- *Prologue*

-- *Exports everything.*

IMPORTS

-- *IPMS upper bounds*

```

ub-auto-forward-comment, ub-free-form-name, ub-ipm-identifier-suffix,
ub-local-imp-identifier, ub-subject-field, ub-telephone-number

```

```

FROM IPMSUpperBounds { joint-iso-ccitt
  mhs-motis(6) ipms(1) modules(0) upper-bounds(10) }

```

-- *DTAM*

```

ProtocolElement

```

```

FROM dTAM

```

--- *MTS abstract service*

```

EncodedInformationTypes, G3FacsimileNonBasicParameters,
MessageDeliveryTime, ORAddress, ORName,
OtherMessageDeliveryFields, SupplementaryInformation,
TeletexNonBasicParameters,

```

```

FROM MTSAbstractService { joint-iso-ccitt
    mhs-motis(6) mts(3) modules(0) mts-abstract-service(1) };

```

```
Time ::= UTCTime
```

```
-- Information object
```

```
InformationObject ::= CHOICE {
    ipm          [0] IPM,
    ipn          [1] IPN }

```

```
-- IPM
```

```
IPM ::= SEQUENCE {
    heading      Heading,
    body         Body }

```

```
-- Heading
```

```
Heading ::= SET {
    this-IPM                ThisIPMField,
    originator               [0] OriginatorField OPTIONAL,
    authorizing-users        [1] AuthorizingUsersField OPTIONAL,
    primary-recipients       [2] PrimaryRecipientsField DEFAULT { },
    copy-recipients          [3] CopyRecipientsField DEFAULT { },
    blind-copy-recipients    [4] BlindCopyRecipientsField OPTIONAL,
    replied-to-IPM           [5] RepliedToIPMField OPTIONAL,
    obsoleted-IPMs          [6] ObsoletedIPMsField DEFAULT { },
    related-IPMs            [7] RelatedIPMsField DEFAULT { },
    subject                  [8] EXPLICIT SubjectField OPTIONAL,
    expiry-time              [9] ExpiryTimeField OPTIONAL,
    reply-time               [10] ReplyTimeField OPTIONAL,
    reply-recipients         [11] ReplyRecipientsField OPTIONAL,
    importance               [12] ImportanceField DEFAULT normal,
    sensitivity              [13] SensitivityField OPTIONAL,
    auto-forwarded           [14] AutoForwardedField DEFAULT FALSE,
    extensions               [15] ExtensionsField DEFAULT { } }

```

```
-- Heading component types
```

```
IPMIdentifier ::= [APPLICATION 11] SET {
    user                   ORAddress OPTIONAL,
    user-relative-identifier LocalIPMIdentifier }

```

```
LocalIPMIdentifier ::= PrintableString
    (SIZE (0..ub-local-ipm-identifier))

```

```
RecipientSpecifier ::= SET {
    recipient               [0] ORDescriptor,
    notification-requests  [1] NotificationRequests DEFAULT { },
    reply-requested         [2] BOOLEAN DEFAULT FALSE }

```

```
NotificationRequests ::= BIT STRING {
    rn(0),
    nrn(1),
    ipm-return(2) }

```

```
ORDescriptor ::= SET {
    formal-name             ORName OPTIONAL,
    free-form-name          [0] FreeFormName OPTIONAL,
    telephone-number       [1] TelephoneNumber OPTIONAL }

```

```
FreeFormName ::= TeletexString (SIZE (0..ub-free-form-name))

```

```
TelephoneNumber ::= PrintableString (SIZE (0..ub-telephone-number))

```

-- This IPM heading field
This IPMField ::= IPMIdentifier

-- Originator heading field
OriginatorField ::= ORDescriptor

-- Authorizing users heading field
AuthorizingdUsersField ::= SEQUENCE OF AuthorizingUsersSubfield
AuthorizingUsersSubfield ::= ORDescriptor

-- Primary recipients heading field
PrimaryRecipientsField ::= SEQUENCE OF PrimaryRecipientsSubfield
PrimaryRecipientsSubfield ::= RecipientSpecifier

-- Copy recipients heading field
CopyRecipientsField ::= SEQUENCE OF CopyRecipientsSubfield
CopyRecipientsSubfield ::= RecipientSpecifier

-- Blind copy recipients heading field
BlindCopyRecipientsField ::= SEQUENCE OF BlindCopyRecipientsSubfield
BlindCopyRecipientsSubfield ::= RecipientSpecifier

-- Replied-to IPM heading field
RepliedToIPMField ::= IPMIdentifier

-- Obsolete IPMs heading field
ObsoleteIPMsField ::= SEQUENCE OF ObsoleteIPMsSubfield
ObsoleteIPMsSubfield ::= IPMIdentifier

-- Related IPMs heading field
RelatedIPMsField ::= SEQUENCE OF RelatedIPMsSubfield
RelatedIPMsSubfield ::= IPMIdentifier

-- Subfield heading field
SubjectField ::= TeletexString (SIZE (0..ub-subject-field))

-- Expiry time heading field
Expiry-TimeField ::= Time

-- Reply time heading field
ReplyTimeField ::= Time

-- Reply recipient heading field

ReplyRecipientsField ::= SEQUENCE OF ReplyRecipientsSubfield

ReplyRecipientsSubfield ::= ORDescriptor

-- Importance heading field

ImportanceField ::= ENUMERATED {
 low (0),
 normal (1),
 high (2) }

-- Sensitivity heading field

SensitivityField ::= ENUMERATED {
 personal (1),
 private (2),
 company-confidential (3) }

-- Auto-forwarded heading field

AutoForwardedField ::= BOOLEAN

-- Extensions heading field

ExtensionsField ::= SET OF HeadingExtension

HeadingExtension ::= SEQUENCE {
 type OBJECT IDENTIFIER,
 value ANY DEFINED BY type DEFAULT NULL NULL }

HEADING-EXTENSION MACRO ::=

BEGIN

 TYPE NOTATION ::= "VALUE" type | empty
 VALUE NOTATION ::= value (VALUE OBJECT IDENTIFIER)

END

-- Body

Body ::= SEQUENCE OF BodyPart

BodyPart ::= CHOICE {
 ia5-text [0] IA5TextBodyPart,
 voice [2] VoiceBodyPart,
 g3-facsimile [3] G3FacsimileBodyPart,
 g4-class1 [4] G4Class1BodyPart,
 teletex [5] TeletexBodyPart,
 videotex [6] VideotexBodyPart,
 encrypted [8] EncryptedBodyPart,
 message [9] MessageBodyPart,
 mixed-mode [11] MixedModeBodyPart,
 bilaterally-defined [14] BilaterallyDefinedBodyPart,
 nationally-defined [7] NationallyDefinedBodyPart,
 externally-defined [15] ExternallyDefinedBodyPart }

-- IA5 text body part

```
IA5TextBodyPart ::= SEQUENCE {
    parameters  IA5TextParameters,
    data        IA5TextData }

IA5TextParameters ::= SET {
    repertoire[0] Repertoire DEFAULT ia5 }

IA5TextData ::= IA5String

Repertoire ::= ENUMERATED {
    ita2(2),
    ia5 (5) }
```

-- Voice body part

```
VoiceBodyPart ::= SEQUENCE {
    parameters  VoiceParameters,
    data        VoiceData }

VoiceParameters ::= SET -- for further study

VoiceData ::= BIT STRING -- for further study
```

-- G3 Facsimile body part

```
G3FacsimileBodyPart ::= SEQUENCE {
    parameters  G3FacsimileParameters,
    data        G3FacsimileData }

G3FacsimileParameters ::= SET {
    number-of-pages      [0]    INTEGER OPTIONAL,
    non-basic-parameters [1]    G3FacsimileNonBasicParameters OPTIONAL }

G3FacsimileData ::= SEQUENCE OF BIT STRING
```

-- G4 class 1 and mixed-mode body parts

```
G4Class1BodyPart ::= SEQUENCE OF ProtocolElement
MixedModeBodyPart ::= SEQUENCE OF ProtocolElement
```

-- Telex body part

```
TeletexBodyPart ::= SEQUENCE {
    parameters  TeletexParameters,
    data        TeletexData }

TeletexParameters ::= SET {
    number-of-pages      [0]    INTEGER OPTIONAL,
    telex-compatible     [1]    BOOLEAN DEFAULT FALSE,
    non-basic-parameters [2]    TeletexNonBasicParameters OPTIONAL }

TeletexData ::= SEQUENCE OF TeletexString
```

-- Videotex body part

```
VideotexBodyPart ::= SEQUENCE {
    parameters  VideotexParameters,
    data        VideotexData }
```

```

VideotexParameters ::= SET {
    syntax      [0] VideotexSyntax OPTIONAL
}
VideotexSyntax ::= INTEGER
    ids          (0),
    data-syntax1 (1),
    data-syntax2 (2),
    data-syntax3 (3) }
VideotexData ::= VideotexString

-- Encrypted body part
EncryptedBodyPart ::= SEQUENCE {
    parameters  EncryptedParameters,
    data        EncryptedData }
EncryptedParameters ::= SET -- for further study
EncryptedData ::= BIT STRING -- for further study

-- Message body part
MessageBodyPart ::= SEQUENCE {
    parameters  MessageParameters,
    data        MessageData }
MessageParameters ::= SET {
    delivery-time      [0]    MessageDeliveryTime OPTIONAL
    delivery-envelope  [1]    OtherMessageDeliveryFields OPTIONAL }
MessageData ::= IPM

-- Bilaterally defined body part
BilaterallyDefinedBodyPart ::= OCTET STRING

-- Nationally defined body part
NationallyDefinedBodyPart ::= ANY

-- Externally defined body part
ExternallyDefinedBodyPart ::= SEQUENCE {
    parameters [0]  ExternallyDefinedParameters OPTIONAL,
    data        ExternallyDefinedData }
ExternallyDefinedParameters ::= EXTERNAL
ExternallyDefinedData ::= EXTERNAL
EXTENDED-BODY-PART-TYPE MACRO ::=
BEGIN
    TYPE NOTATION      ::= Parameters Data
    VALUE NOTATION     ::= value (VALUE OBJECT IDENTIFIER)
    Parameters         ::= "PARAMETERS" type "IDENTIFIED" "BY" value (OBJECT IDENTIFIER) |
                        empty
    Data               ::= "DATA" type
END

```

-- IPN

IPN ::= SET {
 -- common-fields -- COMPONENTS OF CommonFields,
 choice [0] CHOICE {
 non-receipt-fields [0] NonReceiptFields,
 receipt-fields [1] ReceiptFields } }

RN ::= IPN -- with receipt-fields chosen

NRN ::= IPN -- with non-receipt-fields chosen

CommonFields ::= SET {
 subject-ipn SubjectIPNFields,
 ipn-originator [1] IPNOriginatorField OPTIONAL,
 ipn-preferred-recipient [2] IPNPreferredRecipientField OPTIONAL,
 conversion-eits ConversionEITsField OPTIONAL }

NonReceiptFields ::= SET {
 non-receipt-reason [0] NonReceiptReasonField,
 discard-reason [1] DiscardReasonField OPTIONAL,
 auto-forward-comment [2] AutoForwardCommentField OPTIONAL,
 returned-ipm [3] ReturnedIPNField OPTIONAL }

ReceiptFields ::= SET {
 receipt-time [0] RecipientTimeField,
 acknowledgment-mode [1] AcknowledgementModeField DEFAULT manual,
 suppl-receipt-info [2] SupplReceiptInfoField DEFAULT “ ” }

-- Common fields

SubjectIPMField ::= IPMIdentifier

IPMOriginatorField ::= ORDescriptor

IPMPreferredRecipientField ::= ORDescriptor

ConversionEITsField ::= EncodedInformationTypes

-- Non-receipt fields

NonReceiptReasonField ::= ENUMERATED {
 ipm-discarded (0),
 ipm-auto-forwarded (1) }

DiscardReasonField ::= ENUMERATED {
 ipm-expired (0),
 ipm-obsolete (1),
 user-subscription-terminated (2) }

AutoForwardCommentField ::= AutoForwardComment

AutoForwardComment ::= PrintableString
(SIZE (0..ub-auto-forward-comment))

ReturnedIPMField ::= IPM

-- Receipt fields

ReceiptTimeField ::= Time

AcknowledgmentModeField ::= ENUMERATED {
 manual (0),
 automatic (1) }

SupplReceiptInfoField ::= SupplementaryInformation

-- Message store realization

```
ForwardedInfo ::= SET {  
    auto-forwarding-comment [0]  
        AutoForwardComment OPTIONAL,  
    cover-note [1]  
        IA5TextBodyPart OPTIONAL,  
    this-ipm-prefix [2]  
        PrintableString (SIZE (1..ub-ipm-identifier-suffix))  
        OPTIONAL }
```

END -- of IPMSInformationObjects

ANNEX F

(to Recommendation X.420)

Reference definition of functional objects

This Annex is an integral part of this Recommendation.

This Annex, a supplement to §§ 10, 11 and 16, defined for reference purposes the functional objects of interpersonal messaging. It uses the OBJECT and REFINE macros of Recommendation X.407.

```
IPMFunctionalObjects { joint-iso-ccitt
    mhs-motis(6) modules(0) functional-objects(1) }
DEFINITIONS IMPLICIT TAGS ::=
BEGIN

-- Prologue
-- Exports everything.

IMPORTS

    -- IPMS abstract service
        management, origination, reception
        ----
        FROM IPMSAbstractService { joint-iso-ccitt
            mhs-motis(6) ipms(1) modules(0) abstract-service(3) }

    -- IPMS object identifiers
        id-ot-ipme, id-ot-ipms, id-ot-ipms-ms, id-ot-ipms-ua,
        id-ot-ipms-user, id-ot-pdau, id-ot-tima, id-ot-tlxau,
        id-ref-primary, id-ref-secondary
        ----
        FROM IPMSObjectIdentifiers { joint-iso-ccitt
            mhs-motis(6) ipms(1) modules(0) object-identifiers(0) }

    TLMA abstract service
        miscellanea
        ----
        FROM TLMAAbsService { ccitt
            recommendation(0) t(20) 330 tlnaabsservice(0) }

    -- MS abstract service
        retrieval
        ----
        FROM MSAbstractService { joint-iso-ccitt
            mhs-motis(6) ms(4) modules(0) abstract-service(1) }

    -- MTS abstract service
        administration, delivery, mTS, submission
        ----
        FROM MTSAbstractService { joint-iso-ccitt
            mhs-motis(6) mts(3) modules(0) mts-abstract-service(1) }

    -- Abstract service definition conventions
        OBJECT, REFINE
        ----
        FROM AbstractServiceNotation { joint-iso-ccitt
            mhs-motis(6) asdc(2) modules(0) notation(1) }

-- "Root" object type
ipme OBJECT
    ::= id-ot-ipme
```

-- Primary refinement

```
ipme-refinement REFINE ipme AS
  ipms
    origination [S] PAIRED WITH ipms-user
    reception [S] PAIRED WITH ipms-user
    management [S] PAIRED WITH ipms-user
  ipms-user RECURRING
  ::= id-ref-primary
```

-- Primary object types

```
ipms-user OBJECT
  PORTS {
    origination [C],
    reception [C],
    management [C] }
  ::= id-ot-ipms-user

ipms OBJECT
  PORTS }
    origination [S],
    reception [S],
    management [S] }
  ::= id-ot-ipms
```

-- Secondary refinement

```
ipms-refinement REFINE ipms AS
  mTS
    submission [S] PAIRED WITH ipms-ua, ipms-ms
    delivery [S] PAIRED WITH ipms-ua, ipms-ms
    administration [S] PAIRED WITH ipms-ua, ipms-ms
  ipms-ua RECURRING
    origination [S] VISIBLE
    reception [S] VISIBLE
    management [S] VISIBLE
  ipms-ms RECURRING
    submission [S] PAIRED WITH ipms-ua
    retrieval [S] PAIRED WITH ipms-ua
    administration [S] PAIRED WITH ipms-ua
  tlma RECURRING
    origination [S] VISIBLE
    reception [S] VISIBLE
    management [S] VISIBLE
  tlxau RECURRING
    origination [S] VISIBLE
    reception [S] VISIBLE
    management [S] VISIBLE
  pdau RECURRING
    reception [S] VISIBLE
  ::= id-ref-secondary
```

-- Secondary objects

```
ipms-ua OBJECT
  PORTS {
    origination [S],
    reception [S],
    management [S],
    submission [C],
```

```

        delivery      [C],
        retrieval     [C],
        administration [C] }
    ::= id-ot-ipms-ua

ipms-ms OBJECT
    PORTS {
        submission    [S],
        retrieval     [S],
        administration [S],
        submission    [C],
        delivery      [C],
        administration [C] }
    ::= id-ot-ipms-ms

tlma OBJECT
    PORTS {
        origination   [S],
        reception     [S],
        management    [S],
        miscellanea   [S] }
    ::= id-ot-tlma

tlxau OBJECT
    PORTS {
        origination   [S],
        reception     [S],
        management    [S] }
    ::= id-ot-tlxau

pdau OBJECT
    PORTS {
        reception     [S] }
    ::= id-ot-pdau

END -- of IPMSFunctionalObjects

```

ANNEX G

(to Recommendation X.420)

Reference definition of abstract service

This Annex is an integral part of this Recommendation.

This Annex, a supplement to §§ 12 and 13, defines for reference purposes the IPMS abstract service. It uses the PORT and ABSTRACT-OPERATION and -ERROR macros of Recommendation X.407.

```

IPMSAbstractService { joint-iso-ccitt
    mhs-motis(6) ipms(1) modules(0) abstract-service(3) }
DEFINITIONS IMPLICIT TAGS ::=
BEGIN

-- Prologue
-- Exports everything.

IMPORTS
    -- IPMS information objects
    AutoForwardComment, Heading, IPM, NRN, RN
    ----
    FROM IPMSInformationObjects { joint-iso-ccitt
        mhs-motis(6) ipms(1) modules(0) information-objects(2) }

```

```

-- IPMS object identifiers
  id-pt-management, id-pt-origination, id-pt-reception
  ----
  FROM IPMSObjectsIdentifiers { joint-iso-ccitt
    mhs-motis(6) ipms(1) modules(0) objects-identifiers(0) }

-- MTS abstract service
  MessageDeliveryEnvelope, MessageSubmissionEnvelope,
  MessageSubmissionIdentifier, MessageSubmissionTime,
  ProbeSubmissionEnvelope, ProbeSubmissionIdentifier,
  ProbeSubmissionTime, RecipientImproperlySpecified,
  ReportDeliveryEnvelope,
  ----
  FROM MTSAbstractService { joint-iso-ccitt
    mhs-motis(6) mts(3) modules(0) mts-abstract-service(1) }

-- Abstract service definition conventions
  ABSTRACT-ERROR, ABSTRACT-OPERATION, PORT
  ----
  FROM AbstractServiceNotation { joint-iso-ccitt
    mhs-motis(6) asdc(2) modules(0) notation(1) };

```

Time ::= UTCTime

```

-- Ports
origination PORT
  CONSUMER INVOKES {
    OriginateProbe,
    OriginateIPM,
    OriginateRN }
  ::= id-pt-origination

reception PORT
  SUPPLIER INVOKES {
    ReceiveReport,
    ReceiveIPM,
    ReceiveRN,
    ReceiveNRN }
  ::= id-pt-reception

management PORT
  CONSUMER INVOKES {
    ChangeAutoDiscard,
    ChangeAutoAcknowledgment,
    ChangeAutoForwarding }
  ::= id-pt-management

-- Origination abstract operations
OriginateProbe ::= ABSTRACT-OPERATION
  ARGUMENT SET {
    envelope      [0] ProbeSubmissionEnvelope,
    content       [1] IPM }
  RESULT SET {
    submission-identifier [0] ProbeSubmissionIdentifier,
    submission-time     [1] ProbeSubmissionTime }
  ERRORS {
    SubscriptionError,
    RecipientImproperlySpecified }

OriginateIPM ::= ABSTRACT-OPERATION
  ARGUMENT SET {
    envelope      [0] MessageSubmissionEnvelope,
    content       [1] IPM }
  RESULT SET {
    submission-identifier [0] MessageSubmissionIdentifier,
    submission-time     [1] MessageSubmissionTime }

```

```

    ERROR {
        SubscriptionError,
        RecipientImproperlySpecified }
OriginateRN ::= ABSTRACT-OPERATION
    ARGUMENT SET {
        envelope      [0] MessageSubmissionEnvelope,
        content        [1] RN }
    RESULT SET {
        submission-identifier [0] MessageSubmissionIdentifier,
        submission-time      [1] MessageSubmissionTime }
    ERROR {
        SubscriptionError,
        RecipientImproperlySpecified }

-- Reception abstract operations
ReceiptReport ::= ABSTRACT-OPERATION
    ARGUMENT SET {
        envelope      [0] ReportDeliveryEnvelope,
        undelivered-object [1] InformationObject OPTIONAL }
    RESULT
    ERRORS { }

ReceiveIPM ::= ABSTRACT-OPERATION
    ARGUMENT SET {
        envelope      [0] MessageDeliveryEnvelope,
        content        [1] IPM }
    RESULT
    ERRORS { }

ReceiveRN ::= ABSTRACT-OPERATION
    ARGUMENT SET {
        envelope      [0] MessageDeliveryEnvelope,
        content        [1] RN }
    RESULT
    ERRORS { }

ReceiveNRN ::= ABSTRACT-OPERATION
    ARGUMENT SET {
        envelope      [0] MessageDeliveryEnvelope,
        content        [1] NRN }
    RESULT
    ERRORS { }

-- Management abstract operations
ChangeAutoDiscard ::= ABSTRACT-OPERATION
    ARGUMENT SET {
        auto-discard-expired-IPMs [0] BOOLEAN,
        auto-discard-obsolete-IPMs [1] BOOLEAN }
    RESULT
    ERRORS { }

ChangeAutoAcknowledgement ::= ABSTRACT-OPERATION
    ARGUMENT SET {
        auto-acknowledge-IPMs [0] BOOLEAN,
        auto-acknowledge-suppl-receipt-info [1]
        SupplementaryInformation }
    RESULT
    ERRORS {
        SubscriptionError }

ChangeAutoForwarding ::= ABSTRACT-OPERATION
    ARGUMENT SET {
        auto-forward-IPMs [0] BOOLEAN,
        auto-forward-recipients [1] SEQUENCE OF ORName OPTIONAL,
        auto-forward-heading [2] Heading OPTIONAL,

```

```

        auto-forward-comment          [3] AutoForwardComment OPTIONAL }
RESULT
ERRORS {
    SubscriptionError,
    RecipientImproperlySpecified }
-- Abstract errors
SubscriptionError ::= ABSTRACT-ERROR
    PARAMETER SET {
        problem          [0] SubscriptionProblem }
SubscriptionProblem ::= ENUMERATED {
    ipms-eos-not-subscribed          (0),
    mts-eos-not-subscribed          (1) }
END -- of IPMSAbstractService

```

ANNEX H

(to Recommendation X.420)

Reference definition of heading extensions

This Annex is an integral part of this Recommendation.

This Annex, a supplement to Annex A, defines for reference purposes the heading extensions defined for interpersonal messaging. It uses the HEADING-EXTENSION macro of § 12.2.17.

```

IPMSHeadingExtensions { joint-iso-ccitt
    mhs-motis(6) ipms(1) modules(0) heading-extensions(6) }
DEFINITIONS IMPLICIT TAGS ::=
BEGIN
-- Prologue
-- Exports everything.
IMPORTS
    -- IPMS information objects
    HEADING-EXTENSION
    ----
    FROM IPMSInformationObjects { joint-iso-ccitt
        mhs-motis(6) ipms(1) modules(0) information-objects(2) };
    -- IPMS object identifiers
    id-hex-incomplete-copy, id-hex-languages
    ----
    FROM IPMSObjectsIdentifiers { joint-iso-ccitt
        mhs-motis(6) ipms(1) modules(0) objects-identifiers(0) };
-- Incomplete copy
Incomplete copy HEADING-EXTENSION
    ::= id-hex-incomplete-copy
IncompleteCopy ::= NULL
-- Languages
languages HEADING-EXTENSION
    VALUE SET OF Language
    ::= id-hex-languages
Language ::= PrintableString (SIZE (2..2))
END -- of IPMSHeadingExtensions

```

ANNEX I

(to Recommendation X.420)

Reference definition of extended body part types

This Annex is an integral part of this Recommendation.

This Annex, a supplement to Annex B, defines for reference purposes certain extended body part types.

```
IPMSExtendedBodyPartTypes {joint-iso-ccitt
    mhs-motis(6) ipms(1) modules(0) extended-body-part-types(7)}
DEFINITIONS IMPLICIT TAGS ::=
BEGIN

-- Prologue
-- Exports everything.

IMPORTS

-- IPMS information objects
    BilaterallyDefinedBodyPart, EncryptedData,
    EncryptedParameters, EXTENDED-BODY-PART-TYPE, G3FacsimileData,
    G3FacsimileParameters, G4Class1BodyPart, IA5TextData,
    IA5TextParameters, MessageData, MessageParameters,
    MixedModeBodyPart, NationallyDefinedBodyPart, TeletexData,
    TeletexParameters, VideotexData, VideotexParameters,
    VoiceData, VoiceParameters
    ----
    FROM IPMSInformationObjects {joint-iso-ccitt
        mhs-motis(6) ipms(1) modules(0) information-objects(2)}

-- IPMS object identifiers
    id-ep-encrypted,
    id-ep-g3-facsimile,
    id-ep-ia5-text,
    id-ep-message,
    id-ep-teletex,
    id-ep-videotex,
    id-ep-voice,
    id-et-bilaterally-defined, id-et-encrypted, id-et-g3-facsimile,
    id-et-g4-class1, id-et-ia5-text, id-et-message,
    id-et-mixed-mode, id-et-nationally-defined, id-et-teletex,
    id-et-videotex, id-et-voice
    ----
    FROM IPMSObjectsIdentifiers {joint-iso-ccitt
        mhs-motis(6) ipms(1) modules(0) objects-identifiers(0)}

-- Extended IA5 text body part
ia5-text-body-part EXTENDED-BODY-PART-TYPE
    PARAMETERS IA5TextParameters IDENTIFIED BY id-ep-ia5-text
    DATA IA5TextData
    ::= id-et-ia5-text

-- Extended voice body part
voice-body-part EXTENDED-BODY-PART-TYPE
    PARAMETERS VoiceParameters IDENTIFIED BY id-ep-voice
    DATA VoiceData
    ::= id-et-voice
```

```

-- Extended G3 facsimile body part
g3-facsimile-body-part EXTENDED-BODY-PART-TYPE
    PARAMETERS G3FacsimileParameters IDENTIFIED BY id-ep-g3-facsimile
    DATA G3FacsimileData
    ::= id-et-g3-facsimile

-- Extended G4 class 1 body part
g4-class1-body-part EXTENDED-BODY-PART-TYPE
    DATA G4Class1BodyPart
    ::= id-et-g4-class1

-- Extended teletex body part
teletex-body-part EXTENDED-BODY-PART-TYPE
    PARAMETERS TeletexParameters IDENTIFIED BY id-ep-teletex
    DATA TeletexData
    ::= id-et-teletex

-- Extended videotex body part
videotex-body-part EXTENDED-BODY-PART-TYPE
    PARAMETERS VideotexParameters IDENTIFIED BY id-ep-videotex
    DATA VideotexData
    ::= id-et-videotex

-- Extended encrypted body part
encrypted-body-part EXTENDED-BODY-PART-TYPE
    PARAMETERS EncryptedParameters IDENTIFIED BY id-ep-encrypted
    DATA EncryptedData
    ::= id-et-encrypted

-- Extended message body part
message-body-part EXTENDED-BODY-PART-TYPE
    PARAMETERS MessageParameters IDENTIFIED BY id-ep-message
    DATA MessageData
    ::= id-et-message

-- Extended mixed-mode body part
mixed-mode-body-part EXTENDED-BODY-PART-TYPE
    DATA MixedModeBodyPart
    ::= id-et-mixed-mode

-- Extended bilaterally defined body part
bilaterally-defined-body-part EXTENDED-BODY-PART-TYPE
    DATA BilaterallyDefinedBodyPart
    ::= id-et-bilaterally-defined

-- Extended nationally defined body part
nationally-defined-body-part EXTENDED-BODY-PART-TYPE
    DATA NationallyDefinedBodyPart
    ::= id-et-nationally-defined

END -- of IPMSExtendedBodyPartTypes

```

ANNEX J

(to Recommendation X.420)

Reference definition of message store attributes

This Annex is an integral part of this Recommendation.

This Annex, a supplement to Annex C, defines for reference purposes the MS attributes specific to interpersonal messaging. It uses the ATTRIBUTE macro of Recommendation X.500.

```
IPSMMessageStoreAttributes { joint-iso-ccitt
    mhs-motis(6) ipms(1) modules(0) message-store-attributes(8) }
DEFINITIONS IMPLICIT TAGS ::=
BEGIN

-- Prologue
-- Exports everything.

IMPORTS

-- IPMS heading extensions
    IncompleteCopy, Language
----
    FROM IPMSHeadingExtensions { joint-iso-ccitt
        mhs-motis(6) ipms(1) modules(0) heading-extensions(6) }

-- IPMS information objects
    AcknowledgmentModeField, AuthorizingUsersSubfield,
    AutoForwardCommentField, AutoForwardedField,
    BilaterallyDefinedBodyPart, BlindCopyRecipientsSubfield, Body,
    ConversionEITsField, CopyRecipientsSubfield,
    DiscardReasonField, EncryptedBodyPart, EncryptedData,
    EncryptedParameters, ExpiryTimeField,
    ExternallyDefinedParameters, G3FacsimileBodyPart,
    G3FacsimileData, G3FacsimileParameters, G4Class1BodyPart,
    Heading, IA5TextBodyPart, IA5TextData, IA5TextParameters,
    ImportanceField, IPMPreferredRecipientField,
    IPNOriginatorField, MessageBodyPart, MessageData,
    MessageParameters, MixedModeBodyPart,
    NationallyDefinedBodyPart, NonReceiptReasonField,
    ObsoleteIPMsSubfield, ORDescriptor, OriginatorField,
    PrimaryRecipientsSubfield, ReceiptTimeField,
    RelatedIPMsSubfield, RepliedToIPMField,
    ReplyRecipientsSubfield, ReplyTimeField, ReturnedIPMField,
    SensitivityField, SubjectField, SubjectIPMField,
    SupplReceiptInfoField, TeletexBodyPart, TeletexData,
    TeletexParameters, ThisIPMField, VideotexBodyPart,
    VideotexData, VideotexParameters, VoiceBodyPart, VoiceData,
    VoiceParameters
----
    FROM IPMSInformationObjects { joint-iso-ccitt
        mhs-motis(6) ipms(1) modules(0) information-objects(2) }

-- IPMS object identifiers
    id-bat-bilaterally-defined-body-parts, id-bat-body,
    id-bat-encrypted-body-parts, id-bat-encrypted-data,
    id-bat-encrypted-parameters, id-bat-extended-body-part-types,
    id-bat-g3-facsimile-body-parts, id-bat-g3-facsimile-data,
    id-bat-g3-facsimile-parameters, id-bat-g4-class1-body-parts,
    id-bat-ia5-text-body-parts, id-bat-ia5-text-data,
    id-bat-ia5-text-parameters, id-bat-message-body-parts,
    id-bat-message-data, id-bat-message-parameters,
    id-bat-mixed-mode-body-parts,
    id-bat-nationally-defined-body-parts,
```

```

id-bat-teletex-body-parts, id-bat-teletex-data,
id-bat-teletex-parameters, id-bat-videtex-body-parts,
id-bat-videtex-data, id-bat-videtex-parameters,
id-bat-voice-body-parts, id-bat-voice-data,
id-bat-voice-parameters, id-hat-authorizing-users,
id-hat-auto-forwarded, id-hat-blind-copy-recipients,
id-hat-copy-recipients, id-hat-expiry-time, id-hat-heading,
id-hat-importance, id-hat-incomplete-copy, id-hat-languages,
id-hat-nrn-requestors, id-hat-obsoloted-IPMs,
id-hat-originator, id-hat-primary-recipients,
id-hat-related-IPMs, id-hat-replied-to-IPM,
id-hat-reply-recipients, id-hat-reply-requestors,
id-hat-reply-time id-hat-rn-requestors, id-hat-sensitivity,
id-hat-subject, id-hat-this-ipm, id-nat-acknowledgment-mode,
id-nat-auto-forward-comment, id-nat-conversion-eits,
id-nat-discard-reason, id-nat-ipm-preferred-recipient,
id-nat-ipn-originator, id-nat-non-receipt-reason,
id-nat-receipt-time, id-nat-returned-ipm, id-nat-subject-ipm,
id-nat-suppl-receipt-info, id-sat-ipm-entry-type,
id-sat-ipm-synopsis
----
FROM IPMSObjectIdentifiers { joint-iso-ccitt
    mhs-motis(6) ipms(1) modules(0) object-identifiers(0) }
-- MS abstract service
MS-EITs, SequenceNumber
----
FROM MSAbstractService { joint-iso-ccitt
    mhs-motis(6) ms(4) modules(0) abstract-service(1) }
-- MTS abstract service
EncodedInformationTypes
----
FROM MTSAbstractService { joint-iso-ccitt
    mhs-motis(6) mts(3) modules(0) mts-abstract-service(1) };
-- Directory information framework
ATTRIBUTE
----
FROM InformationFramework { joint-iso-ccitt
    ds(5) modules(1) informationFramework(1) };

Time ::= UTCTime

-- SUMMARY ATTRIBUTES

-- IPM entry type
ipm-entry-type ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX IPMEntryType
    MATCHES FOR EQUALITY
    SINGLE VALUE
    ::= id-sat-ipm-entry-type

IPMEntryType ::= ENUMERATED {
    ipm        (0),
    rn         (1),
    nrn        (2) }

-- IPM synopsis
ipm-synopsis ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX IPMSynopsis
    SINGLE VALUE
    ::= id-sat-ipm-synopsis

```

```

IPMSynopsis ::= SEQUENCE OF BodyPartSynopsis
BodyPartSynopsis ::= CHOICE {
    message      [0] MessageBodyPartSynopsis,
    non-message  [1] NonMessageBodyPartSynopsis }
MessageBodyPartSynopsis ::= SEQUENCE {
    number       [0] SequenceNumber,
    synopsis    [1] IPMSynopsis }
NonMessageBodyPartSynopsis ::= SEQUENCE {
    type         [0] OBJECT IDENTIFIER,
    parameters   [1] ExternallyDefinedParameters,
    size         [2] INTEGER,
    processed    [3] BOOLEAN DEFAULT FALSE }
-- HEADING ATTRIBUTES
-- Heading
heading ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX Heading
    SINGLE VALUE
    ::= id-hat-heading
-- Heading analyses
rn-requestors ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX ORDescriptor
    MATCHES FOR EQUALITY
    MULTI VALUE
    ::= id-hat-rn-requestors
nrn-requestors ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX ORDescriptor
    MATCHES FOR EQUALITY
    MULTI VALUE
    ::= id-hat-nrn-requestors
reply-requestors ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX ORDescriptor
    MATCHES FOR EQUALITY
    MULTI VALUE
    ::= id-hat-reply-requestors
-- Heading fields
this-ipm ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX ThisIPMField
    MATCHES FOR EQUALITY
    SINGLE VALUE
    ::= id-hat-this-ipm
originator ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX OriginatorField
    MATCHES FOR EQUALITY
    SINGLE VALUE
    ::= id-hat-originator
replied-to-IPM ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX RepliedToIPMField
    MATCHES FOR EQUALITY
    SINGLE VALUE
    ::= id-hat-replied-to-IPM

```

subject ATTRIBUTE
 WITH ATTRIBUTE-SYNTAX SubjectField
 MATCHES FOR EQUALITY SUBSTRINGS
 SINGLE VALUE
 ::= id-hat-subject

expiry-time ATTRIBUTE
 WITH ATTRIBUTE-SYNTAX ExpiryTimeField
 MATCHES FOR EQUALITY ORDERING
 SINGLE VALUE
 ::= id-hat-expiry-time

reply-time ATTRIBUTE
 WITH ATTRIBUTE-SYNTAX ReplyTimeField
 MATCHES FOR EQUALITY ORDERING
 SINGLE VALUE
 ::= id-hat-reply-time

importance ATTRIBUTE
 WITH ATTRIBUTE-SYNTAX ImportanceField
 MATCHES FOR EQUALITY
 SINGLE VALUE
 ::= id-hat-importance

sensitivity ATTRIBUTE
 WITH ATTRIBUTE-SYNTAX SensitivityField
 MATCHES FOR EQUALITY
 SINGLE VALUE
 ::= id-hat-sensitivity

auto-forwarded ATTRIBUTE
 WITH ATTRIBUTE-SYNTAX AutoForwardedField
 MATCHES FOR EQUALITY
 SINGLE VALUE
 ::= id-hat-auto-forward

-- *Heading sub-fields*

authorizing-users ATTRIBUTE
 WITH ATTRIBUTE-SYNTAX Authorizing-UsersSubfield
 MATCHES FOR EQUALITY
 MULTI VALUE
 ::= id-hat-authorizing-users

primary-recipients ATTRIBUTE
 WITH ATTRIBUTE-SYNTAX PrimaryRecipientsSubfield
 MATCHES FOR EQUALITY
 MULTI VALUE
 ::= id-hat-primary-recipients

copy-recipients ATTRIBUTE
 WITH ATTRIBUTE-SYNTAX CopyRecipientsSubfield
 MATCHES FOR EQUALITY
 MULTI VALUE
 ::= id-hat-copy-recipients

blind-copy-recipients ATTRIBUTE
 WITH ATTRIBUTE-SYNTAX BlindCopyRecipientsSubfield
 MATCHES FOR EQUALITY
 MULTI VALUE
 ::= id-hat-blind-copy-recipients

```

obsoleted-IPMs ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX ObsoletedIPMsSubfield
    MATCHES FOR EQUALITY
    MULTI VALUE
    ::= id-hat-obsoleted-IPMs

related-IPMs ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX RelatedIPMsSubfield
    MATCHES FOR EQUALITY
    MULTI VALUE
    ::= id-hat-related-IPMs

reply-recipients ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX ReplyRecipientsSubfield
    MATCHES FOR EQUALITY
    MULTI VALUE
    ::= id-hat-reply-recipients

-- Heading extensions

incomplete-copy ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX IncompleteCopy
    MATCHES FOR EQUALITY
    SINGLE VALUE
    ::= id-hat-incomplete-copy

languages ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX Language
    MATCHES FOR EQUALITY
    MULTI VALUE
    ::= id-hat-languages

-- BODY ATTRIBUTES

-- Body

body ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX Body
    SINGLE VALUE
    ::= id-bat-body

-- Basic body parts

ia5-text-body-parts ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX IA5TextBodyParts
    MULTI VALUE
    ::= id-bat-ia5-text-body-parts

voice-body-parts ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX VoiceBodyPart
    MULTI VALUE
    ::= id-bat-voice-body-parts

g3-facsimile-body-parts ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX G3FacsimileBodyPart
    MULTI VALUE
    ::= id-bat-g3-facsimile-body-parts

g4-class1-body-parts ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX G4Class1BodyPart
    MULTI VALUE
    ::= id-bat-g4-class1-body-parts

teletex-body-parts ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX TeletexBodyPart
    MULTI VALUE
    ::= id-bat-teletex-body-parts

```

videotex-body-parts ATTRIBUTE
WITH ATTRIBUTE-SYNTAX VideotexBodyPart
MULTI VALUE
::= id-bat-videotex-body-parts

encrypted-body-parts ATTRIBUTE
WITH ATTRIBUTE-SYNTAX EncryptedBodyPart
MULTI VALUE
::= id-bat-encrypted-body-parts

message-body-parts ATTRIBUTE
WITH ATTRIBUTE-SYNTAX SequenceNumber
MULTI VALUE
::= id-bat-message-body-parts

mixed-mode-body-parts ATTRIBUTE
WITH ATTRIBUTE-SYNTAX MixedModeBodyPart
MULTI VALUE
::= id-bat-mixed-mode-body-parts

bilaterally-defined-body-parts ATTRIBUTE
WITH ATTRIBUTE-SYNTAX BilaterallyDefinedBodyPart
MULTI VALUE
::= id-bat-bilaterally-defined-body-parts

nationally-defined-body-parts ATTRIBUTE
WITH ATTRIBUTE-SYNTAX NationallyDefinedBodyPart
MULTI VALUE
::= id-bat-nationally-defined-body-parts

-- *Basic body part parameters component*

ia5-text-parameters ATTRIBUTE
WITH ATTRIBUTE-SYNTAX IA5TextParameters
MULTI VALUE
::= id-bat-ia5-text-parameters

voice-parameters ATTRIBUTE
WITH ATTRIBUTE-SYNTAX VoiceParameters
MULTI VALUE
::= id-bat-voice-parameters

g3-facsimile-parameters ATTRIBUTE
WITH ATTRIBUTE-SYNTAX G3FacsimileParameters
MULTI VALUE
::= id-bat-g3-facsimile-parameters

teletex-parameters ATTRIBUTE
WITH ATTRIBUTE-SYNTAX TeletexParameters
MULTI VALUE
::= id-bat-teletex-parameters

videotex-parameters ATTRIBUTE
WITH ATTRIBUTE-SYNTAX VideotexParameters
MULTI VALUE
::= id-bat-videotex-parameters

encrypted-parameters ATTRIBUTE
WITH ATTRIBUTE-SYNTAX EncryptedParameters
MULTI VALUE
::= id-bat-encrypted-parameters

message-parameters ATTRIBUTE
WITH ATTRIBUTE-SYNTAX MessageParameters
MULTI VALUE
::= id-bat-message-parameters

-- Basic body part data components

ia5-text-data ATTRIBUTE
WITH ATTRIBUTE-SYNTAX IA5TextData
MULTI VALUE
::= id-bat-ia5-text-data

voice-data ATTRIBUTE
WITH ATTRIBUTE-SYNTAX Voice-Data
MULTI VALUE
::= id-bat-voice-data

g3-facsimile-data ATTRIBUTE
WITH ATTRIBUTE-SYNTAX G3FacsimileData
MULTI VALUE
::= id-bat-g3-facsimile-data

teletex-data ATTRIBUTE
WITH ATTRIBUTE-SYNTAX TeletexData
MULTI VALUE
::= id-bat-teletex-data

videotex-data ATTRIBUTE
WITH ATTRIBUTE-SYNTAX VideotexData
MULTI VALUE
::= id-bat-videotex-data

encrypted-data ATTRIBUTE
WITH ATTRIBUTE-SYNTAX EncryptedData
MULTI VALUE
::= id-bat-encrypted-data

message-data ATTRIBUTE
WITH ATTRIBUTE-SYNTAX MessageData
MULTI VALUE
::= id-bat-message-data

-- Extended body part types

extended-body-part-types ATTRIBUTE
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER
MATCHES FOR EQUALITY
MULTI VALUE
::= id-bat-extended-body-part-types

-- Extended body parts

-- (These attributes cannot be enumerated. See § C.3.6)

-- NOTIFICATION ATTRIBUTES

-- Common fields

subject-ipm ATTRIBUTE
WITH ATTRIBUTE-SYNTAX SubjectIPMField
MATCHES FOR EQUALITY SUBSTRINGS
SINGLE VALUE
::= id-nat-subject-ipm

ipn-originator ATTRIBUTE
WITH ATTRIBUTE-SYNTAX IPNOriginatorField
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-nat-ipn-originator

```

ipm-preferred-recipient ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX IPMPreferredRecipientField
    MATCHES FOR EQUALITY
    SINGLE VALUE
    ::= id-nat-ipm-preferred-recipient

conversion-eits ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX MS-EITs
    MATCHES FOR EQUALITY
    MULTI VALUE
    ::= id-nat-conversion-eits

-- Non-receipt fields

non-receipt-reason ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX NonReceiptReasonField
    MATCHES FOR EQUALITY
    SINGLE VALUE
    ::= id-nat-non-receipt-reason

discard-reason ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX DiscardReasonField
    MATCHES FOR EQUALITY
    SINGLE VALUE
    ::= id-nat-discard-reason

auto-forward-comment ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX AutoForwardCommentField
    MATCHES FOR EQUALITY SUBSTRINGS
    SINGLE VALUE
    ::= id-nat-auto-forward-comment

returned-ipm ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX ReturnedIPMField
    SINGLE VALUE
    ::= id-nat-returned-IPM

-- Receipt fields

receipt-time ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX ReceiptTimeField
    MATCHES FOR EQUALITY ORDERING
    SINGLE VALUE
    ::= id-nat-receipt-time

acknowledgment-mode ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX AcknowledgmentModeField
    MATCHES FOR EQUALITY
    SINGLE VALUE
    ::= id-nat-acknowledgment-mode

suppl-receipt-info ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX SupplReceiptInfoField
    MATCHES FOR EQUALITY SUBSTRINGS
    SINGLE VALUE
    ::= id-nat-suppl-receipt-info

END -- of IPMSMessagesStoreAttributes

```

ANNEX K

(to Recommendation X.420)

Reference definition of upper bounds

This Annex is an integral part of this Recommendation but is not a part of the corresponding ISO International Standard.

This Annex defines for reference purposes the upper bounds of various variable-length information items whose abstract syntaxes are defined in the ASN.1 modules of prior annexes.

```
IPMSUpperBounds { joint-iso-ccitt
    mhs-motis(6) ipms(1) modules(0) upper-bounds(10) }
DEFINITIONS IMPLICIT TAGS ::=
BEGIN

-- Prologue
-- Exports everything.

IMPORTS -- nothing --;

-- Upper bounds
ub-auto-forward-comment      INTEGER ::= 256
ub-free-form-name            INTEGER ::= 64
ub-ipm-identifier-suffix     INTEGER ::= 2
ub-local-ipm-identifier      INTEGER ::= 64
ub-subject-field             INTEGER ::= 128
ub-telephone-number         INTEGER ::= 32
END -- of IPMSUpperBounds
```

ANNEX L

(to Recommendation X.420)

Support of the interpersonal messaging service

This Annex is an integral part of this Recommendation.

The interpersonal messaging service which the IPMS provides to users is defined, in non-technical terms, in Recommendation X.400. The service comprises a number of elements of service (**IPM EOS**), each representing one aspect of the service, and each defined in one or two paragraphs of prose. The present Annex indicates in detail how the present, more technical specification realizes each IPM EOS. Equivalently, it identifies the aspects of the specification a UA, e.g., must implement for it to be said to support a particular IPM EOS.

Associated with each IPM EOS are one or more information items that may appear as IPM components. The information item associated with the sensitivity indication IPM EOS, e.g., is the sensitivity heading field. A UA, TLMA, or AU shall be said to support a particular IPM EOS upon origination or reception if, and only if, it supports upon origination or reception (see § 22.1) the information items associated with that IPM EOS.

Note 1 – The task of realizing an IPM EOS may fall, in principle, upon any of the secondary objects that result from the refinement of the IPMS. In the present context, however, it is assumed that the MTS and every MS, by virtue of their application-independence, support every IPM-EOS, and that they do so without having made special provision for any of them.

Note 2 – As described in § 14, a UA makes available to its user many of the capabilities that its MS offers. These capabilities realize the elements of the message retrieval service which is defined in Recommendation X.400. The correspondence between the elements of that service and the associated technical capabilities is given in Recommendation X.413.

Note 3 – As described in § 14, a UA makes available to its user many of the capabilities that the MTS offers. These capabilities realize the elements of the message transfer service which is defined in Recommendation X.400. The correspondence between the elements of that service and the associated technical capabilities is given in Recommendation X.411.

L.1 *Support of recipient specifier components*

Some IPM EOS are realized by means of recipient specifier components. The IPM EOS in this category are listed in the first column of Table L-1/X.420. The second and third columns identify the recipient specifier component, and the particular value of that component, that are the information items associated with each listed IPM EOS.

TABLE L-1/X.420
Support of recipient specifier components

Element of service	Recipient specifier component	Value
Non-receipt notification request	Notification-requests	nrn
Receipt notification request indication	Notification-requests	rn
Reply request indication (see also Table L-2/X.420)	Reply-requested	true

Note 1 - Recipient specifiers appear as sub-fields of the Primary Recipients, Copy Recipients, and Blind Copy Recipients heading fields.

Note 2 - Every IPM EOS except Reply Request Indication falls into exactly one category. The Reply Request Indication IPM EOS falls into two categories, as indicated in the Table.

L.2 *Support of heading fields*

Some IPM EOS are realized by means of heading fields. The IPM EOS in this category are listed in the first column of Table L-2/X.420. The second column identifies the heading fields that are the information items associated with each listed IPM EOS. In the case of the extension field, the second column also identifies, in parentheses, the relevant heading extension.

L.3 *Support of body aspects*

Some IPM EOS are realized by means of aspects of the body. The IPM EOS in this category are listed in the first column of Table L-3/X.420. The second column identifies the body aspect that is the information item associated with each listed IPM EOS.

TABLE L-2/X.420

Support of heading fields

Element of service	Heading field
Authorizing users indication	Authorizing users
Auto-forwarded indication	Auto-forwarded
Blind copy recipient indication	Blind copy recipients
Cross-referencing indication	Related IPMs
Expiry date indication	Expiry time
Importance indication	Importance
IP-message identification	This IPM
Incomplete copy indication	Extensions (incomplete copy)
Language indication	Extensions (languages)
Obsoleting indication	Obsoleted IPMs
Originator indication	Originator
Primary and copy recipients indication	Primary recipients Copy recipients
Reply request indication (see also Table L-3/X.420)	Reply time Reply recipients
Replying IP-message indication	Replied-to IPM
Sensitivity indication	Sensitivity
Subject indication	Subject

Note – Every IPM EOS except Reply Request Indication falls into exactly one category. The Reply Request Indication IMP EOS falls into two categories, as indicated in the Table.

TABLE L-3/X.420

Support of body aspects

Element of service	Body aspect
Body part encryption indication	Encrypted body part
Forwarded IP-message indication	Message body part
Multi-part body	Body with two or more parts
Typed body	Body (itself)

Note – Support of the Typed Body IPM EOS is intrinsic to any implementation of any secondary object.

ANNEX M

Differences between CCITT Recommendation and ISO Standard

This Annex is not a part of this Recommendation.

This Annex lists all but the purely stylistic differences between this Recommendation and the corresponding ISO International Standard.

The following are the differences that exist:

- a) The ISO International Standard corresponding to this Recommendation defines a general text body part, while this Recommendation does not.
- b) The upper bounds of Annex K are an integral part of this Recommendation but are not a part of the corresponding ISO International Standard.
- c) The CCITT text on subject recipient specifier in § 8 states that it may contain either “an O/R name of the preferred recipient” or “an O/R name appearing in the message's DL expansion history”. The ISO Standard has excluded this second possibility.

ANNEX N

Summary of changes to 1984 specification

This Annex is not a part of this Recommendation.

Editorially, this Recommendation differs substantially from Recommendation X.420 (1984). Technically, however, the differences are few. The present Annex lists the technical changes. It is intended as an aid to an implementor of Recommendation X.420 (1984), enabling him to see at a glance how his implementation might be affected by the 1988 specification.

The following, and only the following, substantive changes relevant to interworking between 1984 and 1988 UAs, MSs, TLMAs, and AUs are embodied in the present specification. All but the first are changes to the format of the information objects now defined in the ASN.1 module, IPMSInformationObjects:

- a) The content type assigned to P2 has changed. Formerly identified by the integer 2, P2 now is identified by either the integer 2 or 22, depending upon the functionality employed in a particular instance of communication by means of the MTS (see § 20.2.).
- b) The omission of the user members of IPMIdentifier is now denigrated.
- c) The extensions member has been added to heading. Its grade is optional.
- d) The telex and simple formattable document body part types have been abandoned. (The former had been identified but not defined.)
- e) The syntax member has been added to VideotexParameters. Its grade is optional.
- f) The presence of the delivery-time member of the MessageParameters in the absence of its delivery-envelope member, or vice-versa, is now denigrated.
- g) The bilaterally-defined and externally-defined alternatives have been added to BodyPart.
- h) The following protocol elements, defined in Recommendation X.411 and incorporated in protocol elements of this Recommendation by reference, have changed:
 - i) ORName
 - ii) ORAddress
 - iii) MessageDeliveryEnvelope
 - iv) EncodedInformationTypes
 - v) SupplementaryInformation.
- i) Specifying a value of zero length of any of the following data types is now denigrated:
 - i) LocalIPMIdentifier
 - ii) FreeFormName

- iii) TelephoneNumber
 - iv) SubjectField
 - v) AutoForwardComment.
- j) Upper bounds have been imposed upon certain variable-length protocol elements.

Note – The upper bounds imposed are those found in § 4.3 of Version 6 of the *X.400-series Implementor's Guide*.

ITU-T RECOMMENDATIONS SERIES

Series A	Organization of the work of the ITU-T
Series B	Means of expression: definitions, symbols, classification
Series C	General telecommunication statistics
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	TMN and network maintenance: international transmission systems, telephone circuits, telegraphy, facsimile and leased circuits
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks and open system communications
Series Y	Global information infrastructure and Internet protocol aspects
Series Z	Languages and general software aspects for telecommunication systems