INTERNATIONAL TELECOMMUNICATION UNION

CCITT

X.440

THE INTERNATIONAL
TELEGRAPH AND TELEPHONE
CONSULTATIVE COMMITTEE

(09/92)

DATA COMMUNICATION NETWORKS

MESSAGE HANDLING SYSTEMS: VOICE MESSAGING SYSTEM

Recommendation X.440

FOREWORD

The CCITT (the International Telegraph and Telephone Consultative Committee) is a permanent organ of the International Telecommunication Union (ITU). CCITT is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The Plenary Assembly of CCITT which meets every four years, establishes the topics for study and approves Recommendations prepared by its Study Groups. The approval of Recommendations by the members of CCITT between Plenary Assemblies is covered by the procedure laid down in CCITT Resolution No. 2 (Melbourne, 1988).

Recommendation X.440 was prepared by Study Group VII and was approved under the Resolution No. 2 procedure on the 10th of September 1992.

CCITT NOTE

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

© ITU 1993

All rights reserved. No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the ITU.

CONTENTS

Page

1 Scope 1

2 Normative references 1

2.1 Paired CCITT Recommendations | International Standards equivalent in technical content 1

2.2 Additional references 3

3 Definitions 3

3.1 Common definitions for MHS 3

3.2 Common definitions for Abstract Syntax Notation One 3

3.3 Voice service definitions 3

3.4 Voice Messaging System Definitions 4

4 Abbreviations 4

5 Conventions 5

5.1 Terms 5

5.2 ASN.1 5

5.2.1 Conventions for attribute types in Table 1/X.440 6

5.2.2 Conventions for attribute types in Table 2/X.440 6

6 Information objects 6

7 Common data types 7

7.1 VM Identifier 7

7.2 OR Descriptor 7

7.3 Spoken Name 7

7.4 Extensions 7

8 Voice message 8

8.1 Heading fields 9

8.1.1 This VM 10

8.1.2 Originator 10

8.1.3 Recipients 10

8.1.4 Obsoleted VM 12

8.1.5 VM (Spoken) subject 12

8.1.6 Expiry time 13

8.1.7 Importance 13

8.1.8 Sensitivity 13

8.1.9 Message forwarded 13

8.1.10 VMessage forwarding permitted 13

8.1.11 Language 13

8.1.12 Voice encoding type 14

8.1.13 VM creation time 14

8.1.14 VN receiver 14

8.1.15 VMGS-user security elements 15

8.1.16 Heading extensions 15

8.2 Body part types 15

8.2.1 Voice body part 16

8.2.2 VM body part 16

8.2.3 VM externally defined body parts 18

9 Voice notifications 18

9.1 Common fields 19

9.1.1 Subject VM 20

9.1.2 VN originator 20

9.1.3 First recipient 20

9.1.4 Notice creation time 20

9.1.5 VN voice encoding type 21

Page

9.1.6 Conversion indication ... 21
9.1.7 Notification security elements ... 21
9.1.8 Original VM spoken subject ... 21
9.1.9 Subject VM other recipients ... 21
9.1.10 VN supplementary information ... 22
9.1.11 Notifications extensions ... 22
9.2 Receipt notifications ... 22
9.2.1 RN common fields ... 22
9.2.2 Receipt notification extensions ... 23
9.3 Service notifications ... 23
9.3.1 SN common fields ... 23
9.3.2 Service notification reason ... 23
9.3.3 Service notification extensions ... 24
9.4 Non-receipt notifications ... 24
9.4.1 NRN common fields ... 24
9.4.2 Non-receipt reason codes ... 24
9.4.3 NRN extensions ... 26
10 Primary object types ... 26
10.1 Voice messaging system user ... 27
10.2 Voice messaging system ... 27
11 Primary port types ... 28
11.1 Origination port ... 28
11.2 Reception port ... 28
11.3 Management port ... 28
12 Abstract operations ... 28
12.1 Origination abstract operations ... 29
12.1.1 Originate VM ... 29
12.1.2 Originate voice notifications ... 30
12.2 Reception abstract operations ... 30
12.2.1 Receive Report ... 31
12.2.2 Receive VM ... 31
12.2.3 Receive VN ... 31
12.3 Management abstract operations ... 32
12.3.1 Change auto-discard ... 32
12.3.2 Change auto-acknowledgement ... 32
12.3.3 Change auto-forwarding ... 32
12.3.4 Change auto-SNotice ... 33
12.4 Performance of management operations ... 34
12.4.1 Change auto-discard ... 34
12.4.2 Change auto-acknowledgment ... 34
12.4.3 Change auto-forwarding ... 35
12.4.4 Change auto-SNotice ... 35
13 Abstract errors ... 35
13.1 Subscription error ... 35
13.2 Recipient improperly specified ... 36
14 Other capabilities ... 36
15 Secondary object types ... 36
15.1 VMGS user agent ... 38
15.2 VMGS message store ... 38
15.3 Telephone service access unit ... 38
15.4 Message transfer system ... 39

Page

16 Secondary port types 39
16.1 Submission port 39
16.2 Delivery port 39
16.3 Retrieval port 39
16.4 Administration port 40
16.5 Import port 40
16.6 Export port 40
16.7 Management port 40
17 User agent operation 40
17.1 Performance of origination operations 40
17.1.1 Originate VM 40
17.1.2 Originate VN 41
17.2 Invocation of reception operations 43
17.2.1 Receive report 43
17.2.2 Receive VM 43
17.2.3 Receive VN 43
17.3 Internal procedures 43
17.3.1 Acceptance of the VM 44
17.3.2 VM Service Notification (SN) generation 45
17.3.3 VM Forwarding 46
18 Message Store operation 51
18.1 Binding to the MS 51
18.1.1 Abstract-bind argument 51
18.1.2 Abstract-bind result 51
18.2 Creation of Information Objects 51
18.2.1 Mapping of an MHS message containing a VM into a VMGS-MS 52
18.2.2 Mapping of forwarded messages in MS 52
18.3 Maintenance of attributes 53
18.4 Voice Messaging MS auto forward action types 54
18.4.1 VM-Forwarding-with-NRN 55
18.4.2 VM-Forwarding-with-RN 57
18.4.3 VM-Forwarding-without-notification 59
18.5 Voice messaging MS auto service notification action type 60
18.6 Message store attributes 61
18.6.1 Summary attributes 66
18.6.2 Voice notification indicator 68
18.6.3 Heading attributes 68
18.6.4 Body attributes 71
18.6.5 Notification attributes 73
18.7 Procedures for Voice Messaging MS 76
18.7.1 Additional procedures for message delivery 76
19 Message contents 76
19.1 Content 76
19.2 Content type 77
19.3 Content length 77
19.4 Encoded information types 77
20 Port realization 77
21 Conformance 77
21.1 Origination versus reception 78
21.2 Requirements of a VMG conformance statement 78
21.2.1 Static requirements 78
21.2.2 Dynamic requirements 78

Page

Annex A – VM Object Identifiers – Reference definition 79
Annex B – VM Abstract Information Objects – Reference definition 82
Annex C – VM Message Store attributes – Reference definition 90
Annex D – Message Store VMGS Auto-Action types – Reference definition 98
Annex E – VMGS functional objects - Reference definition 101
Annex F – VMGS abstract service – Reference definition 103
Annex G – VMGS upper bounds parameters – Reference definition 106
Annex H – Index 107

INTRODUCTION

This Recommendation defines an OSI Message Handling System protocol that is intended to be used in the exchange of voice encoded messages in a store and forward manner as specified in Recommendation F.440.

The protocol specified in this Recommendation is based on the CCITT X.400-Series Recommendations and International Standard 10021 Parts 1 to 7.

Recommendation X.440

MESSAGE HANDLING SYSTEMS: VOICE MESSAGING SYSTEM

(1992)

1 Scope

This Recommendation is one of a set of CCITT Recommendations for Message Handling. The entire set provides a comprehensive blueprint for a message handling system (MHS) realized by any number of cooperating 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 (AU) link the MTS to other non-MHS services such as Teletex or the Telephone Service. Effectively, an AU enables a subscriber of a non-MHS service to become an indirect user of MHS. An MHS direct user is assisted in the preparation, storage, and rendering 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 (MTA) which collectively perform the store-and-forward message transfer function.

This Recommendation defines the message handling application called a *voice messaging system* which specifies *voice messaging* (VMG), a form of message handling tailored for exchange of (voice) encoded information between users, a new message content type and associated procedures known as $\mathbf{P_{vm}}$. It is designed to meet the requirements of users of Recommendation G.721 (1988) 32K ADPCM and other commonly used voice syntaxes.

This Recommendation is one of a series on message handling. CCITT Rec. X.402 | ISO/IEC 10021-2 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 CCITT Recommendations | International Standards. CCITT Rec. X.402 | ISO/IEC 10021-2 identifies those documents as well.

2 Normative references

The following CCITT Recommendations and International Standards contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication the editions indicated were valid. All Recommendations | International Standards are subject to revision, and parties to agreements based on this Recommendation are encouraged to investigate the possibility of applying the most recent edition of the Recommendations | Standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards. The CCITT Secretariat maintains a list of currently valid CCITT Recommendations.

2.1 *Paired CCITT Recommendations | International Standards equivalent in technical content*

– CCITT Recommendation X.208 (1988), *Specification of Abstract Syntax Notation One (ASN.1)*.

  ISO/IEC 8824:1990, *Information technology – Open Systems Interconnection, Specification of Abstract Syntax Notation One (ASN.1)*.

– CCITT Recommendation X.209 (1988), *Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1)*

  ISO/IEC 8825:1990, *Information technology – Open Systems Interconnection, Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1)*.

– CCITT Recommendation X.400 (1992), *Message Handling Systems: System and Service Overview*.

  ISO/IEC 10021-1:1990, *Information technology – Text communication – Message-oriented text interchange systems (MOTIS) – Part 1: System and Service Overview*.

– CCITT Recommendation X.402 (1992), *Message Handling Systems: Overall Architecture*.

  ISO/IEC 10021-2:1990, *Information technology – Text communication – Message-oriented text interchange systems (MOTIS) – Part 2: Overall Architecture*.

– CCITT Recommendation X.407 (1988), *Message Handling Systems: Abstract Service Definition Conventions*.

  ISO/IEC 10021-3:1990, *Information technology – Text communication – Message-oriented text interchange systems (MOTIS) – Part 3: Message transfer system: Abstract Service Definition and Procedures*.

– CCITT Recommendation X.411 (1992), *Message Handling Systems: Message Transfer System: Abstract Service Definition and Procedures.*

  ISO/IEC 10021-4:1990, *Information technology – Text communication – Message-oriented text interchange systems (MOTIS) – Part 4: Message transfer system: Abstract Service Definition and Procedures*.

– CCITT Recommendation X.413 (1992), *Message Handling Systems: Message Store: Abstract Service Definition.*

  ISO/IEC 10021-5:1990, *Information technology – Text communication – Message-oriented text interchange systems (MOTIS) – Part 5: Message Store: Abstract Service Definition*.

– CCITT Recommendation X.419 (1992), *Message Handling Systems: Protocol Specifications*.

  ISO/IEC 10021-6:1990, *Information technology – Text communication – Message-oriented text interchange systems (MOTIS) – Part 6: Protocol Specifications*.

– CCITT Recommendation X.420 (1992), *Message Handling Systems: Interpersonal Messaging System.*

  ISO/IEC 10021-7:1990, *Message-oriented text interchange systems: Interpersonal Messaging*.

– CCITT Recommendation X.500 (1988) | ISO/IEC 9594-1:1988, *The Directory: Overview of Concepts, Models, and Service.*

– CCITT Recommendation X.501 (1988) | ISO/IEC 9594-2:1988, *The Directory: Models*.

– CCITT Recommendation X.511 (1988) | ISO/IEC 9594-3:1988, *The Directory: Abstract Service Definition*.

– CCITT Recommendation X.518 (1988) | ISO/IEC 9594-4:1988, *The Directory: Procedures for Distributed Operation*.

– CCITT Recommendation X.519 (1988) | ISO/IEC 9594-5:1988, *The Directory: Protocol Specifications*.

– CCITT Recommendation X.520 (1988) | ISO/IEC 9594-6:1988, *The Directory: Selected Attribute Types*.

– CCITT Recommendation X.521 (1988) | ISO/IEC 9594-7:1988, *The Directory: Selected Object Classes*.

– CCITT Recommendation X.509 (1988) | ISO/IEC 9594-8:1988, *The Directory: Authentication Framework*.

2.2 *Additional references*

– CCITT Recommendation F.440 (1992), *Message Handling Services: The Voice Messaging Service*.

– CCITT Recommendation X.403 (1988), *Message Handling Systems: Conformance Testing*.

– CCITT Recommendation X.408 (1988), *Message Handling Systems: Encoded Information Type Conversion Rules*.

3 Definitions

For the purposes of this Recommendation, the following definitions apply.

3.1 *Common definitions for MHS*

This Recommendation uses terms defined in CCITT Rec. X.400 | ISO/IEC 10021-1, CCITT Rec. X.402 | ISO/IEC 10021-2 and CCITT Rec. X.413 | ISO/IEC 10021-5.

a) access unit;

b) body;

c) content;

d) distribution list;

e) encoded information types;

f) envelope;

g) message handling system;

h) message-oriented text interchange system;

i) message store;

j) message transfer agent;

k) message transfer system;

m) recipient;

n) submission identifier;

o) submission time;

p) synopsis;

q) telematic agent;

r) user;

s) user agent.

3.2 *Common definitions for Abstract Syntax Notation One*

This Recommendation uses the full extent of the Abstract Syntax Notation One (ASN.1) as defined in CCITT Rec. X.208 | ISO/IEC 8824.

3.3 *Voice service definitions*

This Recommendation uses terms defined in Recommendation F.440.

a) VM forwarding;

b) Voice message;

c) Voice messaging;

d) Voice messaging notification;

e) Voice messaging user;

f) Telephone service access unit.

3.4 *Voice Messaging System Definitions*

For the purposes of this Recommendation the following definitions apply:

3.4.1 **voice messaging system message store**

A Voice Messaging System Message Store is a specialised message store for the purposes of voice messaging.

3.4.2 **voice messaging system**

The Voice Messaging System is the functional object by means of which all users communicate with one another in voice messaging.

3.4.3 **voice messaging system user agent**

A Voice Messaging System User Agent is a specialised user agent for the purposes of voice messaging.

4 Abbreviations

ADPCM	Adaptive differential pulse code modulation
AU	Access Unit
ASN.1	Abstract Syntax Notation One
DL	Distribution List
EIT	Encoded Information Types
IA5	International Alphabet No. 5
IPM	Interpersonal messaging
MD	Management Domain
MHS	Message Handling System
MOTIS	Message-oriented Text Interchange System
MS	Message Store
MTA	Message Transfer Agent
MTS	Message Transfer System
NRN	Non-receipt (Forwarded) Notification
PICS	Protocol Implementation Conformance Statement
RN	Receipt Notification
SN	Service Notification
TSAU	Telephone Service Access Unit
UA	User Agent
UTC	Coordinated Universal Time
VM	Voice Message

VMG	Voice Messaging
VMGE	Voice Messaging Environment
VMGS	Voice Messaging System
VMGS-MS	Voice Messaging System Message Store
VMGS-UA	Voice Messaging System User Agent
VMGS-user	Voice Messaging System user
VN	Voice Notification

5 Conventions

5.1 *Terms*

Throughout the rest of this Recommendation, terms that refer to ASN.1 types are written with upper-case letters for all words that actually appear in the ASN.1 type (for example, OR Name).

a) **Voice Messaging System User**: The voice messaging system user (VMGS-user) is normally a person, not an application or computer process. A VMGS-user may be a direct user or indirect user. In MHS, direct users interact with the MHS through a User Agent or Message Store while indirect users interact with the MHS an Access Unit. For brevity, the term user is used throughout the rest of this Recommendation with the meaning of VMGS-user.

b) **Subject Voice Message**: The term *subject Voice Message*, abbreviated as **subject VM** refers to a specific VMG information object, a VM as defined in clause 8, that is either being held, processed or forwarded by a Voice Messaging application entity. This is not to be confused with the Subject VM field found in a VM notification, see 9.1.1.

c) **VM Forwarding**: The term identifies the VMG message forwarding behavior of VMGS-UAs and VMGS-MSs. It covers both manual and automatic forwarding aspects. This term is further refined in 17.3.3.

5.2 *ASN.1*

ASN.1 definitions appear both in the main text and in the annexes. In case of inconsistency between a definition presented in the text, and a definition presented in an annex forming an integral part of this Recommendation, the definition in the annex shall be used. ASN.1 notation is defined in CCITT Rec. X.208 | ISO/IEC 8824.

This Recommendation uses ASN.1-based descriptive conventions:

a) to define the information objects of Voice Messaging, and other data types and values of all kinds, and the ASN.1 itself;

b) to define the functional objects of Voice Messaging, the OBJECT and REFINE macros of CCITT Rec. X.407 | ISO/IEC 10021-4;

c) to define the abstract service of Voice Messaging, the PORT and ABSTRACT-operation and ERROR macros of CCITT Rec. X.407 | ISO/IEC 10021-4;

d) to define the protocol extensions, the VM-EXTENSION macro of this Recommendation;

e) to define extended body part types, the EXTENDED-BODY-PART-TYPE macro of CCITT Rec. X.420 | ISO/IEC 10021-7;

f) to define MS Auto-actions, the AUTO-ACTION macro of CCITT Rec. X.413 | ISO/IEC 10021-5;

g) to define MS attributes, the ATTRIBUTE macro of CCITT Rec. X.413 | ISO/IEC 10021-5.

ASN.1 tags are implicit throughout the ASN.1 modules defined in any annex; the module is definitive in that respect.

Note – The use of ASN.1 to describe a class or piece of information does not in itself imply 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.

5.2.1 *Conventions for attribute types in Table 1/X.440*

This Recommendation uses the conventions listed below in its definition of attribute types for the MS abstract services.

For the column headed *Single/Multi-valued*, the following values can occur:

– S: Single-valued;

– M: Multi-valued.

For the column headed *Support level by MS and UA* (where UA refers only to a UA that accesses an MS), the following values can occur:

– M: Mandatory;

– O: Optional.

For the columns headed *Presence in delivered VM*, *Presence in delivered RN*, *Presence in delivered NRN* and *Presence in delivered SN*, the presence of each attribute type is described by one of the following values:

– P: *Always present* in the entry because it is mandatory for generation by the MS or it is a mandatory or defaulted parameter in the relevant abstract operation.

– C: *Conditionally present* in the entry. It will be present because it is supported by the MS and subscribed to by the user and it was present in an optional parameter in the relevant abstract operation.

– *Always absent*, otherwise.

For the columns headed *Available for list, alert* and *Available for summarize*, the following values can occur:

– N: no

– Y: yes

Note – All attributes are available to the fetch abstract operation, subject to support by the implementation and subscription.

5.2.2 *Conventions for attribute types in Table 2/X.440*

This Recommendation uses the conventions listed below in its definition of attribute types for the MS abstract services.

For the column headed *Generated by* the following values can occur:

– MD: MessageDelivery abstract-operation;

– MS: MessageStore;

– RD: ReportDelivery abstract-operation.

6 Information objects

The information objects that users exchange in voice messaging are of two kinds: voice messages (VM), and voice notifications (VN). Specific details of the information objects used in the Voice Messaging System are defined in this clause and in clauses 7, 8, and 9.

```
InformationObject ::= CHOICE {
   vm                              [0] VM,
   vn                              [1] VN }
```

7 Common data types

Information items of several kinds appears both in voice messages and voice notifications. These common items are defined below.

7.1 *VM Identifier*

A VM Identifier is an information item that unambiguously, globally and forever uniquely identifies a VM.

It comprises an OR Name and a string which may, for example, contain a time or sequence number or other sufficient information to make this VM unique.

```
VMIdentifier ::= SET {
   user                        [0] ORName,
   user-relative-reference     [1] LocalReference }
```

Note – OR Name is defined in 8.5.5 of CCITT Rec. X.411 | ISO/IEC 10021-4.

The VM Identifier shares the same value set with the IPM Identifier defined in CCITT Rec. X.420 | ISO/IEC 10021-7. Therefore a VMGS-UA or VMGS-MS that is capable of handling both IPMs and VMs shall make sure that the Local Reference is unique both for IPMs and VMs.

A VM Identifier has the following components:

a) **user:** Identifies the user who originates the VM. One of the user's OR Names.

b) **user-relative-reference:** Unambiguously identifies the VM, distinguishing it from all other VMs that the user who is identified by the User component originates. A Printable String value shall be of a size from zero to a prescribed number of characters (see Annex G). A length of zero is discouraged.

```
LocalReference ::= PrintableString (SIZE (0..ub-vmg-local-reference))
```

7.2 *OR Descriptor*

An OR Descriptor is an information item that is defined in the IPM protocol specification's 7.1.3 of CCITT Rec. X.420 | ISO/IEC 10021-7. It identifies the user or DL by either its OR Name, free-form-name or telephone number, all of which are text encoded. In VM, it is most likely that the ORDescriptor will identify a user by a telephone number.

7.3 *Spoken Name*

The Spoken Name parameter carries an audio encoding of an associated name. It may be associated with the VMOriginatorField, RecipientField, or VNOriginatorField. The encoding of the value is per the VoiceEncodingType parameter in a VM or the VNVoiceEncodingParameter in a VN. Its size is constrained in seconds of duration.

```
SpokenName ::= ANY      -- defined by VoiceEncodingType
                        -- or VNVoiceEncodingType and not to exceed 10 seconds in duration
```

7.4 *Extensions*

A mechanism is provided which allows for future extensions to this Recommendation.

```
ExtensionField ::= SEQUENCE {
   type                        [0] VM-EXTENSION,
   criticality                 [1] Criticality DEFAULT FALSE,
   value                       [2] ANY DEFINED BY type DEFAULT NULL NULL }
```

An Extension field can be marked critical (Criticality set to TRUE) or non-critical (Criticality set to FALSE) for acceptance of the message. An extension marked as non-critical may be ignored or discarded, while an extension marked as critical must be known and performed for acceptance of a VM.

```
Criticality ::= BOOLEAN
```

As a notation support for future definitions of extensions, a MACRO is defined.

```
VM-EXTENSION MACRO ::=
   BEGIN

   TYPE NOTATION          ::= DataType Critical | empty
   VALUE NOTATION         ::= value (VALUE OBJECT IDENTIFIER)

   DataType               ::= type (X) Default
   Default                ::= "DEFAULT" value (X) | empty
   Critical               ::= "CRITICAL" | empty

   END -- of extension
```

8 Voice message

A Voice Message (VM) is a member of the primary class of information objects conveyed between users in Voice Messaging.

Note – The term ***message*** when used throughout the rest of this Recommendation is a synonym for Voice Message whose spoken components are encoded per Recommendation G.721 (1988) 32K ADPCM.

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

. A Voice Message consists of the following components:

a) **Heading:** A set of Heading Fields (or Fields), each an information item that gives a characteristic of the Voice Message.

b) **Body:** A sequence of one or more body parts.

```
Body ::= SEQUENCE {
   primary-body-part          PrimaryBodyPart,
   additional-body-parts      AdditionalBodyParts OPTIONAL }

PrimaryBodyPart ::= CHOICE {
   vm-body-part               [0] VBodyPart,
   forwarded-VM               [1] VMBodyPart }

AdditionalBodyParts ::= SEQUENCE OF VM-ExternallyDefinedBodyPart
```

Note – VM-Externally Defined Body Part is defined in 8.2.3.

The Body has one Primary Body Part that contains a voice information object. This body part is either the VMGS-user's recorded message itself or a forwarded VM. An example of the encoding of a Voice information object is a 32K ADPCM Encoded Message defined by Recommendation G.721 (1988).

Note – The scope of a voice information object type is rather large and includes, for example, Privately Defined types.

The following rules comply with the requirements stated in clause 3 of Recommendation F.440:

c) When a VM is first created, the Primary Body Part shall contain one Voice Body Part.

d) When a VM is forwarded, its structure shall comply with the rules given in 17.3.3.

The Primary Body Part has one of two basic forms:

– the Primary Body Part contains a voice information object;

– the Primary Body Part contains a forwarded Voice Message.

Additionally, other body parts may be present in a message related to the Primary Body Part but of a different type. Examples of related body parts might be non-voice encoded information, such as drawings/graphics to be used in conjunction with the VM. A VM, generated as a result of VM Forwarding, may contain an additional voice encoded body part for each instance of forwarding.

The structure of a Voice Message is depicted in Figure 1/X.440.

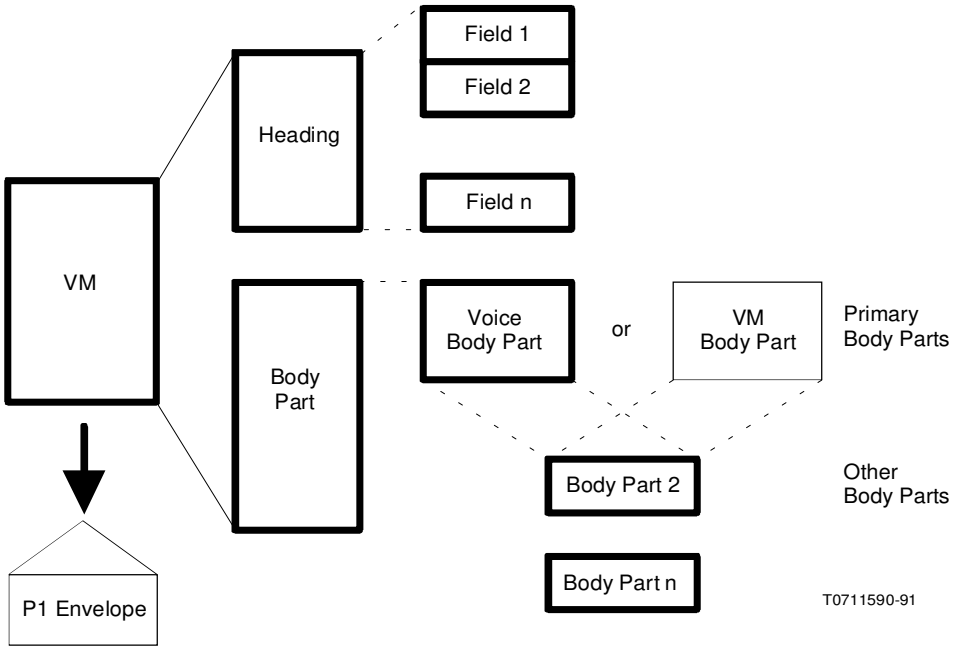

FIGURE 1/X.440
Voice message structure

8.1 *Heading fields*

The fields that may appear in the Heading of a VM are defined and described below.

```
Heading ::= SEQUENCE {
   this-VM                        ThisVMField,
   originator                     [0]  VMOriginatorField OPTIONAL,
   recipients                     [1]  RecipientsField OPTIONAL,
   obsoleted-vm                   [2]  ObsoletedVMField OPTIONAL,
   vm-subject                     [3]  VMSpokenSubjectField OPTIONAL,
   expiry-time                    [4]  ExpiryTimeField OPTIONAL,
   importance                     [5]  ImportanceField DEFAULT {normal}, -- from IPM
   sensitivity                    [6]  SensitivityField OPTIONAL, -- from IPM
   message-forwarded              [7]  MessageForwarded DEFAULT FALSE,
   vm-forwarding-permitted        [8] MessageForwardingPermitted DEFAULT TRUE,
   language                       [9] LanguageField OPTIONAL,
   voice-encoding-type            [10] VoiceEncodingType DEFAULT {id-vmg-g721-32k-adpcm},
   vm-creation-time               [11] VMCreationTime OPTIONAL,
   vn-receiver-field              [12] VNReceiverField OPTIONAL,
   vmgs-user-security-elements    [13] VMGSUserSecurityElementsField OPTIONAL,

   heading-extensions             [14] HeadingExtensionsField OPTIONAL }
```

8.1.1 *This VM*

The field, This VM, identifies the VM. It comprises a VM Identifier which provides a globally and forever unique identification for the VM.

```
ThisVMField ::= VMIdentifier
```

Note – VM Identifier is defined in 7.1.

8.1.2 *Originator*

Identifies the VM's originator. It may comprise an OR Descriptor and optionally a voice encoded version of that name. If the Originator field is not present in the VM Heading, then the P1 envelope shall be used to determine the originator of the VM (see 8.2.1.1.1 of CCITT Rec. X.411 | ISO/IEC 10021-4).

```
VMOriginatorField ::= SET {
   COMPONENTS OF ORDescriptor, -- from IPM
   originator-spoken-name        [3] SpokenName OPTIONAL }
```

Note – OR Descriptor is defined in 7.2 and Spoken Name is defined in 7.3.

8.1.3 *Recipients*

The Recipients field identifies the user(s) and (DL)(s) who are the preferred recipient(s) of the VM. It comprises a set of Recipients subfields, one for each recipient. If the Recipients field is not present in the VM Heading, then the P1 envelope shall be used to determine the recipient of the VM (see 8.2.1.1.2 of CCITT Rec. X.411 | ISO/IEC 10021-4).

Note – The fact that a message can be redirected or forwarded is reflected in the word "preferred" above.

```
RecipientsField ::= SET OF RecipientsSubField SIZE (1..ub-recipients)
```

The Recipients Sub Field is an information item that identifies a recipient of a VM and any special requests made of that specific recipient.

The Recipients Sub Field has the following components:

```
RecipientsSubField ::= SEQUENCE {
   recipient                        [0] RecipientField,
   vn-requests-field                [1] VNotificationRequestsField,
   attendant-assisted-delivery-request  [2] AttendantAssistedDeliveryRequest
   DEFAULT { }, -- default NONE
   recipient-extensions             [3] RecipientExtensionsField OPTIONAL }
```

8.1.3.1 *Recipient*

The heading field Recipient identifies one or more preferred recipients for this VM. For each recipient, it comprises an OR Descriptor, and optionally, a voice encoded version of this recipient's name.

```
RecipientField ::= SEQUENCE {
   recipient-name          COMPONENTS OF ORDescriptor, -- from IPM
   spoken-name             [2] SpokenName OPTIONAL }
```

Note – OR Descriptor is defined in 7.2 and Spoken Name is defined in 7.3.

8.1.3.2 *VNotification requests field*

The field, VNotification Requests Field, is a compound field. It conveys certain requests of the preferred recipient or recipients denoted by the Recipients field.

Note – The fact that a message can be redirected or forwarded is reflected in the word "preferred" above.

This field, represented by the data type *VNotificationRequestsField*, consists of a sequence of three bit strings and one extension mechanism. Of the bit strings, the first selects the type of notification, the second selects which security function should be applied to that notification, and the third may make certain security requests for proof or non-repudiation of reception of this VM by the recipient. Voice Notification Security and Voice Reception Security shall not be requested if Voice Notifications are not requested.

The default semantic of this field is that there are no VM extensions present, and neither VM notification, nor notification security nor reception security services are requested of the recipient.

```
VNotificationRequestsField ::= SEQUENCE {
    vm-notification-requests            [0] VNotificationRequests DEFAULT { }, -- default NONE
    vm-notification-security-requests   [1] VNotificationSecurity DEFAULT { }, -- default NONE
    vm-reception-security-requests      [2] VMReceptionSecurity DEFAULT { }, --  default NONE
    vm-notification-extension-requests  [3] VMNotificationExtensions DEFAULT { } } -- default NONE
```

1) The *VM Notification Requests* bit string may assume any of the following values simultaneously. The absence of the Voice Notification Requests bit string implies that no Voice Notification Requests are made.

   a) **receipt-notice:** A notification of acceptance of the message is requested in the circumstances prescribed in clause 9.

   b) **service-notice:** A service notification which includes the subject VM is requested in the circumstances prescribed in clause 9.

   c) **non-receipt-notice:** A non-receipt notification is requested in the circumstances prescribed in clause 9.

```
VNotificationRequests ::= BIT STRING {
    receipt-notice          (0),
    service-notice          (1),
    non-receipt-notice      (2) } (SIZE (0..ub-bit-options))
```

2) The *Voice Notification Security* bit string may assume any of the following values simultaneously. The absence of the Voice Notification Security bit string implies that no Voice Notification Security requests are made.

   d) **proof:** When submitting the VN to the MTS, content-integrity-check shall be requested in the Message-submission-argument as defined in 8.2.1.1.1.28 in CCITT Rec. X.411 | ISO/IEC 10021-4.

   e) **non-repudiation:** When submitting the VN to the MTS, content-integrity-check shall be requested in the Message-submission-argument as defined in 8.2.1.1.1.28 in CCITT Rec. X.411 | ISO/IEC 10021-4 with a non-repudiable certificate.

```
VNotificationSecurity ::= BIT STRING {
    proof (0),
    non-repudiation (1) } (SIZE (0..ub-bit-options))
```

3) The *Voice Reception Security* bit string may assume any of the following values simultaneously. The absence of the Voice Reception Security field implies that no Voice Reception Security requests are made.

   f) **proof:** When submitting the VN to the MTS, content-integrity-check (possibly in the message token), or the message-origin-authentication-check (depending on the security policy in force) shall be requested. A notification shall contain the security elements and shall be signed on submission to the MTS, using content-integrity-check (possibly in the message token) or message-origin authentication-check (depending on the security policy in force) in the Messagesubmission-argument as defined in 8.2.1.1.1.26, 28 and 29 of CCITT Rec. X.411 | ISO/IEC10021-4.

g) **non-repudiation:** When submitting the VN to the MTS, a non-repudiable content-integrity-check (possibly in the message token) or a message-origin-authentication-check (depending on the security policy in force) shall be requested. A notification shall contain the security elements and shall be signed on submission to the MTS, using non-repudiable content-integrity-check (possibly in the message token) or message-origin-authentication-check (depending on the security policy in force) in the Message-submission-argument as defined in 8.2.1.1.1.26, 28 and 29 of CCITT Rec. X.411 | ISO/IEC 10021-4.

```
VMReceptionSecurity ::= BIT STRING {
   proof (0),
   non-repudiation (1) } (SIZE (0..ub-bit-options))
```

Note – Some security services are available only if the MTA supports secure messaging.

4) The *VM Notification Extensions* field contains notification extensions to the VNotificationRequestsField subfield. There are no extensions defined in this Recommendation.

```
VMNotificationExtensions ::= SET OF VMNotificationExtensionsSubField

VMNotificationExtensionsSubField ::= ExtensionField
```

8.1.3.3 *Attendant-assisted delivery request*

The Attendant-Assisted Delivery Request field allows for the originator of the subject VM to request call assistance services of a Telephone Service attendant. The absence of this field means that Attendant-Assisted Delivery is not requested by the originator.

```
AttendantAssistedDeliveryRequest ::= INTEGER {
   person-to-person (0),
   anyone (1) }
```

The Attendant-Assisted Delivery Request field may assume the values:

a) **person-to-person:** Allows the originator to request person-to-person delivery to the recipient. The Telephone Service attendant will use the RecipientSpokenName component of the RecipientField to ask for the intended person.

b) **anyone:** Allows the originator to request an attendant to assist in the delivery of the message to the anyone at the named recipient's OR-address.

8.1.3.4 *Recipient extensions*

The Recipient Extensions contains extensions to the Recipients subfield.

```
RecipientExtensionsField ::= SET OF RecipientExtensionsSubField

RecipientExtensionsSubField ::= ExtensionField
```

There are no extensions defined in this Recommendation. Support requirements for any future extensions shall be in accordance with the corresponding element of service to which the heading extension applies.

8.1.4 *Obsoleted VM*

The Obsoleted VM field identifies one or more VMs that the present VM obsoletes. It is a sequence of subfields, each a VM Identifier.

```
ObsoletedVMField    ::= SEQUENCE OF ObsoletedVMSubfield

ObsoletedVMSubfield ::= VMIdentifier
```

8.1.5 *VM (Spoken) subject*

The VM Subject field indicates the message subject of the VM Primary Body Part. It is encoded per the algorithm identified in the Voice-Encoding-Type field. The field contains a brief audio description of the subject VM. Its size is expressed in seconds.

Note – A duration of twenty seconds is the upper limit set in Annex G.

VMSpokenSubjectField ::= ANY
-- the encoding is defined by the Voice-Encoding-Type field and not to exceed
-- 20 seconds in duration

Note – The algorithm used to encode this value is indicated by the *Voice Encoding Type* field, see 8.1.12.

8.1.6 *Expiry time*

This field indicates when the originator considers this VM to lose its validity. It comprises a date and time (UTC).

Note – Expiry Time field is defined in 7.2.11 of CCITT Rec. X.420 | ISO/IEC 10021-7.

8.1.7 *Importance*

This field conveys the originator's perceived importance for this message. It may convey one of the levels of importance: low, normal or high. The absence of this element of protocol implies that the originator considers this message to be of **normal** importance.

Note – *Importance* field is defined in 7.2.8 of CCITT Rec. X.420 | ISO/IEC 10021-7.

8.1.8 *Sensitivity*

The Sensitivity field conveys the originator's perceived sensitivity that recipients are to consider in effect for this message. It conveys one of 3 levels of sensitivity: personal, private, company-confidential. The absence of this element of protocol implies that this message is not considered to be sensitive.

Note – *Sensitivity* field is defined in 7.2.15 of CCITT Rec. X.420 | ISO/IEC 10021-7.

8.1.9 *Message forwarded*

The Message Forwarded field is used to indicate whether the message was forwarded. Absence of this field shall be interpreted as the value FALSE, i.e. not forwarded.

MessageForwarded ::= BOOLEAN *-- Default False*

If this field has the value TRUE, it indicates to a receiving UA that the security elements in the inner envelope of the Primary Body Part need to be examined.

Note – Rules regarding the use of this field are contained in 17.3.3.1, 17.3.3.2 and 17.3.3.3.

8.1.10 *VMessage forwarding permitted*

The VMessage Forwarding Permitted field, a Boolean, indicates that the forwarding of this message is permitted by the originator if this field is set to TRUE. Absence of the field shall be interpreted as the value TRUE.

A recipient of a message with the VM Forwarding Permitted field set to FALSE shall originate VNs as requested, and shall not forward this message.

MessageForwardingPermitted ::= BOOLEAN *-- Default TRUE, forwarding is permitted*

8.1.11 *Language*

The Language field indicates the internationally recognized language and optionally the local usage that was used by the originator when creating the message. The absence of this field shall be interpreted as unspecified by the originator.

LanguageField ::= SEQUENCE OF Language *-- from IPM*

The codes used in the language-code and national-usage fields are defined in ISO 639-2 and ISO 3166, respectively.

Note 1 – The syntax of the Language field is imported from Annex H of CCITT Rec. X.420 | ISO/IEC 10021-7. It is defined to be a 5 character field where the first 2 characters indicates the language-code and the last 2 characters indicates the national-usage. The middle character position is a space character as a separator. Usage of the national-usage section of the field is optional.

Note 2 – If the language of the spoken message is English as spoken by a Canadian, then the language-code value would be "en", and the national-usage value would be "CA".

Note 3 – The type LanguageField is encoded as a Sequence to allow for future growth of the protocol to handle VMs of multiple voice encoded body parts.

Note 4 – For this information to be useful for human usage, the receiving system may need to present a spoken version of this information to the user. For example, the value "CA" would be heard in the local language as "canadian" or "Canada".

8.1.12 *Voice encoding type*

Indicates the voice standard used to encode the contents of the Primary Body Part and any other voice encoded data type contained in the subject VM. It is represented by a single object identifier.

Note – This object identifier implements the element of service *VM Encoding Algorithm Identifier (E.10)* defined in Recommendation F.440 (1992).

VoiceEncodingType ::= OBJECT IDENTIFIER *-- default CCITT's G.721 32K ADPCM*

The following standard values have object identifiers defined in this Recommendation:

– 32K-ADPCM: CCITT Rec. G.721 (1988);

– PRIVATE: UNDEFINED OCTETS;

– UNDEFINED: UNDEFINED OCTETS.

The absence of this field shall be interpreted as having the default value set to 32K-ADPCM.

The actual mapping of 32K-ADPCM transmission encoding onto ASN.1 octets is for further study.

Note – The voice encoding referred to by the object identifier is that in which both the Voice Body Part, and those Heading fields whose values are spoken are encoded, notwithstanding the fact that these types are defined as an ASN.1 ANY.

The value of the Voice Body Part Type field shall be used in the Encoded Information Types in the MTS abstract operations (see also 17.1.1). This enables a UA to signal to the MTS which Voice standard the VM's Primary Body Part complies with. The MTS shall make use of this information, if the recipient UA has registered delivery restrictions on Encoded Information Types, to decide if it can deliver the VM or to provide conversion services.

8.1.13 *VM creation time*

Indicates the date and time of creation of the spoken message. It comprises a UTC time.

VMCreationTime ::= UTCTime

8.1.14 *VN receiver*

Identifies the recipient to whom VNs are to be sent. This is created by the originator of the VM when the Recipient of a requested notification is different from the Originator of the message. It consists of a sequence of OR Name, VM Identifier and First Recipient.

This field shall not be present if Voice Notification Requests are not made.

This field shall be present in a forwarded message when the forwarding Voice messaging system user agent (VMGS-UA) or Voice messaging system message store (VMGS-MS) forwards the message. This field may be present when the forwarding VMGS-UA accepts the message. Rules related to the construction of this field are given in 17.3.3.

Note – For brevity, the term user agent (UA) is used throughout the rest of this Recommendation with the meaning of VMGS-UA, and the term message store (MS) is used throughout the rest of this Recommendation with the meaning of VMGS-MS.

```
VNReceiverField ::= SEQUENCE {
    vn-receiver-name            [0] ORName,
    original-vm-identifier      [1] VMIdentifier OPTIONAL,
    first-recipient             [2] FirstRecipientField OPTIONAL }
```

The *first-recipient* field shall not be present if more than one recipient contains Voice Notification Requests.

The *original-vm-identifier* and the *first-recipient* fields shall not be present when the Primary Body Part is a Voice Body Part (that is, when the original originator first creates the VM).

Note – The Original VM Identifier and First Recipient fields are included in order to allow the recipient to construct the VN for a forwarded VM. See 9.1 (more specifically 9.1.3) and 17.3.1.1 for rules related to the construction of a VN; see 17.3.3.4 for rules related to the First Recipient field when constructing a forwarded VM. OR Name is defined in 8.5.5 of CCITT Rec. X.411 | ISO/IEC 10021-4. First Recipient Field is defined in 9.1.3.

8.1.15 *VMGS-user security elements*

The *VMGS-user security elements* field allows a user to exchange security elements having an end-to-end significance.

```
VMGSUserSecurityElementsField ::= SEQUENCE {
    vmgs-user-security-element          [0] VMGSUserSecurityElement OPTIONAL,
    vm-encrypted-primary-bodypart       [1] BOOLEAN OPTIONAL,
    vmgs-user-security-extensions       [2] VMGSUserSecurityExtensions OPTIONAL }

VMGSUserSecurityElement                 ::= BIT STRING (SIZE (0..ub-vmgs-user-security-elements))

VMGSUserSecurityExtensions              ::= SEQUENCE OF VMGSUserSecurityExtension

VMGSUserSecurityExtension               ::= ExtensionField
```

8.1.16 *Heading extensions*

The Heading extensions allows for future extensions to the Heading.

```
HeadingExtensionsField ::= SET OF HeadingExtensionsSubField

HeadingExtensionsSubField ::= ExtensionField
```

There is no Extensions to the Heading defined in this Recommendation. Support requirements for any future extensions shall be in accordance with the corresponding element of service that the heading extension applies.

8.2 *Body part types*

The types of body parts that may appear in the Body of a VM are defined and described below.

8.2.1 *Voice body part*

A Voice Body Part carries a single voice encoded message (object).

```
VBodyPart ::= SEQUENCE {
   voice-parameters          [0] VoiceParameters OPTIONAL,
   voice-data                [1] VoiceData }

VoiceParameters ::= SEQUENCE {
   voice-message-duration    [0] VMDuration OPTIONAL,
   voice-encoding-type       [1] VoiceEncodingType OPTIONAL, -- for use in IPM
   other-parameters          [2] VMSupplementaryInformation OPTIONAL,
   extension-parameters      [3] VBPParameterExtensions OPTIONAL  }

VMDuration ::= INTEGER, -- size indicated in seconds

VBPParameterExtensionsField ::= SET OF VBPExtensionsSubField

VBPExtensionsSubField ::= ExtensionField

VoiceData ::= ANY -- defined by VoiceEncodingType
```

The *VoiceParameters* component of the **VBodyPart** shall be used as follows:

a) **voice-message-duration:** The duration of the voice encoded data component in seconds.

b) **voice-encoding-type:** An object identifier which identifies the voice encoding applied to the data portion of this body part. It is intended for use within IPM where voice encoded body parts are not restricted to be of the same encoding within the body of the message. If used in VM, it shall have the same value as the voice-encoding-type field carried in the Heading.

c) **other-parameters:** Encoded as VMSupplementaryInformation. It conveys additional information necessary for decoding and processing the voice encoded data component of this body part. Its usage is subject to bilateral or local agreements.

d) **extension-parameters:** Conveys parameter extensions needed for processing the voice encoded data component of this body part.

Note – There are no *extension-parameters* defined in this Recommendation.

The reference definition of the encoded voice object used is that used by G.721 32K ADPCM. The object identifier used in the default is defined in Annex A. By employing the use of object identifiers, this Recommendation does not draw a distinction between voice, or audio or music quality encoded objects.

The *VoiceData* component of the **VBodyPart** carries the digital encoding of the voice message. The encoding of the contents of this component is identified in the Voice Encoding Type field, see 9.1.3.

8.2.2 *VM body part*

A VM Body Part contains a forwarded VM, and optionally, its delivery envelope. The delivery envelope shall be present when MHS security services are employed. When a VM is forwarded, its structure shall comply with the rules given in 17.3.3.2.

```
VMBodyPart ::= SEQUENCE {
   parameters                    [0] MessageParameters OPTIONAL,
   data                          [1] MessageData }

MessageParameters ::= SET {
   delivery-time                 [0] MessageDeliveryTime OPTIONAL,
   delivery-envelope             [1] OtherMessageDeliveryFields OPTIONAL,
   -- delivery-time and delivery-envelope shall both be present or both shall be absent.
   other-parameters              [2] VMSupplementaryInformation OPTIONAL }

MessageData ::= VM

VMSupplementaryInformation ::= IA5String (SIZE (1..ub-supplementary-info-length))
```

Note – Primary Body Part is defined in clause 8. Message Delivery Time and Other Message Delivery Fields are defined in 8.3.1.1 of CCITT Rec. X.411 | ISO/IEC 10021-4.

The VM Supplementary Information field is intended to convey additional pre-registered information when performing VM Forwarding.

The structure of a VM Body Part which results from VM Forwarding is depicted in Figure 2/X.440.

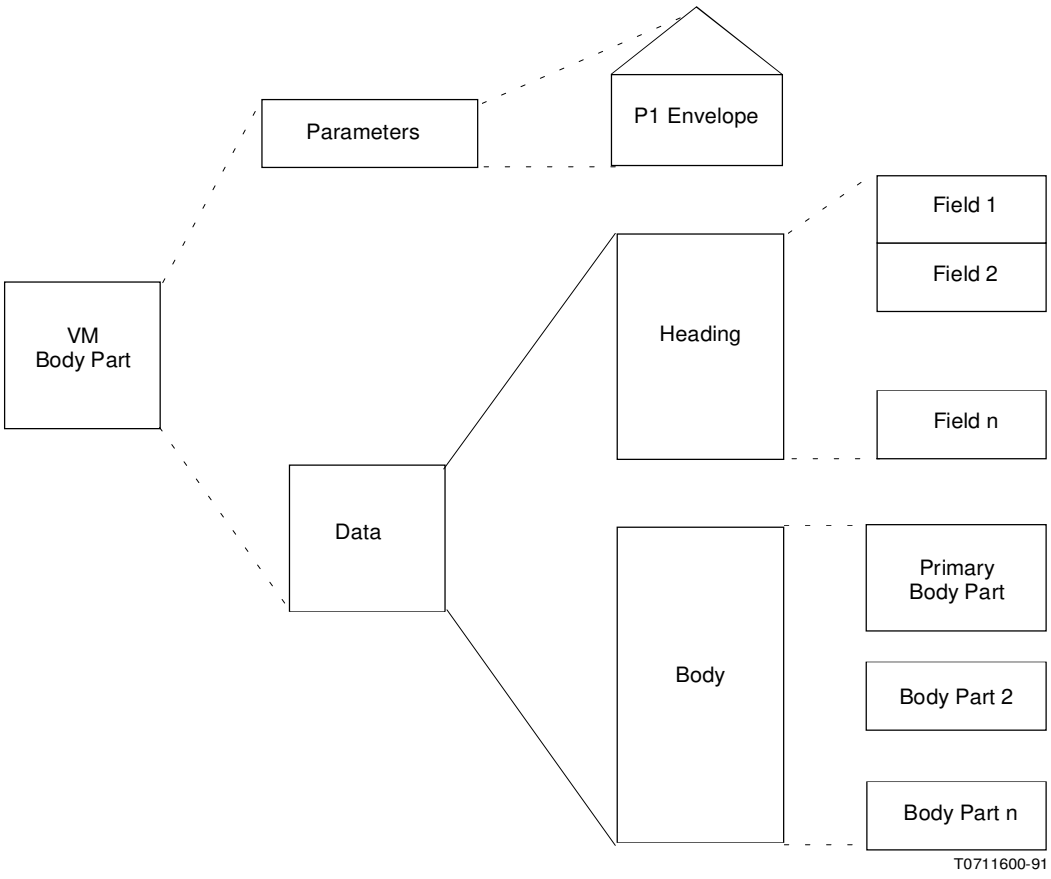

FIGURE 2/X.440

VM body (Forwarded) part structure

8.2.3 *VM externally defined body parts*

Additional body parts, that relate to the Primary Body Part, may be carried together with a Voice Body Part. For submission of the original VM, the body parts shall not be or include voice encoded information objects.

Additional body parts are externally defined and represent information objects whose semantics and abstract syntax are denoted by an object identifier which the body part carries. They have Parameters and Data components.

VM-ExternallyDefinedBodyPart ::= ExternallyDefinedBodyPart *-- from IPMS*

Note –The IPMS Externally Defined Body Part type is defined in 7.3.12 of CCITT Rec. X.420 | ISO/IEC 10021-7.

9 Voice notifications

A Voice Notification (VN) is a member of a secondary class of information object conveyed between users in voice messaging.

Note 1 – The term notification is used throughout the rest of this Recommendation as a synonym for voice notification.

Note 2 – Voice messaging information objects are defined in clause 8.

VN ::= CHOICE {

receipt-notification	**[0] ReceiptNotificationFields,** *-- referred to as RN*
service-notification	**[1] ServiceNotificationFields,** *-- referred to as SN*
non-receipt-notification	**[2] NonReceiptNotificationFields** *-- referred to as NRN* **}**

a) receipt notification (RN): A VN that reports its originator's acceptance of a VM.

b) service notification (SN): A VN that reports its originator's software did not support all of the requested EOSs although the message was received by the recipient.

c) non-receipt notification (NRN): A VN that reports its originator's refusal to accept a VM and that the subject VM has been forwarded without accepting the message or discarded.

Note – RN, SN, and NRN are used in the User Agent section of this Recommendation. However, according to a user's local policy, a UA may delegate to its MS the responsibility for responding to requested VNs.

The VM to which a VN refers is called the subject VM (see 17.3.3).

If VM Forwarding has not occurred, then the recipient of the VN is the Originator of the subject VM, or, if present, the OR Name indicated in the VN Receiver field. However, if VM Forwarding has occurred, then the recipient of the VN may be either the originator of the VM or the originator of the forwarded VM.

There shall be at most one recipient specified for a VN. There shall be at most one RN and SN originated for each subject VM by each recipient from whom these notifications are requested. A NRN may be originated subsequent to a SN, in accordance with c) of 17.3.3.1. One NRN is originated, if and only if requested, by each recipient that forwards a VM. In accordance with the provision of 17.3.3, the subject VM originator shall receive at most one RN or SN for each recipient from whom these notifications were requested. This is true regardless of how many times the VM is forwarded. However, the originator may receive multiple NRNs as a consequence of VM Forwarding.

A VN consists of fields of information specific to Receipt, Service or Non-Receipt Notification. Additionally, each VN type contains the Common fields which are described below. Figure 3/X.440, illustrates the structure of a voice notification.

The specific notification type conveyed by the VNP is also identified in the P1 PerMessageIndicators field as specified in Figure 2 of CCITT Rec. X.411 | ISO/IEC 10021-4.

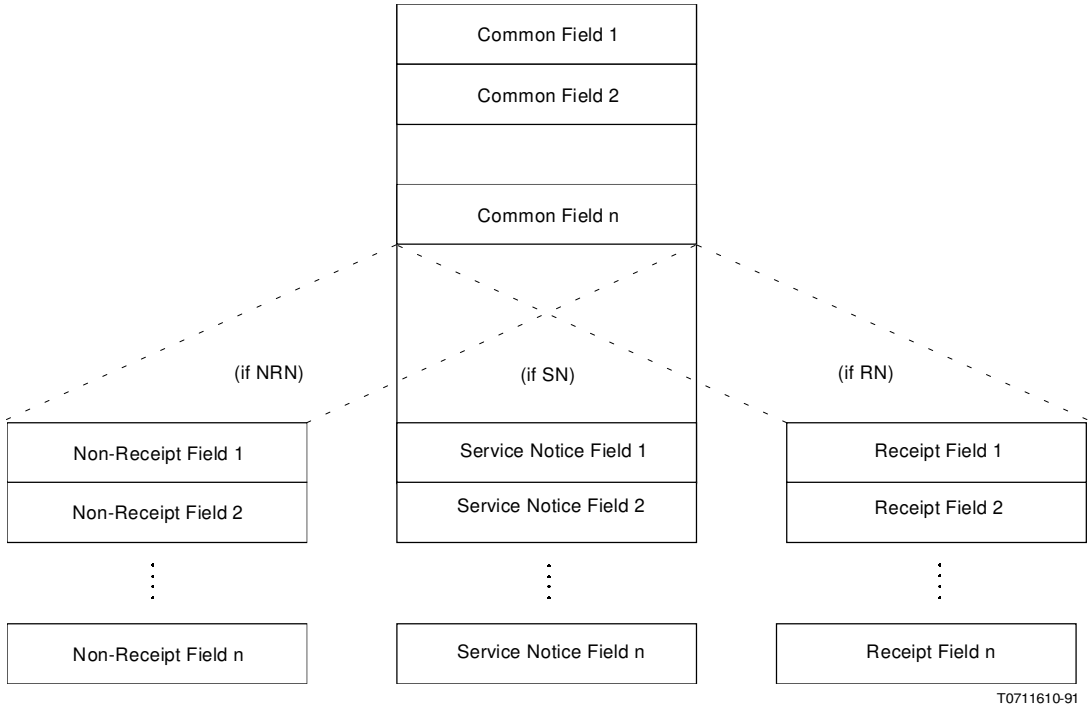

FIGURE 3/X.440

Voice notification structure

9.1 *Common fields*

The Common fields used in the Voice Notifications are defined and described below. The data type which follows is carried in a RN as *rn-common-fields*, in a SN as *sn-common-fields* and in a NRN as *nrn-common-fields*.

```
CommonFields ::= SET {
    subject-vm                  SubjectVMField,
    vn-originator               [1] VNOriginatorField,
    first-recipient             [2] FirstRecipientField OPTIONAL,
    notice-creation-time        [3] NoticeCreationTimeField,
    vn-voice-encoding-type      [4] VNVoiceEncodingType,
    conversion-indication       [5] ConversionElTsField OPTIONAL,
    notification-security-elements  [6] SecurityElementsField OPTIONAL,
    orig-vm-spoken-subject      [7] VMSpokenSubjectField OPTIONAL,
    subject-vm-other-recipients [8] RecipientField OPTIONAL,
    vn-supplementary-info       [9] VNSupplementaryInfo OPTIONAL,
    notifications-extensions    [10] NotificationExtensionsField OPTIONAL }
```

Note – The Common fields appear in the RN, SN and NRN fields defined below.

9.1.1 *Subject VM*

The Subject VM field contains the VM Identifier either passed in the VN Receiver field, if the subject VM has been forwarded, or the This VM field, if not.

```
SubjectVMField ::= VMIdentifier
```

Note – VM Identifier is defined in 7.1. Subject VM is defined in 5.1, b).

9.1.2 *VN originator*

The VN Originator contains the OR Name component of the Originator field for the UA constructing the notification. Additionally, it may contain a voice encoded representation of this originator and the identity of the VM application entity that initiated the notice.

```
VNOriginatorField ::= SEQUENCE {
   originator-name              [0] VMOriginatorField,
   vn-initiator                 [1] VNInitiatorField OPTIONAL }
```

Note – OR Name is defined in 8.5.5 of CCITT Rec. X.411 | ISO/IEC 10021-4. The VMOriginatorField parameter is defined in 8.1.2. The audio encoded component of the VMOriginator field, the SpokenName, is defined in 7.3.

The VN Initiator field can take one of the following values:

a) **internal-UA** means that the UA generated the VN either for local reasons or because the generation had been delegated to it by the user.

b) **internal-MS** means that the MS generated the VN either for local reasons or because the generation had been delegated to it by the user.

c) **external-UA** means that the generation of the VN was requested by the user via the abstract operation Originate VN (see 12.1.2).

d) **internal-tsau** means that the TSAU generated the VN either for local reasons or because the generation had been delegated to it by the user.

```
VNInitiatorField ::= ENUMERATED {
   internal-ua (0),
   external-ua (1),
   internal-ms (2),
   internal-tsau (3) }
```

Origination of a Receipt Notification implies that the message has been accepted, regardless of the value of this field.

The value of this field shall be consistent with the choice (UA/MS, user) of the Reason Code field for SNs and NRNs.

9.1.3 *First recipient*

The First Recipient field contains the OR Name of the first recipient in a forwarding chain. This field, together with other fields, is used by the recipient of the notification to correlate the notification and the original message.

```
FirstRecipientField ::= ORName
```

Note – OR Name is defined in 8.5.5 of CCITT Rec. X.411 | ISO/IEC 10021-4. If the receiver of this VN desires the audio version of this recipient it is extracted from a locally held copy of the original subject VM.

If the originator of the VN is not the preferred recipient specified by the VBodyPart originator, then the First Recipient field shall be present in the VN (see 17.3 and more specifically 17.3.1.1).

9.1.4 *Notice creation time*

Notice Creation Time contains the date and time, in UTC format, at which the notification for the subject VM was generated.

```
NoticeCreationTimeField ::= UTCTime
```

9.1.5 *VN voice encoding type*

The VN Voice Encoding Type field conveys the identity of the voice encoding algorithm used to encode spoken data element fields contained in this notification as well as any additional information needed to decode any spoken data elements by the recipient of this VN. If conversion was performed on the subject VM that this VN references, then its value can be different from the *Voice Encoding Type* field found in the subject VM.

```
VNVoiceEncodingType ::= VoiceEncodingType
```

Note – Its syntax is specified in 8.1.12.

9.1.6 *Conversion indication*

The Conversion Indication common field identifies the EITs contained in the subject VM upon delivery to the VN's originator. It's value is an EIT descriptor. This field is present only if the subject VM was subjected to MTA conversion services for delivery to the VN originator.

The encoding of this field is found in 8.1.4 of CCITT Rec. X.420 (1988) | ISO/IEC 10021-7, and more specifically in 19.4 below.

9.1.7 *Notification security elements*

The Notification Security Elements field is used to provide "proof/non repudiation of content received", and "Voice security" services.

```
SecurityElementsField ::= SEQUENCE {
   original-content                [0] Content OPTIONAL, -- from P1
   original-content-integrity-check [1] ContentIntegrityCheck OPTIONAL, -- from P1
   vmgs-user-security-elements     [2] VMGSUserSecurityElementsField OPTIONAL,
   security-extensions             [3] SecurityExtensionsField OPTIONAL }

SecurityExtensionsField ::= SET OF SecurityExtensionsSubField

SecurityExtensionsSubField ::= ExtensionField
```

Note – The *VMGS-user Security Elements* field is defined in 8.1.15. Content and Content Integrity Check are defined in, respectively, 8.2.1.1.1.37 and 8.2.1.1.1.28 of CCITT Rec. X.411 | ISO/IEC 10021-4. Security services are available only if the MTA supports secure messaging.

Subclause 17.1.2 below specifies how these fields are filled in.

9.1.8 *Original VM spoken subject*

The Original VM Spoken Subject field conveys the value of the *VM Subject* field received in the subject VM. The subject VM is the cause of this VN being generated. If the *VM Subject* field is not provided in the subject VM, then this notification field shall not have a value.

Note – VM Subject field which contains voice encoded information is defined in 8.1.5.

9.1.9 *Subject VM other recipients*

The *Subject VM Other Recipients* conveys the identities of the other intended recipients of the subject VM that prompted this notification. If the subject VM was originally addressed to only one recipient and VM Forwarding was not performed, then this field shall be omitted. If the original subject VM had undergone VM Forwarding, then the union of all recipients shall be present in the value of this field.

```
SubjectVMOtherRecipients ::= SEQUENCE  OF RecipientField
```

Note – Recipient Field is defined in 8.1.3.1.

9.1.10 *VN supplementary information*

The VN Supplementary Information field carried in a notification, if present, may be encoded in either IA5 or voice. When present, the duration of (audio encoded) *Spoken Supplementary Information* is not to exceed the upper bounds set in Annex G.

When VN Supplementary Information contains voice encoded information voice, the encoding is that of the original subject VM. If subject VM had been subject to conversion prior to delivery then the *SpokenSupplementaryInfo* field shall be absent.

```
VNSupplementaryInfo ::= SEQUENCE {
   supplementary-info          [0] VMSupplementaryInformation OPTIONAL,
   v-supplementary-info        [1] SpokenSupplementaryInfo OPTIONAL }

SpokenSupplementaryInfo ::= ANY (SIZE (0..ub-vmg-spoken-supplemental-info))
-- The encoding of the value carried in v-supplementary-info is
-- identified by the value of the Voice Encoding Types field.
```

Note 1 – VM Supplementary Information field is defined in 8.2.2.

Note 2 – Annex G has set the upper bounds for *SpokenSupplementaryInfo* field to be twenty seconds.

9.1.11 *Notifications extensions*

The Notification Extensions allows for future extensions to the VN.

```
NotificationExtensionsField ::= SET OF NotificationExtensionsSubField

NotificationExtensionsSubField ::= ExtensionField
```

There are no extensions to the VN defined in this Recommendation. Support requirements for any future extensions shall be in accordance with the corresponding element of service to which the heading extension applies.

Extensions shall not be critical in VNs.

9.2 *Receipt notifications*

A Receipt Notification (RN) indicates that the subject VM has been accepted by a recipient. It is generated, if, and only if, the originator has requested one and the subject VM has been accepted by the user.

The exact procedures which constitute acceptance of message are a local matter; for example, the UA may construct the RN as soon as it passes the message to the user or it may wait for an external stimulus from the user that the message has been accepted and therefore can generate the RN.

A RN may be generated as a result of VM Forwarding. RNs generated as a result of VM Forwarding is described in 17.3.3 and in 18.4.2 when a VMGS-MS is involved.

A RN is composed of Receipt Notification Fields which consist of VN common fields, and may consist of supplementary information, receipt extension fields and the subject VM.

The structure of an RN is defined as follows:

```
ReceiptNotificationFields ::= SEQUENCE {
   rn-common-fields            [0] CommonFields,
   rn-extensions               [1] NotificationExtensionsField OPTIONAL }
```

9.2.1 *RN common fields*

The RN Common Fields carried in this notification are those fields relevant to Receipt notices as specified in 17.3.1.1. The structure of this data type is found in 9.1.

9.2.2 *Receipt notification extensions*

The Receipt Notification Extensions allows for future extensions to the RN.

There are no extensions to the RN defined in this Recommendation. Support requirements for any future extensions shall be in accordance with the corresponding element of service to which the heading extension applies.

Extensions shall not be critical in RNs.

Note – The structure of NotificationExtensionsField is defined in 9.1.11.

9.3 *Service notifications*

A Service Notification (SN) is sent by a UA, if and only if the originator has requested one. It conveys the semantic that the VMGS-UA or VMGS-MS has been presented the subject VM although the subject VM contained *Optional Additional* elements of service that were not available to the recipient.

Note – This notice is not supported in IPM.

Service Notification fields are defined and described below.

```
ServiceNotificationFields ::= SEQUENCE {
   sn-common-fields              [0] CommonFields,
   sn-reason-code-field          [1] SNReasonCode,
   sn-extensions                 [2] NotificationExtensionsField OPTIONAL }
```

9.3.1 *SN common fields*

The SN Common Fields carried in this notification are those fields relevant to a Service Notification as specified in 17.3.2.1. The structure of this data type is found in 9.1.

9.3.2 *Service notification reason*

Generally, the Service Notification Reason indicates that the subject VM was passed to the VMGS-UA or VMGS-MS generating this notice and that one or more requested elements of service are not available to that user. If the SN Reason Field has the value "*unspecified"*, additional information may be carried in any combination of a Diagnostic field or the VN Supplementary Info field. Depending on the security policy in force, the security error diagnostic code may or may not be present. Any SN Reason Code value set to **TRUE** indicates that this requested optional element of service is not available to the VMGS-user. The receiving VMGS-UA or VMGS-MS may be authorized to initiate a SN on behalf of the receiving user.

```
SNReasonCode ::= SEQUENCE {
   sn-reason                     [0] SNReasonField,
   sn-diagnostic                 [1] SNDiagnosticField OPTIONAL }
-- Service Notification Basic Reason Codes from a VMGS-UA or VMGS-MS or VMGS-TSAU. These codes are those
-- specified in Annex B of Recommendation F.440 for the Element of Service "VM Notification Request".

SNReasonField ::= BITSTRING {
   unspecified                          (0),
   auto-forwarding-ind                  (1),
   language-ind                         (2),
   obsoleting-ind                       (3),
   attendant-assisted-delivery-request  (4),
   expiry-date-ind                      (5),
   body-part-encryption-ind             (6) } (SIZE (2..ub-sn-reasons))
```

-- Service Notification Diagnostic Codes from a VMGS-UA or VMGS-MS

SNDiagnosticField ::= INTEGER {

-- This field may be used to further specify the error signalled in sn-ua-ms-basic-code.

-- Additional information may be indicated in sn-supplementary-information.

-- general diagnostic codes

language-national-usage-problem (1),

-- used if the national usage of the language included is incompatible

vm-language-not-understood (2),

-- language not understood by this recipient

vm-unsupported-voice-encoding (3),

-- recipient does not support the VM's encoding

-- security error diagnostic codes

local-security-not-supported (4) } (1..ub-vn-reason-code)

9.3.3 *Service notification extensions*

The Service Notification Extensions allows for future extensions to the SN.

There are no extensions to the SN defined in this Recommendation. Support requirements for any future extensions shall be in accordance with the corresponding element of service to which the heading extension applies.

Extensions shall not be critical in SNs.

Note – SN Extensions, which is defined to be the field NotificationExtensionsField, is defined in 9.1.11.

9.4 *Non-receipt notifications*

A Non-Receipt Notification (NRN) is sent by a UA, if and only if the originator has requested one, when the UA determines that it cannot accept the message, or decides to forward the VM and the Voice Notification Requests contained in the VM, to another UA.

Non-Receipt Notification fields are defined and described below.

NonReceiptNotificationFields ::= SEQUENCE {

nrn-common-fields	**[0] CommonFields,**
nrn-reason-codes	**[1] VNRNReasonCodeField,**
nrn-extensions	**[2] NotificationExtensionsField OPTIONAL }**

9.4.1 *NRN common fields*

The NRN Common Fields carried in this notification are those fields relevant to a Non-Receipt Notification as specified in 17.3.3.6. The structure of NRNCommonFields is found in 9.1.

9.4.2 *Non-receipt reason codes*

A VN Non-Receipt Reason Codes field indicates why the subject VM could not be accepted by the user of the UA originating the VN. If any Basic Code field has the value "unspecified", additional information may be carried in any combination of a Diagnostic field or the VN Supplementary Information field. Depending on the security policy in force, the security error diagnostic may or may not be present.

```
VNRNReasonCodeField ::= CHOICE {
   nrn-ua-ms-reason-code          [0] VNRNUAMSReasonCodeField,
   nrn-user-reason-code           [1] VNRNUserReasonCodeField,
   nrn-tsau-reason-code           [2] VNRNTSAUReasonCodeField }

-- NRN Reason Codes from a VMGS-UA or VMGS-MS.

VNRNUAMSReasonCodeField ::= SEQUENCE {
   nrn-ua-ms-basic-codes          [0] VNRNUAMSBasicCodeField,
   nrn-ua-ms-diagnostics          [1] VNRNUAMSDiagnosticField OPTIONAL }

-- VN NRN Basic Reason Codes from a VMGS-UA or VMGS-MS.

VNRNUAMSBasicCodeField ::= INTEGER {
   unspecified (0),
   auto-forwarded (1),
   can-not-pass-to-mhs-user (2),
   delivery-timeout (3),
   message-discarded (4),
   subscription-terminated (5),
   forwarding-error (6),
   security-error (7),
   message-forwarded (8),
   voice-encoding-not-supported (9) } (0..ub-vn-reason-code)

-- NRN Diagnostic Codes from a VMGS-UA or VMGS-MS or TSAU.

VNRNUAMSDiagnosticField ::= INTEGER {
   -- This field may be used to further specify the error
   -- signaled in the field "nrn-ua-ms-basic-codes".
   -- Additional information may be
   -- indicated in the "vn-supplementary-info" field.

  protocol-violation (0) -- this is a place holder
  } (0..ub-vn-reason-code)

-- NRN Reason Codes from a VMGS-User.

VNRNUserReasonCodeField ::= SEQUENCE {
   vn-user-basic-codes            [0] VNRNUserBasicCodeField,
   vn-user-diagnostics            [1] VNRNUserDiagnosticField OPTIONAL }

-- VN NRN Basic Reason Codes from a VMGS-User.

VNRNUserBasicCodeField ::= INTEGER {
   unspecified (0),
   user-defined-reason (1) -- this is a place holder
   } (0..ub-vn-reason-code)

-- VN NRN Basic Reason Codes from a VMGS-User.

VNRNUserDiagnosticField ::= INTEGER
   --  Contains the reason passed by the user when the value
   -- of "nrn-user-basic-code" is "user-defined-reason".
   -- Additional information may be indicated in the
   -- "vn-supplementary-info" field. The values used in this field
   -- are outside the scope of this Recommendation.

-- NRN Reason Codes from a TSAU-User.

VNRNTSAUReasonCodeField ::= SEQUENCE {
   nrn-user-basic-codes           [0] VNRNTSAUBasicCodeField,
   nrn-user-diagnostics           [1] VNRNTSAUDiagnosticField OPTIONAL }
```

```
VNRNTSAUReasonBasicCodeField ::= INTEGER {
   attendant-assisted-delivery-failure (0),
   unknown-telephone-number (1),
   attendant-assisted-delivery-not-provided (2),
   delivery-timeout (3),
   security-error (4),
   message-forwarded (5),
   unspecified (6),
   inappropriate-voice-encoding (7),
   telephone-number-unreachable (8),
   recipient-refused-message (9),
   no-answer-on-every-attempt (10),
   busy-on-every-attempt (11),
   no-answer-or-busy-on-every-attempt (12),
   sensitivity-not-supported (13),
   importance-not-supported (14),
   busy-on-every-attempt (15)
   -- TSAU providers may define additional values above (1000). }

VNRNTSAUDiagnosticField ::= INTEGER {
   person-to-person-spoken-name-not-provided (0),
   person-unavailable (1),
   number-not-in-service (2),
   message-expired (3),
   importance-not-conveyed (4)
   } (0..ub-vn-reason-code)
```

9.4.3 *NRN extensions*

The NRN Extensions field allows for future extensions to the structure of an NRN.

There are no extensions to the NRN defined in this Recommendation. Support requirements for any future extensions shall be in accordance with the corresponding element of service to which the heading extension applies.

Extensions shall not be critical in NRNs.

Note – NRN Extensions, which is defined to be the field NotificationExtensionsField, is defined in 9.1.11.

10 Primary object types

The environment in which Voice Messaging takes place can be modelled as an abstract object which is hereafter referred to as the Voice Messaging Environment (VMGE).

```
vmge OBJECT ::= id-ot-vmge
```

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

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

The lesser objects are referred to as the primary objects of Voice Messaging. They include a single, central object, the Voice Messaging System (VMGS), and numerous peripheral objects called Voice Messaging System users (users). The structure of the VMGE is depicted in Figure 4/X.440.

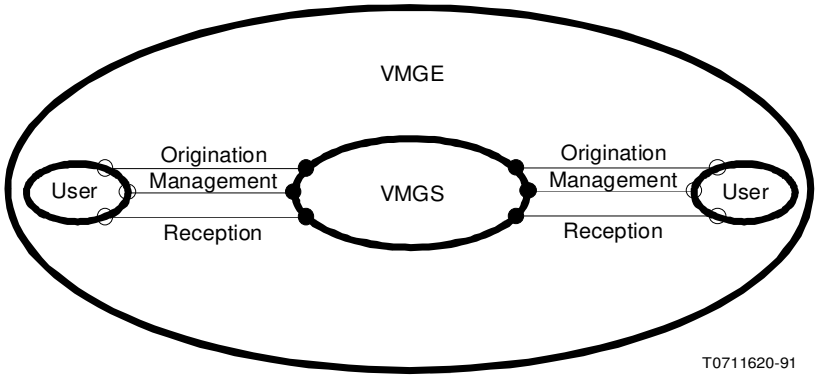

FIGURE 4/X.440

The voice messaging environment

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

10.1 *Voice messaging system user*

A Voice Messaging System user (VMGS-user) is typically a human that engages in Voice Messaging. Such a person is referred to by the term "user" in this Recommendation as identified earlier in 7.1. A user originates, receives, or both originates and receives Information Objects of the types defined in clause 6. Hence, a VMGS-user object is defined as follows:

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

The VMGE comprises any number of users.

Note – Voice messaging is typically an activity between two or more humans. The primary access device is anticipated to be a telephone hand set. This does not preclude the possibility of human interaction with the information processing systems which are performing voice messaging adjunct services, or more direct interaction of a human user with the VMGS. These information processes are referred to as Voice applications.

10.2 *Voice messaging system*

The Voice Messaging System is the object by means of which all users communicate a voice message (VM) with one another in Voice Messaging. Users may choose to use interpersonal messaging systems to exchange voice encoded messages which is described in Recommendation X.420. As the primary communication object for Voice Messaging, VMGS supplies the services of message origination, reception, and management through *ports.* The VMGS object is defined as follows:

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

The VMGE comprises exactly one VMGS.

11 Primary port types

The primary objects of Voice Messaging are joined to and interact with one another by means of ports. These ports, which the VMGS supplies, are referred to as the primary ports of Voice Messaging. They are of the types defined below.

Note – In clause 15, the VMGS is decomposed into still lesser objects, among which is the MTS. This fact is anticipated here by the inclusion of certain MTS capabilities in the VMGS Abstract Service.

11.1 *Origination port*

An Origination Port is the means by which a single user conveys to the VMGS messages containing Information Objects of the types defined in clause 6. Through such a port the user originates a Voice Message and a Voice Notification. A Voice Notification is a Receipt Notification (RN), a Non-Receipt Notification (NRN), and a Service Notification (SN). The abstract operations available at the origination port are defined as:

```
origination PORT
   CONSUMER INVOKES {
      OriginateVM,
      OriginateVN }
      ::= id-pt-origination
```

The VMGS supplies one Origination Port to each user (with the exception of indirect users served by TSAUs (see 15.3).

11.2 *Reception port*

A Reception Port is the means by which the VMGS conveys to a single user messages containing Information Objects of the types defined in clause 6. Through such a port the user receives Voice Messages and Voice Notifications. In addition, the user may receive Delivery and Non-Delivery reports through such a port. The abstract operations available at the reception port are defined to be:

```
reception PORT
   SUPPLIER INVOKES {
      ReceiveReport,
      ReceiveVM,
      ReceiveVN }
      ::= id-pt-reception
```

The VMGS supplies one Reception Port to each user.

11.3 *Management port*

A Management Port is the means by which a single user changes personally managed information that is recorded within the VMGS. The specific types of information is defined in clause 6. By means of such a port the user enables and disables auto-discard, -acknowledgement, and -SNs, and -forwarding. The abstract operations available at the management port are defined as:

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

12 Abstract operations

What follows defines the abstract service that characterizes Voice Messaging, and describes the environment in which that service is supplied and consumed. It does both using the abstract service definition conventions of CCITT Rec. X.407 | ISO/IEC 10021-3.

The VMGS Abstract Service is the set of capabilities that the VMGS provides to each user by means of one Origination Port, one Reception Port and one Management Port. These capabilities are modelled as abstract operations, which may encounter abstract errors when invoked.

The purpose of the VMGS Abstract Service definition is not to prescribe the interface between the VMG-user and the VMGS-UA, but rather to clarify the meaning and intended use of the Information Objects. A user interface need not provide commands in one-to-one correspondence to the service's abstract operations, nor indeed even divide the labour between the user and the VMGS as the service does.

The abstract operations available at the Origination Port and Reception Port are defined and described below. The abstract errors they may provoke are the subject of clause 13.

The VMGS Abstract Service involves neither abstract bind nor abstract unbind operations.

The VMGS authenticates (i.e. establishes the identity of) the typical user before offering the VMGS Abstract Service to him. By this means it can verify, e.g. that the user is a VMG subscriber. Authentication, where required, is implicit (rather than explicit) in the definition of the VMGS Abstract Service.

Note – In clause 15, the VMGS is decomposed into objects among which is the MTS. The text here reflects this fact by its inclusion of various MTS-defined information items in the VMGS Abstract Service.

12.1 *Origination abstract operations*

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

12.1.1 *Originate VM*

The Originate VM abstract operation originates a message whose content is a VM.

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

This abstract operation has the following arguments:

a) **Envelope:** A message submission envelope as defined by the MTS Abstract Service. The UA supplies all but the following envelope components, which the user provides:

   1) The desired per-message options (i.e. priority, per-message indicators, deferred delivery time, and extensions).

   2) The OR Names of the preferred recipients and the per-recipient options (i.e. originator report request, explicit conversion, and extensions) desired for each.

b) **Content:** The VM being originated. If application to application security services are required, the user shall supply the value for the Voice Application Security Elements field.

This abstract operation has the following results:

c) **Submission-identifier:** The message submission identifier the MTS assigns to the submission.

d) **Submission-time:** The date and time the message was directly submitted.

12.1.2 *Originate voice notifications*

The Originate VN abstract operation originates a message whose content is a VN. A VN may be either a RN, NRN or SN.

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

VNType ::= CHOICE {
      [0]  RN,
      [1]  NRN,
      [2]  SN }
```

A user may, if notifications are requested, invoke an Originate VN abstract operation to indicate to the UA that it should accept, refuse or forward the subject VM, or generate a SN. The exact type of VN to be generated (RN or NRN or SN) is determined from the VN structure conveyed in the Content argument.

A VN shall be originated only by an actual recipient of the subject VM from whom a VN is requested by means of the Voice Notification Request field of the subject VM's Recipient field.

The user shall not have previously originated a VN in response to the subject VM requested voice notice type, by means of either the present abstract operation of auto-acknowledgement.

A user may delegate the task of generating VNs to the UA. In this case, this abstract operation is not present at the abstract interface between the UA and the user, that is, the operation is not available at the Origination Port. In this case the UA behaves as described in 17.3.

This abstract operation has the following arguments:

a) **Envelope:** A message submission envelope as defined by the MTS Abstract Service. The UA supplies all but the following envelope components, which the user provides:

   1) The desired per-message options (i.e. priority, per-message indicators, and extensions). Implicit conversion shall be prohibited; priority is that of the subject VM.

   2) The OR Names of the preferred recipients and the per-recipient options (i.e. explicit conversion and extensions) desired for each. The preferred recipient of the VN is the originator of the subject VM or, if present, the OR Name indicated in the VN Receiver field. Reports shall not be requested.

b) **Content:** The VN type (RN/NRN/SN) being originated.

   1) The user shall supply the value for the VN type being originated as part of the notification structure.

   2) If UA- to -UA security services are required, the user shall supply the value for the VM Application Security Elements field and in accordance with the user's security policy.

The VN shall be constructed as described in 17.3.

This abstract operation has the following results:

c) **Submission-identifier:** The message submission identifier the MTS assigns to the submission.

d) **Submission-time:** 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 VMGS and performed by the user.

Note – As abstractly defined, the VMGS provides no storage for received messages because the provisioning of said storage does not impact that user's ability to communicate with other VMGS-users. Thus the provision of storage is a local matter.

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 { }
```

A report may be received as a result of either:

a) The invocation of an Originate-VM abstract operation of a VM or by forwarding.

b) A message whose content was a VN that was originated as a result of a previously received message. The VN could be any of RN, SN or NRN.

This abstract operation has the following arguments:

c) **Envelope:** A report delivery envelope, whose make-up the MTS Abstract Service defines.

d) **Undelivered-object:** The content of the message whose status is being reported. A VM or VN.

If the report was provoked by a previous Originate VM abstract operation invocation, the argument shall be present if, and only if, P1 content return was requested.

This abstract operation has no results.

12.2.2 *Receive VM*

The Receive VM abstract operation receives a message whose content is a VM.

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

This abstract operation has the following arguments:

a) **Envelope:** The message's delivery envelope.

b) **Content:** The VM that is the message's content.

This abstract operation has no results.

When the received VM contains a VM Body Part (that is, when the original VM has been forwarded), it may be necessary to scan several levels of nested Heading fields in order to determine the correct original value for optional Heading fields (see 8.2.2 for the nested structure of a forwarded VM and 17.3.3 for rules related to Heading fields).

12.2.3 *Receive VN*

The Receive VN abstract operation receives a message whose content is a VN. The VN is provoked by a VM originated with the Originate VM abstract operation. A voice notification may be a RN, NRN or SN.

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

This abstract operation has the following arguments:

a) **Envelope:** The message's delivery envelope.

b) **Content:** The VN that is the message's content. A voice notification may be a RN, NRN or SN.

This abstract operation has no results.

12.3 *Management abstract operations*

12.3.1 *Change auto-discard*

The Change auto-discard abstract operation enables or disables auto-discard, the automatic discard by the Voice Messaging System of expired or obsolete messages delivered to, but not yet received by the user.

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

When it auto-discards a message, the Voice Messaging System originates a NRN on the user's behalf if, and only if, one was requested by the originator. This request is indicated by means of the V-notification-requests component of the subject VM's VNotificationRequestsField.

This abstract operation has the following arguments:

a) **auto-discard-expired-VM:** Whether or not *expired* VMs are to be automatically discarded by the Voice Messaging System. A Boolean.

b) **auto-discard-obsolete-VM:** Whether or not *obsolete* VMs are to be automatically discarded by the Voice Messaging System. A Boolean.

This abstract operation does not have any results.

12.3.2 *Change auto-acknowledgement*

The Change auto-acknowledgement abstract operation enables or disables auto-acknowledgement, the automatic origination of RNs by the VMGS-MS on the user's behalf. The automatic origination of RNs occurs on delivery of VMs that request RNs of the user. This request is conveyed by means of the V-notification-requests component of the subject VM's VNotificationRequestsField.

```
ChangeAutoAcknowledgements ::= ABSTRACT-OPERATION
   ARGUMENT SET {
      auto-acknowledge-VM                        [0] BOOLEAN,
      auto-acknowledge-suppl-recipient-info      [1] VNSupplementaryInfo OPTIONAL }
   RESULT
   ERRORS { SubscriptionError }
```

This abstract operation has the following arguments:

a) **auto-acknowledge-VM:** Whether or not the VMs are to be auto-acknowledged. This is a mandatory, Boolean argument.

b) **auto-acknowledge-suppl-recipient-info:** The *supplemental receipt-information* receipt field of each RN provoked by auto-acknowledgment.

   The conditional argument, *auto-acknowledge-suppl-receipt-info*, shall be present if, and only if, the auto-acknowledge-VMs argument has the value *true*.

This abstract operation does not have any results.

12.3.3 *Change auto-forwarding*

The Change auto-forwarding abstract operation enables or disables auto-forwarding, i.e. the automatic forwarding of VMs by the Voice Messaging System to pre-specified users or DLs. Such forwarding occurs after delivery of the VM.

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

The Body for each VM, namely the VM originated as a result of auto-forwarding, comprises a single body part of type *message*, as in the IPMS. The content of the message represented by that body part is the forwarded VM.

When it auto-forwards a VM, the VMG application entity originates a NRN on the user's behalf if, and only if, one was requested of the recipient by means of the V-notification-requests component of the subject VM's V Notification Requests Field.

This abstract operation has the following arguments:

a) **auto-forward-VM:** Whether or not VMs are to be auto-forwarded. This is a mandatory, Boolean argument.

b) **auto-forward-recipients:** The user or DLs to which VMs are to be auto-forwarded. Optionally, the recipient spoken name may be included. A sequence of O/R names.

   The conditional argument, auto-forward-recipients, shall be present if, and only if, the auto-forward-VMs argument has a value of **true**.

c) **auto-forward-heading:** The VM heading that is to be used for each forwarded VM. The heading's *auto-forwarded* field shall have the value *true*.

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

d) **auto-forward-comment:** 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 VM.

   The conditional argument, auto-forward-comment, shall be present if, and only if, the auto-forwarded-VMs agreement has the value **true**.

This abstract operation does not have any results.

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

12.3.4 *Change auto-SNotice*

The Change auto-SNotice abstract operation enables or disables auto-SNotice, the automatic origination of SNs by the Voice Messaging System on the user's behalf. The automatic origination of SNs occurs on delivery of VMs that request SNs of the user. This request is conveyed by means of the V-notification-requests component of the subject VM's V Notification Requests Field.

```
ChangeAutoSNotice ::= ABSTRACT-OPERATION
  ARGUMENT SET {
    auto-SNotice                 [0] BOOLEAN,
    auto-service-status-info     [1] EOSSupportIndicator,
    auto-SN-suppl-recipient-info [2] VNSupplementaryInfo OPTIONAL }
  RESULT
  ERRORS { SubscriptionError }
```

```
EOSSupportIndicator ::= BIT STRING {
    unspecified                        (0),
    auto-forwarding-ind                (1),
    language-ind                       (2),
    obsoleting-ind                     (3),
    attendant-assisted-delivery-request (4),
    expiry-date-ind                    (5),
    body-part-encryption-ind           (6) } (SIZE (2..ub-sn-reasons)) -- setting any bit on, indicates
                                                                       -- non-support
```

This abstract operation has the following arguments:

a) **auto-SNotice:** Determines whether or not the SNs are to be automatically generated. This is a mandatory, Boolean argument.

b) **auto-service-status-info:** The value for SN Reason Field to be included in each SN provoked by auto-SNotice.

The conditional argument, *auto-service-status-info*, shall be present if, and only if, the auto-SNotice argument has the value *true*.

c) **auto-SN-suppl-recipient-info:** The value that is to be supplied as the comment of each SN conveyed to the originator of the subject VM.

The conditional argument, *auto-SN-suppl-recipient-info*, shall be present if, and only if, the **auto-SNotice** argument has the value **true**.

This abstract operation does not have any results.

12.4 *Performance of management operations*

A UA shall perform the abstract operation it makes available at its management port as specified below The UA alters one or more of its state variable (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 error should be repeated is beyond the scope of this Recommendation.

12.4.1 *Change auto-discard*

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

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

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

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

12.4.2 *Change auto-acknowledgment*

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

a) **auto-acknowledge-VMs:** A Boolean that indicates whether or not *auto-acknowledge* is in effect.

b) **auto-acknowledge-suppl-receipt-info:** The supplemental receipt information 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-VMs* 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.

12.4.3 *Change auto-forwarding*

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

a) **auto-forward-VMs:** 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 VMs are being auto-forwarded.

c) **auto-forward-heading:** The VM heading of each forwarded VM 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 VM.

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

12.4.4 *Change auto-SNotice*

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

a) **auto-SNotice:** A Boolean that indicates whether or not *auto-SNotices* is in effect.

b) **auto-service-status-info:** The value to be used to determine the SN information to the originator of the subject VM when the SN is automatically generated.

c) **auto-SN-suppl-recipient-info:** The value that is to be supplied as the comment of each SN conveyed to the originator of the subject VM.

A UA shall perform the *change auto-SNotice* abstract operation by recording the value of the *auto-SNotice* argument in the correspondingly named state variable. If the value is **true**, the UA also shall record the values of the *auto-service-status-info* and *auto-SN-suppl-recipient-info* arguments in the correspondingly named state variables.

13 Abstract errors

The abstract errors that may be reported in response to the invocation of the abstract operations available at the Origination Port, Reception Port and Management Ports are defined and described below or as part of the MTS Abstract Service definition.

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 in his invocation of the abstract operation whose performance is aborted.

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

This abstract error has the parameter:

– **problem:** The subscription-related problem encountered.

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

This parameter may assume any one of the following values:

i) **vm-eos-not-subscribed:** A VM element of service is not subscribed.

ii) **mts-eos-not-subscribed:** A 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 OR 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 VMGS Abstract Service, defined above, the VMGS shall transparently extend to each user the other MS (see CCITT Rec. X.413 | ISO/IEC 10021-5) and MTS (see CCITT Rec. X.411 | ISO/IEC 10021-4) capabilities identified below. (The enumeration of these capabilities necessarily anticipates the fact, stated in clause 15, that MSs and the MTS are among the VMGS' component parts.)

The following additional capabilities shall be provided:

a) **Submission:** Capabilities of the MS' or MTS' submission port not embodied in the VMGS Abstract Service, e.g. the ability to cancel delivery of a previously originated message whose content is a VM (but not a VN), if deferred delivery was selected.

b) **Delivery:** Capabilities of the MTS' delivery port not embodied in the VMGS 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 VMGS 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 above are excluded from the formal definition of the VMGS 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.

15 Secondary object types

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

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

These lesser objects are referred to as the secondary objects of Voice Messaging. They include a single, central object, the MTS, and numerous peripheral objects: Voice messaging system user agents (VMGS-UA), Voice messaging system message stores (VMGS-MS), and Telephone service access unit(s) (TSAU).

The structure of the VMGS is depicted in Figure 5/X.440. As shown by the figure, VMGS-UAs, VMGS-MSs, and TSAUs are the instruments by means of which the VMGS provides the VMGS Abstract Service to users.

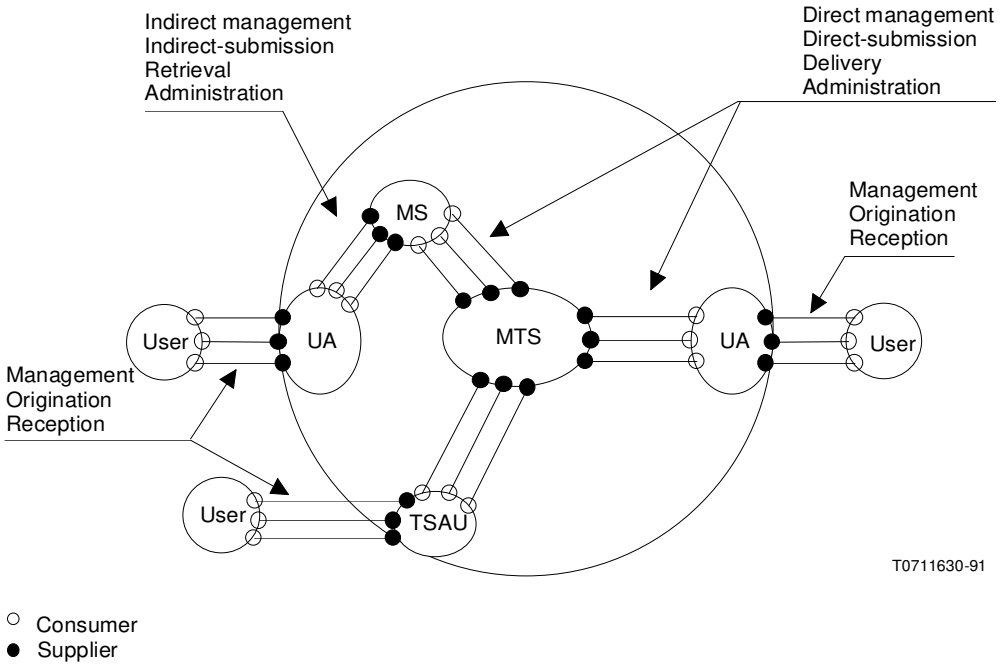

FIGURE 5/X.440

The voice messaging system

The secondary object types are defined and described below. The types of ports by means of which they interact are discussed in clause 16.

The refinement above encompasses all possible interconnections of all possible objects. It ignores the possible absence of objects of a particular type (e.g. TSAU), and specific logical configurations of the MS. The latter are identified in CCITT Rec. X.402 | ISO/IEC 10021-2.

The MTS supplies import and export ports. However, since those ports are not formally defined (in CCITT Rec. X.411 | ISO/IEC 10021-4), they are not included in the formal refinement above.

15.1 *VMGS user agent*

An VMGS user agent (VMGS-UA) is a UA tailored so as to better assist a single user to engage in Voice Messaging. It helps the user originate, receive, or both originate and receive messages containing Information Objects of the types defined in clause 6.

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

The VMGS comprises any number of VMGS-UAs.

Note – As noted above, the term user agent (UA) is used throughout this Recommendation with the meaning of VMGS-UA.

15.2 *VMGS message store*

An VMGS message store (VMGS-MS) is an MS tailored so as to better assist a single UA engaged in Voice Messaging. It helps it submit, take delivery of, or both submit and take delivery of messages containing Information Objects of the types defined in clause 6.

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

The VMGS comprises any number of VMGS-MSs.

Note – As noted above, the term message store (MS) is used throughout this Recommendation with the meaning of VMGS-MS.

15.3 *Telephone service access unit*

A Telephone Service access unit (TSAU) is an AU that enables a Telephone System direct user be an indirect user of Voice Messaging from a Telephone network's terminal. A TSAU helps the user originate, receive, or both originate and receive messages containing Information Objects of the types defined in clause 6.

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

Note 1 – The TSAU consumes *Import* and *Export*. However, since they are not formally defined in CCITT Rec. X.411 | ISO/IEC 10021-4, they are not included in this formal definition of a TSAU.

Note 2 – It is unlikely that a TSAU will be able to handle non-voice encoded body parts on reception from VM direct users. In other words, sending a message consisting of non-voice encoded body parts may cause MTS Non-Delivery reports to be returned.

15.4 *Message transfer system*

In the present context, the Message Transfer System (MTS) conveys Information Objects of the types defined in clause 6 between UAs, MSs, and AUs.

The VMGS comprises a single MTS.

16 Secondary port types

The secondary objects of Voice 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 Voice Messaging. They are of the types identified below.

The capabilities embodied in one Submission Port, one Retrieval Port, and one Administration Port constitute the MS Abstract Service. They are defined in CCITT Rec. X.413 | ISO/IEC 10021-5.

The capabilities embodied in one Submission Port, one Delivery port, and one Administration Port constitute the MTS Abstract Service. They are defined in CCITT Rec. X.411 | ISO/IEC 10021-4.

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.

16.1 *Submission port*

In the present context, a Submission Port is the means by which a UA (directly or indirectly) or an MS (directly) submits messages containing Information Objects of the types defined in clause 6.

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.

16.2 *Delivery port*

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 clause 6.

The MTS supplies one Delivery Port to each UA configured without an MS and to each MS.

16.3 *Retrieval port*

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 clause 6.

An MS supplies one Retrieval Port to its UA.

16.4 *Administration port*

The MTS supplies one Administration Port to each UA configured without an MS and to each MS.

16.5 *Import port*

In the present context, an Import Port is the means by which the MTS imports reports concerning, and messages containing, VM Information Objects of the types defined in clause 6.

The MTS supplies one Import Port to each AU.

16.6 *Export port*

In the present context, an Export Port is the means by which the MTS exports messages containing VM Information Objects of the types defined in clause 6.

The MTS supplies one Export Port to each AU. For each TSAU, the MTS exports P1 reports, VMs and VNs.

16.7 *Management port*

In the present context, a Management Port is the means by which a single user changes personally managed information that is on file with the Voice Messaging System. By means of such a port the user enables and disables auto-discard, -acknowledgement, and -VSNotice, and -forwarding. These managed Information Objects are of the types defined in clause 6.

17 User agent operation

A UA must employ the MTS in a particular way in order to (correctly) provide the VMGS 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 what follows.

Note – The purpose of the following is not to dictate or constrain the implementation of a UA application entity, but rather to specify the effect to be achieved between the UA and an MTA.

17.1 *Performance of origination operations*

A UA shall perform the abstract operations it makes available at its Origination Port as prescribed below.

In the performance of these operations, the UA invokes the following abstract operation of the MTS Abstract Service (which, for what follows, are unqualified as to their source):

– Message Submission.

In response to the invocation of this abstract operation, 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.

Note – Probe Submission is not used in Voice Messaging.

17.1.1 *Originate VM*

A UA shall perform the Originate VM 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 components not explicitly mentioned below shall be as specified by the user through the Originate VM Envelope argument:

   1) **Originator-name**: The OR Name of the UA's user.

2) **Content-type** and **Original-encoded-information-types**: Determined from Originate VM's Content argument as specified in 19.2 and 19.4, respectively.

3) **Content-identifier** and **Content-correlator**: Specified or omitted as a local matter.

4) The *security arguments* on message submission are subject to the security policy in force. When the security policy specifies the support of *Content Integrity* message transfer security element of service, and when *VNotificationSecurity* is requested, the UA shall generate and submit the **content-integrity-check** Security Argument as defined in 8.2.1.1.1.28 in CCITT Rec. X.411 | ISO/IEC 10021-4.

The components of the *Envelope* argument that constitute *per-recipient fields* shall also be as specified through this abstract operation.

To prevent an unknown number of VNs from being sent to the message's original originator as a result of forwarding, the originator may set, if available, ***dl-expansion prohibited*** to **Prohibited** when requesting any VNs.

b) *Content:* Determined from Originate VM's Content argument (identified as a VM) as specified in 19.1.

1) If **proof** or **non-repudiation** for a notification is requested, the UA shall set the *VNotificationSecurity* field accordingly for each recipient as required.

2) If **proof** or **non-repudiation** of content received advice is requested, the UA shall set the *VMReceptionSecurity* field accordingly for each recipient as required.

The **results** of Originate VM shall be as follows:

c) *Submission-identifier:* Message Submission's Message-submission-identifier result.

d) *Submission-time:* Message Submission's Message-submission-time result.

How the UA uses the result of Message Submission's **content-identifier** or **content-correlator** is a local matter.

The inclusion of Message Submission *extensions* among Originate VM's results is proper and may be the subject for future standardization.

17.1.2 *Originate VN*

A UA shall perform the Originate VN abstract operation, if the UA makes it available to its user, by invoking Message Submission with the arguments indicated below, and by returning to its user the results indicated below. This abstract operation is only performed in response to a request made by the subject VM's originator.

Note – Subject VM is defined in clause 9 and subclause 17.3.3.

A user may delegate the task of generating VNs to the UA. In this case, this abstract operation is not present at the abstract interface between the UA and the user, that is, the operation is not available at the Origination Port. In this case, the UA behaves as if the abstract operation would have been invoked. The UA may not accept the VM at will, but shall accept the VM when the VM is made available to the user, or when it forwards a VM with content changed. In this context, "content changed" means that the forwarding UA adds or removes body parts from the VM being forwarded, in accordance with 17.3.3.

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 components not explicitly mentioned below shall be as specified by the Originate VN Envelope argument:

1) **Originator-name**: The OR Name of the UA's user.

2) **Content-type** and **Original-encoded-information-types**: Determined from the VN as specified in 19.2 and 19.4, respectively.

3) **Content-identifier** and **Content-correlator**: Specified or omitted as a local matter.

4) **Deferred-delivery-time**: Omitted.

5) **Priority**: Same as that of the subject VM.

6) **PerMessageIndicators**: If supported by the MTA, the notification type shall to be set to Type 1 for RN, to Type 2 for NRN or Type 3 for SN.

Note – MTA support for setting the notification type indication may become mandatory in a future version of MHS.

b) *Content:* Determined from Originate VN's Content argument (identified as a RN, SN or NRN) as specified in 19.1.

1) If in the subject VM, *VMReceptionSecurity* is set to *non-repudiation* and *VNotificationSecurity* is set to *non-repudiation* and the **content-integrity-check** security argument ***is present*** in the delivery envelope of the subject VM, then the **content-integrity-check** security argument is copied into the **content-integrity-check** field of the VN. The UA shall submit the VN with a non-repudiable security element **content-integrity-check** (possibly in the message token) or a **message-origin-authentication-check** (depending on the security policy in force).

2) If in the subject VM, *VMReceptionSecurity* is set to *non-repudiation* and *VNotificationSecurity* is set to *non-repudiation* and the **content-integrity-check** security argument is ***not present*** in the P1 delivery envelope of the subject VM, then the Content of the subject message shall be copied into the **original-content** field of the VN. The UA shall submit the VN with a non-repudiable security element **content-integrity-check** (possibly in the message token) or a **message-origin-authentication-check** (depending on the security policy in force).

3) If in the subject VM, *VMReceptionSecurity* is set to *proof* and *VNotificationSecurity* is set to *proof* and the *content-integrity-check* security argument ***is present*** in the delivery envelope of the subject VM, then the *content-integrity-check* security argument is copied into the **content-integrity-check** field of the VN. The UA shall submit the VN with the security element **content-integrity-check** (possibly in the message token) or a **message-origin-authentication-check** (depending on the security policy in force).

4) If in the subject VM, *VNotificationSecurity* is set to *proof* the UA shall submit the VN with the security element **content-integrity-check** (possibly in the message token) or the **message-origin-authentication-check**, according to the security policy in force.

5) If in the subject VM, *VNotificationSecurity* is set to *non-repudiation* the UA shall submit the VN with a non-repudiable security argument **content-integrity-check** (possibly in the message token) or a **message-origin-authentication-check**, according to the security policy in force.

6) If in the subject VM, *VMReceptionSecurity* is set to *proof* and *VNotificationSecurity* is set to *proof* and the **content-integrity-check** security argument ***is not present*** in the delivery envelope of the subject VM, then the Content of the subject VM is copied into the **original-content** field of the VN. The UA shall submit the VN with the security element **content-integrity-check** (possibly in the message token) or a **message-origin-authentication-check** (depending on the security policy in force).

7) If the UA's MTA does not support secure messaging and *VMReceptionSecurity* or *VNotificationSecurity* services are requested, the VN shall contain an appropriate Reason Code.

8) *Content-integrity-check* of the subject VM shall always be checked for validity by the receiving UA before a VN is created.

The **results** of Originate VN shall be as follows:

c) *Submission-identifier:* Message Submission's Message-submission-identifier result.

d) *Submission-time:* Message Submission's Message-submission-time result.

How the UA employs the result of Message Submission's **Content-identifier** or **Content-correlator** is a local matter.

17.2 *Invocation of reception operations*

A UA shall invoke the abstract operations available at its Reception Port as specified below.

The UA invokes these operations in response to the MTS's invocation of the following abstract operations of the MTS Abstract Service (which, for what follows, are unqualified as to their source):

a) Report Delivery;

b) Message Delivery.

The abstract operations of a Reception Port report have neither results nor errors.

17.2.1 *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 19.1.

If the report was provoked by a previous Originate VM abstract operation invocation, the argument shall be present if, and only if, content return was requested through P1's *per message indicators*. Otherwise, the argument shall be absent.

Note – Return-of-Contents is discouraged in P1 Delivery Reports per Recommendation F.400 and not provided in any VN defined in this Recommendation unless specifically requested for secure notifications as provided in 17.1.2.

How the UA employs the **Content-identifier** or the **Content-correlator** component of Report Delivery's Envelope argument is a local matter.

17.2.2 *Receive VM*

When the MTS invokes Message Delivery at a UA's Delivery Port, and its Content argument encodes a VM as specified in clause 8, the UA shall invoke the Receive VM 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 19.1 (but no longer marked as a VM).

Note – Even in the case of forwarding, the user accepts the message through the Receive VM operation.

17.2.3 *Receive VN*

Whenever the MTS invokes Message Delivery at a UA's Delivery Port, and its Content argument encodes a VN as specified in 19.1, the UA shall invoke the Receive VN 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 19.1.

17.3 *Internal procedures*

A UA shall perform, as specified below, the internal procedures that relate to *acceptance of the VM, refusal of the VM and forwarding*.

A user may instruct its UA to accept or refuse the VM of incoming messages based on certain criteria. In addition, a user may instruct its UA to forward incoming messages based on certain criteria.

Because of forwarding, redirection or DL expansion, it is possible for a UA to receive the same VM more than once. Mechanisms for detecting such duplicate receptions are not required, but may be a matter of local implementation by the UA. If these mechanisms exist, and notifications are requested, the UA shall not generate duplicate notifications for the duplicate messages. If they do not exist, and notifications are requested, the UA shall originate the requested VNs, as appropriate.

The procedures involve the following abstract operations of the MTS Abstract Service (which, for what follows, 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:

c) The MTS has conveyed the message to the UA by invoking Message Delivery at the UA's Delivery Port;

d) The UA has not conveyed the message to the user by invoking Receive VM at the UA's Reception Port;

e) The message contains a VM (rather than a VN).

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

Upon the transfer of responsibility of a VM from the MTS to its MTS-user, SNs shall be actioned, provided that a SN was requested and that the delivered message satisfies SN creation criteria.

17.3.1 *Acceptance of the VM*

A UA is said to have accepted the VM when responsibility for the message is successfully passed from the UA to the VMG-user. The UA shall follow the procedures below for each candidate message with respect to whose content the following condition holds:

– The VM requests a RN by means of the Voice Notification Request field of the appropriate Recipients Sub field in the VM's Recipients field.

The UA may forward a message that it has accepted the VM. See also 17.3.3 on forwarding.

17.3.1.1 *Construction of RN*

The UA shall construct a RN if, and only if, one is requested by means of the Voice Notification Requests field of the appropriate Recipients Sub field in the VM's Recipients field and in accordance with 17.3.1.

The RN shall have the following **common fields**:

a) *Subject VM:* The VM's *This VM* field or, if present, the *Original VM Identifier* in the VN Receiver field.

b) *VN Originator:* The OR Name of the UA which submits the VN. If the UA is a preferred recipient of the subject VM, the value shall be precisely that which is the Recipient field of the subject VM.

c) *Notice Creation Time*: The current date and time.

d) *VN Voice Encoding Type:* The encoding algorithm used to create any spoken component of this VN. It may also indicate any additional parameters needed by the recipient to decode the spoken information.

e) Optionally, *Conversion Indication:* The converted encoded information types component of the message delivery's envelope argument.

f) Optionally, *Notification Security Elements Field:* Notification Security elements carried in any component field shall follow the rules of b) in 17.1.2.

g) Optionally, *Original VM Spoken Subject:* The value of the subject VM's spoken subject field.

h) Optionally, *Subject VM Other Recipients:* The OR Names (and when present in the subject-VM, the spoken encoding of the recipient name) of the other recipients that the subject VM was sent to.

i) Optionally, *Supplementary Information* that adds information to the reason given.

j) Optionally, *First Recipient:* The OR Name of the UA which the originator of the VM specified as *recipient* in the subject VM's heading or, if present, the First Recipient field in the VN Receiver field. If the Recipient Spoken Name component is present, then it is included and its encoding is indicated in the *spoken-name-encoding-field*. If the VN Receiver field is not present, the First Recipient OR Name is the OR Name of the UA creating the VM, unless the MTA has performed redirection or DL-expansion. In case of redirection, the correct First Recipient OR Name must be obtained from the Intended Recipient Name field of the P1 envelope (see 8.3.1.1.1.4 of CCITT Rec. X.411 | ISO/IEC 10021-4). In case of DL-expansion, the correct First Recipient OR Name must be obtained from the DL Expansion History field of the Delivery envelope (see 8.3.1.1.1.7 of CCITT Rec. X.411 | ISO/IEC 10021-4).

k) Optionally, *Notifications Extensions:* Extensions relevant to notifications.

17.3.1.2 *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 VN abstract operation with the following exceptions:

1) **Priority**: As specified by subject VMs Message Delivery Envelope's *priority* argument.

2) **Per-message-indicators**: The RN indication bit and the conversion-prohibited shall be among the values specified. The setting of other bits is a local matter.

3) **Per-recipient-fields**: A single field whose Recipient-name component shall be the Originator-name component of Message Delivery's Envelope argument.

b) *Content*: Determined from the RN as specified in 19.1.

17.3.2 *VM Service Notification (SN) generation*

A UA shall generate a VM Service Notification when a message is accepted by the intended user even though the UA does not support all elements of service in the message and an SN was requested by the originator of the subject VM. The UA shall follow the procedures below for each candidate message under the following conditions:

a) The VM requests an SN of the UA's user by means of the *VN Requests* field of the subject VM's *Recipients Sub field*.

b) The VM is not forwarded.

c) The UA does not support one or more requested additional elements of service on reception.

Note – See also 17.3.3 on forwarding.

17.3.2.1 *Construction of SN*

The UA shall construct an SN if, and only if, one is requested by means of the *VN Requests* field of the subject VM's Recipients Sub field and in accordance with 17.3.2.

The SN shall have the common fields prescribed for Construction of RN (see 17.3.1.1).

The SN shall have the following fields:

a) *SN Reason Code:* Identifies the requested element of service that was not available to the intended recipient. It may also include a diagnostic code to further explain the reason value.

b) Optionally, *SN Extensions:* Extensions relevant to an SN.

17.3.2.2 *Submission of SN*

The UA shall submit the SN created as per the above by invoking Message Submission. Its Envelope argument shall be as prescribed for Acceptance of the VM (see 17.3.1) except that the PerMessageIndicators bit for SN Notification type shall be set. Its Content argument is determined from the SN as specified in 19.1.

17.3.2.3 *Handling of received VM*

The handling of a received VM for which the UA generates a SN (or not) is a local matter and is not defined in this Recommendation.

17.3.3 *VM Forwarding*

The procedures defined in this subclause describe **VM Forwarding**.

Note – For brevity, the term **forwarding** is used in this Recommendation as a synonym for VM Forwarding.

A user may instruct its UA to forward received messages based on local criteria. A user may also instruct its UA to automatically forward requests for notifications together with the forwarded message.

In order to forward a VM, the UA creates a new VM with a new Heading and encapsulates in the Primary Body Part the received subject VM (Heading and Body), and optionally, components of the P1 envelope of the received message, using the body part type VM Body Part (see 8.2.2). The term forwarded-VM refers to the outermost element of the message structure, the VM Body Part, of the new VM being forwarded by the UA and contains all or part of the original subject VM.

Figure 6/X.440 illustrates forwarding with an example.

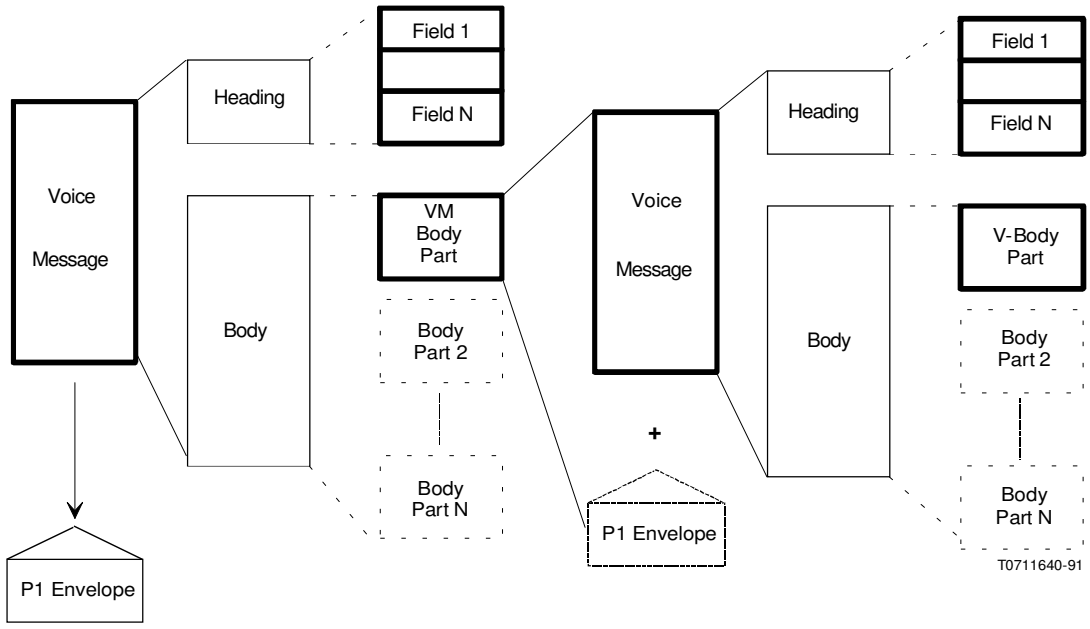

FIGURE 6/X.440

Forwarded VM structure

The term **forwarding VM** refers to the new VM that is being created, and that will include all or part of the subject VM, in accordance with 5.1.

The following rules apply to the creation of the Heading fields of the forwarding VM:

– Optional fields whose values are changed with respect to the values present in the subject VM shall be present;

– Heading fields whose values are unchanged shall be copied from the subject VM Heading to the forwarding VM Heading if the field is present in the subject VM Heading and the value in the field is other that the value specified as DEFAULT in 8.1;

– Other fields need not be copied.

VM Forwarding is done by the MS if the UA has an MS, otherwise by the UA.

VM Forwarding may take two forms:

a) forwarding of message not accepted or accessed by the user;

b) forwarding of message when accepted by the user.

VM Forwarding may take place even if no notifications have been requested. This is equivalent to form b) above.

A message may be forwarded to one or more recipients. However, the request for VNs for any recipient depends upon the form of VM Forwarding actioned.

The UA shall, subject to the instructions given by the user, forward messages as follows.

17.3.3.1 *VM Forwarding of message without accepting the VM*

VM Forwarding of a message without accepting the VM implies the following:

a) The *Primary Body Part* of the new message is the content of the subject message unchanged. The delivery envelope of the received VM shall be included if security notifications are requested.

b) If the *VN Requests* field of the subject VM's *Recipients Sub* field requests Receipt Notification (RN) or Service Notification (SN), the *VN Request* is forwarded unchanged with the new message to one, and only one, of the recipients of the new message. The value of the *Message Forwarded* field shall be set to **TRUE**.

c) If **forwarding fails** (i.e a Non-delivery report on the forwarded message is returned) within a given period of time (either specified by the originator in Expiry Time or as a local decision in the MS or UA, with priority given to the Expiry Time), the UA may send a NRN back to the originator provided a NRN was requested. The ability to generate this NRN may be hindered if the message was deleted when VM Forwarding is actioned.

Note – Return-of-Contents is discouraged in P1 Delivery Reports per Recommendation F.400 and not provided in any VN defined in this Recommendation unless specifically requested for secure notifications as provided in 17.1.2.

d) If the *VN Requests* field of the subject *Recipients Sub* field requests NRN, the Non-Receipt Notification shall be sent back to the originator of the subject VM.

The delivery envelope of the received message shall be included in the new VM if the received VM's Primary Body Part is not a Forwarded VM. Body parts shall not be added or deleted during VM Forwarding. However, supplementary information may be added through the *other-parameters* component of the VM Body Part.

A message may be forwarded to multiple recipients as specified in a) and b) except that the VN-request shall be passed to only one recipient. The recipient to which the VN-request is passed shall also have the P1 element of protocol, *DL expansion indication*, set to *DL expansion prohibited*.

If a MTS Non-Delivery Notification is returned for the recipient to which VN-request was forwarded, then the forwarding UA shall fulfill the VN-request itself. For example, the UA may choose to accept responsibility itself or to not accept responsibility which in turn will determine if a RN or NRN should be created, had one been requested.

Note 1 – Since the F.400-Series Recommendations discourage the use of *Return of content* element of service, there is an implication that the forwarding UA has retained the identity of the recipient to which the VN-request was forwarded.

Note 2 – Failure of the MTS to deliver a forwarded-VM to the intended recipient to which VN-request was *not* passed, *would not* cause VN-request to be acted upon. For example, failure of the MTS to deliver a forwarded-VM because an intended recipient of the forwarded-VM is actually a distribution list, does not cause VN's to be generated.

The originator of a message may prohibit VM Forwarding by setting the Message Forwarding Permitted field to FALSE. In this case, if the receiving UA cannot accept the VM and the NRN is requested, the receiving UA shall submit a NRN with appropriate reason code. If the receiving UA cannot accept the VM and the NRN is not requested, then a NRN shall not be submitted.

If VM Forwarding is permitted by the originator, then it is possible to forward a message more than once.

17.3.3.2 *Forwarding of message after accepting the message*

Forwarding a message after accepting the message implies the following:

a) The *Primary Body Part* of the new message is the content of the subject message changed or unchanged. This type of forwarding is less restricted and may include removal or addition of body parts. However, the forwarded VM Body Part shall not be removed.

   Note – If the delivery envelope of the received message is included in the forwarded message, and if that envelope contained security fields, and if body parts are added or removed, then the security fields may no longer be valid.

b) The addition and removal of Additional-Body-Parts is permitted. However, not more than one voice encoded body part (AdditionalBodyParts) may be added to the resulting forwarded message per instance of forwarding.

   Note – The addition of more than one voice encoded body part per instance of forwarding is not permitted. However, a forwarded-VM may contain more than one voice encoded body part. Voice encoded information contained in a forwarded-VM that is added during forwarding may be encoded differently from the original subject VM.

c) *Message Forwarded* shall not be requested.

d) If the *Notifications Requests Field* of the subject VM requests Receipt Notification, a RN shall be sent back to the addressee specified in the field VNReceiver, or to the originator of the VM, if no VN Receiver is specified. Receipt notification may be requested either with or without a request for the return of the subject VM.

e) A *Non-Receipt Notification*, NRN, **shall not** be sent back to the originator of the message.

Note – By scanning the successive nested Headings of a forwarded VM, the final recipient UA can determine, from the setting of the *Message Forwarded* indication field, which UA or MS in the forwarding chain accepted the message.

17.3.3.3 *Prevention of loops*

The UA may use the *Message Forwarded* field of the heading to initially detect that loop suppression must be performed. The UA shall suppress forwarding if the VM about to be forwarded contains a forwarded VM that the UA previously created. In other words, *VM forwarding* shall be suppressed whenever the UA that is about to forward the same subject VM determines that it had forwarded this same message earlier to someone else. It may need to check through a nested series of VM body parts to make this determination.

The UA shall consider itself to have created a forwarding VM if, and only if, the OR Name component of a This VM field in the forwarded VM matches the OR Name of the UA's user.

Note – Forwarding a VM of the kind described above would constitute a VM Forwarding "loop".

17.3.3.4 *Construction of Forwarding VM*

The UA shall construct a forwarding VM whose Primary Body Part comprises a body part of type VM Body Part. Additionally, other body parts may be added, and body parts may be removed.

The Heading shall have the following components:

a) *This VM:* New value generated.

b) *Originator:* OR Name of the forwarding user.

c) *Recipients:* The recipients to which the VM is being forwarded.

If the message is *not* accepted, the following rules relating to the components of the VM Heading apply:

d) *VN Requests* (sub-field of Recipients): The UA may forward the VM to several recipients by simply adding recipients to the Recipients field. If the UA does not accept the message, and if VN Requests are present in the subject VM, the UA shall set identical VN Requests for one, and only one, of the recipients, and notifications are not requested of any other recipient.

Note – VMGS-MS also follow these rules, see 18.4.

e) *Expiry Time:* may be set to a value different from the value indicated in the subject VM.

f) All other Heading fields shall follow the general rules in 17.3.3.

If the VM is accepted, the VM Heading shall comply with a), b) and c) above and with the following rules:

g) Other fields may be added (including VN Requests).

Other fields apart from those especially mentioned above may, but need not, be copied from the Heading of the subject VM to the Heading of the forwarding VM. There is one exception, the *Original VM Identifier* shall not be present.

VN Requests set for any recipient is at the discretion of the VMG-user. The VN Receiver field is set to the forwarding user or to some OR address other than the originator of the forwarded VM.

The *PrimaryBodyPart* shall be of type VM Body Part and shall have the following components:

h) *Parameters:* Specified or omitted in the field MessageParameters as a local matter.

i) *Data:* The VM to be forwarded in the field MessageData.

The *AdditionalBodyParts* shall be of type VM Externally Defined BodyPart and shall have the components relevant to the encoded information object being carried in that body part.

17.3.3.5 *Submission of a Forwarding VM*

The UA shall submit the forwarding VM it constructed above by invoking Message Submission with the following arguments:

a) *Envelope:* The components of this argument shall be as follows:

1) **Originator-name**: The OR Name of the UA's user.

2) **Content-type** and **Original-encoded-information-types:** Determined from the VM as specified in 19.2 and 19.4, respectively.

3) **Content-identifier**: Specified or omitted as a local matter.

4) **Priority**: As specified by the subject VM's Message Delivery's Envelope argument unless forwarding of message occurs after accepting the message. In this case, the setting of Priority is at the user's discretion.

5) **Per-message-indicators** and **Extensions**: A local matter, except that the NRN bit will be set.

6) **DeferredDeliveryTime**: Omitted.

7) **Per-recipient-fields**: The Recipient-name components shall be the OR Name(s) that the message shall be forwarded to. Their other components are a local matter.

b) *Content:* Determined from the VM as specified in 19.1.

17.3.3.6 *Construction of NRN*

The UA shall construct a NRN if, and only if, one is requested by means of the Voice Notification Requests field of the subject VM's Recipients field and the user is not willing to accept the message.

The NRN shall have the common fields as prescribed for non-acceptance of the message (see 17.3.3.1).

The NRN shall also have the following fields:

a) *Non-receipt UA/MS Reason Code:* The reason why the subject message was not received by the intended recipient's MS or UA. This information is conveyed in the field VNRNUAMSReasonCodeField.

b) Optionally, *Non-receipt User Reason Code:* The reason why the subject message was not received by the intended recipient. This information is conveyed in the field VNRNUserReasonCodeField.

c) Optionally, *Non-receipt TSAU Reason Code:* The reason why the subject message was not received by the intended recipient associated with a TSAU. This information is conveyed in the field VNRNTSAUReasonCodeField.

d) Optionally, *NRN Extensions:* The set of NRN extension sub fields that are to be associated with the VN.

17.3.3.7 *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 acceptance of the VM (see 17.3.1) except that the *notification type* shall be set to type 2 (NRN), its *Content* argument determined from the NRN as specified in 19.1.

Note – If the OR Name in the field VNReceiverField is not valid, then the UA may be unable to submit the VN. Guidelines regarding the use of the VNReceiverField are found in j) of 17.3.1.1.

18 Message Store operation

CCITT Rec. X.413 | ISO/IEC 10021-5 defines the abstract service for a general content independent Message Store (MS). The MS is an optional system component in an MHS. The MS is asssociated with a single user's UA. The user can submit messages through it and retrieve messages that have been delivered to the MS. In addition, the MS can perform certain predefined auto actions on the UA's behalf.

Note — Because the MS is an optional system component in an MHS, use of the word "shall" with respect to MS specifications is to be construed as mandating a provision of an MS or the services it provides, if an MS is provided.

All the abstract operations, general attribute types and general auto actions types defined in CCITT Rec. X.413 | ISO/IEC 10021-5 are also available for use by Voice messages.

An MS may optionally offer additional support for the Voice messaging specific attribute types and auto actions, which would qualify it as a voice messaging specific MS (Voice MS). These additional definitions are given in what follows.

18.1 *Binding to the MS*

Binding to the MS is described in 7.1 of CCITT Rec. X.413 | ISO/IEC 10021-5. Attention should be given to the following points when using the MS for Voice messaging.

18.1.1 *Abstract-bind argument*

The following parameters from 7.1.1 of CCITT Rec. X.413 | ISO/IEC 10021-5 have special meaning in this Recommendation:

a) Fetch-restrictions

   The name of the object identifier for the Voice content type is **id-mct-pvm**, the value is defined in Annex A.

b) Allowed-EITs

   The names of the object identifiers so far standardized in this Recommendation are defined in Annex A. See also 8.1.12.

18.1.2 *Abstract-bind result*

The following parameter from 7.1.2 of CCITT Rec. X.413 | ISO/IEC 10021-5 has special meaning for this Recommendation:

— Available-auto-actions

Note — The use of the general auto action **auto-forward** is discouraged for use with VMs. Instead the Voice messaging specific auto actions **vm-forwarding-with-RN** and **vm-forwarding-with-NRN** and **vm-forwarding-without-notification** should be used as defined in 18.4.

18.2 *Creation of Information Objects*

An MS 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 a VM or VN that is delivered to it.

b) The MS shall maintain as a separate information object not only each message containing a forwarding VM [pursuant to Item a)] but also each message containing a forwarded VM (recursively).

c) The MS shall assign sequence numbers to the messages in the hierarchy formed by a forwarding VM and its forwarded VMs.

The general (content independent) attributes that may occur in a stored-messages information-base are documented in CCITT Rec. X.413 | ISO/IEC 10021-5. All content-independent MS attributes can be used for the content defined in this Recommendation. The Voice specific attributes for stored-messages are defined in 18.6. All general attribute types classified as "mandatory" in Table 1 of CCITT Rec. X.413 | ISO/IEC 10021-5 shall be supported.

18.2.1 *Mapping of an MHS message containing a VM into a VMGS-MS*

When a VM or VN gets delivered into the VMGS-MS, a corresponding MS entry is generated in the stored-messages information-base. The MS generates some attributes for administration purposes such as Sequence number, a Creation Time for the entry, the V-Bodypart Length, etc. It then generates attributes based on protocol elements in the MHS Envelope, in the Heading and one attribute containing the whole Voice object, e.g., the message. The attribute Voice Encoding Type signals which Voice Standard has been used to encode the message. Similarly, other Body Parts will be mapped into one or several additional attributes.

Note — In what follows, reference is made to an "MHS message". This is not be confused with the term "message", which refers to a VM.

Figure 7/X.440 describes how an MHS message with a VM is mapped into a corresponding MS entry.

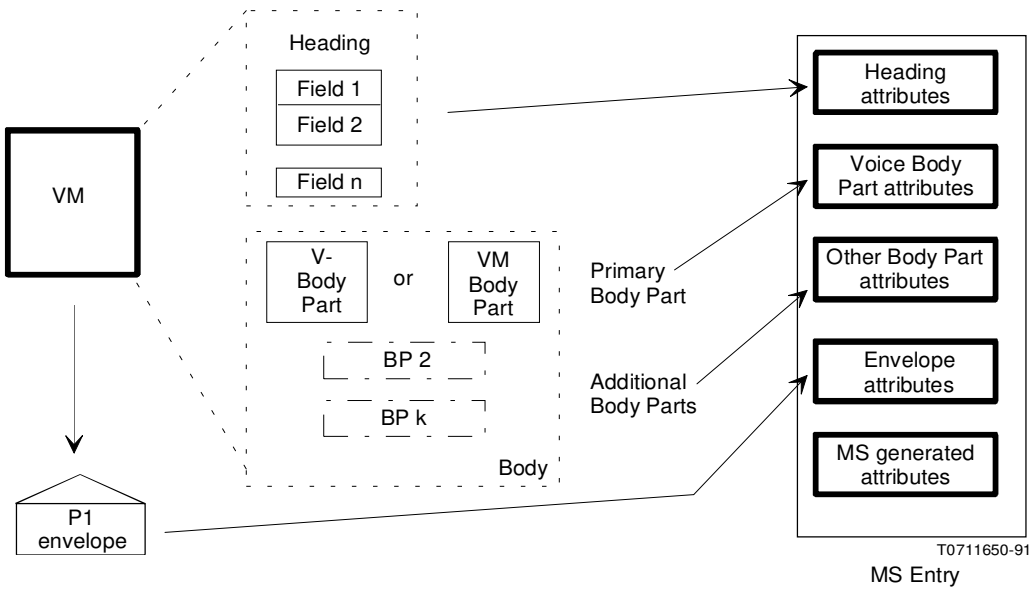

FIGURE 7/X.440

MHS message containing a VM mapped into a VMGS-MS

18.2.2 *Mapping of forwarded messages in MS*

A forwarded VM is mapped into the Message Store as one main entry and one or more linked child entries. The final child entry will contain the original VM (with its message and any additional body parts).

The MS Structure of a forwarded message such as the message in Figure 6/X.440 is depicted in Figure 8/X.440.

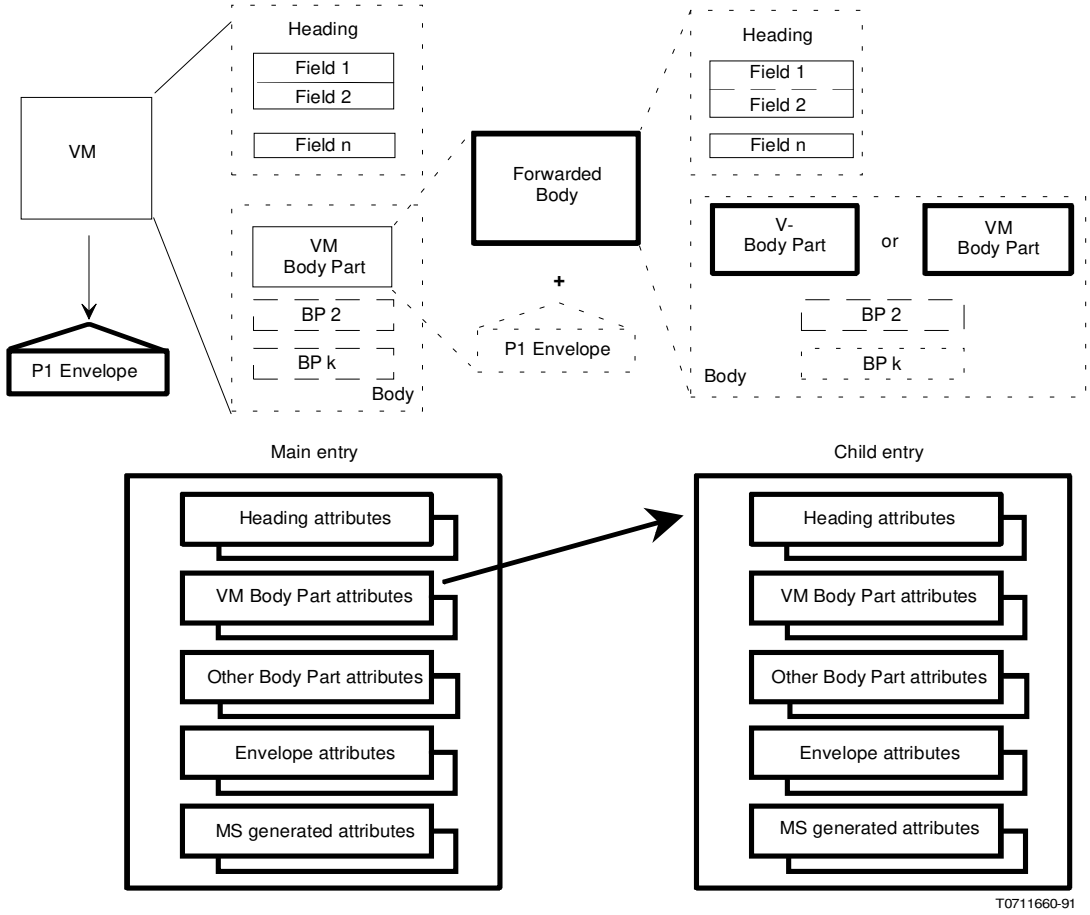

FIGURE 8/X.440

Forwarded message mapped into a MS

18.3 *Maintenance of attributes*

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

a) For each VM or VN it holds, the MS shall support the attributes as required in 18.6.

b) For each VM it holds, the MS shall give the following meanings to the defined values of the MS-status attribute:

1) **new**: No attribute values have been conveyed to the UA.

2) **listed**: At least one attribute value has been conveyed to the UA, and at least one body part value has not been conveyed to the UA.

3) **processed**: All body parts have been conveyed to the UA or the MS has performed an auto action and the definition of that auto action causes a change to the entry's status.

c) For each VN it holds, the MS shall give the following meanings to the defined values of the MS-status attribute:

1) **new**: No attribute values have been conveyed to the UA.

2) **listed**: At least one attribute value has been conveyed to the UA, and at least one attribute value has not been conveyed to the UA.

3) **processed**: All attributes have been conveyed to the UA or the MS has performed an auto action and the definition of that auto action causes a change to the entry's status.

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 a VM or VN that is delivered to the MS shall have as value the Object Identifier *id-mct-pvm* (see Annex A).

18.4 *Voice Messaging MS auto forward action types*

The concept of auto actions is described in 6.5 and 12 of CCITT Rec. X.413 | ISO/IEC 10021-5. This defines two general auto action types, which can potentially be used for all content-types. However, the "auto-forward" auto action defined there is not well suited for the VM content-type and its use for Voice messaging is discouraged. Instead, a specific auto action, called **vm-auto-forward-actions**, is defined below for VM Forwarding. Also, VM offers MS users a new auto action for the automatic generation of VN's Service Notificaton. This new auto action, called **vm-auto-sn-action**, is defined in 18.5.

The auto-alert auto action defined in 12.2 of CCITT Rec. X.413 | ISO/IEC 10021-5 can be used for Voice messaging without any restrictions.

Auto actions are registered and deregistered using the Register-MS abstract operation as described in 8.6 of CCITT Rec. X.413 | ISO/IEC 10021-5.

The *vm-auto-forward-actions* auto action is described in this clause. The operation of this auto action may be affected by the implementation of a security policy. Body Parts may not be added or removed through *vm-auto-forward-actions*.

The MS auto action, described below, together with its abstract syntax using the AUTO-ACTION macro is defined in 6.5 of CCITT Rec. X.413 | ISO/IEC 10021-5.

Note 1 — Substrings in filters cannot be defined for composite attributes (attributes with further ASN.1 structure in the attribute value) in the present version of CCITT Rec. X.413 | ISO/IEC 10021-5.

Note 2 — This version of this Recommendation does not provide for matching on voice encoded information objects.

The *vm-auto-forward-actions* allows the MS to perform VM Forwarding. The forwarding of VMs occurs in one of three ways:

a) the **vm-forwarding-with-NRN,** which means that the VM responsibility is forwarded and an NRN VN is returned to the originator. See a) of 18.4.1;

b) the **vm-forwarding-with-RN,** which means that the VM responsibility is accepted and an RN VN is returned to the originator. See b) of 18.4.2;

c) the **vm-forward-without-VNs,** which means that no VNs were requested and that no VN is returned to the originator. See a) of 18.4.3.

If Voice Security Requests are present, then the VM-auto-forward actions defined above may be prohibited, subject to the security policy in force. If Voice Security Requests are present then the VM-auto-forward action a) of 17.3.3.1 shall not be performed.

The *VM-auto-forward-actions* allows one or more sets of *VMActionRegistrationParameters* to be registered with the MS, each identified by its registration-identifier. Each *VMActionRegistrationParameter* value specifies criteria to determine whether it applies to a delivered VM, and if so, a copy of the message is VM-auto-forwarded and the status of the entry is set to processed, and an RN or NRN may be generated when requested, using the Message-submission abstract operation. In cases a) and b) above, a VN is also returned to the originator using the Message-submission abstract operation. The delivered VM may be automatically deleted afterwards. The ASN.1 definition of the **vm-auto-forward-actions** AUTO ACTION is as follows:

```
vm-auto-forward-actions AUTO-ACTION
    REGISTRATION PARAMETER IS VMActionRegistrationParameter
    ::= id-act-vm-auto-actions

VMActionRegistrationParameter ::= SEQUENCE {
    filter   [0] Filter OPTIONAL,
    vm-supplementary-information          [1] VMSupplementaryInformation OPTIONAL,
    delete-after-forwarding     [2] BOOLEAN DEFAULT FALSE,
    vm-auto-forwarding-mode     CHOICE {
        vm-forwarding-with-message-not-accepted     [3] ForwardWithNonReceipt,
        vm-forwarding-with-message-accepted         [4] ForwardWithMessageAccepted,
        vm-forwarding-with-no-VNs       [5] ForwardWithoutVNs } }
```

Note — The data types Filter, Per Message Auto Forward Fields and Per Recipient Auto Forward Fields are defined in 12.1 of CCITT Rec. X.413 | ISO/IEC 10021-5.

The common parameters of the Voice Forward Registration Parameter have the following meanings:

a) *Filter*: This is a set of criteria which a new entry representing a delivered VM shall satisfy for the MS abstract service provider to auto forward it using this set of parameters.

   The absence of this parameter indicates that all new entries are to be examined for potential auto actions, e.g. auto forwarding.

b) *Vm-supplementary-information*: This parameter can contain text to be included in the *other-parameters* field of a forwarded VM.

c) *Delete-after-forwarding*: This parameter indicates whether an MS entry should be deleted or not, once the auto-forward submission has succeeded. If not specified, no deletion takes place.

Note — If the message is deleted upon successful auto-forward submission and the forwarded message is non-delivered, it is not possible to reconstruct the entry in the MS. The F.400-Series of Recommendations discourages the use of the Return of Contents feature in a MTS Non-Delivery report. Thus the creation of a NRN may not be possible.

d) *VM-auto-forwarding-mode*: This is a choice between:

   1) **VM-forwarding-with-NRN,**

   2) **VM-forwarding-with-RN,**

   3) **VM-forwarding-without-notification.**

The remaining parameters are described separately below for these three cases.

18.4.1 *VM-Forwarding-with-NRN*

The *vm-forwarding-with-NRN* case enables the MS abstract service provider to automatically forward any VM to one or more recipients that has been delivered into the stored-messages information base and indicate to the originator of the subject VM that the message was not accepted. An assumption for this case is that a NRN has been requested.

The following limitations apply to *vm-forwarding-with-NRN*, when compared to the general rules for forwarding contained in 17.3.3:

a) The MS shall contruct and forward a VM whose primary body part comprises a body part of type VM body part as described in clause 8; the other-parameters component of the VM (Forwarded) Body Part will contain any registered supplementary information; the original delivery envelope shall be included, and the components of the original Heading shall be copied to the Heading of the submitted VM according to the rules in 17.3.3 with the following exceptions:

   1) the **recipient** parameter value is set to the next recipient;

   2) the **message-forwarded** parameter value is set to TRUE.

b) The *vm-forwarding-with-NRN* auto action type shall only be performed once for a particular VM by the same MS.

c) If VN requests are present in the subject VM, the MS shall set identical VN requests for only the first recipient named in the per-recipient-arguments field of this parameter. VN requests shall not be made of any other recipients.

If the VM Heading contains a request to return a Non-Receipt Notification (NRN), the MS generates an NRN according to the rules described in 17.3.3.6, using the Message-submission abstract operation. The value **nrn-sent** is added in the V Notification Indicator attributes.

The following ASN.1 data type defines the parameters specific to this case:

```
ForwardWithMessageNonReceipt ::= SET {
        COMPONENTS OF PerMessageAutoForwardFields, -- from MS auto-forward Abstract Operation
       per-recipient-no-accept-arguments       [3] SEQUENCE OF
       PerRecipientNoAcceptArguments,
        notification-argument       [4] NotificationArguments OPTIONAL }

PerRecipientNoAcceptArguments ::= SEQUENCE {
        per-recipient-field    [0] PerRecipientAutoForwardFields,
        heading-next-recipient       [1] RecipientField }

NotificationArguments ::= SET {
        COMPONENTS OF PerMessageAutoForwardFields, -- from MS auto-forward Abstract Operation
        per-recipients-field  [3] COMPONENTS OF per RecipientAutoForwardFields
                        WITH RecipientName ABSENT }
```

a) *PerMessageAutoForwardFields*

This is a set of arguments registered to be used for each message submission abstract operation (see 8.2.1.1.1 of CCITT Rec. X.411 | ISO/IEC 10021-4). Any argument which is not registered, not mandatory, and not specifically mentioned below, will be absent from each message submission.

If conversion-with-loss-prohibited is registered with the value "zero" indicating that conversion-with-loss is allowed [as defined in Figure 2/X.411 (1988) (part 21 of 41)] either by explicit registration of the value, or if it is not registered and thus assumes this value by default, the value used for each Message-submission abstract-operation shall be the value of the corresponding Message-delivery argument. If conversion-with-loss-prohibited is registered with the value "one" indicating that conversion-with-loss is prohibited, this value shall be used for each Message-submission abstract-operation.

If implicit-conversion-prohibited is registered with the value "zero", indicating that implicit-conversion is allowed, or if no value is registered, the value used for each Message-submission abstract-operation shall be the value of the corresponding Message-delivery argument. If implicit-conversion-prohibited is registered with the value "one", indicating that implicit-conversion is prohibited, this value shall be used for each Message-submission abstract-operation.

If the following arguments are not registered, their presence as message submission arguments depends upon the presence of the corresponding message delivery arguments, their values being transformed where appropriate: *content-confidentiality-algorithm-identifier*, *message-origin-authentication-check*, *message-security-label* and *priority*.

DL-expansion-prohibited shall have the fixed value DL-expansion-prohibited in the Message-submission abstract-operation.

Certain message submission arguments may not be registered. These are: *proof-of-submission-request*, *original-encoded-information-types* and *content-type*.

b) *PerRecipientNoAcceptArguments*

This is a set of arguments registered to be used for each message submission abstract operation (see 8.2.1.1.1 of CCITT Rec. X.411 | ISO/IEC 10021-4). Any argument which is not registered, not mandatory, and not specifically mentioned below, will be absent from each message submission.

If the following arguments are not registered, their presence as message submission arguments depends upon the presence of the corresponding message delivery arguments, their values being transformed where appropriate: *message-token* and *content-integrity-check*.

The following argument has a fixed value:

1) *originator-report-request:* This shall have either the value non-delivery-report or the value report.

Only one recipient is allowed for this case.

c) *Notification-argument*

This contains a subset of the parameters as in a) and b) above, but the actual values can differ from the values in the forwarded VM.

Notification-argument is used in the submission abstract operation of any VN generated by the MS.

Note — The Recipient Name component is not used because a VN may be sent to the originator of the subject VM or to the user identified in the VN-Receiver-field.

18.4.2 *VM-Forwarding-with-RN*

The *vm-forwarding-with-RN* case enables the MS abstract service provider to automatically forward to one or more preferred recipients, and automatically generate a RN if requested, any VM that has been delivered into the stored-messages information base. An assumption for this case is that a RN has been requested and that the MS has accepted responsibility for the subject VM even though it is being forwarded to another recipient.

The following limitations apply to *vm-forwarding-with-RN*, when compared to the general rules for forwarding contained in 17.3.3:

a) The MS shall contruct and forward a VM whose primary body part comprises a body part of type VM body part as described in 17.3.3.4; body parts may be added, the original delivery envelope shall be included, and the components of the original Heading shall be copied to the Heading of the submitted VM according to the rules in 17.3.3 with the following exceptions:

1) the **recipient** parameter value is set to the next recipient;

2) the **message-forwarded** parameter value is set to TRUE.

b) The *vm-forwarding-with-RN* auto action type shall only be performed once for a particular VM by the same MS.

c) At least one preferred recipient shall be specified for the forwarding auto action. Additional recipients may be specified. Setting of VN request bits is at the discretion of the user and the VN-Receiver-field is set to indicate that VNs are not returned to the originator of the subject VM.

If the VM Heading contains a request to return a Receipt Notification (RN) the MS generates an RN according to the rules described in 17.3.1.1, using the Message-submission abstract operation. The value ***rn-sent*** is added in the V Notification Indicator attributes.

The following ASN.1 data type defines the parameters specific to this case:

```
ForwardWithRespAccepted ::= SET {
        COMPONENTS OF PerMessageAutoForwardFields, -- from MS auto-forward Abstract Operation
        per-recipient-arguments    [3] SEQUENCE SIZE (1..ub-recipients) OF
                                       Per RecipientArguments, -- P1 + VMP information
        notification-argument      [4] NotificationArguments OPTIONAL -- RN arguments }

PerRecipientArguments ::= SEQUENCE {
        per-recipients-p1-info     [0] PerRecipientAutoForwardFields,
                                       -- P1 Per Recipient information
        vm-heading-fields    [1] HeadingFields OPTIONAL, -- VMP Per Recipient information
        new-vn-receiver-name       [2] ORName } -- for all VMP recipients

HeadingFields ::= SEQUENCE {
        next-recipient         [0] RecipientField,
        next-recipient-vn-requests-field        [1] VNotificationRequestsField OPTIONAL,
        next-message-forwarding-permitted   [2] MessageForwardingPermitted DEFAULT
                                       FALSE }
```

a) *PerMessageAutoForwardFields*

This is a set of arguments registered to be used for each message submission abstract operation (see 8.2.1.1.1 of CCITT Rec. X.411 | ISO/IEC 10021-4). Any argument which is not registered, not mandatory, and not specifically mentioned below, will be absent from each message submission.

If conversion-with-loss-prohibited is registered with the value “zero” indicating that conversion-with-loss is allowed [as defined in Figure 2/X.411 (1988) (part 21 of 41)], either by explicit registration of the value, or if it is not registered and thus assumes this value by default, the value used for each Message-submission abstract-operation shall be the value of the corresponding Message-delivery argument. If conversion-with-loss-prohibited is registered with the value “one” indicating that conversion-with-loss is prohibited, this value shall be used for each Message-submission abstract-operation.

If implicit-conversion-prohibited is registered with the value “zero”, indicating that implicit-conversion is allowed, or if no value is registered, the value used for each Message-submission abstract-operation shall be the value of the corresponding Message-delivery argument. If implicit-conversion-prohibited is registered with the value "one", indicating that implicit-conversion is prohibited, this value shall be used for each Message-submission abstract-operation.

If the following arguments are not registered, their presence as message submission arguments depends upon the presence of the corresponding message delivery arguments, their values being transformed where appropriate: *content-confidentiality-algorithm-identifier*, *message-origin-authentication-check*, *message-security-label* and *priority*.

DL-expansion-prohibited shall have the fixed value DL-expansion prohibited in the Message-submission abstract-operation.

Certain message submission arguments may not be registered. These are: *proof-of-submission-request*, *original-encoded-information-types* and *content-type*.

b) *PerRecipientArguments*

This set of arguments registers the values to be used for each recipient of a forwarded message. Messages are forwarded to one or more preferred recipients.

1) **PerRecipientAutoForwardFields**

This is a set of arguments registered to be used for each message submission abstract operation (see 8.2.1.1.1 of CCITT Rec. X.411 | ISO/IEC 10021-4). Any argument which is not registered, not mandatory, and not specifically mentioned below, will be absent from each message submission.

If the following arguments are not registered, their presence as message submission arguments depends upon the presence of the corresponding message delivery arguments, their values being transformed where appropriate: *message-token* and *content-integrity-check*.

The following argument has a fixed value:

— *originator-report-request*: this shall have either the value non-delivery-report or the value report.

— Multiple recipients are allowed for this case.

2) **New VN receiver name** (same for all recipients)

— *new-vn-receiver-name* to replace *vn-receiver-field* in heading.

3) **HeadingFields**:

— *next-recipient* to replace *recipient* in Recipients Sub Field. This field is mandatory.

— *next-recipient-vn-requests-field* to replace *vn-requests-field* in Recipients Sub Field. This field is optional.

— *next-message-forwarding-permitted* to replace *message-forwarding-permitted* in Recipients Sub Field. This field is optional.

c) *Notification-argument:* This contains a subset of the parameters as in a) and b) above, but the actual values can differ from the values in the forwarded VM.

Notification-argument is used in the submission abstract-operation of any VN generated by the MS.

Note — The Recipient Name component is not used because a VN may be sent to the originator of the subject VM or to the user identified in the VN-Receive-field.

18.4.3 *VM-Forwarding-without-notification*

The *vm-forwarding-without-notification* case enables the MS abstract service provider to automatically forward any VM that has been delivered into the stored-messages information base. An assumption for this case is that neither an NRN or a RN has been requested.

The following limitations apply to *vm-forwarding-without-notification*, when compared to the general rules for forwarding contained in 17.3.3:

a) The MS shall contruct and forward a VM whose primary body part comprises a body part of type VM body part as described in 8.2.2. The original delivery envelope may be included. Components of the original Heading shall be copied to the Heading of the submitted VM according to the rules in 18.4.2, to the forwarded VM with the following exceptions:

1) The *recipient* parameter value is set to the next recipient.

2) Any registered values for Heading fields shall replace the old values in the new Heading.

3) The **message-forwarded** parameter value is set to TRUE.

4) If VNs are requested of any recipient, then the *VN-receiver* field, as defined in 8.1.14, is set to indicate that VNs are not returned to the originator of the subject VM.

b) At least one preferred recipient shall be specified for the forwarding auto action. Additional recipients may be specified.

The following ASN.1 data type defines the parameters specific to this case:

```
ForwardWithoutVNs ::= SET {
    COMPONENTS OF PerMessageAutoForwardFields, -- from MS auto-forward Abstract Operation
    per-recipient-no-vn-arguments        [3] SEQUENCE SIZE (1..ub-recipients) OF
    PerRecipientArguments -- P1 + VM information }
```

a) **PerMessageAutoForwardFields**

The description is the same as in a) of 18.4.1;

b) **PerRecipientArguments**

The description is the same as in b) of 18.4.2;

Multiple recipients are allowed for this case.

18.5 *Voice messaging MS auto service notification action type*

VM offers MS users a new auto action for the automatic generation of an SN. This new auto action, called **vm-auto-sn-action** is defined below for interacting with a VMGS-MS. This is a new auto action that is not defined in CCITT Rec. X.413 | ISO/IEC 10021-5.

Note — VM's service notification provides the element of service E.15 defined in Recommendation F.440.

The *vm-auto-sn-action* allows one or more sets of *VNSNActionRegistrationParameters* to be registered with the MS; each is identified by its registration-identifier. Each instance of *VNSNActionRegistrationParameter* specifies criteria to be used by a VMGS-MS to determine whether it applies to a delivered VM. If so and the originator of the subject VM requests an SN, the MS is instructed to automatically generate an SN using the Message-submission abstract operation.

The service-notification case enables the MS abstract service provider to automatically generate a requested SN for any VM that has been delivered into the stored-messages information base. Once an SN is successfully generated by the MS, the value **sn-sent** is added in the V Notification Indicator attribute. The MS only generates a single SN for this named recipient. SNs shall be acted upon by a VMGS-MS before engaging in VM Forwarding.

Note — The delivery of a message into a VMGS-MS and generation of a SN is independent of VM Forwarding actions.

The following limitations apply to MS auto SN generation, when compared to the general rules for generating a SN contained in 17.3.2:

a) The MS shall construct a VN whose structure is a SN as described in 17.3.2. Components of the original Heading shall be copied to the Common Fields component of the SN according to the rules in 17.3 and 17.3.2.1, with the following exceptions:

— The "recipient" parameter value is set to the value of vn-receiver-name of the vn-receiver-field. If the vn-receiver-field is absent, then the parameter is set to the value of the originator-field.

The following AUTO-ACTION and ASN.1 data type defines the parameters specific to this case:

```
vm-auto-sn-action AUTO-ACTION
    REGISTRATION PARAMETER IS VNSNActionRegistrationParameter
    ::= id-act-vn-sn-auto-action

VNSNActionRegistrationParameter ::= SEQUENCE {
    filter   [0] Filter OPTIONAL,
    vn-service-notice-info      [1] ServiceNotificationInfo }
```

The *Filter* specifies the criteria upon which the MS is to invoke this auto-action. The data type *Filter* is defined in 12.1 of CCITT Rec. X.413 | ISO/IEC 10021-5.

The *ServiceNotificationInfo* to be registered has the following arguments:

a) **SN Notification Argument**

This argument registers the set of possible SN reasons that may be generated by an MS. Its syntax is that of the *SNReasonCode* field. The description is found in 9.3.2.

b) **VN Supplementary Info**

This argument registers the value to be used in VNSupplementaryInfo field for SNs generated automatically by the MS. Its syntax is that of the *VNSupplementaryInfo* field. The description is found in 9.1.10.

c) **Extensions**

This argument is used to pass additional elements of protocol not specified in this version of this Recommendation. The syntax of this field is that of *NotificationExtensionsField*. The description is found in 9.1.11.

```
ServiceNotificationInfo ::= SET {
    sn-notification-argument   [4] SNReasonCode,
    sn-supplementary-info      [5] VNSupplementaryInfo OPTIONAL,
    extensions    [6] NotificationExtensionsField OPTIONAL }
```

18.6 *Message store attributes*

As described in CCITT Rec. X.413 | ISO/IEC 10021-5, an MS maintains and provides access to certain attributes 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, for example, VM messaging kind.

The following clauses define the MS attributes specific to voice messaging. These clauses also stipulate if support for any VM specific attribute is mandatory or optional, see Table 1/X.440.

All of the attributes defined in this Recommendation, except those corresponding to extended body part types (which cannot be enumerated), are listed alphabetically, for reference, in the first column of Table 1/X.440. This table records their presence in a delivered message entry. None of them will appear in a delivered report entry. Additional, unnamed attributes are described in 18.6.4.5. Table 2/X.440 identifies the source of each attribute value and describes how the VM attributes are generated.

Note — See 5.2.1 and 5.2.2 for an elaboration of the legend of the tables.

TABLE 1/X.440 (Sheet 1 of 2)

Summary of the VM specific MS attribute types

Attribute	Single/ Multi valued	Support level by MS and UA	Presence in delivered VM	Presence in delivered RN	Presence in delivered NRN	Presence in delivered SN	Available for list, alert	Available for summarize
body	S	M	P	—	—	—	N	N
conversion-indication	S	M	—	C	C	C	Y	N
expiry-time	S	O	C	—	—	—	Y	N
externally-defined-body-part-types	M	O	C	—	—	—	Y	N
first-recipient	S	O	C	C	C	C	Y	N
heading	S	M	P	—	—	—	N	N
heading-extensions	M	M	C	—	—	—	Y	N
importance	S	O	C	—	—	—	Y	Y
language	M	O	C	—	—	—	Y	Y
message-data	S	O	C	—	—	—	N	N
message-parameters	S	O	C	—	—	—	N	N
notification-security-elements	S	O	—	C	C	C	Y	N
notification-creation-time	S	O	—	P	P	P	Y	N
notification-extensions	M	O	—	C	C	C	Y	N
nrn-extensions	M	O	—	—	C	—	Y	N
nrn-tsau-reason-code	M	O	—	—	C	—	Y	N
nrn-ua-ms-reason-code	M	O	—	—	C	—	Y	N
nrn-user-reason-code	M	O	—	—	C	—	Y	N
obsoleted-vm	S	O	C	—	—	—	Y	N
originator	S	O	C	—	—	—	Y	N
orig-vm-spoken-subject	S	O	—	C	C	C	Y	N
recipient-extensions-for-this-recipient	M	O	C	—	—	—	Y	N
rn-extensions	M	O	—	C	—	—	Y	N
sensitivity	S	O	C	—	—	—	Y	N
sn-extensions	M	O	—	—	—	P	Y	N
sn-reason-code	S	O	—	—	—	P	Y	N
subject-vm	S	M	—	P	P	P	Y	N
subject-vm-other-recipients	M	O	—	C	C	C	N	N
this-vm	S	M	P	—	—	—	Y	N
this-recipient	S	O	C	—	—	—	Y	N
v-body-message-length	S	M	—	—	—	—	N	Y
v-body-part	S	M	P	—	—	—	N	N

TABLE 1/X.440 (Sheet 2 of 2)

Attribute	Single/ Multi valued	Support level by MS and UA	Presence in delivered VM	Presence in delivered RN	Presence in delivered NRN	Presence in delivered SN	Available for list, alert	Available for summarize
vm-bodypart-sequence-number	S	M	—	—	—	—	N	N
vm-creation-time	S	M	C	—	—	—	Y	N
vm-encrypted-primary-bodypart	S	O	C	—	—	—	—	N
vm-entry-type	S	M	P	P	P	P	Y	Y
vm-forwarded-indication	S	M	C	—	—	—	Y	N
vm-forwarding-permitted	S	O	C	—	—	—	Y	N
vm-notification-indicator	S	O	—	—	—	—	Y	N
vm-notification-extension-requests-for-this-recipient	M	O	C	—	—	—	Y	N
vm-notification-requests-for-this-recipient	S	O	C	—	—	—	Y	N
vm-notification-security-requests-for-this-recipient	S	O	C	—	—	—	Y	N
vm-originator-spoken-name	S	O	C	—	—	—	Y	Y
vm-reception-security-requests-for-this-recipient	S	O	C	—	—	—	Y	N
vm-spoken-subject	S	O	C	—	—	—	Y	Y
vm-synopsis	S	O	P	—	—	—	N	N
vm-this-recipient-spoken-name	S	O	C	—	—	—	Y	Y
vmgs-user-security-element	S	O	C	—	—	—	Y	N
vmgs-user-security-extensions	M	O	C	—	—	—	Y	N
vn-initiator	S	O	—	P	P	P	Y	N
vn-originator-text	S	O	—	P	P	P	Y	N
vn-originator-spoken-name	M	O	—	C	C	C	Y	N
vn-receiver-field	S	O	C	—	—	—	Y	N
vn-supplementary-information	S	O	—	C	C	C	N	N
vn-voice-encoding-type	S	M	—	P	P	P	Y	Y
voice-encoding-type	S	M	C	—	—	—	Y	N
voice-message-duration	S	M	P	—	—	—	Y	Y
voice-message-other-parms	S	M	C	—	—	—	Y	Y

TABLE 2/X.440 (Sheet 1 of 3)

Generation of the VM specific MS attribute types

Attribute-type-name	Source parameters	Generated by	Generation rules
body	Body	MD	The value of the parameter is the attribute value.
conversion-indication	ConversionEITsField	MD	The value of the parameter is the attribute value when found in a VN
expiry-time	ExpiryTimeField	MD	The value of the parameter is the attribute value.
externally-defined-body-part-types	AdditionalBodyParts	MD	From each component of the SEQUENCE, one value is generated from the value of the ExternallyDefinedData components direct-reference and one is generated from the value of the ExternallyDefinedParameters components direct-reference, if present.
first-recipient	FirstRecipient	MD	The value of the parameter is the attribute value.
heading	Heading	MD	The value of the parameter is the attribute value.
heading-extensions	HeadingExtensionsSub Field	MD	A value is generated from each value of the SET.
importance	ImportanceField	MD	The value of the parameter is the attribute value.
language	Language	MD	The value of the parameter is the attribute value.
message-data	MessageData	MD	The value of the parameter is the attribute value.
message-parameters	MessageParameters	MD	The value of the parameter is the attribute value.
notification-creation-time	NoticeCreationTime Field	MD	The value of the parameter is the attribute value.
notification-extensions	NotificationExtensions SubField	MD	A value is generated from each value of the SET.
notification-security-elements	SecurityElements Field	MD	The value of the parameter is the attribute value.
nrn-extensions	NRNExtensionsSub Field	MD	A value is generated from each value of the SET when found in a VN.
nrn-tsau-reason-code	VNRNTSAUReason CodeField	MD	The value of the parameter is the attribute value when found in a VN.
nrn-ua-ms-reason-code	VNRNUAMSReason CodeField	MD	The value of the parameter is the attribute value when found in a VN.
nrn-user-reason-code	VNRNUSERReason CodeField	MD	The value of the parameter is the attribute value when found in a VN.
obsoleted-vm	ObsoletedVMFields	MD	A value is generated from each value of the SEQUENCE.
originator	Originator	MD	The value of the parameter is the attribute value.
orig-vmg-spoken-subject	VMSpokenSubject Field	MD	The value of the parameter is the attribute value when found in a VN.
recipient-extensions-for-this-recipient	RecipientExtensions	MD	A value is generated from each value of the SET in the recipient-sub-field for this recipient.
rn-extensions	RNExtensions	MD	A value is generated from each value of the SET when found in a VN.
sensitivity	SensitivityField	MD	The value of the parameter is the attribute value.
sn-extensions	SNExtensionsSub Field	MD	A value is generated from each member of the SET when found in a VN.

TABLE 2/X.440 (Sheet 2 of 3)

Attribute-type-name	Source parameters	Generated by	Generation rules
sn-reason-code	SNUAMSReasonBasic Code	MD	The value of the parameter is the attribute value when found in a VN.
subject-vm	SubjectVM	MD	The value of the parameter is the attribute value when found in a VN.
subject-vmg-other-recipients	RecipientField	MD	A value is generated from each member of the SET when found in a VN.
this-vm	ThisVM	MD	The value of the parameter is the attribute value.
this-recipient	Recipient	MD	The attribute-value is the value of the parameter in the recipient-sub-field for this recipient.
v-body-message-length	None	MS	The total length of the subject-VM
v-body-part	VBodyPart	MD	The value of the parameter is the attribute value.
vm-body-part	NONE	MS	The value is the sequence-number of the entry created for the forwarded VM.
vm-creation-time	VMCreationTime	MD	The value of the parameter is the attribute value when found in a VN.
vm-entry-type	NONE	MS	Provided by the MS when it creates the entry in the MS information base. If the information object is an VM, the value is set to “vm”. If the information object is a VN, the value is set according to the type of the VN, e.g., rn, sn, nrn.
vm-message-forwarded-indication	MessageForwarded	MD	The value of the parameter is the attribute value. If the source parameter is missing, an attribute with the default value shall be generated.
vm-forwarding-permitted	MessageForwarding Permitted	MD	The value of the parameter is the attribute value when found. If the source parameter is missing, an attribute with the default value shall be generated.
vm-notification-indicator	NONE	MS	A value is added when an VN is submitted from the MS. A default value of no notifications sent is set when the entry is created.
vm-notification-requests-for-this-recipient	VNotificationRequests	MD	The attribute-value is the value of the parameter in the recipient-sub-field for this recipient.
vm-notification-security-requests-for-this-recipient	VMNotification Security	MD	The attribute-value is the value of the parameter in the recipient-sub-field for this recipient.
vm-originator-spoken-name	OriginatorSpoken Name	MD	The value of the parameter is the attribute value when found in VN.
vm-reception-security-requests-for-this-recipient	VMReceptionSecurity	MD	The attribute-value is the value of the parameter in the recipient-sub-field for this recipient.
vm-spoken-subject	VMSpokenSubject Field	MD	The value of the parameter is the attribute value when found in VN.
vm-synopsis	see 18.6.1.2	MS	see 18.6.1.2
vm-this-recipient-spoken-name	RecipientSpoken Name	MD	The value of the parameter is the attribute value.
vmgs-user-security-element	VMGSUserSecurity Element	MD	The value of the parameter is the attribute value.

TABLE 2/X.440 (Sheet 3 of 3)

Attribute-type-name	Source parameters	Generated by	Generation rules
vm-user-security-extensions	VMGSUserSecurity Extensions	MD	A value is generated from each value of the SET.
vn-initiator	VNInitiator	MD	The value of the parameter is the attribute value.
vn-originator-spoken-name	OriginatorSpokenName	MD	The value of the parameter is the attribute value when found in a VN.
vn-originator-text	ORDescriptor	MD	The value of the parameter is the attribute value component of the VMOriginatorField found in the VN.
vn-receiver-field	VNReceiverField	MD	The value of the parameter is the attribute value.
vn-supplementary-information	VNSupplementary	MD	The value of the parameter is the attribute value when found in a VN.
vn-voice-encoding-type	VNVoiceEncodingType	MD	The value of the parameter is the attribute value when found in a VN.
voice-encoding-type	VoiceEncodingType	MD	The value of the parameter is the attribute value.
voice-message-duration	VMDuration	MD	The attribute-value is the value of the voice-message-duration component of the VoiceParameters field.
voice-message-other-parameters	VMSupplementary Information	MD	The attribute-value is the value of the other-parameters component of the VoiceParameters field.

18.6.1 *Summary attributes*

Some attributes summarize a Voice Messaging information object. These attributes are defined and described below.

18.6.1.1 *VM Entry Type*

The MS VM Entry Type attribute identifies an information object's type.

```
vm-entry-type ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX EntryType
        MATCHES FOR EQUALITY
        SINGLE VALUE
        ::= id-sat-vmg-entry-type

EntryType ::= ENUMERATED {
        voice-message (0),
        receipt-notification (1),
        non-receipt-notification (2),
        service-notification (3) }
```

This attribute may assume any one of the following values:

a) **voice-message**: The information object is a VM.

b) **receipt-notification**: The information object is a RN.

c) **non-receipt-notification**: The information object is an NRN.

d) **service-notification**: The information object is an SN.

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 a VM or VN.

18.6.1.2 *VM Synopsis*

The MS VM Synopsis attribute gives the structure, characteristics, size, and processing status of a VM at the granularity of individual body parts.

```
vm-synopsis ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX VMSynopsis
    SINGLE VALUE
    ::= id-sat-vm-synopsis
```

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

```
VMSynopsis ::= SEQUENCE OF BodyPartSynopsis
```

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

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

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

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

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

a) MS Number: The sequence number that the MS assigns to the entry that the Message body part represents.

b) MS Synopsis: The synopsis of the VM 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 it was so conveyed to the MS:

c) MS Type: The body part's extended type, i.e. the Direct-reference component of the body part's Data component.

d) MS Parameters: The body part's format and control parameters, i.e. the body part's Parameters component.

e) MS Size: The size in octets of the encoding component of the body part's Data component when the Basic Encoding Rules of CCITT Rec. X.209 | ISO/IEC 8825 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.

f) MS Processed (default *false*): An indication of whether the body part has been conveyed to the UA by means of the MS Fetch abstract operation. A default of *false* indicates that the body part has not been conveyed.

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 a VM.

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

18.6.2 *Voice notification indicator*

The Voice Notification Indicator attribute contains information about whether any Voice Notifications have been sent in response to a VM, and if so which type of Voice Notifications were sent. The MS creates this attribute for each new VM and maintains the attribute values, depending on the user invoked or auto actions performed.

```
vm-notification-indicator ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX VNotificationIndicator
        MATCHES FOR EQUALITY
        SINGLE VALUE
        ::=id-sat-vm-notification-indicator

VNotificationIndicator ::= VNIndicatorBits DEFAULT { }  -- default is NO Notifications Sent

VNIndicatorBits ::= BIT STRING {
        rn-sent         (0),
        sn-sent         (1),
        nrn-sent        (2) } (SIZE (1..ub-bit-options))
```

Each value of this attribute may assume one of the following values:

a) **rn-sent**: This value means that the MS has generated and sent a Receipt Notification (RN) in response to a request for a RN.

b) **sn-sent**: This value means that the MS has generated and sent a Service Notification (SN) in response to a request for an SN.

c) **nrn-sent**: This value means that the MS has generated and sent a Non-Receipt Notification (NRN) in response to a request for an NRN.

The MS ensures that when a VN is generated the appropriate bit shall be set correctly. The default for this attribute is no-notifications have been sent, i.e. all bits are set to zero.

18.6.3 *Heading attributes*

Some attributes are derived form the Heading of a VM. These attributes are defined and described below.

18.6.3.1 *Heading*

The Heading attribute is the (entire) Heading of a VM.

```
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 a VM.

18.6.3.2 *Heading fields*

Some attributes bear the names of Heading fields and have those fields as their values. Some attributes bear the names of Heading fields and have sub-fields of those fields as their values. See 8.1 for semantics.

```
this-vm ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX ThisVMField
        MATCHES FOR EQUALITY
        SINGLE VALUE
        ::= id-hat-this-VM

originator ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX COMPONENTS OF ORDescriptor
        MATCHES FOR EQUALITY
        SINGLE VALUE
        ::= id-hat-originator
```

obsoleted-vm ATTRIBUTE
WITH ATTRIBUTE-SYNTAX ObsoletedVMField
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-hat-obsoleted-vm

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

vm-forwarded-indication ATTRIBUTE
WITH ATTRIBUTE-SYNTAX MessageForwarded
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-hat-vm-forwarded-indication

vm-forwarding-permitted ATTRIBUTE
WITH ATTRIBUTE-SYNTAX MessageForwardingPermitted
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-hat-vm-forwarding-permitted

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

voice-encoding-type; ATTRIBUTE *-- Identifies the Encoding of the voice object*
WITH ATTRIBUTE-SYNTAX VoiceEncodingType
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-hat-voice-encoding-type

vm-creation-time ATTRIBUTE
WITH ATTRIBUTE-SYNTAX VMCreationTime
MATCHES FOR EQUALITY ORDERING
SINGLE VALUE
::= id-hat-vm-creation-time

vn-receiver-field ATTRIBUTE
WITH ATTRIBUTE-SYNTAX VNReceiverField
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-hat-vn-receiver

Voice Messaging user security element attributes from the originator of the subject message:

vmgs-user-security-element ATTRIBUTE
WITH ATTRIBUTE-SYNTAX VMGSUserSecurityElement
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-hat-vmgs-user-security-elements

vm-encrypted-primary-bodypart ATTRIBUTE
WITH ATTRIBUTE-SYNTAX BOOLEAN
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-hat-vm-encrypted-primary-bodypart

vmgs-user-security-extensions ATTRIBUTE
WITH ATTRIBUTE-SYNTAX VMGSUserSecurityExtensions
MATCHES FOR EQUALITY
MULTI VALUE
::= id-hat-vmgs-user-security-extensions

Voice encoded attributes from the originator of the subject message:

vm-spoken-subject ATTRIBUTE
WITH ATTRIBUTE-SYNTAX VMSpokenSubjectField
SINGLE VALUE
::= id-hat-vm-spoken-subject

vm-originator-spoken-name ATTRIBUTE
WITH ATTRIBUTE-SYNTAX OriginatorSpokenName
SINGLE VALUE
::= id-hat-vm-originator-spoken-name

vm-this-recipient-spoken-name ATTRIBUTE
WITH ATTRIBUTE-SYNTAX RecipientSpokenName
SINGLE VALUE
::= id-hat-vm-this-recipient-spoken-name

Heading extensions:

heading-extensions ATTRIBUTE
WITH ATTRIBUTE-SYNTAX HeadingExtensionsSubField
MATCHES FOR EQUALITY
MULTI VALUE
::= id-hat-heading-extensions

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 a VM whose Heading contains the field or sub-field whose name the attribute bears.

18.6.3.3 *Recipient sub-field*

Some attributes bear the names of Recipient fields and have sub-fields of those fields as their values. See 8.1.3 for semantics.

this-recipient ATTRIBUTE
WITH ATTRIBUTE-SYNTAX COMPONENTS OF ORDescriptor
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-rat-this-recipient

vm-notification-requests-for-this-recipient ATTRIBUTE
WITH ATTRIBUTE-SYNTAX VNotificationRequests
MATCHES FOR EQUALITY
MULTI VALUE
::= id-rat-vm-notification-requests-for-this-recipient

vm-notification-security-requests-for-this-recipient ATTRIBUTE
WITH ATTRIBUTE-SYNTAX VNotificationSecurity
MATCHES FOR EQUALITY
MULTI VALUE
::= id-rat-vm-notification-security-requests-for-this-recipient

vm-reception-security-requests-for-this-recipient ATTRIBUTE
WITH ATTRIBUTE-SYNTAX VMReceptionSecurity
MATCHES FOR EQUALITY
MULTI VALUE
::= id-rat-vm-reception-security-requests-for-this-recipient

vm-notification-extension-requests-for-this-recipient ATTRIBUTE
WITH ATTRIBUTE-SYNTAX VMNotificationExtensionsSubField
MATCHES FOR EQUALITY
MULTI VALUE
::= id-rat-vm-notification-extension-requests-for-this-recipient

Extensions specific to this recipient:

recipient-extensions-for-this-recipient ATTRIBUTE
WITH ATTRIBUTE-SYNTAX RecipientExtensionsSubField
MATCHES FOR EQUALITY
MULTI VALUE
::= id-rat-recipient-extensions-for-this-recipient

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 a VM whose Heading Recipients field contains the field whose name the attribute bears. It shall maintain one attribute value for each sub-field.

18.6.4 *Body attributes*

Some attributes are derived from the Body of a VM. These attributes are defined and described below.

18.6.4.1 *Body*

The Body attribute is the (entire) Body of a VM.

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 a VM.

18.6.4.2 *Body analyses*

Some attributes have as their values information about the body parts contained in the body of the message.

The voice message length attribute is created by the Message Store when it receives a VM. Its value indicates the length, in octets, of the subject Voice Body carried in the Primary Body Part of the message.

v-body-message-length ATTRIBUTE
WITH ATTRIBUTE-SYNTAX VoiceMessageLength
MATCHES FOR EQUALITY ORDERING
SINGLE VALUE
::= id-bat-v-body-message-length

VoiceMessageLength ::= INTEGER *-- in octets*

The VM Message Length gives the number of octets occupied by the voice encoded message.

The voice message duration attribute is extracted from the Parameters segment of the subject voice message body part. Its value indicates the duration, in seconds, of the encoded voice message.

voice-message-duration ATTRIBUTE
WITH ATTRIBUTE-SYNTAX VMDuration
MATCHES FOR ORDERING
SINGLE VALUE
::= id-bat-voice-message-duration

The other parameters attribute contains additional information that may be used by the recipient VMG-user when processing the voice encoded information.

voice-message-other-parms ATTRIBUTE
WITH ATTRIBUTE-SYNTAX VMSupplementaryInformation
MATCHES FOR EQUALITY SUBSTRINGS
SINGLE VALUE
::= id-bat-voice-message-other-parms

18.6.4.3 *Primary Body Parts*

Some attributes bear the names of the Primary Body Part types and have such body parts as their values. See clause 8 for semantics.

v-body-part ATTRIBUTE
WITH ATTRIBUTE-SYNTAX VBodyPart
SINGLE VALUE
::= id-bat-v-body-part

An MS holds each forwarded VM (i.e. each Message body part) as an information object in its own right, separate from the forwarding VM (stored as a separate child entry in the stored-messages information base). That information object is a message whose content is a VM. The VM Body Parts attribute below, therefore, has as its values the sequence numbers the MS assigns to those child entries. See 8.2.2 for semantics.

vm-bodypart-sequence-number ATTRIBUTE
WITH ATTRIBUTE-SYNTAX SequenceNumber *-- sequence number of the forwarded VM entry.*
SINGLE VALUE
::= id-bat-vm-bodypart-sequence-number

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 a VM. It shall maintain one attribute value for such a body part.

Some attributes bear the names of the Parameters and Data components of a VM Body Part and have the Parameters and Data components as their values.

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

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

An MS that supports these attributes shall maintain it for an information object that it holds if, and only if, that object is a message whose content is a VM whose Body contains a VM Body Part.

18.6.4.4 *VM Externally Defined Body Part Types*

The VM Externally Defined Body Part Types attribute identifies the externally defined body part types represented in a VM.

externally-defined-body-part-types ATTRIBUTE
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER
MATCHES FOR EQUALITY
MULTI VALUE
::= id-bat-externally-defined-body-part-types

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 a VM 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 type as specified in 7.3.12 of CCITT Rec. X.420 | ISO/IEC 10021-7.

Note — Some Externally Defined body part types are defined in CCITT Rec. X.420 | ISO/IEC 10021-7. Each value of this attribute identifies both an extended body part type represented in IPM or VM, and the attribute type generated for the data component(s) of the body part(s) of that body part type, as specified in C.3.6 in CCITT Rec. X.420 | ISO/IEC 10021-7. Although this Recommendation provides a mechanism for defining extended body parts in the future, no additional VM extension body parts are defined.

18.6.4.5 *Externally Defined Body Parts*

Some attributes have as their values the encoding components (see 8.2.3) or the ASN.1 Externals that constitute the Data components of Externally Defined Body Parts. To each externally defined body part type there are two corresponding attributes.

The first attribute is denoted by the object identifier that is the *direct-reference* component (again see 8.2.3) of the External which constitutes the *DATA* component of a Body Part of that Type. The first attribute contains only the Externally Defined Body Part which is encoded as a VMExternallyDefinedBodyPart (see 8.2.3).

The second attribute is denoted by the object identifier that is the *direct-reference* component of the External which constitutes the *PARAMETERS* component of a Body Part of that Type. The content of this second attribute is that Parameters component.

Where a Parameters type is defined for an extended body part type, the sequence of values in the attribute generated from the Data components of body parts of a given extended body part type corresponds to the sequence of values in the attribute generated from the Parameters components of the same body parts. Thus the value created for the Data component of a body part occupies the same position in the first attribute as the value created for the Parameters component occupies in the second attribute.

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

The Externally Defined Body Part Type attribute determines the Externally Defined Body Part attributes for a particular VM.

Note — Some Externally Defined body part types are defined in CCITT Rec. X.420 | ISO/IEC 10021-7. In principal the externally defined body part attributes can not be enumerated because the corresponding types can not be so enumerated.

18.6.5 *Notification attributes*

Some attributes are derived from a VN. These attributes are defined and described below.

18.6.5.1 *Common fields*

Some attributes bear the names of Common fields and have those fields as their values. See 6.1 for semantics.

```
subject-vm ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX SubjectVMField
    MATCHES FOR EQUALITY
    SINGLE VALUE
    ::= id-nat-subject-vm
```

vn-originator-text ATTRIBUTE
WITH ATTRIBUTE-SYNTAX COMPONENTS OF ORDescriptor *-- sub component of VNOriginatorField*
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-nat-vn-originator

vn-originator-spoken-name ATTRIBUTE
WITH ATTRIBUTE-SYNTAX OriginatorSpokenName
SINGLE VALUE
::= id-nat-vn-originator

vn-initiator ATTRIBUTE
WITH ATTRIBUTE-SYNTAX VNInitiatorField
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-nat-vn-initiator

first-recipient ATTRIBUTE
WITH ATTRIBUTE-SYNTAX FirstRecipientField
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-nat-first-recipient

notification-creation-time ATTRIBUTE
WITH ATTRIBUTE-SYNTAX NoticeCreationTimeField
MATCHES FOR EQUALITY ORDERING
SINGLE VALUE
::= id-nat-notification-creation-time

notification-security-elements ATTRIBUTE
WITH ATTRIBUTE-SYNTAX SecurityElementsField
SINGLE VALUE
::= id-nat-notification-security-elements

conversion-indication ATTRIBUTE
WITH ATTRIBUTE-SYNTAX ConversionEITsField
SINGLE VALUE
::= id-nat-conversion-indication

orig-vm-spoken-subject ATTRIBUTE
WITH ATTRIBUTE-SYNTAX VMSpokenSubjectField
SINGLE VALUE
::= id-nat-orig-vm-spoken-subject

subject-vm-other-recipients ATTRIBUTE
WITH ATTRIBUTE-SYNTAX RecipientField
MULTI VALUE
::= id-nat-subject-vm-other-recipients

vn-supplementary-info ATTRIBUTE
WITH ATTRIBUTE-SYNTAX VNSupplementaryInfo
SINGLE VALUE
::= id-nat-vn-supplementary-info

vn-voice-encoding-type ATTRIBUTE
WITH ATTRIBUTE-SYNTAX VNVoiceEncodingType
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-nat-vn-voice-encoding-type

Some attributes bear the names of notification fields and have sub-fields of the Common fields of a notification as their values.

notification-extensions ATTRIBUTE
WITH ATTRIBUTE-SYNTAX NotificationExtensionsSubField
MATCHES FOR EQUALITY
MULTI VALUE
::= id-nat-notification-extensions

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 a VN that contains the field or sub-field whose name the attribute bears.

18.6.5.2 *Receipt Notification fields*

Some attributes bear the names of RN VN fields and have those fields as their values. Some attributes bear the names of notification fields and have sub-fields of the RN fields of a notification as their values. Other than the VN common attributes and rn-extensions field, no other attributes are defined for VM. See 9.2 for semantics.

rn-extensions ATTRIBUTE
WITH ATTRIBUTE-SYNTAX RNExtensionsSubField
MATCHES FOR EQUALITY
MULTI VALUE
::= id-nat-rn-extensions

Note — No RN extensions are defined in this version of this Recommendation.

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 a RN that contains the field whose name the attribute bears. It shall maintain one attribute value for each field or sub-field.

18.6.5.3 *Service Notification fields*

Some attributes bear the names of SN VN fields and have those fields as their values. Some attributes bear the names of notification fields and have sub-fields of the SN fields of a notification as their values. See 9.3 for semantics.

sn-reason-code ATTRIBUTE
WITH ATTRIBUTE-SYNTAX SNReasonCode
SINGLE VALUE
::= id-nat-sn-ua-ms-basic-reason-code

SN extensions are defined as follows:

sn-extensions ATTRIBUTE
WITH ATTRIBUTE-SYNTAX SNExtensionsSubField
MATCHES FOR EQUALITY
MULTI VALUE
::= id-nat-sn-extensions

Note — No SN extensions are defined in this version of this Recommendation.

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 SN that contains the field whose name the attribute bears. It shall maintain one attribute value for each field or sub-field.

18.6.5.4 *Non-Receipt Notification fields*

Some attributes bear the names of NRN VN fields and have those fields as their values. Some attributes bear the names of notification fields and have sub-fields of the NRN fields of a notification as their values. See 9.4 for semantics.

nrn-ua-ms-reason-code ATTRIBUTE
WITH ATTRIBUTE-SYNTAX VNRNUAMSReasonCodeField
SINGLE VALUE
::= id-nat-nrn-ua-ms-reason-basic-code

nrn-user-reason-code ATTRIBUTE
WITH ATTRIBUTE-SYNTAX VNRNUserReasonCodeField
SINGLE VALUE
::= id-nat-nrn-user-reason-basic-code

nrn-tsau-reason-code ATTRIBUTE
WITH ATTRIBUTE-SYNTAX VNRNTSAUReasonCodeField
SINGLE VALUE
::= id-nat-nrn-tsau-reason-code

NRN extensions are defined as follows:

nrn-extensions ATTRIBUTE
WITH ATTRIBUTE-SYNTAX NRNExtensionsSubField
MATCHES FOR EQUALITY
MULTI VALUE
::= id-nat-nrn-extensions

Note — No NRN extensions are defined in this version of this Recommendation.

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 NRN that contains the field whose name the attribute bears. It shall maintain one attribute value for each field or sub-field.

18.7 *Procedures for Voice Messaging MS*

The procedures for a general MS are specified in clauses 14 and 15 of CCITT Rec. X.413 | ISO/IEC 10021-5. This reference gives complementary information for MS systems that also explicitly support VMG.

18.7.1 *Additional procedures for message delivery*

How the MS consumes the MTS abstract service is described in clause 14 of CCITT Rec. X.413 | ISO/IEC 10021-5. The following text describes additional information about the procedures needed for Voice messaging.

Addition to 14.1.1 item 2) a) of CCITT Rec. X.413 | ISO/IEC 10021-5:

— If auto forwarding criteria are registered by the Register-MS abstract operation, the new entry shall be matched against the criteria specified. The matching shall always start with any registrations against the "vm-forward-with-NRN" and "forwarding-without-notifications" auto actions. If this results in forwarding being performed, no further forwarding with-NRN shall be performed by the same MS. Further, registered criteria against "vm-forward-with-RN" shall be examined, possibly resulting in one or several forwardings being performed. Requested notifications shall be returned for each forwarding that is performed. If a forwarding does not succeed, e.g. through a non-delivery, a NRN notification may be returned to the originator. When a VN is submitted, a value reflecting the type of VN shall be added to the "vm-notification-indicator" attribute.

19 Message contents

As has already been seen, various secondary objects (e.g. UAs) have occasion to convey the VMG Information Objects of clause 6 as the contents of messages. What follows 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, constitute Voice Messaging.

19.1 *Content*

A secondary object that submits a message containing a VM or VN shall supply, as the octets of the Octet String that constitutes the content of the message, the result of encoding the Information Object of clause 6 in accordance with the Basic Encoding Rules of CCITT Rec. X.209 | ISO/IEC 8825.

19.2 *Content type*

A secondary object that submits a message containing a VM or VN shall assign the integer "40" to the Content Type. The object identifier *id-mct-pvm* defined in Annex A is provided for use with a VMGS-MS. The integer "40" shall be used if the MTA conforms to CCITT Rec. X.411 | ISO/IEC 10021-4.

A secondary object that receives a message containing a VM or VN shall accept the integer "40" as the P1 Content Type.

Note — Future versions of this Recommendation may require support for the object identifier id-mct-pvm identifying the Content Type.

19.3 *Content length*

A secondary object, submitting a message containing a VM or VN, shall specify as the length of the message's content the size in octets of the encoding of the instance in question of the Information Object of clause 6 (a choice of a VM or a VN) when the Basic Encoding Rules of CCITT Rec. X.209 | ISO/IEC 8825 are followed. If those rules permit several (e.g. both primitive and constructed) encodings of that Information Object, the content length may reflect any one of them.

19.4 *Encoded information types*

A secondary object that submits a message containing a VM or VN shall specify the Encoded Information Types (EIT) of the message as follows.

In the case of a VN, the basic EITs shall be unspecified.

In the case of a VM, the EITs shall be the logical union of the EITs of the VM's body parts specified in accordance with the following rules:

a) Voice Body Part: The EIT (if any) of the Voice Body Part shall have the same values as the Heading field Voice Body Part Type.

b) VM Body Part(Forwarded Message): The EITs (if any) of a VM Body Part shall be those of the forwarded message.

c) Additional body parts: The EIT of additional body parts (if any) shall be the logical union of the individual body parts EITs.

An Externally Defined body part whose extended type corresponds to a basic type shall be indicated using the built-in EIT.

The Voice Body Part Type may be indicated in the external EITs.

A secondary object that submits a message containing a VM or VN to an MTA that conforms to CCITT Rec. X.411 | ISO/IEC 10021-4 in its 1988 version shall use the object identifier *id-mct-pvm* (see Annex A) for all "original-encoded-information-types".

20 Port realization

The manner in which an MS or the MTS concretely realizes the secondary ports it supplies is specified in CCITT Rec. X.419 | ISO/IEC 10021-6.

The manner in which a UA, MS, or AU concretely realizes the primary ports it supplies is beyond the scope of this Recommendation.

21 Conformance

The requirements that 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 distinguishs between support upon origination and support upon reception.

21.1 *Origination versus reception*

A UA or AU shall be said to support upon origination a particular Heading field, Heading extension, VM Body Part type or Externally Defined Body Part type if, and only if, the UA or AU accepts, preserves, and emits, in full accord with this Recommendation, that particular Heading field, Heading extension, VM Body type or Externally Defined Body Part type, whenever a user calls upon the UA or AU to convey a VM or VN containing them to the MTS or the user's MS (the latter only in the case of a UA).

A UA or AU shall be said to support upon reception a particular Heading field, Heading extension, VM Body Part type or Externally Defined Body Part type if, and only if, the UA or AU accepts, preserves, and emits, in full accord with this Recommendation, that particular Heading field, Heading extension, VM Body Part type or Externally Defined Body Part type, whenever the MTS or a user's MS (the latter only in the case of a UA) calls upon the UA or AU to convey to the user a VM or VN containing them.

21.2 *Requirements of a VMG conformance statement*

The implementor of a VM application entity (UA, MS or AU) shall separately state the following for both origination and reception. This may be done by including a completed VM PICS in accordance with the VM PICS-proforma contained in Recommendation X.485.

a) The Heading fields for which it claims conformance.

b) The body part types for which it claims conformance.

c) In the case of an MS or a UA with MS, the Voice Messaging-specific MS attributes for which it claims conformance.

d) In the case of an MS or a UA with MS, whether it supports the Voice messaging-specific auto actions.

21.2.1 *Static requirements*

A UA, MS or AU shall satisfy the following static requirements:

a) A UA, MS or AU shall implement the Heading fields and the body part types for which conformance is claimed.

b) An MS or UA with MS shall support the Voice messaging-specific MS attribute-types for which conformance is claimed, but including as a minimum those designated mandatory in 18.6. In addition, it shall support the mandatory attributes identified in Table 1 of CCITT Rec. X.413 | ISO/IEC 10021-5.

c) A UA, MS or AU shall concretely realize its abstract ports as specified in clause 20.

d) A UA or MS shall be able to both submit and receive messages of the content types of 19.2. An AU shall be able to both import and export such messages.

e) An MS, or UA accessing an MS, shall conform to at least one of the MS Access protocols specified in CCITT Rec. X.419 | ISO/IEC 10021-6.

21.2.2 *Dynamic requirements*

A VM application entity (UA, MS or AU) shall satisfy the following dynamic requirements:

a) A UA or MS shall follow the rules of operation specified in subclause 15.1 or clause 18, respectively.

b) A UA, MS or AU shall submit and receive messages whose contents are as specified in clause 19.

c) A UA, MS or AU shall register with the MTS its ability to accept delivery of messages of both of the VMG content type of 19.2 and EITs as specified in 19.4.

ANNEX A

(to Recommendation X.440)

VM Object Identifiers — Reference definition

(This annex forms 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 which this Recommendation assigns are assigned in this annex. The annex is definitive for all but those for ASN.1 modules, the object VMG application (VMGE) itself and the VMGS' use of Directories. The definitive assignments for the former occur in the modules themselves; other references to them appear in IMPORT statements. For the VMGS' use of Directories object identifiers, this annex only defines a base object identifier.

```
VMGSObjectIdentifiers { joint-iso-ccitt
        mhs-motis(6) vmgs(8) modules(9) object-identifiers(0) }
        DEFINITIONS IMPLICIT TAGS ::=
        BEGIN

-- Prologue

-- Exports everything

IMPORTS  --  nothing --;
        ID ::= OBJECT IDENTIFIER

-- VM Messaging (definitive)

        id-vmgs  ID ::= { joint-iso-ccitt mhs-motis(6) vmgs(8) } -- This is definitive

-- Categories

        id-mod      ID ::= {id-vmgs  0} -- vm auto-action indentifier types
        id-act      ID ::= {id-vmgs  1} -- vm auto-action indentifier types
        id-bat      ID ::= {id-vmgs  2} -- body attributes
        id-bp       ID ::= {id-vmgs  3} -- vm body part types
        id-dir      ID ::= {id-vmgs  4} -- vm use of directories
        id-for      ID ::= {id-vmgs  5} -- vm action indicator types
        id-hat      ID ::= {id-vmgs  6} -- heading attributes
        id-ipe      ID ::= {id-vmgs  7} -- ipm's vm extensions
        id-mct      ID ::= {id-vmgs  8} -- message content types
        id-mod      ID ::= {id-vmgs  9} -- modules
        id-nat      ID ::= {id-vmgs 10} -- notification attributes
        id-nt       ID ::= {id-vmgs 11} -- vm notification types
        id-ot       ID ::= {id-vmgs 12} -- object types
        id-pt       ID ::= {id-vmgs 13} -- port types
        id-rat      ID ::= {id-vmgs 14} -- recipient attributes
        id-ref      ID ::= {id-vmgs 15} -- refinements
        id-sat      ID ::= {id-vmgs 16} -- summary attributes
        id-syn      ID ::= {id-vmgs 17} -- vm synopsis type
        id-vmg      ID ::= {id-vmgs 18} -- reserved

-- Modules

        id-mod-object-identifiers   ID ::= {id-mod  0}
        id-mod-functional-objects   ID ::= {id-mod  1}
        id-mod-information-objects          ID ::= {id-mod  2}
        id-mod-abstract-service     ID ::= {id-mod  3}
        id-mod-message-store-attributes ID ::= {id-mod  4}
        id-mod-upper-bounds         ID ::= {id-mod  5}
        id-mod-vmg-directory-cl-att         ID ::= {id-mod  6}
        id-mod-message-store-auto-actions                   ID ::= {id-mod  7}
```

-- Object types

```
id-ot-vmge    ID ::= {id-ot 0}
id-ot-vmgs-user      ID ::= {id-ot 1}
id-ot-vmgs    ID ::= {id-ot 2}
id-ot-vmg-ua ID ::= {id-ot 3}
id-ot-vmg-ms         ID ::= {id-ot 4}
id-ot-tsau     ID ::= {id-ot 5}
```

-- 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}
```

-- VM-Notification Types (for use in P1 notification extension field)

```
id-nt-vmg-rn  ID ::= {id-nt 0}
id-nt-vmg-sn ID ::= {id-nt 1}
id-nt-vmg-nrn         ID ::= {id-nt 2}
```

-- Message content type

```
id-mct-pvm    ID ::= {id-mct 0} -- Pvm
```

-- VM Body Part type (and P1 EIT)

```
id-bp-g721-32k-adpcm       ID ::= {id-bp 0} -- Recommendation G.721 32K ADPCM
                                               -- encoding
id-bp-private-octet   ID ::= {id-bp 1}
id-bp-undefined-octet       ID ::= {id-bp 2}
```

-- VMGS Specific Register Auto Actions

```
id-act-vmg-auto-forward    ID ::= {id-act 0}
id-act-vn-sn-auto-action    ID ::= {id-act 1}
```

-- VM Synopsis (MS)

```
id-syn-removed       ID ::= {id-syn 0}
id-syn-place-holder ID ::= {id-syn 1}
```

-- MESSAGE STORE ATTRIBUTES

-- Summary attributes

```
id-sat-vmg-entry-type       ID ::= {id-sat 0}
id-sat-vm-synopsis  ID ::= {id-sat 1}
id-sat-vm-notification-indicator    ID ::= {id-sat 2}
```

-- Heading attributes

```
id-hat-heading        ID ::= {id-hat 0}
id-hat-this-vm        ID ::= {id-hat 1}
id-hat-originator     ID ::= {id-hat 2}
id-hat-vm-originator-spoken-name                  ID ::= {id-hat 3}
id-hat-vm-this-recipient-spoken-name              ID ::= {id-hat 4}
id-hat-vm-creation-time     ID ::= {id-hat 5}
id-hat-obsoleted-vm         ID ::= {id-hat 6}
id-hat-vm-spoken-subject  ID ::= {id-hat 7}
id-hat-expiry-time    ID ::= {id-hat 8}
id-hat-importance    ID ::= {id-hat 9}
```

```
id-hat-sensitivity-field        ID ::= {id-hat 10}
id-hat-vm-forwarded-indication  ID ::= {id-hat 11}
id-hat-language         ID ::= {id-hat 12}
id-hat-vm-forwarding-permitted  ID ::= {id-hat 13}
id-hat-voice-encoding-type      ID ::= {id-hat 14}
id-hat-vm-creation-time     ID ::= {id-hat 15}
id-hat-vm-encrypted-primary     ID ::= {id-hat 16}
id-hat-vmgs-user-security ID ::= {id-hat 17}
id-hat-vmgs-user-security-extensions            ID ::= {id-hat 18}
id-hat-vn-receiver  ID ::= {id-hat 19}
id-hat-heading-extensions ID ::= {id-hat 20}

-- Per Recipient attributes

id-rat-this-recipient           ID ::= {id-rat 0}
id-rat-vm-notification-extension-requests-for-this-recipient  ID ::= {id-rat 1}
id-rat-vm-notification-requests-for-this-recipient            ID ::= {id-rat 2}
id-rat-vm-notification-security-requests-for-this-recipientID ::= {id-rat 3}
id-rat-vm-reception-security-requests-for-this-recipient  ID ::= {id-rat 4}
id-rat-recipient-extensions-for-this-recipient            ID ::= {id-rat 5}

-- Body attributes

id-bat-body  ID ::= {id-bat 0}
   -- Originator's subject-VM Bodypart
id-bat-voice-message-duration   ID ::= {id-bat 1}
id-bat-voice-message-other-parms            ID ::= {id-bat 2}
id-bat-v-body-message-length    ID ::= {id-bat 3}
id-bat-v-body-part  ID ::= {id-bat 4}
   -- Forwarded VM Bodypart or Additional Bodyparts
id-bat-vm-bodypart-sequence-number          ID ::= {id-bat 5}
id-bat-message-parameters       ID ::= {id-bat 6}
id-bat-message-data         ID ::= {id-bat 7}
id-bat-externally-defined-body-part-types       ID ::= {id-bat 8}
id-bat-ipm-parameters       ID ::= {id-bat 9}
id-bat-ipm-data         ID ::= {id-bat 10}

-- Notification attributes

id-nat-subject-vm   ID ::= {id-nat 0}
id-nat-vn-originator ID ::= {id-nat 1}
id-nat-subject-vm-other-recipients          ID ::= {id-nat 2}
id-nat-first-recipientID ::= {id-nat 3}
id-nat-notification-creation-time  ID ::= {id-nat 4}
id-nat-notification-security-elements       ID ::= {id-nat 5}
id-nat-notification-extensions      ID ::= {id-nat 6}
id-nat-conversion-indication        ID ::= {id-nat 7}
id-nat-vn-initiator     ID ::= {id-nat 8}
id-nat-orig-vmg-spoken-subject  ID ::= {id-nat 9}
id-nat-vn-originator-spoken-name            ID ::= {id-nat 10}
id-nat-vn-originator-text   ID ::= {id-nat 11}
id-nat-vn-voice-encoding-type   ID ::= {id-nat 12}
id-nat-vn-supplementary-information         ID ::= {id-nat 13}
   -- RN attributes
id-nat-rn-extensions        ID ::= {id-nat 14}
   -- SN attributes
id-nat-sn-extensions        ID ::= {id-nat 15}
id-nat-sn-ua-ms-basic-reason-code           ID ::= {id-nat 16}
   -- NRN attributes
id-nat-nrn-extensions       ID ::= {id-nat 17}
id-nat-nrn-tsau-reason-code     ID ::= {id-nat 18}
id-nat-nrn-ua-ms-reason-basic-code          ID ::= {id-nat 19}
id-nat-nrn-user-reason-basic-code           ID ::= {id-nat 20}

-- MESSAGE STORE ATTRIBUTES — END

END -- of VMGSObjectIdentifiers
```

ANNEX B

(to Recommendation X.440)

VM Abstract Information Objects — Reference definition

(This annex forms an integral part of this Recommendation.)

This annex defines for reference purposes the abstract information objects of Voice Messaging. It defines a Body Part for VMGS that includes a body part reference number while importing the IPM externally defined MACRO for specifying non-VM body parts. It also defines a VM-EXTENSION macro.

```
VMGSInformationObjects { joint-iso-ccitt
        mhs-motis(6) vmgs(8) modules(9) information-objects(2) }

    DEFINITIONS IMPLICIT TAGS ::=
    BEGIN

-- Prologue

-- Exports everything

    IMPORTS

-- VMGS Upper bounds

    ub-vmg-local-reference, ub-vn-reason-code, ub-vmgs-spoken-name,
        ub-vmg-spoken-subject, ub-vmgs-user-security-elements, ub-vmg-spoken-
        supplemental-info
        ----
        FROM VMGSUpperBounds { joint-iso-ccitt mhs-motis(6) vmgs(8) modules(9) upper-
        bounds(5) }

-- VMGS Object Identifiers

    id-vmg-g721-32k-adpcm
        -----
        FROM VMGSObjectIdentifiers { joint-iso-ccitt mhs-motis(6) vmgs(8) modules(9) object-
        identifiers(0) }

--  MTS Upper Bounds

    ub-bit-options, ub-integer-options, ub-recipients, ub-supplementary-info-length
        ----
        FROM MTSUpperBounds { joint-iso-ccitt mhs-motis(6) mts(3) modules(0) upper-bounds(3) }

-- MTS Abstract Service

    Content, ContentIntegrityCheck, MessageDeliveryTime, ORAddress, ORName,
        OtherMessageDeliveryFields
        ----
        FROM MTSAbstractService { joint-iso-ccitt mhs-motis(6) mts(3) modules(0) mts-abstract
        service(1) }

-- IPM Abstract Service

    EXTENDED-BODY-PART-TYPE, ExternallyDefinedBodyPart, Sensitivity,
        ImportanceField, ORDescriptor, ConversionEITsField
        ----
        FROM IPMSInformationObjects { joint-iso-ccitt mhs-motis(6) ipms(1) modules(0)
        information-objects(2) }

-- IPM Heading Extensions

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

```
-- END Imports

-- ABSTRACT INFORMATION OBJECTS

-- Overview

    InformationObject ::= CHOICE {
        vm          [0] VM,
        vn          [1] VN }

-- Common data types

-- Extensions

    ExtensionField ::= SEQUENCE {
        type    [0] VM-EXTENSION,
        criticality     [1] Criticality DEFAULT FALSE,
        value   [2] ANY DEFINED BY type DEFAULT NULL NULL }

    Criticality ::= BOOLEAN

-- VMGS Extension MACRO

    VM-EXTENSION MACRO ::=
        BEGIN

        TYPE NOTATION     ::= DataType Critical | empty
        VALUE NOTATION    ::= value(VALUE OBJECT IDENTIFIER)

        DataType      ::= type (X) Default
        Default       ::= "DEFAULT" value (X) | empty
        Critical      ::= "CRITICAL" | empty

        END -- of extension

-- VM Messages

    VM ::= SEQUENCE {
        heading     Heading,
        body    Body }

-- Heading Field Component Types

-- VM Identifier

    VMIdentifier ::= SET {
        user    [0] ORName,
        user-relative-reference     [1] LocalReference }

-- Local Reference

    LocalReference ::= PrintableString (SIZE (0..ub-vmg-local-reference))

-- Originator

    VMOriginatorField ::= SET {
        COMPONENTS OF ORDescriptor,  -- from IPM
        originator-spoken-name    [3] SpokenName OPTIONAL }

-- Spoken Name

    SpokenName ::= ANY      -- defined by VoiceEncodingType or VNVoiceEncodingType and not to exceed
                            -- 20 seconds in duration

-- Recipient

    RecipientField ::= SEQUENCE {
        recipient-name      COMPONENTS OF ORDescriptor,  -- from IPM
        spoken-name         [2] SpokenName OPTIONAL }
```

```
-- Heading Fields

    Heading ::= SEQUENCE {
    this-VM         ThisVMField,
    originator      [0]  VMOriginatorField OPTIONAL,
    recipients      [1]  RecipientsField OPTIONAL,
    obsoleted-vm          [2]  ObsoletedVMField OPTIONAL,
    vm-subject      [3]  VMSpokenSubjectField OPTIONAL,
    expiry-time     [4]  ExpiryTimeField OPTIONAL,
    importance      [5]  ImportanceField DEFAULT {normal},  -- from IPM
    sensitivity     [6]  SensitivityField OPTIONAL,  -- from IPM
    message-forwarded [7]  MessageForwarded DEFAULT FALSE,
    vm-forwarding-permitted   [8] MessageForwardingPermitted DEFAULT TRUE,
    language        [9] LanguageField OPTIONAL,
    voice-encoding-type          [10] VoiceEncodingType DEFAULT { id-vmg-g721-32k-
                                 adpcm },
    vm-creation-time      [11] VMCreationTime OPTIONAL,
    vn-receiver-field       [12] VNReceiverField OPTIONAL,
    vmgs-user-security-elements       [13] VMGSUserSecurityElementsField OPTIONAL,
    heading-extensions [14] HeadingExtensionsField OPTIONAL }

-- This VM field

    ThisVMField ::= VMIdentifier

-- Recipients

    RecipientsField ::= SET OF RecipientsSubField SIZE (1..ub-recipients)

    RecipientsSubField ::= SEQUENCE {
        recipient       [0] RecipientField,
        vn-requests-field     [1] VNotificationRequestsField,
        attendant-assisted-delivery-request     [2] AttendantAssistedDeliveryRequest
                            DEFAULT { },  -- default NONE
        recipient-extensions          [3] RecipientExtensionsField OPTIONAL }

-- VMGS Notification Requests

    VNotificationRequestsField ::= SEQUENCE {
        vm-notification-requests    [0] VNotificationRequests DEFAULT { },  -- default NONE
        vm-notification-security-requests         [1] VNotificationSecurity DEFAULT { },  -- default NONE
        vm-reception-security-requests            [2] VMReceptionSecurity DEFAULT { },  -- default NONE
        vm-notification-extension-requests      [3] VMNotificationExtensions DEFAULT { } }  -- default NONE

    VNotificationRequests ::= BIT STRING {
        receipt-notice          (0),
        service-notice          (1),
        non-receipt-notice    (2) }(SIZE (0..ub-bit-options))

    VNotificationSecurity ::= BIT STRING {
        proof (0),
       non-repudiation (1) } (SIZE (0..ub-bit-options))   -- Note — These requests are for further study per
                    -- Recommendation F.440.

    VMReceptionSecurity ::= BIT STRING {
        proof (0),
       non-repudiation (1) } (SIZE (0..ub-bit-options))   -- Note — These requests are for further study per
                    -- Recommendation F.440.

    VMNotificationExtensions ::= SET OF VMNotificationExtensionsSubField

    VMNotificationExtensionsSubField ::= ExtensionField
```

```
-- Attendant Assisted Delivery Request Indication

    AttendantAssistedDeliveryRequest ::= INTEGER {
        person-to-person (0),
        anyone (1) }

-- Recipient Extensions

    RecipientExtensionsField ::= SET OF RecipientExtensionsSubField

    RecipientExtensionsSubField ::= ExtensionField

-- Auto Forwarded indication

    MessageForwarded ::= BOOLEAN -- Default False

-- Voice Encoding Types - identifies 'Voice' Standard used to encode a VM data type that carries
-- voice encoded information.

    VoiceEncodingType ::= OBJECT IDENTIFIER -- default CCITT's G.721 32K ADPCM

--  Message Forwarding Permitted

    MessageForwardingPermitted ::= BOOLEAN -- Default TRUE, forwarding is permitted

-- Expiry time

    ExpiryTimeField ::= UTCTime

-- Obsoleted VM

    ObsoletedVMField    ::= SEQUENCE OF ObsoletedVMSubfield

    ObsoletedVMSubfield ::= VMIdentifier

-- VMG User Security Elements

    VMGSUserSecurityElementsField ::= SEQUENCE {
        vmgs-user-security-element          [0] VMGSUserSecurityElement OPTIONAL,
        vm-encrypted-primary-bodypart       [1] BOOLEAN OPTIONAL,
        vmgs-user-security-extensions       [2] VMGSUserSecurityExtensions OPTIONAL }

    VMGSUserSecurityElement         ::= BIT STRING (SIZE (0..ub-vmgs-user-security-elements))

    VMGSUserSecurityExtensions      ::= SEQUENCE OF VMGSUserSecurityExtension

    VMGSUserSecurityExtension       ::= ExtensionField

-- VM Spoken Subject Field

    VMSpokenSubjectField ::= ANY    -- the encoding is defined by the Voice-Encoding-Type field and not to
                                    -- exceed 10 seconds in duration.

-- Spoken Language Identifier

    LanguageField ::= SEQUENCE OF Language -- from IPM

-- VM Creation Date/Time

    VMCreationTime ::= UTCTime

-- VN Receiver Field

    VNReceiverField ::= SEQUENCE {
        vn-receiver-name    [0] ORName,
        original-vmg-identifier     [1] VMIdentifier OPTIONAL,
        first-recipient         [2] FirstRecipientField OPTIONAL }
```

```
-- Heading Extensions

    HeadingExtensionsField ::= SET OF HeadingExtensionsSubField

    HeadingExtensionsSubField ::= ExtensionField

-- VM body

    Body ::= SEQUENCE {
        primary-body-part   PrimaryBodyPart,
        additional-body-parts       AdditionalBodyParts OPTIONAL }

    PrimaryBodyPart ::= CHOICE {
        vm-body-part        [0] VBodyPart,
        forwarded-VM        [1] VMBodyPart }

    AdditionalBodyParts ::= SEQUENCE OF VM-ExternallyDefinedBodyPart

-- VM body part

    VBodyPart ::= SEQUENCE {
        voice-parameters    [0] VoiceParameters OPTIONAL,
        voice-data      [1] VoiceData }

    VoiceParameters ::= SEQUENCE {
        voice-message-duration      [0] VMDuration OPTIONAL,
        voice-encoding-type         [1] VoiceEncodingType OPTIONAL, -- for use in IPM
        other-parameters    [2] VMSupplementaryInformation OPTIONAL,
        extension-parameters        [3] VBPParameterExtensions OPTIONAL  }

    VMDuration ::= INTEGER, -- size indicated in seconds

    VBPParameterExtensionsField ::= SET OF VBPExtensionsSubField

    VBPExtensionsSubField ::= ExtensionField

    VoiceData ::= ANY -- defined by VoiceEncodingType

-- Forwarded VM body part

    VMBodyPart ::= SEQUENCE {
        parameters    [0] MessageParameters OPTIONAL,
        data    [1] MessageData }

    MessageParameters ::= SET {
        delivery-time [0] MessageDeliveryTime OPTIONAL,
        delivery-envelope    [1] OtherMessageDeliveryFields OPTIONAL,
       -- delivery-time and delivery-envelope shall both be present or both shall be absent.
        other-parameters    [2] VMSupplementaryInformation OPTIONAL }

    MessageData ::= VM

-- VM Externally Defined Body Parts

    VM-ExternallyDefinedBodyPart ::= ExternallyDefinedBodyPart -- from IPMS --

-- Supplementary Info

    VMSupplementaryInformation ::= IA5String (SIZE (1..ub-supplementary-info-length))

-- VM Notifications (VNs)

    VN ::= CHOICE {
        receipt-notification  [0] ReceiptNotificationFields,  -- Referred to as RN
        service-notification [1] ServiceNotificationFields,  -- Referred to as SN
        non-receipt-notification     [2] NonReceiptNotificationFields   -- Referred to as NRN }
```

-- Common fields

```
CommonFields ::= SET {
        subject-vm    SubjectVMField,
        vn-originator [1] VNOriginatorField,
        first-recipient         [2] FirstRecipientField OPTIONAL,
        notice-creation-time    [3] NoticeCreationTimeField,
        vn-voice-encoding-type  [4] VNVoiceEncodingType,
        conversion-indictation  [5] ConversionEITsField OPTIONAL,
        notification-security-elements          [6] SecurityElementsField OPTIONAL,
        orig-vm-spoken-subject  [7] VMSpokenSubjectField OPTIONAL,
        subject-vm-other-recipients             [8] RecipientField OPTIONAL,
        vn-supplementary-info   [9] VNSupplementaryInfo OPTIONAL,
        notifications-extensions    [10] NotificationExtensionsField OPTIONAL }
```

-- Subject VM Identifier

```
SubjectVMField ::= VMIdentifier
```

-- VM Notification Originator

```
VNOriginatorField ::= SEQUENCE {
        originator-name     [0] VMOriginatorField,
        vn-initiator    [1] VNInitiatorField OPTIONAL }
```

-- First Recipient

```
FirstRecipientField ::= ORName
```

-- Notification Time

```
NoticeCreationTimeField ::= UTCTime
```

-- VN Initiator

```
VNInitiatorField ::= ENUMERATED {
        internal-ua (0),
        external-ua (1),
        internal-ms (2),
        internal-tsau (3) }
```

-- Subject VM Other Recipients

```
SubjectVMOtherRecipients ::= SEQUENCE  OF RecipientField
```

-- Voice Notification Supplementary Information

```
VNSupplementaryInfo ::= SEQUENCE {
        supplementary-info [0] VMSupplementaryInformation OPTIONAL,
        v-supplementary-info        [1] SpokenSupplementaryInfo OPTIONAL }

SpokenSupplementaryInfo ::= ANY (SIZE (0..ub-vmg-spoken-supplemental-info))
        -- The encoding of the value carried in v-supplementary-info is identified by the value of the Voice Encoding
        -- Types field.
```

-- Security Elements

```
SecurityElementsField ::= SEQUENCE {
        original-content        [0] Content OPTIONAL,  -- from P1
        original-content-integrity-check            [1] ContentIntegrityCheck OPTIONAL,  -- from P1
        vmgs-user-security-elements                 [2] VMGSUserSecurityElementsField OPTIONAL,
        security-extensions [3] SecurityExtensionsField OPTIONAL }

SecurityExtensionsField ::= SET OF SecurityExtensionsSubField

SecurityExtensionsSubField ::= ExtensionField
```

```
-- Voice Encoding of Spoken information in this VN

    VNVoiceEncodingType ::= VoiceEncodingType

-- Notification Extensions

    NotificationExtensionsField ::= SET OF NotificationExtensionsSubField

    NotificationExtensionsSubField ::= ExtensionField

-- Receipt Notification fields

    ReceiptNotificationFields ::= SEQUENCE {
        rn-common-fields    [0] CommonFields,
        rn-extensions       [1] NotificationExtensionsField OPTIONAL }

-- Service notification fields

    ServiceNotificationFields ::= SEQUENCE {
        sn-common-fields    [0] CommonFields,
        sn-reason-code-field        [1] SNReasonCode,
        sn-extensions       [2] NotificationExtensionsField OPTIONAL }

-- Service Notification Reason Codes from a UA or MS or TSAU

    SNReasonCode ::= SEQUENCE {
        sn-reason     [0] SNReasonField,
        sn-diagnostic       [1] SNDiagnosticField OPTIONAL }

-- Service Notification Basic Reason Codes from a VMGS-UA or VMGS-MS or VMGS-TSAU. These codes
-- are those specified in Annex B of Recommendation F.440 for the Element of Service "VM Notification
-- Request".

    SNReasonField ::= BITSTRING {
        unspecified   (0),
        auto-forwarding-ind(1),
        language-ind (2),
        obsoleting-ind      (3),
        attendant-assisted-delivery-request     (4),
        expiry-date-ind     (5),
        body-part-encryption-ind   (6) } (SIZE (2..ub-sn-reasons))

-- Service Notification Diagnostic Codes from a VMGS-UA or VMGS-MS

    SNDiagnosticField ::= INTEGER {
        -- This field may be used to further specify the error signalled in sn-ua-ms-basic-code. Additional
        -- information may be indicated in sn-supplementary-information.

        -- General diagnostic codes

        language-national-usage-problem (1),
        -- used if the national usage of the language included is incompatible
        vm-language-not-understood (2),
        -- language not understood by this recipient
        vm-unsupported-voice-encoding (3),
        -- recipient does not support the VM's encoding

        -- security error diagnostic codes
        local-security-not-supported (4)
        } (1..ub-vn-reason-code)
```

-- Non-Receipt Notification Fields

```
NonReceiptNotificationFields ::= SEQUENCE {
        nrn-common-fields  [0] CommonFields,
        nrn-reason-codes   [1] VNRNReasonCodeField,
        nrn-extensions     [2] NotificationExtensionsField OPTIONAL }

VNRNReasonCodeField ::= CHOICE {
        nrn-ua-ms-reason-code     [0] VNRNUAMSReasonCodeField,
        nrn-user-reason-code      [1] VNRNUserReasonCodeField,
        nrn-tsau-reason-code      [2] VNRNTSAUReasonCodeField }
```

-- NRN Reason Codes from a VMGS-UA or VMGS-MS

```
VNRNUAMSReasonCodeField ::= SEQUENCE {
        nrn-ua-ms-basic-codes     [0] VNRNUAMSBasicCodeField,
        nrn-ua-ms-diagnostics     [1] VNRNUAMSDiagnosticField OPTIONAL }
```

-- VN NRN Basic Reason Codes from a VMGS-UA or VMGS-MS

```
VNRNUAMSBasicCodeField ::= INTEGER {
        unspecified (0),
        auto-forwarded (1),
        can-not-pass-to-mhs-user (2),
        delivery-timeout (3),
        message-discarded (4),
        subscription-terminated (5),
        forwarding-error (6),
        security-error (7),
        message-forwarded (8),
        voice-encoding-not-supported (9)
                } (0..ub-vn-reason-code)
```

-- NRN Diagnostic Codes from a VMGS-UA or VMGS-MS or TSAU.

```
VNRNUAMSDiagnosticField ::= INTEGER {
        -- This field may be used to further specify the error signalled in the field "nrn-ua-ms-basic-codes".
        -- Additional information may be indicated in the "vn-supplementary-info" field.
        protocol-violation (0) -- this is a place holder
        } (0..ub-vn-reason-code)
```

-- NRN Reason Codes from a VMGS-User

```
VNRNUserReasonCodeField ::= SEQUENCE {
        vn-user-basic-codes       [0] VNRNUserBasicCodeField,
        vn-user-diagnostics[1] VNRNUserDiagnosticField OPTIONAL }
```

-- VN NRN Basic Reason Codes from a VMGS-User.

```
VNRNUserBasicCodeField ::= INTEGER {
        unspecified (0),
        user-defined-reason (1) -- this is a place holder
        } (0..ub-vn-reason-code)
```

-- VN NRN Basic Reason Codes from a VMGS-User.

```
VNRNUserDiagnosticField ::= INTEGER
        -- Contains the reason passed by the user when the value of "nrn-user-basic-code" is "user-defined-
        -- reason". Additional information may be indicated in the "vn-supplementary-info" field. The values used in
        -- this field are outside the scope of this Recommendation.
```

```
-- NRN Reason Codes from a TSAU-User

    VNRNTSAUReasonCodeField ::= SEQUENCE {
        nrn-user-basic-codes        [0] VNRNTSAUBasicCodeField,
        nrn-user-diagnostics        [1] VNRNTSAUDiagnosticField OPTIONAL }

    VNRNTSAUReasonBasicCodeField ::= INTEGER {
        attendant-assisted-delivery-failure (0),
        unknown-telephone-number (1),
        attendant-assisted-delivery-not-provided (2),
        delivery-timeout (3),
        security-error (4),
        message-forwarded (5),
        unspecified (6),
        inappropriate-voice-encoding (7),
        telephone-number-unreachable (8),
        recipient-refused-message (9),
        no-answer-on-every-attempt (10),
        busy-on-every-attempt (11),
        no-answer-or-busy-on-every-attempt (12),
        sensitivity-not-supported (13),
        importance-not-supported (14),
        busy-on-every-attempt (15)
       -- TSAU providers may define additional values above (1000). }

    VNRNTSAUDiagnosticField ::= INTEGER {
        person-to-person-spoken-name-not-provided (0),
        person-unavailable (1),
        number-not-in-service (2),
        message-expired (3),
        importance-not-conveyed (4)
        } (0..ub-vn-reason-code)

    END -- of VMGSInformationObjects
```

ANNEX C

(to Recommendation X.440)

VM Message Store attributes — Reference definition

(This annex forms an integral part of this Recommendation.)

This annex defines for reference purposes the MS attributes specific to VMGS. It uses the ATTRIBUTE macro of CCITT Rec. X.501 | ISO 9594-2.

```
    VMGSMessageStoreAttributes { joint-iso-ccitt
            mhs-motis(6) vmgs(8) modules(9) message-store-attributes(4) }

        DEFINITIONS IMPLICIT TAGS ::=
        BEGIN

-- Prologue

-- Exports everything.

        IMPORTS
```

-- VMGS Object Identifiers

id-bat-body, id-bat-externally-defined-body-part-types, id-bat-message-data,
id-bat-message-duration, id-bat-message-parameters,id-bat-message-other-params,
id-bat-body-message-length, id-bat-v-body-part, id-bat-vm-bodypart-sequence-number,
id-hat-vm-creation-time, id-hat-expiry-time, id-hat-heading,
id-hat-heading-extensions, id-hat-importance,
id-hat-language, id-hat-obsoleted-vm, id-hat-originator,
id-hat-sensitivity-field, id-hat-this-VM, id-hat-forwarded-indication,
id-hat-forwarding-permitted, id-hat-vm-encrypted-primary-bodypart,
id-hat-vm-originator-spoken-name, id-hat-vm-spoken-subject,
id-hat-vm-this-recipient-spoken-name, id-hat-vmgs-security-elements,
id-hat-vmgs-user-security-extensions, id-hat-vn-receiver,
id-hat-voice-encoding-type, id-nat-conversion-indication,
id-nat-first-recipient, id-nat-notification-extensions,
id-nat-notification-security-elements, id-nat-notification-creation-time,
id-nat-nrn-extensions, id-nat-nrn-tsau-reason-code, id-nat-nrn-ua-ms-reason-basic-code,
id-nat-nrn-user-reason-basic-code, id-nat-orig-vm-spoken-name, id-nat-orig-vmg-spoken-subject,
id-nat-rn-extensions, id-nat-sn-extensions, id-nat-sn-reason-code,
id-nat-spoken-subject, id-nat-subject-vm, id-nat-subject-vmg-other-recipients,
id-nat-vn-initiator, id-nat-vn-originator-text,
id-nat-vn-originator-spoken-name, id-nat-vn-supplementary-info,
id-nat-vn-voice-encoding-type,
id-rat-message-forwarding-permitted, id-rat-recipient-extensions-for-this-recipient,
id-rat-this-recipient, id-rat-vm-notification-requests-for-this-recipient,
id-rat-vm-notification-extension-requests-for-this-recipient,
id-rat-vm-notification-security-requests-for-this-recipient,
id-rat-vm-reception-security-requests-for-this-recipient,
id-sat-vmg-entry-type, id-sat-vmg-synopsis

FROM VMGSObjectIdentifiers { joint-iso-ccitt mhs-motis(6) vmgs(8) modules(9) object-identifiers(0)
}

-- MTS Upper Bounds

ub-bit-options, ub-integer-options, ub-supplementary-info-length
----
FROM MTSUpperBounds { joint-iso-ccitt mhs-motis(6) mts(3) modules(0) upper-bounds(3)
}

-- MS Abstract Service

SequenceNumber

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

-- VMGS Information Objects

Body, BodyPartReference, DateAndTimeOfPreparationField, ExtensionField,
FirstRecipientField, Heading, HeadingExtensionsSubField, MessageData,
MessageForwarded, MessageForwardingPermitted, MessageParameters,
NoticeCreationTimeField, NotificationExtensionsSubField, NRNExtensionsSubField,
NRNReasonCodeField, ObsoletedVMSubfield, OriginatorSpokenName, ReceiptNotificationFields,
Recipient, RecipientExtensionsSubField, RecipientSpokenName, RecipientsSubField,
RNExtensionsSubField, SecurityElementsField, SecurityExtensionsSubField,
SNExtensionsSubField, SNReasonCodeField, SpokenSupplementaryInfo, SubjectVMField,
ThisVMField, VMBodyPart, VMCreationTime, VMDuration, VMGSUserSecurityElements,
VMGSUserSecurityExtensions, VMSpokenSubjectField, VMSupplementaryInformation,
VMNotificationExtensionsSubField, VNInitiatorField, VNOriginatorSpokenName,
VNotificationRequests,
VNotificationSecurity, VNReceiverField, VNRNUAMSBasicCodeField,
VNRNUAMSDiagnosticCodeField,
VNRNUserBasicCodeField, VNRNUserDiagnosticField, VoiceEncodingType,
VReceptionSecurity

```
        ----
        FROM VMGSInformationObjects { joint-iso-ccitt mhs-motis(6) vmgs(8) modules(9)
        information-objects(2) }

-- IPMS Information Objects

    ConversionEITsField, ExternallyDefinedParameters, ExpiryTimeField, ImportanceField,
        ORDescriptor, SensitivityField
        ----
        FROM IPMSInformationObjects { joint-iso-ccitt mhs-motis(6) ipms(1) modules(0)
        information-objects(2) }

-- IPM Heading Extensions

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

-- Directory Information Framework

    ATTRIBUTE
        ----
        FROM InformationFramework { joint-iso-ccitt ds(5) modules(1) informationFramework(1)
        };

--  END imports

-- MESSAGE STORE ATTRIBUTES

-- Summary Attributes

-- VM Entry Type

    vm-entry-type ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX EntryType
        MATCHES FOR EQUALITY
        SINGLE VALUE
        ::= id-sat-vmg-entry-type

    EntryType ::= ENUMERATED {
        voice-message (0),
        receipt-notification (1),
        non-receipt-notification (2),
        service-notification (3) }

-- VM Synopsis

    vm-synopsis ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX VMSynopsis
        SINGLE VALUE
        ::= id-sat-vm-synopsis

    VMSynopsis ::= SEQUENCE OF BodyPartSynopsis

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

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

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

-- VM Notification Indicator

```
vm-notification-indicator ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX VNotificationIndicator
        MATCHES FOR EQUALITY
        SINGLE VALUE
        ::=id-sat-vm-notification-indicator

VNotificationIndicator ::= VNIndicatorBits DEFAULT { }  -- default is NO Notifications Sent

VNIndicatorBits ::= BIT STRING {
        rn-sent         (0),
        sn-sent         (1),
        nrn-sent        (2) } (SIZE (1..ub-bit-options))
```

-- Heading Attributes

-- Heading

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

-- Heading Subfield Attributes

```
this-vm ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX ThisVMField
        MATCHES FOR EQUALITY
        SINGLE VALUE
        ::= id-hat-this-VM

originator ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX COMPONENTS OF ORDescriptor
        MATCHES FOR EQUALITY
        SINGLE VALUE
        ::= id-hat-originator

obsoleted-vm ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX ObsoletedVMField
        MATCHES FOR EQUALITY
        SINGLE VALUE
        ::= id-hat-obsoleted-vm

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

vm-forwarded-indication ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX MessageForwarded
        MATCHES FOR EQUALITY
        SINGLE VALUE
        ::= id-hat-vm-forwarded-indication
```

vm-forwarding-permitted ATTRIBUTE
WITH ATTRIBUTE-SYNTAX MessageForwardingPermitted
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-hat-vm-forwarding-permitted

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

voice-encoding-type; ATTRIBUTE *-- Identifies the Encoding of the voice object*
WITH ATTRIBUTE-SYNTAX VoiceEncodingType
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-hat-voice-encoding-type

vm-creation-time ATTRIBUTE
WITH ATTRIBUTE-SYNTAX VMCreationTime
MATCHES FOR EQUALITY ORDERING
SINGLE VALUE
::= id-hat-vm-creation-time

vn-receiver-field ATTRIBUTE
WITH ATTRIBUTE-SYNTAX VNReceiverField
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-hat-vn-receiver

-- User Security Attributes

vmgs-user-security-element ATTRIBUTE
WITH ATTRIBUTE-SYNTAX VMGSUserSecurityElement
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-hat-vmgs-user-security-elements

vm-encrypted-primary-bodypart ATTRIBUTE
WITH ATTRIBUTE-SYNTAX BOOLEAN
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-hat-vm-encrypted-primary-bodypart

vmgs-user-security-extensions ATTRIBUTE
WITH ATTRIBUTE-SYNTAX VMGSUserSecurityExtensions
MATCHES FOR EQUALITY
MULTI VALUE
::= id-hat-vmgs-user-security-extensions

-- Originator-Generated Spoken Attributes

vm-spoken-subject ATTRIBUTE
WITH ATTRIBUTE-SYNTAX VMSpokenSubjectField
SINGLE VALUE
::= id-hat-vm-spoken-subject

vm-originator-spoken-name ATTRIBUTE
WITH ATTRIBUTE-SYNTAX OriginatorSpokenName
SINGLE VALUE
::= id-hat-vm-originator-spoken-name

vm-this-recipient-spoken-name ATTRIBUTE
WITH ATTRIBUTE-SYNTAX RecipientSpokenName
SINGLE VALUE
::= id-hat-vm-this-recipient-spoken-name

-- Heading Extensions

heading-extensions ATTRIBUTE
WITH ATTRIBUTE-SYNTAX HeadingExtensionsSubField
MATCHES FOR EQUALITY
MULTI VALUE
::= id-hat-heading-extensions

-- Recipient Sub-field

this-recipient ATTRIBUTE
WITH ATTRIBUTE-SYNTAX COMPONENTS OF ORDescriptor
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-rat-this-recipient

vm-notification-requests-for-this-recipient ATTRIBUTE
WITH ATTRIBUTE-SYNTAX VNotificationRequests
MATCHES FOR EQUALITY
MULTI VALUE
::= id-rat-vm-notification-requests-for-this-recipient

vm-notification-security-requests-for-this-recipient ATTRIBUTE
WITH ATTRIBUTE-SYNTAX VNotificationSecurity
MATCHES FOR EQUALITY
MULTI VALUE
::= id-rat-vm-notification-security-requests-for-this-recipient

vm-reception-security-requests-for-this-recipient ATTRIBUTE
WITH ATTRIBUTE-SYNTAX VMReceptionSecurity
MATCHES FOR EQUALITY
MULTI VALUE
::= id-rat-vm-reception-security-requests-for-this-recipient

vm-notification-extension-requests-for-this-recipient ATTRIBUTE
WITH ATTRIBUTE-SYNTAX VMNotificationExtensionsSubField
MATCHES FOR EQUALITY
MULTI VALUE
::= id-rat-vm-notification-extension-requests-for-this-recipient

recipient-extensions-for-this-recipient ATTRIBUTE
WITH ATTRIBUTE-SYNTAX RecipientExtensionsSubField
MATCHES FOR EQUALITY
MULTI VALUE
::= id-rat-recipient-extensions-for-this-recipient

-- Body Attributes

-- Body

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

-- Body Analyses

v-body-message-length ATTRIBUTE
WITH ATTRIBUTE-SYNTAX VoiceMessageLength
MATCHES FOR EQUALITY ORDERING
SINGLE VALUE
::= id-bat-v-body-message-length

VoiceMessageLength ::= INTEGER *-- in octets*

voice-message-duration ATTRIBUTE
WITH ATTRIBUTE-SYNTAX VMDuration
MATCHES FOR ORDERING
SINGLE VALUE
::= id-bat-voice-message-duration

voice-message-other-parms ATTRIBUTE
WITH ATTRIBUTE-SYNTAX VMSupplementaryInformation
MATCHES FOR EQUALITY SUBSTRINGS
SINGLE VALUE
::= id-bat-voice-message-other-parms

-- PRIMARY BODY PARTS

-- Subject Voice Message

v-body-part ATTRIBUTE
WITH ATTRIBUTE-SYNTAX VBodyPart
SINGLE VALUE
::= id-bat-v-body-part

-- Forwarded Voice Message(s)

vm-bodypart-sequence-number ATTRIBUTE
WITH ATTRIBUTE-SYNTAX SequenceNumber *-- sequence number of the forwarded VM entry.*
SINGLE VALUE
::= id-bat-vm-bodypart-sequence-number

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

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

-- EXTERNALLY DEFINED BODY Part Types

externally-defined-body-part-types ATTRIBUTE
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER
MATCHES FOR EQUALITY
MULTI VALUE
::= id-bat-externally-defined-body-part-types

-- Description of the externally-defined-body-part-types attribute syntax for parameter portion only

VMExternallyDefinedBodyPartParameterAttribute ::= ExternallyDefinedBodyPart *-- from IPMS*

-- Notification Attributes

-- Common Fields

subject-vm ATTRIBUTE
WITH ATTRIBUTE-SYNTAX SubjectVMField
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-nat-subject-vm

vn-originator-text ATTRIBUTE
WITH ATTRIBUTE-SYNTAX COMPONENTS OF ORDescriptor *-- sub component of VNOriginatorField*
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-nat-vn-originator

vn-originator-spoken-name ATTRIBUTE
WITH ATTRIBUTE-SYNTAX OriginatorSpokenName
SINGLE VALUE
::= id-nat-vn-originator

vn-initiator ATTRIBUTE
WITH ATTRIBUTE-SYNTAX VNInitiatorField
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-nat-vn-initiator

first-recipient ATTRIBUTE
WITH ATTRIBUTE-SYNTAX FirstRecipientField
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-nat-first-recipient

notification-creation-time ATTRIBUTE
WITH ATTRIBUTE-SYNTAX NoticeCreationTimeField
MATCHES FOR EQUALITY ORDERING
SINGLE VALUE
::= id-nat-notification-creation-time

notification-security-elements ATTRIBUTE
WITH ATTRIBUTE-SYNTAX SecurityElementsField
SINGLE VALUE
::= id-nat-notification-security-elements

conversion-indication ATTRIBUTE
WITH ATTRIBUTE-SYNTAX ConversionEITsField
SINGLE VALUE
::= id-nat-conversion-indication

orig-vm-spoken-subject ATTRIBUTE
WITH ATTRIBUTE-SYNTAX VMSpokenSubjectField
SINGLE VALUE
::= id-nat-orig-vm-spoken-subject

subject-vm-other-recipients ATTRIBUTE
WITH ATTRIBUTE-SYNTAX RecipientField
MULTI VALUE
::= id-nat-subject-vm-other-recipients

vn-supplementary-info ATTRIBUTE
WITH ATTRIBUTE-SYNTAX VNSupplementaryInfo
SINGLE VALUE
::= id-nat-vn-supplementary-info

vn-voice-encoding-type ATTRIBUTE
WITH ATTRIBUTE-SYNTAX VNVoiceEncodingType
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-nat-vn-voice-encoding-type

notification-extensions ATTRIBUTE
WITH ATTRIBUTE-SYNTAX NotificationExtensionsSubField
MATCHES FOR EQUALITY
MULTI VALUE
::= id-nat-notification-extensions

-- Receipt Notification Fields

rn-extensions ATTRIBUTE
WITH ATTRIBUTE-SYNTAX RNExtensionsSubField
MATCHES FOR EQUALITY
MULTI VALUE
::= id-nat-rn-extensions

```
-- Service Notification Fields for any VMGS-user

    sn-reason-code ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX SNReasonCode
        SINGLE VALUE
        ::= id-nat-sn-ua-ms-basic-reason-code

    sn-extensions ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX SNExtensionsSubField
        MATCHES FOR EQUALITY
        MULTI VALUE
        ::= id-nat-sn-extensions

-- Non-receipt Fields

    nrn-ua-ms-reason-code ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX VNRNUAMSReasonCodeField
        SINGLE VALUE
        ::= id-nat-nrn-ua-ms-reason-basic-code

    nrn-user-reason-code ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX VNRNUserReasonCodeField
        SINGLE VALUE
        ::= id-nat-nrn-user-reason-basic-code

    nrn-tsau-reason-code ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX VNRNTSAUReasonCodeField
        SINGLE VALUE
        ::= id-nat-nrn-tsau-reason-code

    nrn-extensions ATTRIBUTE
        WITH ATTRIBUTE-SYNTAX NRNExtensionsSubField
        MATCHES FOR EQUALITY
        MULTI VALUE
        ::= id-nat-nrn-extensions

    END -- of VMGSMessageStoreAttributes
```

ANNEX D

(to Recommendation X.440)

Message Store VMGS Auto-Action types — Reference definition

(This annex forms an integral part of this Recommendation.)

This annex, a supplement to Annex C, defines the MS auto-action registration ASN.1 types specific to Voice Messaging. The definitions contained herein use the Auto-Action macro of CCITT Rec. X.413 | ISO 10021-5.

```
    VMGSAutoActionTypes { joint-iso-ccitt mhs-motis(6)
            vmgs(8) modules(9) message-store-auto-actions(7) }
        DEFINITIONS ::=
        BEGIN

-- Prologue

-- Exports everything.

    IMPORTS
```

```
-- VMGS Object Identifiers

    id-act-vmg-auto-forward, id-act-vn-sn-auto-action
        ----
        FROM VMGSObjectIdentifiers { joint-iso-ccitt mhs-motis(6) vmgs(8) modules(9) object-
        identifiers(0) }

-- VMGS Information Objects

    VMSupplementaryInformation, RecipientField, ActionRequestField
        VNotificationRequestsField, MessagePassingAllowed
        ----
        FROM VMGSInformationObjects { joint-iso-ccitt mhs-motis(6) vmgs(8) modules(9)
        information-objects(2) }

-- MS Abstract Service

    AUTO-ACTION, Filter
        ----
        FROM MSAbstractService { joint-iso-ccitt mhs-motis(6) ms(4) modules(0) abstract-
        service(1) }

-- MS General Auto Actions

    PerMessageAutoForwardFields, PerRecipientAutoForwardFields
        ----
        FROM MSGeneralAutoActionTypes { joint-iso-ccitt mhs-motis(6) ms(4) modules(0)
        general-auto-action-types(3) }

-- MTS Upper Bounds

    ub-recipients
        ----
        FROM MTSUpperBounds { joint-iso-ccitt mhs-motis(6) mts(3) modules(0) upper-bounds(3)
        }

-- MTS Abstract Service Definition

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

-- END Imports

-- Auto-Action Types

-- VMGS Auto Action Registration

    vm-auto-forward-actions AUTO-ACTION
        REGISTRATION PARAMETER IS VMActionRegistrationParameter
        ::= id-act-vm-auto-actions

    VMActionRegistrationParameter ::= SEQUENCE {
        filter   [0] Filter OPTIONAL,
        vm-supplementary-information          [1] VMSupplementaryInformation OPTIONAL,
        delete-after-forwarding     [2] BOOLEAN DEFAULT FALSE,
        vm-auto-forwarding-mode     CHOICE {
            vm-forwarding-with-message-not-accepted [3] ForwardWithNonReceipt,
            vm-forwarding-with-message-accepted      [4] ForwardWithMessageAccepted,
            vm-forwarding-with-no-VNs        [5] ForwardWithoutVNs } }
```

```
-- Auto Action Registration Parameters for NonReceipt with Message not Accepted

    ForwardWithMessageNonReceipt ::= SET {
        COMPONENTS OF PerMessageAutoForwardFields, -- from MS auto-forward Abstract Operation
        per-recipient-no-accept-arguments     [3] SEQUENCE OF
        PerRecipientNoAcceptArguments,
        notification-argument       [4] NotificationArguments OPTIONAL }

    PerRecipientNoAcceptArguments ::= SEQUENCE {
        per-recipient-field   [0] PerRecipientAutoForwardFields,
        heading-next-recipient      [1] RecipientField }

    NotificationArguments ::= SET {
        COMPONENTS OF PerMessageAutoForwardFields, -- from MS auto-forward Abstract Operation
        per-recipients-field  [3] PerRecipientAutoForwardFields }

-- Auto Action Registration Parameters for NonReceipt with Message Accepted

    ForwardWithRespAccepted ::= SET {
        COMPONENTS OF PerMessageAutoForwardFields, -- from MS auto-forward Abstract Operation
        per-recipient-arguments   [3] SEQUENCE SIZE (1..ub-recipients) OF
        PerRecipientArguments, -- P1 + VMP information
        notification-argument       [4] NotificationArguments OPTIONAL -- RN arguments }

    PerRecipientArguments ::= SEQUENCE {
        per-recipients-p1-info      [0] PerRecipientAutoForwardFields,
                        -- P1 Per Recipient information
        vm-heading-fields   [1] HeadingFields OPTIONAL, -- VMP Per Recipient information
        new-vn-receiver-name      [2] ORName } -- for all VMP recipients

    HeadingFields ::= SEQUENCE {
        next-recipient        [0] RecipientField,
        next-recipient-vn-requests-field          [1] VNotificationRequestsField OPTIONAL,
        next-message-forwarding-permitted   [2] MessageForwardingPermitted DEFAULT FALSE }

-- Auto Action Registration Parameters for MS auto Forwarding without RN or NRN.

    ForwardWithoutVNs ::= SET {
        COMPONENTS OF PerMessageAutoForwardFields, -- from MS auto-forward Abstract Operation
        per-recipient-no-vn-arguments           [3] SEQUENCE SIZE (1..ub-recipients) OF
        PerRecipientArguments -- P1 + VM information }

-- Auto Action Registration Parameters for auto generation of Service Notification.

    vm-auto-sn-action AUTO-ACTION
        REGISTRATION PARAMETER IS VNSNActionRegistrationParameter
        ::= id-act-vn-sn-auto-action

    VNSNActionRegistrationParameter ::= SEQUENCE {
        filter   [0] Filter OPTIONAL,
        vn-service-notice-info      [1] ServiceNotificationInfo }

    ServiceNotificationInfo ::= SET {
        sn-notification-argument   [4] SNReasonCode,
        sn-supplementary-info      [5] VNSupplementaryInfo OPTIONAL,
        extensions   [6] NotificationExtensionsField OPTIONAL }

    END -- of VMGSAutoActionTypes
```

ANNEX E

(to Recommendation X.440)

VMGS functional objects - Reference definition

(This annex forms an integral part of this Recommendation.)

This annex defines for reference purposes the functional objects of Voice Messaging. It uses the OBJECT and REFINE macros of CCITT Rec. X.407 | ISO 10021-3.

```
VMGSFunctionalObjects { joint-iso-ccitt
        mhs-motis(6) vmgs(8) modules(9) functional-objects(1) }
    DEFINITIONS IMPLICIT TAGS ::=
    BEGIN

-- Prologue

-- Exports everything.

    IMPORTS

-- VMGS Abstract Service

    origination, reception, management
        ----
        FROM VMGSAbstractService { joint-iso-ccitt mhs-motis(6) vmgs(8) modules(9) abstract-service(3) }

-- VMGS Object Identifiers

    id-ot-vmge, id-ot-vmgs, id-ot-vmg-ms, id-ot-vmg-ua,
        id-ot-vmgs-user, id-ot-tsau,
        id-ref-primary, id-ref-secondary
        ----
        FROM VMGSObjectIdentifiers { joint-iso-ccitt mhs-motis(6) vmgs(8) modules(9) object-
        identifiers(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) };

-- END imports

-- "Root" Object Type

    vmge OBJECT ::= id-ot-vmge
```

```
-- Primary Refinement

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

-- Primary Object Types

-- VMGS User

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

-- VMGS Messaging System

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

-- Secondary Refinement

        vmgs-refinement REFINE vmgs AS
                mTS
                        submission     [S] PAIRED WITH vmg-ua, vmg-ms
                        delivery       [S] PAIRED WITH vmg-ua, vmg-ms
                        administration         [S] PAIRED WITH vmg-ua, vmg-ms
                vmg-ua RECURRING
                        origination    [S] VISIBLE
                        reception      [S] VISIBLE
                        management [S] VISIBLE
                vmg-ms RECURRING
                        submission     [S] PAIRED WITH vmg-ua
                        retrieval      [S] PAIRED WITH vmg-ua
                        administration         [S] PAIRED WITH vmg-ua
                tsau RECURRING
                        origination    [S] VISIBLE
                        reception      [S] VISIBLE
                        management [S] VISIBLE
                ::= id-ref-secondary

-- Secondary Object Types

-- VMGS User Agent

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

```
-- VMGS Message Store

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

--  Telephone Service Access Unit

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

    END -- of VMGS-Functional-Objects
```

ANNEX F

(to Recommendation X.440)

VMGS abstract service — Reference definition

(This annex forms an integral part of this Recommendation.)

This annex defines for reference purposes the VMGS Abstract Service. It uses the PORT and ABSTRACT-OPERATION and ABSTRACT-ERROR macros of CCITT Rec. X.407 | ISO/IEC 10021-3.

```
    VMGSAbstractService { joint-iso-ccitt mhs-motis(6)
            vmgs(8) modules(9) abstract-service(3) }
    DEFINITIONS IMPLICIT TAGS ::=
    BEGIN

-- Prologue

-- Exports everything.

    IMPORTS

-- VMGS Information Objects

    InformationObject, RecipientField, VMP, VNP
            ----
        FROM VMGSInformationObjects { joint-iso-ccitt mhs-motis(6) vmgs(8) modules(9)
        information-objects(2) }

-- VMGS Object Identifiers

    id-pt-origination, id-pt-reception id-pt-management
        ----
        FROM VMGSObjectIdentifiers { joint-iso-ccitt mhs-motis(6) vmgs(8) modules(9) object-
        identifiers(0) }
```

```
-- MTS Abstract Service

    MessageDeliveryEnvelope, MessageSubmissionEnvelope,
        MessageSubmissionIdentifier, MessageSubmissionTime,
        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) };

-- Primary Port Types

-- Origination

    origination PORT
        CONSUMER INVOKES {
            OriginateVM,
            OriginateVN }
            ::= id-pt-origination

-- Reception

    reception PORT
        SUPPLIER INVOKES {
            ReceiveReport,
            ReceiveVM,
            ReceiveVN }
            ::= id-pt-reception

-- Management

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

-- ABSTRACT OPERATIONS

-- Origination Abstract Operations

-- Originate VM

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

```
-- Originate VN

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

    VNType ::= CHOICE {
            [0] RN,
            [1] NRN,
            [2] SN }

-- Reception Abstract Operations

-- Receive Report

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

-- Receive VM

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

-- Receive VNP

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

-- Change auto-discard

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

-- Change auto-acknowledgements

    ChangeAutoAcknowledgements ::= ABSTRACT-OPERATION
        ARGUMENT SET {
            auto-acknowledge-VM             [0] BOOLEAN,
            auto-acknowledge-suppl-recipient-info          [1] VNSupplementaryInfo OPTIONAL }
        RESULT
        ERRORS { SubscriptionError }
```

```
-- Change auto-forwarding

ChangeAutoForwarding ::= ABSTRACT-OPERATION
        ARGUMENT SET {
                auto-forward-VM     [0] BOOLEAN,
                auto-forward-recipients          [1] SEQUENCE OF RecipientField OPTIONAL,
                auto-forward-heading             [2] Heading OPTIONAL,
                auto-forward-comment             [2] VNSupplementaryInfo OPTIONAL }
        RESULT
        ERRORS {
                SubscriptionError,
                RecipientImproperlySpecified }

-- Change auto-SNotice

ChangeAutoSNotice ::= ABSTRACT-OPERATION
        ARGUMENT SET {
        auto-SNotice [0] BOOLEAN,
        auto-service-status-info     [1] EOSSupportIndicator,
        auto-SN-suppl-recipient-info             [2] VNSupplementaryInfo OPTIONAL }
        RESULT
        ERRORS { SubscriptionError }

        EOSSupportIndicator ::= BIT STRING {
                unspecified   (0),
                auto-forwarding-ind (1),
                language-ind (2),
                obsoleting-ind        (3),
                attendant-assisted-delivery-request     (4),
                expiry-date-ind       (5),
                body-part-encryption-ind   (6) } (SIZE (2..ub-sn-reasons))
                                -- setting any bit on indicates non-support

-- Subscription Error

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

-- Subscription Problem

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

        END -- of VMGSAbstractService
```

ANNEX G

(to Recommendation X.440)

VMGS upper bounds parameters — Reference definition

(This annex forms an integral part of this Recommendation.)

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.

```
VMGSUpperBounds { joint-iso-ccitt
                mhs-motis(6) vmgs(8) modules(9) upper-bounds(5) }
        DEFINITIONS ::=
        BEGIN
```

```
-- Prologue

-- Exports everything.

    IMPORTS -- nothing -- ;

-- Upper bounds

    ub-sn-reasons        INTEGER ::= 4

    ub-vmg-local-reference     INTEGER ::= 64

    ub-vmg-spoken-name        INTEGER ::= 10

    ub-vmg-spoken-subject     INTEGER ::= 20

    ub-vmg-spoken-supplemental-info              INTEGER ::= 20

    ub-vmgs-user-security-elements  INTEGER ::= 1024

    END -- of VMGSUpperBounds
```

ANNEX H

(to Recommendation X.440)

Index

(This annex forms an integral part of this Recommendation.)

This annex indexes this Recommendation. It gives the number(s) of the page(s) on which each item in each of several categories is found.

This annex indexes items in the following categories:

a) Abbreviations;

b) ASN.1 Macros;

c) ASN.1 Modules;

d) ASN.1 Modules;

e) Information Items;

f) MS ASN.1 Types;

g) MS Attributes:

h) MS VM Auto Actions;

i) NRN Reason Codes;

j) Object Identifiers;

k) P1 ASN.1 Types;

l) SN Reason Codes;

m) Terms;

n) Upper Bounds;

o) VMGS Abstracts;

p) VMGS Error Reason Codes;

q) VMGS Operation Reasons;

r) VMGS Operations.

Page

H.1 *Abbreviations*

AU 6,9
DL 9,15
EIT 9,85
MD 9
MHS 6,9
MOTIS 9
MS 6,9,20
MTA 6,9
MTS 6,9,44
NRN 9,23,29
RN 9,23,27
SN 9,23,28
TSAU 9,42
UA 6,9,20
VM 9,11,13,32
VMG 6,10
VMGE 10,31
VMGS 10
VMGS-MS 10,20,42
VMGS-UA 10,20,42
VMGS-user 10,32
VN 10,11,23
voice 6

H.2 *ASN.1 Macros*

VM-EXTENSION 12,13,93

H.3 *ASN.1 Modules*

VMGSAbstractService 113
VMGSAutoActionTypes 108
VMGSFunctionalObjects 111
VMGSInformationObjects 92
VMGSMessageStoreAttributes 100
VMGSObjectIdentifiers 89
VMGSUpperBounds 116

H.4 *ASN.1 Types*

AdditionalBodyParts 13,55,56,96
AttendantAssistedDeliveryRequest 17,95
Body 13,93,96
BodyPartSynopsis 75,102
BOOLEAN 63,75,102,109
CommonFields 24,27,28,29,97,98,99
COMPONENTS OF per RecipientAutoForwardFields WITH RecipientName ABSENT 64
ConversionEITsField 24,51,97
Criticality 12,93
EntryType 74,102

Page

EOSSupportIndicator 38,39,116
ExpiryTimeField 14,56,94,95
ExtensionField 12,17,20,21,26,27,93,94,95,97,98
ExternallyDefinedBodyPart 23,96,106
ExternallyDefinedParameters 75,102
Filter 63,69,109,110
FirstRecipientField 20,24,25,52,82,95,97,107
ForwardWithMessageAccepted 63,109
ForwardWithMessageNonReceipt 64,110
ForwardWithNonReceipt 63,109
ForwardWithoutVNs 63,68,109,110
ForwardWithRespAccepted 66,110
Heading 13,14,38,93,94,116
HeadingExtensionsField 14,20,94,96
HeadingExtensionsSubField 20,96
HeadingFields 66,110
ID 89
ImportanceField 14,94
InformationObject 11,36,93,115
INTEGER 75,79,102,105
Language 18,95
LanguageField 14,18,94,95
LocalReference 12,93
MessageBodyPartSynopsis 75,102
MessageData 22,56,96
MessageDeliveryTime 22,96
MessageForwarded 14,18,55,94,95
MessageForwarding Permitted 14,18,55,66,94,95,110
MessageParameters 22,56,96
NonMessageBodyPartSynopsis 75,102
NonReceiptNotificationFields 29,96,99
NoticeCreationTimeField 24,25,51,97
NotificationArguments 64,66,110
NotificationExtensions Field 24,27,28,29,31,69,97,98,99,110
NotificationExtensionsSubField 27,52,98
NRN 35,115
OBJECT IDENTIFIER 75,102
ObsoletedVMField 14,17,94,95
ObsoletedVMSubfield 17,95
ORDescriptor 15,82,93,106
OriginatorField 56
ORName 66,110
OtherMessageDeliveryFields 22,96
PerMessageAutoForwardFields 64,66,68,110
PerRecipientArguments 66,68,110
PerRecipientAutoForwardFields 64,66,110
PerRecipientNoAcceptArguments 64,110
PrimaryBodyPart 13,54,56,96
ReceiptNotificationFields 23,27,96,98

	Page
RecipientExtensionsField	15,17,94,95
RecipientExtensionsSubField	17,95
RecipientField	15,24,38,51,64,66,93,94,97,110,116
RecipientsField	14,15,56,94
RecipientsSubField	15,54,94
RN	35,115
SecurityElementsField	24,26,47,49,51,97
SecurityExtensionsField	26,97
SecurityExtensionsSubField	26,97
SensitivityField	14,94
SequenceNumber	75,80,102,106
ServiceNotificationFields	23,28,96,98
ServiceNotificationInfo	69,110
SN	35,115
SNDiagnosticField	28,29,98
SNReasonCode	28,69,98,110
SNReasonCodeField	52
SNReasonField	28,98
SpokenName	12,15,93
SpokenSupplementaryInfo	27,97
SubjectVMField	24,25,51,97
SubjectVMOtherRecipients	26,97
SubscriptionProblem	41,116
ThisVMField	14,15,56,94
VBodyPart	13,21,96
VBPExtensionsSubField	21,96
VBPParameterExtensions	21,96
VBPParameterExtensionsField	21,96
VM-ExternallyDefinedBodyPart	13,23,96
VMActionRegistrationParameter	63,109
VMActionRegistrationParameters	63
VMBodyPart	13,22,96
VMCreationTime	14,19,94,95
VMDuration	21,96
VMExternallyDefinedBodyPart ParameterAttribute	106
VMGSUserSecurityElement	20,95
VMGSUserSecurityElements Field	14,20,26,94,95,97
VMGSUserSecurityExtension	20,95
VMGSUserSecurityExtensions	20,95
VMIdentifier	12,15,17,20,25,93,94,95,97
VMNotificationExtensions	16,17,94
VMNotificationExtensionsSubField	17,94
VMOriginatorField	14,15,25,93,94,97
VMP	11,13,22,34,36,93,96,114,115
VMReceptionSecurity	16,17,47,49,94
VMSpokenSubjectField	14,18,24,26,51,94,95,97
VMSupplementary Information	21,22,27,63,96,97,109
VMSynopsis	75,102
VNIndicatorBits	76,103
VNInitiatorField	25,97
VNOriginatorField	24,25,97
VNotificationIndicator	76,103
VNotificationRequests	16,94
VNotificationRequestsField	15,16,55,56,66,94,110
VNotificationSecurity	16,47,49,94
VNP	11,23,36,93,96,115
VNReceiverField	14,20,58,94,95
VNRNReasonCodeField	29,30,99
VNRNTSAUBasicCodeField	30,100
VNRNTSAUDiagnosticField	30,31,100
VNRNTSAUReasonBasicCodeField	31,100
VNRNTSAUReasonCodeField	30,58,99,100
VNRNUAMSBasicCodeField	30,99
VNRNUAMSDiagnosticField	30,99
VNRNUAMSReasonCodeField	30,58,99
VNRNUserBasicCodeField	30,99
VNRNUserDiagnosticField	30,99
VNRNUserReasonCodeField	30,58,99
VNSNActionRegistrationParameter	68,69,110
VNSNActionRegistrationParameters	68
VNSupplementary Info	24,27,37,38,52,69,97,110,115,116
VNType	35,115
VNVoiceEncodingType	12,24,26,51,97,98
VoiceData	21,96
VoiceEncodingType	14,19,21,26,93,94,95,96,98
VoiceMessageLength	79,105
VoiceParameters	21,96

H.5 *Information Items*

auto-acknowledge-suppl-receipt-info	37
auto-acknowledge-VM	37
auto-discard-expired-VM	37
auto-discard-obsolete-VM	37
auto-forward-comment	38
auto-forward-heading	38
auto-forward-recipients	38
auto-forward-VM	38
auto-service-status-info	39
auto-SN-suppl-recipient-info	39
auto-VSNotice	39
Body	13
Change auto-acknowledgement	37
Change auto-discard	37
Change auto-forwarding	37
Change auto-SNotice	38
Content	34,35,36,37
Envelope	34,35,36,37
Extensions	17
Externally Defined Body Part	23
mts-eos-not-subscribed	41

	Page
Originate VM	34
Originate VN	35
problem	41
Receive Report	36
Receive VM	36
Receive VN	36
Recipient	15
Recipient Improperly Specified	41
Submission-identifier	34,35
Submission-time	34,35
Undelivered-object	36
user	12
user-relative-identifier	12
vm-eos-not-subscribed	41
VM-Externally Defined	13

H.6 *MS ASN.1 Types*

Number	75
Parameters	75
Processed	75
Size	75
Synopsis	75
Type	75
VM Entry Type	74
VM Synopsis	75

H.7 *MS Attributes*

body	79,105
conversion-indication	82,107
externally-defined-body-part-types	80,106
first-recipient	82,107
heading	76,103
heading-extensions	78,105
importance	77,103
language	77,104
message-data	80,106
message-parameters	80,106
notification-creation-time	82,107
notification-extensions	83,107
notification-security-elements	82,107
nrn-extensions	84,108
nrn-tsau-reason-code	84,108
nrn-ua-ms-reason-code	83,108
nrn-user-reason-code	84,108
obsoleted-vm	77,103
orig-vm-spoken-subject	82,107
originator	76,103
recipient-extensions-for-this-recipient	79,105
rn-extensions	83,107
sensitivity	77,103
sn-extensions	83,108
sn-reason-code	83,108
subject-vm	81,106
subject-vm-other-recipients	82,107
this-recipient	78,105
this-vm	76,103
v-body-message-length	79,105
v-body-part	80,106
vm-bodypart-sequence-number	80,106
vm-creation-time	77,104
vm-encrypted-primary-bodypart	78,104
vm-entry-type	74,102
vm-forwarded-indication	77,103
vm-forwarding-permitted	77,104
vm-notification-extension-requests-for-this-recipient	79,105
vm-notification-indicator	76,103
vm-notification-requests-for-this-recipient	78,105
vm-notification-security-requests-for-this-recipient	78,105
vm-originator-spoken-name	78,104
vm-reception-security-requests-for-this-recipient	79,105
vm-spoken-subject	78,104
vm-synopsis	75,102
vm-this-recipient-spoken-name	78,104
vmgs-user-security-element	77,104
vmgs-user-security-extensions	78,104
vn-initiator	82,107
vn-originator-spoken-name	82,107
vn-originator-text	82,106
vn-receiver-field	77,104
vn-supplementary-info	82,107
vn-voice-encoding-type	82,107
voice-message-duration	79,106
voice-message-other-parms	80,106

H.8 *MS VM Auto Actions*

vm-auto-forward-actions	63,109
vm-auto-sn-action	69,110

H.9 *NRN Reason Codes*

attendant-assisted-delivery-failure	31,100
attendant-assited-delivery-not-provided	31,100
auto-forwarded	30,99
busy-on-every-attempt	31,100
can-not-pass-to-mhs-user	30,99
delivery-timeout	30,31,99,100
forwarding-error	30,99
importance-not-conveyed	31,100
importance-not-supported	31,100
inappropriate-voice-encoding	31,100
message-discarded	30,99
message-expired	31,100
message-forwarded	30,31,99,100

no-answer-on-every-attempt 31,100
no-answer-or-busy-on-every-attempt 31,100
number-not-in-service 31,100
person-to-person-spoken-name-not-provided 31,100
person-unavailable 31,100
protocol-violation 30,99
recipient-refused-message 31,100
security-error 30,31,99,100
sensitivity-not-supported 31,100
subscription-terminated 30,99
telephone-number-unreachable 31,100
unknown-telephone-number 31,100
unspecified 30,31,99,100
user-defined-reason 30,99
voice-encoding-not-supported 30,99

H.10 *Object Identifiers*

id-act 89
id-act-vm-auto-actions 63,109
id-act-vmg-auto-forward 90,109
id-act-vn-sn-auto-action 69,90,110
id-bat 89
id-bat-body 79,91,105
id-bat-externally-defined-body 80,106
id-bat-externally-defined-body-part-types 91
id-bat-ipm-data 91
id-bat-ipm-parameters 91
id-bat-message-data 80,91,106
id-bat-message-parameters 80,91,106
id-bat-v-body-message-length 79,91,105
id-bat-v-body-part 80,91,106
id-bat-vm-bodypart-sequence 80,106
id-bat-vm-bodypart-sequence-number 91
id-bat-voice-message-duration 91,106
id-bat-voice-message-other-parms 91,106
id-bp 89
id-bp-g721-32k-adpcm 19,90
id-bp-private-octet 19,90
id-bp-undefined-octet 19,90
id-dir 89
id-for 89
id-hat 89
id-hat-expiry-time 77,90,103
id-hat-heading 76,90,103
id-hat-heading-extensions 78,91,105
id-hat-importance 77,90,103
id-hat-language 77,91,104
id-hat-obsoleted-vm 77,90,103
id-hat-originator 76,90,103
id-hat-sensitivity-field 77,91,103
id-hat-this-VM 76,103
id-hat-this-vm 90

Page
id-hat-vm-creation-time 77,90,91,104
id-hat-vm-encrypted-primary 78,91,104
id-hat-vm-forwarded-indication 77,91,103
id-hat-vm-forwarding-permitted 77,91,104
id-hat-vm-originator-spoken-name 78,90,104
id-hat-vm-spoken-subject 78,90,104
id-hat-vm-this-recipient-spoken-name 78,90,104
id-hat-vmgs-user-security 77,91,104
id-hat-vmgs-user-security-extensions 78,91,104
id-hat-vn-receiver 77,91,104
id-hat-voice-encoding-type 77,91,104
id-ipe 89
id-mct 89
id-mct-pvm 59,62,85,90
id-mod 89
id-mod-abstract-service 89
id-mod-functional-objects 89
id-mod-information-objects 89
id-mod-message-store-attributes 89
id-mod-message-store-auto-actions 89
id-mod-object-identifiers 89
id-mod-upper-bounds 89
id-mod-vmg-directory-cl-att 89
id-nat 89
id-nat-conversion-indication 82,91,107
id-nat-first-recipient 82,91,107
id-nat-notification-creation 82,107
id-nat-notification-creation-time 91
id-nat-notification-extensions 83,91,107
id-nat-notification-security-elements 82,91,107
id-nat-nrn-extensions 84,91,108
id-nat-nrn-tsau-reason-code 84,91,108
id-nat-nrn-ua-ms-reason-basic-code 83,91,108
id-nat-nrn-user-reason-basic-code 84,91,108
id-nat-orig-vm-spoken-subject 82,107
id-nat-orig-vmg-spoken-subject 91
id-nat-rn-extensions 83,91,107
id-nat-sn-extensions 83,91,108
id-nat-sn-ua-ms-basic-reason-code 83,91,108
id-nat-subject-vm 81,91,106
id-nat-subject-vm-other-recipients 82,91,107
id-nat-vn-initiator 82,91,107
id-nat-vn-originator 82,91,106,107
id-nat-vn-originator-spoken-name 91
id-nat-vn-originator-text 91
id-nat-vn-supplementary-info 82,107
id-nat-vn-supplementary-information 91
id-nat-vn-voice-encoding-type 82,91,107
id-nt 89
id-nt-vmg-nrn 90
id-nt-vmg-rn 90
id-nt-vmg-sn 90
id-ot 89

	Page
id-ot-tsau	44,90,113
id-ot-vmg-ms	43,90,113
id-ot-vmg-ua	43,90,112
id-ot-vmge	31,90,111
id-ot-vmgs	32,90,112
id-ot-vmgs-user	32,90,112
id-pt	89
id-pt-management	33,90,114
id-pt-origination	33,90,114
id-pt-reception	33,90,114
id-rat	89
id-rat-recipient-extensions-for-this-recipient	79,91,105
id-rat-this-recipient	78,91,105
id-rat-vm-notification-extension-requests-for-this-recipient	79,91,105
id-rat-vm-notification-requests-for-this-recipient	78,105
id-rat-vm-notification-security-requests-for-this-recipient	78,91,105
id-rat-vm-reception-security-requests-for-this-recipient	79,91,105
id-ref	89
id-ref-primary	31,90,112
id-ref-secondary	42,90,112
id-sat	89
id-sat-vm-notification-indicator	76,90,103
id-sat-vm-synopsis	75,90,102
id-sat-vmg-entry-type	74,90,102
id-syn	89
id-syn-place-holder	90
id-syn-removed	90
id-vmg	89
id-vmg-g721-32k-adpcm	14,94
id-vmgs	89

H.11 *P1 ASN.1 Types*

Content	26,50,97
content-confidentiality-algorithm-identifier	65,66
content-type	65,67
ContentCorrelator	47,49,50
ContentIdentifier	47,49,50
ContentIntegrityCheck	26,47,49,97
ContentType	47,49
conversion-prohibited	52
DeferredDeliveryTime	49,58
DL-expansion-prohibited	47,67
message-origin-authentication-check	65,66
message-security-label	65,66
MessageDeliveryEnvelope	36,50,115
MessageOriginAuthenticationCheck	49
MessageSubmissionEnvelope	34,35,45,47,114,115
MessageSubmissionIdentifier	34,35,47,49,114,115
MessageSubmissionTime	34,35,47,49,114,115
original-encoded-information-types	65,67
OriginalEncodedInformation Types	47,49
ORName	12,20,25,45,47,93,95,97
PerMessageIndicators	24,49,50,53,58
PerRecipientMessageSubmissionFields	47,58
Priority	49,58
priority	52
proof-of-submission-request	65,67
RecipientImproperlySpecified	34,35,38,114,115,116
ReportDeliveryEnvelope	36,50,115

H.12 *SN Reason Codes*

attendand-assisted-delivery-request	28,39,98,116
auto-forwarding-ind	28,39,98,116
body-part-encryption-ind	28,39,98,116
expiry-date-ind	28,39,98,116
language-ind	28,39,98,116
language-national-usage-problem	29,98
local-security-not-supported	29,98
obsoleting-ind	28,39,98,116
unspecified	28,39,98,116
vm-language-not-understood	29,98
vm-unsupported-voice-encoding	29,98

H.13 *Terms*

32K ADPCM Encoded Message	13
ASN.1 definitions	10
Body	79
Encoded Information Types	85
First Recipient	25
forwarding VM	54
Heading	76
Management Port	33
message	13
Message Transfer System	44
message store	43
Non-Receipt Notification	29
non-receipt notification	23
notification	23
OR Descriptor	12
Origination Port	33
primary object	31
primary port	33
Pvm	6
Receipt Notification	27

	Page
receipt notification	23
Reception Port	33
secondary object	42
secondary port	44
Service Notification	28
service notification	23
subject VM	10
subject VM	23
support upon origination	86
support upon reception	86
Telephone Service access unit	43
user	10
user agent	43
VM Externally Defined Body Part Types	80
VM Forwarding	54,55,62
vm-forward-without-VNs	62
vm-forwarding-with-NRN	62
vm-forwarding-with-RN	62
VM Forwarding	53
VM Identifier	12
VMGS Abstract Service	34
VMGS user agent	43
VMGS-user	10
VMGS message store	43
Voice applications	32
Voice messaging user	32
Voice Message	13
Voice Messaging	84
Voice Messaging Environment	31
Voice Messaging System	32
voice messaging system user	10
Voice Notification	23

H.14 *Upper-Bounds*

ub-bit-options	16,17,76,92,94,103
ub-integer-options	92
ub-recipients	92
ub-sn-reasons	117
ub-supplementary-info-length	22,96
ub-vmg-local-reference	12,92,93,117
ub-vmg-spoken-name	117
ub-vmg-spoken-subject	92,117
ub-vmg-spoken-supplemental-info	27,97,117
ub-vmgs-spoken-name	92
ub-vmgs-user-security-elements	20,92,95,117
ub-vn-reason-code	29,30,31,92,98,99,100

H.15 *VMGS Abstracts*

management	33,114
mTS	42
origination	33,114
reception	33,114
tsau	42,44,112,113
vmg-ms	42,43,112,113
vmg-ua	42,43,112
vmge	31,111,112
vmge-refinement	31,112
vmgs	31,32,42,112
vmgs-refinement	42,112
vmgs-user	31,32,112

H.16 *VMGS Error Reason Codes*

mts-eos-not-subscribed	41,116
vm-eos-not-subscribed	41,116

H.17 *VMGS Operation Errors*

SubscriptionError	34,35,37,38,41,114,115,116

H.18 *VMGS Operations*

ChangeAutoAcknowledge	33,114
ChangeAutoAcknowledgements	37,115
ChangeAutoDiscard	33,37,114,115
ChangeAutoForward	33,114
ChangeAutoForwarding	38,116
ChangeAutoSNotice	38,116
ChangeAutoVSNotice	33,114
management	31,32,112
OriginateVM	33,34,45,50,114
OriginateVN	33,35,114,115
origination	31,32,112
ReceiveReport	33,36,114,115
ReceiveVM	33,36,114,115
ReceiveVN	33,36,114,115
reception	31,32,112