INTERNATIONAL STANDARD

ISO/IEC 10021-9

First edition
1995-08-01

Information technology — Message Handling Systems (MHS) —

Part 9: Electronic Data Interchange Messaging System

Technologies de l'information — Systèmes de messagerie (MHS) — Partie 9: Système de messagerie avec échange de données informatisé

ISO IEC

Reference number
ISO/IEC 10021-9:1995(E)

Contents

ISO/IEC Copyright Office • Case postale 56 • CH-1211 Genève 20 • Switzerland

Printed in Switzerland

ANNEXES

TABLES

FIGURES

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

International Standard ISO/IEC 10021-9 was prepared by ITU-T (as CCITT Recommendation X.435) and was adopted, under a special "fast-track procedure", by Joint Technical Committee ISO/IEC JTC 1, *Information technology,* in parallel with its approval by national bodies of ISO and IEC.

ISO/IEC 10021 consists of the following parts, under the general title *Information technology — Message Handling Systems (MHS)*:

— *Part 1: System and Service Overview*

— *Part 2: Overall Architecture*

— *Part 3: Abstract Service Definition Conventions*

— *Part 4: Message Transfer System: Abstract Service Definition and Procedures*

— *Part 5: Message Store: Abstract Service Definition*

— *Part 6: Protocol Specification*

— *Part 7: Interpersonal Messaging System*

— *Part 8: Electronic Data Interchange Messaging Service*

— *Part 9: Electronic Data Interchange Messaging System*

Annexes A to J form an integral part of this part of ISO/IEC 10021. Annexes K, L and M are for information only.

Introduction

This part of ISO/IEC 10021 is one of a number of parts of ISO/IEC 10021 (the International Standard for Message Handling Systems (MHS)).

MHS provides for the exchange of messages between users on a store-and-forward basis. A message submitted by one user (the *originator*) is transferred through the Message Transfer System (MTS), and delivered to one or more other users (the *recipients*). A user may interact directly with the MTS, or indirectly via a message store (MS).

The MTS comprises a number of message-transfer-agents (MTAs), which transfer messages and deliver them to their intended recipients.

This part of ISO/IEC 10021 was developed and published by the ITU-T in 1991. The equivalent ITU-T document is CCITT Recommendation X.435 (1991) as amended by the MHS Implementor's Guide (version 12).

Information technology - Message Handling Systems (MHS) -

Part 9 :
Electronic Data Interchange Messaging System

Section 1 - Introduction

1 Scope

This part of ISO/IEC 10021 is one of a set of standards for message handling. The entire set provides a comprehensive blueprint for a message handling system (MHS) realized by any number of cooperating open systems.

NOTE - The Message-Oriented Text Interchange System (MOTIS) was formerly the title of 10021:1990 parts and has been superseded by amendment to become Message Handling Systems (MHS). MHS is also published by the ITU-T as part of the X.400 series of Recommendations.

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 communication systems of other kinds (e.g., postal systems). A user is assisted in the preparation, storage, and display of messages by a user agent (UA). Optionally, it is assisted in the storage of messages by a message store (MS). The MTS comprises a number of message transfer agents (MTA) which collectively perform the store-and-forward message transfer function.

This part of ISO/IEC 10021 defines the message handling application called EDI messaging (EDIMG), a form of message handling tailored for exchange of electronic data interchange (EDI) information, a new message content type and associated procedures known as P_{edi}. It is designed to meet the requirements of users of ISO 9735 (EDIFACT), and other commonly used EDI systems.

This part of ISO/IEC 10021 is one of a series on message handling. ISO/IEC 10021-2 | CCITT Recommendation X.402 constitutes the introduction to the series and identifies the other documents in it.

The architectural basis and foundation for message handling are defined in still other parts. ISO/IEC 10021-2 | CCITT Recommendation X.402 identifies those documents as well.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 10021. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this part of ISO/IEC 10021 are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO/IEC 8824:1990, *Information technology – Open Systems Interconnection – Specification of Abstract Syntax Notation One (ASN.1).*
(See also CCITT Recommendation X.208 (1988))

ISO/IEC 8825:1990, *Information technology – Open Systems Interconnection – Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1).*
(See also CCITT Recommendation X.209 (1988))

ISO/IEC 8859, *Information processing - 8-bit single-byte coded graphic character sets.*

ISO/IEC 9594-1:1990, *Information technology – Open Systems Interconnection – The Directory - Part 1: Overview of concepts, models, and service.*
(See also CCITT Recommendation X.500 (1988))

ISO/IEC 9594-2:1990, *Information technology – Open Systems Interconnection – The Directory – Part 2: Models.*
(See also CCITT Recommendation X.501 (1988))

ISO/IEC 9594-3:1990, *Information technology – Open Systems Interconnection – The Directory – Part 3: Abstract service definition.*
(See also CCITT Recommendation X.511 (1988))

ISO/IEC 9594-4:1990, *Information technology – Open Systems Interconnection – The Directory – Part 4: Procedures for distributed operation.*
(See also CCITT Recommendation X.518 (1988))

ISO/IEC 9594-5:1990, *Information technology – Open Systems Interconnection – The Directory – Part 5: Protocol specifications.*
(See also CCITT Recommendation X.519 (1988))

ISO/IEC 9594-6:1990, *Information technology – Open Systems Interconnection – The Directory – Part 6: Selected attribute types.*
(See also CCITT Recommendation X.520 (1988))

ISO/IEC 9594-7:1990, *Information technology – Open Systems Interconnection – The Directory – Part 7: Selected object classes.*
(See also CCITT Recommendation X.521 (1988))

ISO/IEC 9594-8:1990, *Information technology – Open Systems Interconnection – The Directory – Part 8: Authentication framework.*
(See also CCITT Recommendation X.509 (1988))

ISO 9735:1988, *Electronic data interchange for administration, commerce and transport (EDIFACT) - Application level syntax rules.*

ISO 9735:1988/Amd.1:1990, *Electronic data interchange for administration, commerce and transport (EDIFACT) - Application level syntax rules – Amendment 1.*

ISO/IEC 10021-1:1990, *Information technology – Text Communication – Message-Oriented Text Interchange Systems (MOTIS) - Part 1: System and Service Overview.*
(See also CCITT Recommendation F.400 (1992) | X.400 (1993))

ISO/IEC 10021-2:1990, *Information technology – Text Communication – Message-Oriented Text Interchange Systems (MOTIS) - Part 2: Overall Architecture.*
(See also CCITT Recommendation X.402 (1992))

ISO/IEC 10021-3:1990, *Information technology – Text Communication – Message-Oriented Text Interchange Systems (MOTIS) - Part 3: Abstract Service Definition.*
(See also CCITT Recommendation X.407 (1988))

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.*
(See also CCITT Recommendation X.411 (1992))

ISO/IEC 10021-5:1994, *Information technology – Text Communication – Message-Oriented Text Interchange Systems (MOTIS) - Part 5: Message Store: Abstract Service Definition.*
(See also CCITT Recommendation X.413 (1992))

ISO/IEC 10021-6:1990, *Information technology – Text Communication – Message-Oriented Text Interchange Systems (MOTIS) - Part 6: Protocol Specifications.*
(See also CCITT Recommendation X.419 (1992))

ISO/IEC 10021-7:1990, *Information technology – Text Communication – Message-Oriented Text Interchange Systems (MOTIS) - Part 7: Interpersonal Messaging System.*
(See also CCITT Recommendation X.420 (1992))

ISO/IEC 10021-8:1995, *Information technology – Message Handling Systems (MHS) - Part 8: Electronic Data Interchange Messaging Service.*
(See also CCITT Recommendation F.435 (1991))

CCITT Recommendation X.408 (1988), *Message handling systems: Encoded information type conversion rules.*

ANSI X12.5-1987: *American National Standard for electronic business data interchange - interchange control structure.*

NOTES

1 - ITU-T Recommendations are published subsequent to March 1993 and CCITT Recommendations were published prior to March 1993. The term *CCITT Recommendation* is used because all of the editions referred to in this part of ISO/IEC 10021 were published before March 1993.

2 - ISO/IEC 8859 is a multi-part standard.

3 Definitions

3.1 Common definitions for MHS

This part of ISO/IEC 10021 uses the following terms defined in ISO/IEC 10021-1 | CCITT Recommendation X.400, ISO/IEC 10021-2 | CCITT Recommendation X.402 and ISO/IEC 10021-5 | CCITT Recommendation X.413:

a) Access unit

b) Body

c) Content

d) Distribution list

e) Encoded information type

f) Envelope

g) Message handling system

h) Message-oriented text interchange system (defined and used prior to 1993)

i) Message store

j) Message transfer agent

k) Message transfer system

l) Physical delivery access unit

m) Recipient

n) Submission identifier

o) Submission time

p) Synopsis

q) Telematic agent

r) Telex access unit

s) User

t) User agent

3.2 Common definitions for abstract syntax notation one

This part of ISO/IEC 10021 uses the full extent of the abstract syntax notation one (ASN.1) as defined in ISO/IEC 8824 | CCITT Recommendation X.208.

3.3 Terms imported from EDI service

This part of ISO/IEC 10021 uses the following terms defined in ISO/IEC 10021-8 | CCITT Recommendation F.435:

a) EDI forwarding

b) EDI message

c) EDI notification

d) EDI user

e) EDIM responsibility

3.4 Other EDI definitions

The terms listed below may assume different meanings in the standards referenced.

3.4.1 Terms imported from EDI for administration, commerce and transport

This part of ISO/IEC 10021 uses the following terms defined in ISO 9735 (EDIFACT - Application level syntax rules):

- Acknowledgement request
- Address for reverse routing
- Application reference
- Communications agreement
- Component data element separator
- Data element separator
- Date/time of preparation
- Decimal notation
- Functional group header
- Identification code
- Identification code qualifier
- Interchange control reference
- Interchange control header
- Interchange recipient
- Interchange sender
- Message header
- Processing priority code
- Recipient identification code
- Recipients reference qualifier
- Recipients reference, password
- Release indicator
- Routing address
- Segment terminator
- Sender identification
- Service string advice
- Syntax identifier
- Syntax version
- Test indicator
- UNA segment
- UNB segment

- UNH segment

3.4.2 Terms imported from United Nations trade data interchange

This part of ISO/IEC 10021 uses the following terms defined in the United Nations trade data interchange (UNTDI) syntax rules (developed from the earlier syntax Recommendation UNGTDI), unanimously accepted by the United Nations Economic Commission for Europe, Working Party 4, in September 1985:

- Application reference
- Date and time of transmission
- Message header
- MHD segment
- Recipients reference/password
- Start of transmission
- Transmission priority code
- Transmission recipient
- Transmission sender

3.4.3 Terms imported from ANSI X12.5

- Authorization information qualifier
- Authorization information
- Functional group header
- Interchange date
- Interchange header
- Interchange receiver id
- Interchange sender id
- Interchange time
- ISA segment
- Test indicator
- Transactional set header, ST segment

3.5 EDI messaging system definitions

For the purposes of this part of ISO/IEC 10021, the following definitions apply.

3.5.1 EDI message store: A specialized message store for the purposes of EDI messaging.

3.5.2 EDI messaging system: The functional object by means of which all users communicate with one another in EDI messaging.

3.5.3 EDI user agent: A specialized user agent for the purposes of EDI messaging.

4 Abbreviations

ANSIX12	American National Standards Institute, Accredited Committee X12
AU	Access unit
DL	Distribution list
EDI	Electronic data interchange
EDI-MS	EDI message store
EDI-UA	EDI user agent

EDIFACT	Electronic data interchange for administration, commerce and transport
EDIM	EDI message
EDIME	EDI messaging environment
EDIMG	EDI messaging
EDIMG user	EDI messaging user
EDIMS	EDI messaging system
EDIN	EDI notification
EIT	Encoded information type
FN	Forwarded notification
MD	Management domain
MHS	Message handling system
MS	Message store
MTA	Message transfer agent
MTS	Message transfer system
NN	Negative notification
PDAU	Physical delivery access unit
PDS	Physical delivery system
PN	Positive notification
TLMA	Telematic agent
UA	EDI user agent
UNTDI	United Nations/trade data interchange

5 Conventions

5.1 Terms

Throughout the rest of this part of ISO/IEC 10021, terms that refer to ASN.1 types are written with initial upper-case letters for all words in the ASN.1 type (for example, ORName or OR Name).

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 a normative annex, the definition in the annex shall be used. ASN.1 notation is defined in ISO/IEC 8824 | CCITT Recommendation X.208.

This part of ISO/IEC 10021 uses for the indicated purposes the following ASN.1-based descriptive conventions:

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

b) to define the functional objects of EDI Messaging, the OBJECT and REFINE macros of ISO/IEC 10021-3 | CCITT Recommendation X.407;

c) to define the abstract service of EDI Messaging, the PORT and ABSTRACT-operation and ERROR macros of ISO/IEC 10021-3 | CCITT Recommendation X.407;

d) to define the protocol extensions, the EDIM-EXTENSION macro of this part of ISO/IEC 10021;

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

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

g) to define MS attributes, the ATTRIBUTE macro of ISO/IEC 9594-2 | CCITT Recommendation X.501.

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.

5.3 Conventions for Attribute Types in Table 1

This part of ISO/IEC 10021 uses the conventions listed below in its definition of attribute types for the MS abstract services.

For the columns headed “Single/Multi-valued” the following values can occur:

- S: single-valued,
- M: multi-valued.

For the columns 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 EDIM”, “Presence in delivered PN”, “Presence in delivered NN” and “Presence in delivered FN”, 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.
- - a hyphen (-) indicates “always absent”, otherwise.

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

- N: No
- Y: Yes

5.4 Conventions for Attribute Types in Table 2

This part of ISO/IEC 10021 uses the conventions listed below in its definition of attribute types for the MS abstract services.

For the columns headed “Source generated by”, the following values can occur:

- MD: Message Delivery abstract-operation
- MS: Message Store
- RD: Report Delivery abstract-operation

6 Information objects

The information objects that users exchange in EDI messaging are of two kinds: EDI messages (EDIM), and EDI notifications (EDIN).

NOTE – The EDI messaging user (*EDIMG user*) is normally an EDI application or computer process, not a person. For brevity, the term *user* is used throughout the rest of this part of ISO/IEC 10021 with the meaning of *EDIMG user*.

```
InformationObject ::= CHOICE {
  edim                    [0] EDIM,
  edin                    [1] EDIN }
```

7 Common data types

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

7.1 EDIM identifier

An EDIM Identifier is an information item that unambiguously, globally and forever uniquely identifies an EDIM.

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 EDIM unique.

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

NOTE – OR-name is defined in 8.5.5 of ISO/IEC 10021-4 | CCITT Recommendation X.411 and ORName is defined in figure 2 of ISO/IEC 10021-4 | CCITT Recommendation X.411.

The EDIM Identifier shares the same value set with the IPM Identifier defined in ISO/IEC 10021-7 | CCITT Recommendation X.420. Therefore, an EDI user agent or EDI message store that is capable of handling both IPM and EDIM shall make sure that the Local Reference is unique both for IPMs and EDIMs.

An EDIM Identifier has the following components:

a) *User:* Identifies the user who originates the EDIM. One of the user's OR-names.

b) *User-relative-identifier:* Unambiguously identifies the EDIM, distinguishing it from all other EDIMs that the user who is identified by the User component originates. Its syntax is that of Local Reference, a Printable String of from zero to a prescribed number of characters (see annex G). A length of zero is discouraged.

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

7.2 Extensions

A mechanism is provided which allows for future extensions to this part of ISO/IEC 10021.

```
ExtensionField ::= SEQUENCE {
  type                    [0] EDIM-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 Responsibility. An extension marked as non-critical for Responsibility may be ignored or discarded, while an extension marked as critical must be known and performed for acceptance of Responsibility of an EDIM.

NOTE – The term *EDIM Responsibility* is defined in 3.5 of ISO/IEC 10021-8 | CCITT Recommendation F.435. Throughout this document, the term "Responsibility" refers to the term defined in ISO/IEC 10021-8 | CCITT Recommendation F.435, and not to the everyday use of the word.

```
Criticality ::= BOOLEAN
```

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

```
EDIM-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 EDI message

An EDI Message (EDIM) is a member of the primary class of information objects conveyed between users in EDI messaging.

NOTE 1 – The term message when used throughout the rest of this part of ISO/IEC 10021 is a synonym for EDI Message where the context admits.

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

An EDI 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 EDI Message.

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

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

PrimaryBodyPart ::= CHOICE {
  edi-body-part          [0] EDIBodyPart,
  forwarded-EDIM         [1] EDIMBodyPart }

OtherBodyParts ::= SEQUENCE OF EDIM-ExternallyDefinedBodyPart
```

NOTE 2 – EDIM-Externally Defined Body Part is defined in 8.3.3. EDI Body Part is defined in 8.3.1. EDIM Body Part is defined in 8.3.2.

The Body has one Primary Body Part that contains an EDI information object. This body part is either an EDI interchange itself or a forwarded EDIM. Examples of types of EDI information objects are EDI Interchanges defined by ISO 9735, Electronic data interchange for administration, commerce and transport (EDIFACT), by United Nations trade data interchange (UNTDI) and by American National Standards Institute Committee X12 (ANSIX12).

NOTE 3 – The scope of an EDI information object type is rather large and includes for example Privately Defined types. For brevity, the term interchange is used throughout the rest of this part of ISO/IEC 10021 with the meaning of EDI Interchange.

The following rules comply with the requirements stated in 7.4 of ISO/IEC 10021-8 | CCITT Recommendation F.435:

c) When an EDIM is first created, the Primary Body Part shall contain one EDI Body Part.

d) When an EDIM is forwarded, its structure shall comply with the rules given in 17.3.3.2.

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 textual information, voice annotation or graphics to be used in conjunction with the interchange.

The structure of an EDI Message is depicted in figure 1.

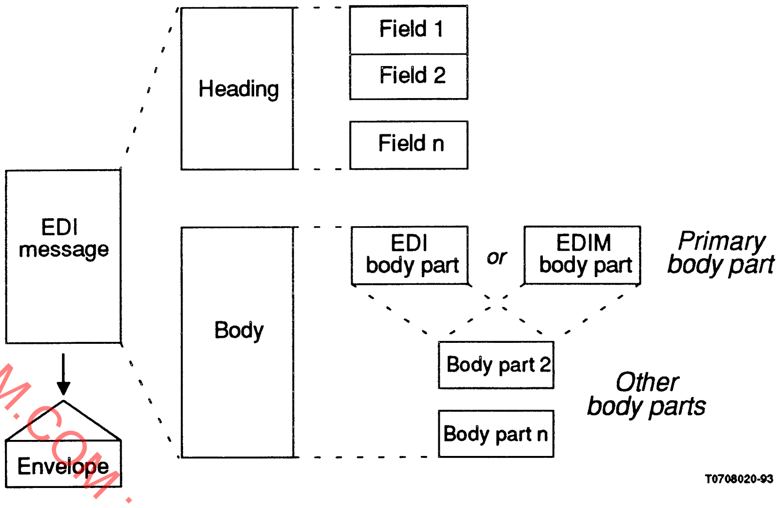

Figure 1 - EDI message structure

8.1 Heading field component types

Information items of several kinds appear throughout the Heading. These common items are defined below.

In the text that follows, reference is made to EDIFACT segments and data elements. Annex K explains this in relation to UNTDI and ANSIX12. Values copied from EDI data elements and represented as T.61 Strings are semantically equivalent to the characters used to form the EDI data elements in EDIFACT, UNTDI and ANSIX12.

8.1.1 Interchange recipient/sender

The Interchange Recipient and Interchange Sender fields have some data types in common. They are defined below.

8.1.1.1 Identification code

The Identification Code identifies a sender/recipient of an interchange. This is semantically identical to the "Sender identification / recipient identification" component of the Interchange sender/recipient of the EDIFACT UNB segment.

```
IdentificationCode ::= TeletexString (SIZE (1..ub-identification-code))
```

8.1.1.2 Identification code qualifier

The Identification Code Qualifier, if present, is a qualifier to the Identification Code of a sender/recipient. This is semantically identical to the "Identification code qualifier" component of the Interchange sender/recipient of the EDIFACT UNB segment.

```
IdentificationCodeQualifier ::=
        TeletexString (SIZE (1..ub-identification-code-qualifier))
```

8.1.1.3 Routing address

The Routing Address, if present, is an address for routing to the sender/recipient specified in the Identification Code. This is semantically identical to the "Address for reverse routing / Routing address" component of the Interchange sender/recipient of the EDIFACT UNB segment.

```
RoutingAddress ::= TeletexString (SIZE (1..ub-routing-address))
```

8.2 Heading fields

The fields that may appear in the Heading of an EDIM are defined and described below.

```
Heading ::= SEQUENCE {
  this-EDIM                        [1] ThisEDIMField,
  originator                       [2] OriginatorField OPTIONAL,
  recipients                       [3] RecipientsField OPTIONAL,
  edin-receiver                    [4] EDINReceiverField OPTIONAL,
  responsibility-forwarded         [5] ResponsibilityForwarded DEFAULT FALSE,
  edi-bodypart-type                [6] EDIBodyPartType DEFAULT {id-bp-edifact-ISO646},
  incomplete-copy                  [7] IncompleteCopyField DEFAULT FALSE,
  expiry-time                      [8] ExpiryTimeField OPTIONAL,
  related-messages                 [9] RelatedMessagesField OPTIONAL,
  obsoleted-EDIMs                  [10] ObsoletedEDIMsField OPTIONAL,
  edi-application-security-elements  [11] EDIApplicationSecurityElementsField OPTIONAL,
  cross-referencing-information      [12] CrossReferencingInformationField OPTIONAL,
  -- Begin Fields from EDIFACT Interchange
  edi-message-type                 [13] EDIMessageTypeField OPTIONAL,
  service-string-advice            [14] ServiceStringAdviceField OPTIONAL,
  syntax-identifier                [15] SyntaxIdentifierField OPTIONAL,
  interchange-sender               [16] InterchangeSenderField OPTIONAL,
  date-and-time-of-preparation [17]  DateAndTimeOfPreparationField OPTIONAL,
  application-reference            [18] ApplicationReferenceField OPTIONAL,
  -- End Fields from EDIFACT
  heading-extensions               [19] HeadingExtensionsField OPTIONAL }
```

NOTE – The names of the Heading fields derived from EDI standards are taken directly from the relevant standards. See also annex K.

8.2.1 This EDIM

The This EDIM Field identifies the EDIM. It comprises an EDIM Identifier which provides a globally and forever unique identification for the EDIM.

```
ThisEDIMField ::= EDIMIdentifier
```

NOTE – EDIM Identifier is defined in 7.1.

8.2.2 Originator

Identifies the EDIM's originator. It comprises an OR-name. If the Originator field is not present in the EDIM Heading on reception, then the *originator-name* of the delivery envelope shall be used to determine the originator of the EDIM (see 8.2.1.1.1.1 of ISO/IEC 10021-4 | CCITT Recommendation X.411).

```
OriginatorField ::= ORName
```

NOTE – OR-name is defined in 8.5.5 of ISO/IEC 10021-4 | CCITT Recommendation X.411 and ORName is defined in figure 2 of ISO/IEC 10021-4 | CCITT Recommendation X.411.

8.2.3 Recipients

The Recipients Field identifies the user(s) and distribution lists (DLs) who are the (preferred) recipient(s) of the EDIM. It comprises a set of Recipients subfields, one for each recipient. If the Recipients field is not present in the EDIM Heading on reception, then the This-recipient-name of the delivery envelope shall be used to determine the recipient of the EDIM (see 8.3.1.1.1.3 of ISO/IEC 10021-4 | CCITT Recommendation X.411).

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

```
RecipientsField ::= SET OF RecipientsSubField
```

The Recipients Sub Field is an information item that identifies a recipient of an EDIM and that may make certain requests of him.

```
RecipientsSubField ::= SEQUENCE {
  recipient                        [1] RecipientField,
  action-request                   [2] ActionRequestField DEFAULT {id-for-action},
  edi-notification-requests-field  [3]  EDINotificationRequestsField OPTIONAL,
  responsibility-passing-allowed [4] ResponsibilityPassingAllowedField DEFAULT FALSE,
  -- Begin Fields from EDIFACT UNB
  interchange-recipient            [5] InterchangeRecipientField OPTIONAL,
  recipient-reference              [6] RecipientReferenceField OPTIONAL,
  interchange-control-reference    [7] InterchangeControlReferenceField OPTIONAL,
  processing-priority-code         [8] ProcessingPriorityCodeField OPTIONAL,
acknowledgement-request        [9] AcknowledgementRequestField DEFAULT FALSE,
communications-agreement-id    [10] CommunicationsAgreementIdField OPTIONAL,
test-indicator                 [11] TestIndicatorField DEFAULT FALSE,
-- End Fields from EDIFACT UNB
-- Begin Fields from ANSIX12 ISA
authorization-information      [12]  AuthorizationInformationField OPTIONAL,
-- End Fields from ANSIX12 ISA
recipient-extensions           [13] RecipientExtensionsField OPTIONAL }
```

The Recipients subfield has the following components:

8.2.3.1 Recipient

A Recipient identifies the preferred recipient in question. It comprises an OR-name.

```
RecipientField ::= ORName
```

NOTE – OR-name is defined in 8.5.5 of ISO/IEC 10021-4 | CCITT Recommendation X.411 and ORName is defined in figure 2 of ISO/IEC 10021-4 | CCITT Recommendation X.411.

8.2.3.2 Action request

An Action Request indicates what action the originator requests from the recipient. Its value is an object identifier.

```
ActionRequestField ::= OBJECT IDENTIFIER
```

The following standard values have object identifiers defined in this part of ISO/IEC 10021:

- For Action,
- Copy.

The absence of this field shall be interpreted as having the default value set to *For Action*.

NOTE – Additional values for this field can be defined by any interested parties.

8.2.3.3 EDI notification requests

The EDI Notification Requests component (Default: no notifications, no notification security and no reception security) may make certain requests of the preferred recipient denoted by the Recipient field.

NOTE 1 – The fact that a message can be redirected or forwarded is reflected in the word "preferred" above.

```
EDINotificationRequestsField ::= SEQUENCE {
  edi-notification-requests        [0] EDINotificationRequests DEFAULT {},
  edi-notification-security        [1] EDINotificationSecurity DEFAULT {},
  edi-reception-security           [2] EDIReceptionSecurity DEFAULT {}  }

EDINotificationRequests ::= BIT STRING {
  pn(0),
  nn(1),
  fn(2) } (SIZE (0..ub-bit-options))

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

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

NOTE 2 – Only the following combinations of EDI Notification Security and EDI Reception Security bits have a defined behaviour:

EDI Notification Security {proof(0)} *and* EDI Reception Security {proof(0)};
EDI Notification Security {non-repudiation(1)} *and* EDI Reception Security {non-repudiation(1)};
EDI Notification Security {proof(0)} *and* EDI Reception Security { };
EDI Notification Security {non-repudiation(1)} *and* EDI Reception Security { };
EDI Notification Security { } *and* EDI Reception Security { }.

The EDI Notification Requests field consists of a sequence of three optional bit strings of which the first selects the type of notification, the second selects what 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 EDIM by the recipient. EDI Notification Security and EDI Reception Security shall not be requested if EDI Notifications are not requested.

The EDI Notification Requests bit string may assume any of the following values simultaneously.

a) *PN:* A notification of acceptance of Responsibility is requested in the circumstances prescribed in clause 9.

b) *NN:* A notification of refusal of Responsibility for a message is requested in the circumstances prescribed in clause 9.

c) *FN:* A forwarded notification is requested in the circumstances prescribed in clause 9.

The absence of the EDI Notification Requests bit string implies that no EDI Notification requests are made.

The EDI Notification Security bit string may assume any of the following values simultaneously. Each of these values places requirements as indicated below on an EDI-UA submitting a subsequent EDIN in response to the EDI Notification Requests.

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

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

The absence of the EDI Notification Security bit string implies that no EDI Notification Security requests are made.

The EDI Reception Security bit string may assume any of the following values simultaneously. Each of these values places requirements as indicated below on an EDI-UA submitting a subsequent EDIN in response to the EDI Notification Requests.

f) *Proof:* When submitting the EDIN 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 Message-submission-argument as defined in 8.2.1.1.1.26, 8.2.1.1.1.28 and 8.2.1.1.1.29 of ISO/IEC 10021-4 | CCITT Recommendation X.411.

g) *Non-repudiation:* When submitting the EDIN 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, 8.2.1.1.1.28 and 8.2.1.1.1.29 of ISO/IEC 10021-4 | CCITT Recommendation X.411.

The absence of the EDI Reception Security field implies that no EDI Reception Security requests are made.

NOTE 3 – Security services are available only if the MTS supports secure messaging.

8.2.3.4 Responsibility passing allowed

The Responsibility Passing Allowed Field indicates that forwarding Responsibility is allowed if this field is set to TRUE. Absence of the field shall be interpreted as the value FALSE.

A recipient of a message with the Responsibility Passing Allowed Field set to FALSE shall originate EDIN's as requested, and shall not forward Responsibility.

```
ResponsibilityPassingAllowedField ::= BOOLEAN -- Default FALSE
```

If allowed, Responsibility may be forwarded to at most one recipient.

8.2.3.5 Interchange recipient

The Interchange Recipient identifies the EDI Interchange recipient. This is semantically identical to the "Interchange recipient" of the EDIFACT UNB segment.

```
InterchangeRecipientField ::= SEQUENCE {
  recipient-identification        [0] IdentificationCode,
  identification-code-qualifier   [1] IdentificationCodeQualifier OPTIONAL,
  routing-address                 [2] RoutingAddress OPTIONAL }
```

NOTE – The above fields are defined in 8.1.1.

8.2.3.6 Recipient reference

The Recipient Reference identifies a reference meaningful to the recipient's EDI application. This is semantically identical to the "Recipient's Reference, Password" of the EDIFACT UNB segment. It consists of two strings.

```
RecipientReferenceField ::= SEQUENCE {
  recipient-reference            [0] RecipientReference,
  recipient-reference-qualifier  [1] RecipientReferenceQualifier OPTIONAL }

RecipientReference ::= TeletexString (SIZE (1..ub-recipient-reference))

RecipientReferenceQualifier ::=
         TeletexString (SIZE (1..ub-recipient-reference-qualifier))
```

8.2.3.7 Interchange control reference

Indicates the Interchange Control Reference as assigned by the Interchange sender. This is semantically identical to the "Interchange control reference" of the EDIFACT UNB segment.

```
InterchangeControlReferenceField ::=
         TeletexString (SIZE (1..ub-interchange-control-reference))
```

8.2.3.8 Processing priority code

Indicates the EDI application Processing Priority Code. This is semantically identical to the "Processing priority code" in the EDIFACT UNB segment. It consists of a string.

```
ProcessingPriorityCodeField ::=
         TeletexString (SIZE (1..ub-processing-priority-code))
```

8.2.3.9 Acknowledgement request

The Acknowledgement Request indicates the request for EDI acknowledgement as indicated by the interchange sender. This is semantically identical to the "Acknowledgement request" in the EDIFACT UNB segment. Its value is a Boolean, where the value TRUE indicates a request for acknowledgement. Absence of this field shall be interpreted as the value FALSE.

```
AcknowledgementRequestField ::= BOOLEAN -- default FALSE
```

8.2.3.10 Communications agreement id

The Communications Agreement Id indicates the type of communications agreement controlling the interchange, e.g. Customs or other agreement. This is semantically identical to the "Communications agreement id" in the EDIFACT UNB segment.

```
CommunicationsAgreementIdField ::=
         TeletexString (SIZE (1..ub-communications-agreement-id))
```

8.2.3.11 Test indicator

Indicates that the EDI Interchange is a test. This is semantically identical to the "test indicator" in the EDIFACT UNB segment. It is a Boolean where the value TRUE indicates that the EDI Interchange is a test. Absence of this field shall be interpreted as the value FALSE.

```
TestIndicatorField ::= BOOLEAN  -- default FALSE
```

8.2.3.12 Authorization information

The Authorization Information indicates who authorized the interchange. This is semantically identical to the "Authorization information" in the ANSIX12 Interchange.

```
AuthorizationInformationField ::= SEQUENCE {
  authorization-information           [0] AuthorizationInformation,
  authorization-information-qualifier [1] AuthorizationInformationQualifier OPTIONAL
  }

AuthorizationInformation ::= TeletexString (SIZE (1..ub-authorization-information))

AuthorizationInformationQualifier ::=
        TeletexString (SIZE (1..ub-authorization-information-qualifier))
```

NOTE – In the above text reference is made to ANSIX12 segments and data elements. Annex K explains this in relation to UNTDI and EDIFACT (ISO 9735), being the other two widely used syntaxes.

8.2.3.13 Recipient extensions

The Recipient Extensions contains extensions to the Recipients subfield.

```
RecipientExtensionsField ::= SET OF RecipientExtensionsSubField

RecipientExtensionsSubField ::= ExtensionField
```

There are no extensions defined in this part of ISO/IEC 10021.

8.2.4 EDIN receiver

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

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

This field shall be present in a forwarded message when the forwarding EDI user agent (EDI-UA) or EDI message store (EDI-MS) forwards Responsibility. This field may be present when the forwarding EDI-UA accepts Responsibility. Rules related to the construction of this field are given in 17.3.3.4.

NOTE 1 – For brevity, the term user agent (UA) is used throughout the rest of this part of ISO/IEC 10021 with the meaning of EDI-UA, and the term message store (MS) is used throughout the rest of this part of ISO/IEC 10021 with the meaning of EDI-MS.

```
EDINReceiverField ::= SEQUENCE {
  edin-receiver-name          [0] ORName,
  original-edim-identifier    [1] EDIMIdentifier OPTIONAL,
  first-recipient             [2] FirstRecipientField OPTIONAL }
```

The "first-recipient" field shall not be present if more than one Recipents Subfield contains EDI Notification Requests.

The "original-edim-identifier" and the "first-recipient" fields shall not be present when the Primary Body Part is an EDI Body Part (that is, when the original originator first creates the EDIM).

NOTE 2 – The Original EDIM Identifier and First Recipient fields are included in order to allow the recipient to construct the EDIN for a forwarded EDIM. See subclauses 9.1 (more specifically 9.1.3) and 17.3.1.1 for rules related to the construction of an EDIN; see 17.3.3.4 for rules related to the First Recipient field when constructing a forwarded EDIM. OR-name is defined in 8.5.5 of ISO/IEC 10021-4 | CCITT Recommendation X.411 and ORName is defined in figure 2 of ISO/IEC 10021-4 | CCITT Recommendation X.411. First Recipient Field is defined in 9.1.3.

8.2.5 Responsibility forwarded

The Responsibility Forwarded field is used to indicate whether Responsibility was forwarded. Absence of this field shall be interpreted as the value FALSE.

```
ResponsibilityForwarded ::= BOOLEAN -- Default FALSE
```

If this field has the value TRUE it indicates to a receiving UA that Responsibility was forwarded. If this field has the value FALSE (or is absent) it indicates to a receiving UA that the security elements of the inner envelope have been checked.

Subject to the security policy in force, the security elements may have been checked when the message was forwarded. However, when Responsibility is accepted, the security elements shall be checked.

NOTE – Rules regarding the use of this field are contained in 17.3.3.1 and 17.3.3.2.

8.2.6 EDI body part type

Indicates the EDI standard and EDI character sets used in the Primary Body Part. It is represented by a single object identifier.

```
EDIBodyPartType ::= OBJECT IDENTIFIER -- default EDIFACT-ISO646
```

The following standard values have object identifiers defined in annex A of this part of ISO/IEC 10021:

- EDIFACT: ISO646|CCITT Recommendation T61|ISO8859|UNDEFINED OCTETS
- ANSIX12: ISO646|CCITT Recommendation T61|EBCDIC|UNDEFINED OCTETS
- UNTDI: ISO646|CCITT Recommendation T61|UNDEFINED OCTETS
- PRIVATE: UNDEFINED OCTETS
- UNDEFINED: UNDEFINED OCTETS

The absence of this field shall be interpreted as having the default value set to EDIFACT, ISO/IEC 646.

NOTES

1 – The character set referred to by the object identifier is that in which both the EDI Body Part, and those Heading fields that are OCTET STRINGS and are derived from the EDI Interchange are encoded, notwithstanding the fact that these types are defined as OCTET STRING.

2 - The PRIVATE and UNDEFINED object identifiers are provided as an interim measure and rely on the existence of bilateral agreements. The PRIVATE object identifier should be used in preference to the UNDEFINED, as it conveys a semantic of being understood according to private arrangements between the communicating parties, i.e., the originator and the intended recipient.

3 - Instead of using one of the object identifiers listed above, a privately defined object identifier may be used indicatng a provatly defined EDI syntax and character set. Such an object identifier should be acquired from a local registration authority and used in accordance with the practices and policies of that registation authority.

The object identifier root is defined in annex A of this part of ISO/IEC 10021 for EDIFACT body types whose character repertoire is encoded as defined in ISO/IEC 8859. ISO/IEC 8859 is composed of several parts, where each part specifics a specific character repertoire. The specific part number shall form the leaf value of the object identifier used in the EDIMG protocol.

This is the same technique used for indicating character repertoires used in IPM's General Text bodypart. For example, an EDIFACT message encoded per ISO/IEC 8859-6 would be represented with the object identifier:

`{joint-iso-ccitt mhs-motis(6) edims(7) id-bp(11) id-bp-edifact-8859(12) iso-8859-6(6)}`, or alternatively, `{ 2 6 7 11 12 6 }`.

The value of the EDI Body Part Type field shall be used in the Encoded Information Types in the MTS abstract operations (in accordance with 19.4). This enables a UA to signal to the MTS what type of EDI standard the EDIM's Primary Body Part complies with. The MTS makes use of this information, if the recipient UA has registered delivery restrictions on Encoded Information Types, to decide if it can deliver the EDIM.

NOTE 4 – The term Encoded Information Type is defined in 8.1 of ISO/IEC 10021-2 | CCITT Recommendation X.402. See also 8.2.1.1.1.23 of ISO/IEC 10021-4 | CCITT Recommendation X.411.

8.2.7 Incomplete copy

The Incomplete Copy field indicates that the forwarded EDIM is an incomplete copy of an EDIM. Its value is a Boolean. This field shall have the value TRUE if body parts are removed when an EDIM is forwarded. The absence of this field shall be interpreted as having the value FALSE.

```
IncompleteCopyField ::= BOOLEAN   -- Default FALSE
```

NOTE – The term "forwarded EDIM" is defined in 17.3.3.

8.2.8 Expiry time

Indicates when the originator considers this EDIM loses its validity. It comprises a date and time (UTC).

```
ExpiryTimeField ::= UTCTime
```

8.2.9 Related messages

Identifies messages, EDIMs or other (for example IPMs), that the originator of this EDIM considers related to it. It comprises a sequence of one or more message references, one for each related message.

```
RelatedMessagesField ::= SEQUENCE OF RelatedMessageReference

RelatedMessageReference ::= CHOICE {
  edi-message-reference          [0] EDIMIdentifier,
  external-message-reference     [1] ExternalMessageReference }

ExternalMessageReference ::= EXTERNAL
```

NOTES

1 – If the related message identifies messages from other services the user component of the message identifier (EDIM Identifier) must be present.

2 – Message identifier values of the referenced message of other service types than EDIMG are carried in the EDIM Identifier field.

8.2.10 Obsoleted EDIMs

The Obsoleted EDIMs Field identifies one or more EDIMs that the present EDIM obsoletes. It is a sequence of subfields, each an EDIM Identifier.

```
ObsoletedEDIMsField ::= SEQUENCE OF ObsoletedEDIMsSubfield

ObsoletedEDIMsSubfield ::= EDIMIdentifier
```

8.2.11 EDI application security elements

The EDI Application Security Elements field allows an EDI application to exchange security elements having an end-to-end significance.

```
EDIApplicationSecurityElementsField ::= SEQUENCE {
  edi-application-security-element     [0] EDIApplicationSecurityElement OPTIONAL,
  edi-encrypted-primary-bodypart       [1] BOOLEAN OPTIONAL,
  edi-application-security-extensions [2] EDIApplicationSecurityExtensions OPTIONAL }

EDIApplicationSecurityElement ::=
          BIT STRING (SIZE (0..ub-edi-application-security-elements))

EDIApplicationSecurityExtensions   ::= SET OF EDIApplicationSecurityExtension

EDIApplicationSecurityExtension    ::= ExtensionField
```

8.2.12 Cross referencing information

The Cross Referencing Information allows an EDI application to reference individual body parts within the same EDIM and within other EDIMs. It contains a set of cross reference data. Its usage is outside the scope of this part of ISO/IEC 10021.

```
CrossReferencingInformationField    ::= SET OF CrossReferencingInformationSubField
```

```
CrossReferencingInformationSubField ::= SEQUENCE {
  application-cross-reference     [0] ApplicationCrossReference,
  message-reference               [1] MessageReference OPTIONAL,
  body-part-reference             [2] BodyPartReference }

ApplicationCrossReference            ::= OCTET STRING

MessageReference                     ::= EDIMIdentifier
```

If the Message Reference is absent, the message referred to is the current one.

NOTES

1 – Body Part Reference is defined in 8.3.3.

2 – The character set used in the Application Cross Reference field is indicated by the value of the field EDI Body Part Type.

8.2.13 EDI message type

Indicates the Message type(s) present in the EDI Interchange. It consists of a set of distinct strings.

NOTE – "Message" is to be understood as message types that are defined in EDI standards and shall not be confused with "message" used elsewhere in this part of ISO/IEC 10021.

```
EDIMessageTypeField ::= SET OF EDIMessageTypeFieldSubField

EDIMessageTypeFieldSubField ::= TeletexString (SIZE (1..ub-edi-message-type))
```

The values for this field shall be:

- EDIFACT: Message Type from the UNH segment;
- ANSIX12: Transaction Set ID from the ST segment;
- UNTDI: Message Type from the MHD segment.

8.2.14 Service string advice

Indicates the Service String Advice of the EDI Interchange. This is semantically identical to the "UNA, Service string advice" of the EDIFACT Interchange.

```
ServiceStringAdviceField ::= SEQUENCE {
  component-data-element-separator   [0] ComponentDataElementSeparator,
  data-element-separator             [1] DataElementSeparator,
  decimal-notation                   [2] DecimalNotation,
  release-indicator                  [3] ReleaseIndicator OPTIONAL,
  reserved                           [4] Reserved OPTIONAL,
  segment-terminator                 [5] SegmentTerminator }

ComponentDataElementSeparator          ::= OCTET STRING (SIZE (1))

DataElementSeparator          ::= OCTET STRING (SIZE (1))

DecimalNotation               ::= OCTET STRING (SIZE (1))

ReleaseIndicator              ::= OCTET STRING (SIZE (1))

Reserved                      ::= OCTET STRING (SIZE (1))

SegmentTerminator             ::= OCTET STRING (SIZE (1))
```

8.2.15 Syntax identifier

Indicates the syntax used. This is semantically identical to the "Syntax identifier" of the EDIFACT UNB segment.

It consists of a sequence of the Syntax Identifier and the Syntax Version.

```
SyntaxIdentifierField ::= SEQUENCE {
  syntax-identifier     SyntaxIdentifier,
  syntax-version        SyntaxVersion }

SyntaxIdentifier ::= TeletexString (SIZE (1..ub-syntax-identifier))
```

```
SyntaxVersion ::= PrintableString (SIZE (1..ub-syntax-version))
```

8.2.16 Interchange sender

Indicates the sender of the EDI Interchange. This is semantically identical to the "Interchange sender" of the EDIFACT UNB segment.

```
InterchangeSenderField ::= SEQUENCE {
  sender-identification          [0] IdentificationCode,
  identification-code-qualifier  [1] IdentificationCodeQualifier OPTIONAL,
  address-for-reverse-routing    [2] RoutingAddress OPTIONAL } -- EDIFACT Routing
Information
```

NOTE – The above fields are defined in 8.1.1.

8.2.17 Date and time of preparation

Indicates the Date/Time of preparation of the EDI Interchange. This is in UTC Time and is derived from the "Date and time of preparation" of the EDIFACT UNB segment. It comprises a UTC Time.

```
DateAndTimeOfPreparationField ::= UTCTime
```

8.2.18 Application reference

Provides a general reference to an application or function. This is semantically identical to the "Application reference" segment of the EDIFACT UNB segment. It consists of a string.

```
ApplicationReferenceField ::= TeletexString (SIZE (1..ub-application-reference))
```

8.2.19 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 part of ISO/IEC 10021.

NOTE – The Heading Extensions may be used to implement the element of service "Services Indication" defined in ISO/IEC 10021-8 | CCITT Recommendation F.435.

8.3 Body part types

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

8.3.1 EDI body part

An EDI Body Part carries a single EDI Interchange.

```
EDIBodyPart ::= OCTET STRING
```

The reference definition of EDI Interchange used is that used by EDIFACT (ISO 9735). Annex K describes equivalent terms in other EDI standards.

8.3.2 EDIM body part

An EDIM Body Part contains an EDIM, and optionally, its delivery envelope. It is used for forwarding of EDIMs. When an EDIM is forwarded, its structure shall comply with the rules given in 17.3.3.2.

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

MessageParameters ::= SET {
  delivery-time        [0] MessageDeliveryTime OPTIONAL,
  delivery-envelope    [1] OtherMessageDeliveryFields OPTIONAL,
  other-parameters     [2] EDISupplementaryInformation OPTIONAL }
-- MessageDeliveryTime and OtherMessageDeliveryFields shall both be present or both
-- be absent. EDISupplementaryInformation is used in the case of ms-auto-actions,
-- see subclause 18.6.

MessageData ::= SEQUENCE {
  heading                Heading,
  body                   BodyOrRemoved }

BodyOrRemoved ::= SEQUENCE {
  primary-or-removed    PrimaryOrRemoved,
  additional-body-partsAdditionalBodyParts OPTIONAL }

PrimaryOrRemoved ::= CHOICE {
  removed-edi-body       [0] NULL,
  primary-body-part      [1] EXPLICIT PrimaryBodyPart }

AdditionalBodyParts ::= SEQUENCE OF CHOICE {
  external-body-part [0] EDIM-ExternallyDefinedBodyPart,
  place-holder       [1] BodyPartPlaceHolder }  -- This type is for Body Part Removal

BodyPartPlaceHolder ::= EDIM-ExternallyDefinedBodyPart  -- Only the data
-- portion of the Externally Defined Body shall be removed.
-- See text in 8.3.2.

EDISupplementaryInformation ::=
        TeletexString (SIZE (1..ub-supplementary-info-length))
```

NOTE – Primary Body Part is defined in clause 8. Body Part Reference is defined in 8.3.3. The data types Message Delivery Time and Other Message Delivery Fields are defined in 8.3.1.1 (and figure 2) of ISO/IEC 10021-4 | CCITT Recommendation X.411.

The Body Part Place Holder shall only be used for removal of body parts, i.e., it indicates a "removed-EDI-body". It may consist of only the Body Part Reference, or a modified Externally Defined Body Part. In the latter case, the Object Identifier and Body Part Reference of the removed body part are preserved; from the parameter (if present) and data portions of the removed body part, only the Object Identifier and the identifier octets of the "encoding" field of the EXTERNAL are preserved. That is, the EXTERNAL type shall have an encoding field of zero length and hence no content.

The delivery envelope shall be present if security services are invoked.

The structure of an EDIM Body Part is depicted in figure 2.

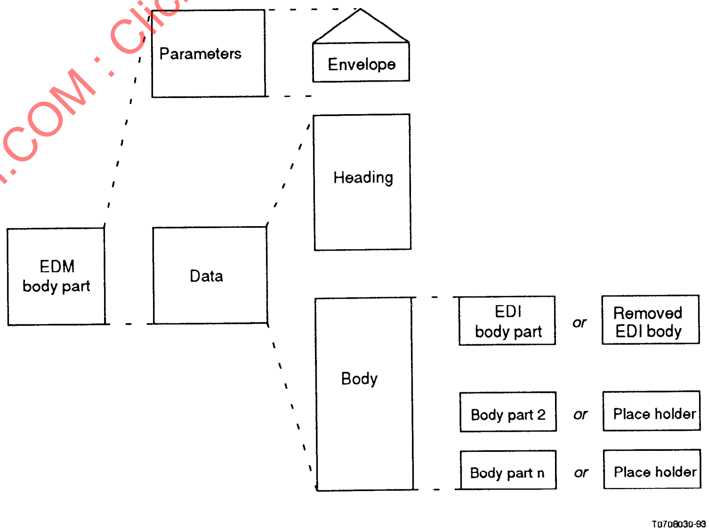

Figure 2 - EDIM body part structure

8.3.3 Externally defined body parts

Additional body parts, that relate to the Primary Body Part, may be carried together with an EDI Body Part. These body parts shall not be or include EDI Interchanges.

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 and optionally a Body Part Reference that may be used for cross-referencing to a body part.

```
EDIM-ExternallyDefinedBodyPart ::= SEQUENCE {
  body-part-reference   [0] BodyPartReference OPTIONAL,
  external-body-part    [1] ExternallyDefinedBodyPart -- from IPMS --}

BodyPartReference ::= INTEGER -- is unique within a EDIM
```

Body-part-reference is assigned when the body part is created, and is not modified subsequently. Its value shall be unique within an EDIM. It shall be present if the originator wishes to cross-reference the body part at creation or in the future.

NOTE – Some Externally Defined body part types are defined in 7.3.12 of ISO/IEC 10021-7 | CCITT Recommendation X.420.

9 EDI notifications

An EDI Notification (EDIN) is a member of a secondary class of Information Object conveyed between users in EDI Messaging.

NOTE – The term *notification* is used throughout the rest of this part of ISO/IEC 10021 as a synonym for EDI Notification.

```
EDIN ::= CHOICE {
  positive-notification    [0] PositiveNotificationFields,
  negative-notification    [1] NegativeNotificationFields,
  forwarded-notification   [2] ForwardedNotificationFields }
```

a) *Positive notification:* An EDIN that reports its originator's acceptance of Responsibility of an EDIM.

b) *Negative notification:* An EDIN that reports its originator's refusal to accept Responsibility of an EDIM.

c) *Forwarded notification:* An EDIN that reports that Responsibility of an EDIM has been forwarded together with the subject EDIM.

The EDIM to which an EDIN refers is called the subject EDIM (see also 17.3.3).

The recipient of the EDIN is the Originator of the subject EDIM, or, if present, the OR-name indicated in the EDIN Receiver field. There shall be at most one recipient specified for an EDIN. There shall be at most one PN, NN or FN originated for each subject EDIM by each recipient of whom notifications are requested (except that an NN may be originated by the same UA subsequent to an FN, in accordance with c) of 17.3.3.1). One FN is originated, if and only if requested, by each recipient that forwards an EDIM. In accordance with the provisions of 17.3.3, the original originator shall receive at most one PN or NN for each recipient of whom notifications were requested, regardless of how many times the EDIM is forwarded, and may receive multiple FNs.

An EDIN consists of Positive, Negative or Forwarded Notification fields. Each of these contains the Common Fields which are described below.

The structure of an EDIN is depicted in figure 3.

9.1 Common fields

The Common Fields are defined and described below.

```
CommonFields ::= SEQUENCE {
  subject-edim                     [1] SubjectEDIMField,
  edin-originator                  [2] EDINOriginatorField,
  first-recipient                  [3] FirstRecipientField OPTIONAL,
  notification-time                [4] NotificationTimeField,
  notification-security-elements   [5] SecurityElementsField OPTIONAL,
  edin-initiator                   [6] EDINInitiatorField,
  notifications-extensions         [7] NotificationExtensionsField OPTIONAL }
```

NOTE – The Common Fields appear in the Positive Notification, Negative Notification and Forwarded Notification fields defined below.

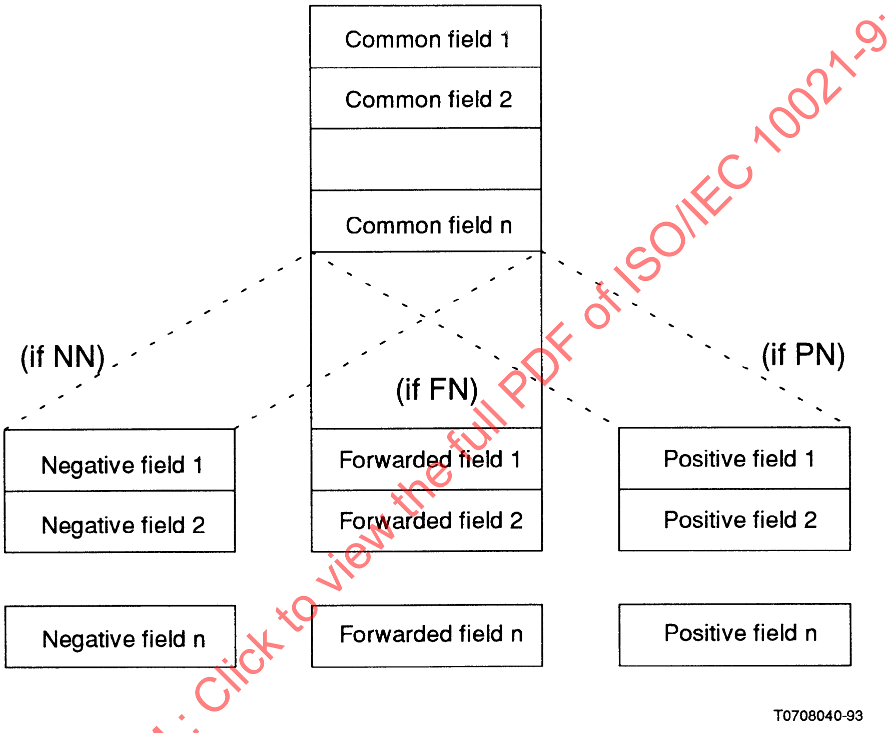

Figure 3 - EDI notification structure

9.1.1 Subject EDIM

The Subject EDIM Identifier is the EDIM Identifier either passed in the EDIN Receiver field, if Responsibility has been forwarded, or the This EDIM field, if not.

```
SubjectEDIMField   ::= EDIMIdentifier
```

NOTE – EDIM Identifier is defined in 7.1. Subject EDIM is defined in clause 9.

9.1.2 EDI notification originator

The EDI Notification Originator contains the OR-name of the UA constructing the notification.

```
EDINOriginatorField  ::= ORName
```

NOTE – OR-name is defined in 8.5.5 of ISO/IEC 10021-4 | CCITT Recommendation X.411, and ORName is defined in figure 2 of ISO/IEC 10021-4 | CCITT Recommendation X.411.

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 ISO/IEC 10021-4 | CCITT Recommendation X.411, and ORName is defined in figure 2 of ISO/IEC 10021-4 | CCITT Recommendation X.411.

If the originator of the EDIN is not the recipient specified by the original originator, then the First Recipient Field shall be present in the EDIN (see 17.3 and more specifically 17.3.1.1).

9.1.4 Notification time

Notification Time contains the date and time, in UTC format, at which the notification for the subject EDIM was generated.

```
NotificationTimeField ::= UTCTime
```

9.1.5 Security elements

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

```
SecurityElementsField ::= SEQUENCE {
  original-content                    [0] Content OPTIONAL,
  original-content-integrity-check    [1] ContentIntegrityCheck OPTIONAL,
  edi-application-security-elements [2] EDIApplicationSecurityElementsField OPTIONAL,
  security-extensions                 [3] SecurityExtensionsField OPTIONAL }

SecurityExtensionsField ::= SET OF SecurityExtensionsSubField

SecurityExtensionsSubField ::= ExtensionField
```

NOTE – The EDI Application Security Elements Field is defined in 8.2.11. Content and Content Integrity Check are defined in, respectively, subclauses 8.2.1.1.1.37 and 8.2.1.1.1.28 (and figure 2) of ISO/IEC 10021-4 | CCITT Recommendation X.411. Security services are available only if the MTS supports secure messaging.

Subclause 17.1.3 specifies how these fields are filled in.

9.1.6 EDIN initiator

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

a) "internal-UA" means that the UA generated the EDIN 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 EDIN 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 EDIN was requested by the user via the abstract operation Originate EDIN (see 17.1.3).

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

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

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

NOTE – Physical delivery access unit (PDAU) is defined in 15.4.

9.1.7 Notification extensions

The Notification Extensions allows for future extensions to the EDIN.

```
NotificationExtensionsField ::= SET OF NotificationExtensionsSubField

NotificationExtensionsSubField ::= ExtensionField
```

There are no extensions to the EDIN defined in this part of ISO/IEC 10021.

Extensions shall not be critical in EDINs.

9.2 Positive notifications

A Positive Notification (PN) is sent by the recipient UA, if and only if the originator has requested one, when Responsibility for the EDIM has been accepted by the UA.

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

Positive Notification Fields are defined and described below.

```
PositiveNotificationFields ::= SEQUENCE {
  pn-common-fields                [0] CommonFields,
  pn-supplementary-information    [1] EDISupplementaryInformation OPTIONAL,
  pn-extensions                   [2] PNExtensionsField OPTIONAL }
```

9.2.1 PN supplementary information

The PN Supplementary Information field may be used to return further information to the EDIN recipient to clarify the Positive Notification.

NOTE – EDI Supplementary Information field is defined in 8.3.2.

9.2.2 Positive notification extensions

The Positive Notification Extensions allows for future extensions to the PN.

```
PNExtensionsField ::= SET OF PNExtensionsSubField

PNExtensionsSubField ::= ExtensionField
```

There are no extensions to the PN defined in this part of ISO/IEC 10021.

Extensions shall not be critical in PNs.

9.3 Negative notifications

A Negative Notification (NN) is sent by a UA, if and only if the originator has requested one, when it determines that it can neither accept Responsibility, nor forward the EDIM and the EDI Notification Request contained in the EDIM to another UA.

Negative Notification Fields are defined and described below.

```
NegativeNotificationFields ::= SEQUENCE {
  nn-common-fields                [0] CommonFields,
  nn-reason-code                  [1] NNReasonCodeField,
  nn-supplementary-information    [2] EDISupplementaryInformation OPTIONAL,
  nn-extensions                   [3] NNExtensionsField OPTIONAL }
```

9.3.1 Negative notification reason

The Negative Notification Reason indicates why the subject EDIM could not be passed to the user by the UA originating the EDIN. Additional information may be carried in any combination of a diagnostic field or the NN Supplementary Information field. Depending on the security policy in force, the security error diagnostic code may or may not be present.

NOTE - The value "unspecified(0)" is provided for use in any basic code field when other code values do not apply.

```
    NNReasonCodeField ::= CHOICE {
      nn-ua-ms-reason-code  [0] NNUAMSReasonCodeField,
      nn-user-reason-code   [1] NNUserReasonCodeField,
      nn-pdau-reason-code   [2] NNPDAUReasonCodeField }

-- Negative Notification Reason Codes from an EDI-UA or EDI-MS

    NNUAMSReasonCodeField ::= SEQUENCE {
      nn-ua-ms-basic-code   [0] NNUAMSBasicCodeField,
      nn-ua-ms-diagnostic   [1] NNUAMSDiagnosticField OPTIONAL }

-- Negative Notification Basic Reason Codes from an EDI-UA or EDI-MS. These codes are
-- those specified in annex B of ISO/IEC 10021-8 | CCITT Recommendation F.435
-- for the element of service "EDI Notification Request".

    NNUAMSBasicCodeField ::= INTEGER {
      unspecified (0),
      cannot-deliver-to-user (1),
      -- the EDI Interchange can not be passed on to the user
      delivery-timeout (2),
      -- the EDI Interchange could not be passed on to the user within
      -- a specified time limit
      message-discarded (3),
      -- the UA/MS discarded the message before handoff to user
      subscription-terminated (4),
      -- recipient's subscription terminated after delivery but before
      -- handoff to user
      forwarding-error (5),
      -- EDI Forwarding was attempted, but failed
      security-error (6)
      -- security error
      -- physical delivery errors indicated by "cannot-deliver-to-user"
       } (0..ub-reason-code)

-- Negative Notification Diagnostic Codes from an EDI-UA or EDI-MS

    NNUAMSDiagnosticField ::= INTEGER {
    -- This field may be used to further specify the error signalled in nn-ua-ms basic-
    -- code. Additional information may be indicated in nn-supplementary-information
      -- general diagnostic codes
      protocol-violation (1),      -- used if the UA detects a protocol error
      edim-originator-unknown (2),
      edim-recipient-unknown (3),
      edim-recipient-ambiguous (4), -- used if the EDIM recipients or
      -- originator are not valid
      action-request-not-supported (5),
      -- used when the action requested by the recipient is not performed
      edim-expired (6),
      -- used when the expiry date of the received EDIM occurred before the subject EDIM
      -- was successfully passed to the user or forwarded by the EDI-UA
      edim-obsoleted (7),
      -- used when the EDIM Identifier of the received EDIM was contained in the
      -- Obsoleted EDIM field of a previously received EDIM
      duplicate-edim (8),
      -- used when the same EDIM is received more than once from the same originator.
      unsupported-extension (9),
      -- used if the EDIM contains an extension which is not supported by the UA.
      incomplete-copy-rejected (10), -- used if the EDI-UA does not accept
      --  EDIMs with the Incomplete Copy Indication true.
      edim-too-large-for-application (11),
      -- used if the EDIM cannot be delivered to the user due to length constraints.

      -- forwarding error diagnostic codes.
      forwarded-edim-not-delivered (12),
      -- used when an Non-Delivery Report is received for forwarded EDIM.
      forwarded-edim-delivery-time-out (13),
      -- used when no Delivery Report is received within a given period.
```

```
     forwarding-loop-detected (14),
     -- used if the UA receives an EDIM which contains a previously forwarded EDIM.
     unable-to-accept-responsibility (15),
     -- used if the EDI-UA cannot accept or forward responsibility.
     -- INTERCHANGE HEADER DIAGNOSTIC CODES
     interchange-sender-unknown (16), -- used when the UA does not
     -- recognize the interchange-sender of the EDI interchange.
     interchange-recipient-unknown (17), -- used when the UA cannot find
     -- a valid interchange recipient in the Recipient Specifier.
     invalid-heading-field (18),
     invalid-bodypart-type (19),
     invalid-message-type (20),
     invalid-syntax-id (21),

   -- SECURITY ERROR DIAGNOSTIC CODES
     message-integrity-failure (22),
     forwarded-message-integrity-failure (23),
     unsupported-algorithm (24),
     decryption-failed (25),
     token-error (26),
     unable-to-sign-notification (27),
     unable-to-sign-message-receipt (28),
     authentication-failure (29),
     security-context-failure (30),
     message-sequence-failure (31),
     message-security-labelling-failure (32),
     repudiation-failure (33),
     proof-service-failure (34)
     } (1..ub-reason-code)

-- Negative Notification Reason Codes from a user

   NNUserReasonCodeField ::= SEQUENCE {
     nn-user-basic-code    [0]   NNUserBasicCodeField,
     nn-user-diagnostic    [1]   NNUserDiagnosticField OPTIONAL }

-- Negative Notification Basic Reason Codes from a user

   NNUserBasicCodeField ::= INTEGER {
     unspecified (0),
     syntax-error (1),      -- used when the user discovers a syntax error
                            -- within the EDI interchange
     interchange-sender-unknown (2),
     interchange-recipient-unknown (3),-- used when the UA cannot find a valid
                            -- interchange recipient in the Recipient Specifier
     invalid-heading-field (4),
     invalid-bodypart-type (5),
     invalid-message-type (6),
     functional-group-not-supported (7),
     subscription-terminated (8),     -- unknown to EDIMS-User service
     no-bilateral-agreement (9),
     user-defined-reason (10)
   } (0..ub-reason-code)

-- Negative Notification Diagnostic Codes from a user

   NNUserDiagnosticField ::= INTEGER (1..ub-reason-code)
   -- Contains reason passed by user when the value of nn-user-basic-code is
   -- user-defined-reason. Additional information may be indicated in
   -- nn-supplementary-information

-- Negative Notification Reason Codes from a PDAU

   NNPDAUReasonCodeField ::= SEQUENCE {
     nn-pdau-basic-code    [0]   NNPDAUBasicCodeField,
     nn-pdau-diagnostic    [1]   NNPDAUDiagnosticField OPTIONAL }
```

```
-- Negative Notification Basic Reason Codes from a PDAU

   NNPDAUBasicCodeField ::= INTEGER {
     unspecified (0),
     undeliverable-mail (1),         -- used if the PDAU determines that it cannot
                                     -- perform physical delivery of the EDIM
     physical-rendition-not-performed (2) -- used if the PDAU cannot perform
                                     -- the physical rendition of the EDIM
    } (0..ub-reason-code)

-- Negative Notification Diagnostic Codes from a PDAU

   NNPDAUDiagnosticField ::= INTEGER {
   -- This field may be used to further specify the error signalled in
   -- nn-pdau-basic-code. Additional information may be indicated in the
   -- nn-supplementary-information.
     undeliverable-mail-physical-delivery-address-incorrect (32),
     undeliverable-mail-physical-delivery-office-incorrect-or-invalid (33),
     undeliverable-mail-physical-delivery-address-incomplete (34),
     undeliverable-mail-recipient-unknown (35),
     undeliverable-mail-recipient-deceased (36),
     undeliverable-mail-organization-expired (37),
     undeliverable-mail-recipient-refused-to-accept (38),
     undeliverable-mail-recipient-did-not-claim (39),
     undeliverable-mail-recipient-changed-address-permanently (40),
     undeliverable-mail-recipient-changed-address-temporarily (41),
     undeliverable-mail-recipient-changed-temporary-address (42),
     undeliverable-mail-new-address-unknown (43),
     undeliverable-mail-recipient-did-not-want-forwarding (44),
     undeliverable-mail-originator-prohibited-forwarding (45),
     physical-rendition-attributes-not-supported (31)
     } (1..ub-reason-code)
```

9.3.2 NN supplementary information

The NN Supplementary Information field may be used to return further information to the EDIN recipient to clarify the Negative Notification.

NOTE – EDI Supplementary Information is defined in 8.3.2.

9.3.3 Negative notification extensions

The Negative Notification Extensions allows for future extensions to the NN.

```
   NNExtensionsField ::= SET OF NNExtensionsSubField

   NNExtensionsSubField ::= ExtensionField
```

There are no extensions to the NN defined in this part of ISO/IEC 10021.

Extensions shall not be critical in NNs.

9.4 Forwarded notifications

A Forwarded Notification (FN) is sent by a UA, if and only if the originator has requested one, when it determines that it cannot accept Responsibility and decides to forward the EDIM, and the EDI Notification Requests contained in the EDIM, to another UA.

Forwarded Notification Fields are defined and described below.

```
   ForwardedNotificationFields ::= SEQUENCE {
     fn-common-fields               [0] CommonFields,
     forwarded-to                   [1] ForwardedTo,
     fn-reason-code                 [2] FNReasonCodeField,
     fn-supplementary-information   [3] EDISupplementaryInformation OPTIONAL,
     fn-extensions                  [4] FNExtensionsField OPTIONAL }
```

9.4.1 Forwarded to

The Forwarded To field indicates the new recipient of the (forwarded) subject EDIM. Its value is an OR-name.

```
ForwardedTo ::= ORName
```

NOTE – OR-name is defined in 8.5.5 of ISO/IEC 10021-4 | CCITT Recommendation X.411, and ORName is defined in figure 2 of ISO/IEC 10021-4 | CCITT Recommendation X.411.

9.4.2 Forwarded notification reason

The Forwarded Reason Code indicates the reason why the Responsibility of the subject EDIM was forwarded. It also indicates the appropriate source of forwarding, e.g., the UA or MS, the user, or the PDAU. Additional information may be carried in any combination of a diagnostic field and the FN Supplementary Info field.

```
    FNReasonCodeField ::= CHOICE {
      fn-ua-ms-reason-code  [0] FNUAMSReasonCodeField,
      fn-user-reason-code   [1] FNUserReasonCodeField,
      fn-pdau-reason-code   [2] FNPDAUReasonCodeField }

-- Forwarding Notification Reason Codes from an EDI-UA or EDI-MS
    FNUAMSReasonCodeField ::= SEQUENCE {
      fn-ua-ms-basic-code   [0] FNUAMSBasicCodeField,
      fn-ua-ms-diagnostic   [1] FNUAMSDiagnosticField OPTIONAL,
      fn-security-check     [2] FNUAMSSecurityCheckField DEFAULT FALSE }

-- Forwarding Notification Basic Reason Codes from an EDI-UA or EDI-MS
    FNUAMSBasicCodeField ::= INTEGER {
      unspecified (0),
      onward-routing (1), -- used whenever the UA decides to re-route the
      -- subject EDIM for local reasons
      EDIM-recipient-unknown (2),
      EDIM-originator-unknown (3),
      forwarded-by-edi-ms (4)
       } (0..ub-reason-code)

-- Forwarding Notification Diagnostic Reason Codes from an EDI-UA or EDI-MS
    FNUAMSDiagnosticField ::= INTEGER {
    -- This field may be used to further specify the error signalled in fn-ua-ms-basic-
    -- code. Additional information may be indicated in fn-supplementary-information
      recipient-name-changed (1),
      recipient-name-deleted (2)
       } (1..ub-reason-code)

-- Forwarding Notification Security Check Codes from an EDI-UA or EDI-MS.
    --     This field may be used, with a value of TRUE, to indicate that all security
    -- features present have been validated, or a value of FALSE to indicate that
    -- the security features have not been validated.

    FNUAMSSecurityCheckField ::= BOOLEAN

-- Forwarding Notification Reason Codes from a user

    FNUserReasonCodeField ::= SEQUENCE {
      fn-user-basic-code    [0] FNUserBasicCodeField,
      fn-user-diagnostic    [1] FNUserDiagnosticField OPTIONAL }

-- Forwarding Notification Basic Reason Codes from a user

    FNUserBasicCodeField ::= INTEGER {
      unspecified (0),
      forwarded-for-archiving (1),
      forwarded-for-information (2),
      forwarded-for-additional-action (3),
      subscription-changed (4),
      heading-field-not-supported (5),
      bodypart-type-not-supported (6),
      message-type-not-supported (7),
      syntax-identifier-not-supported (8),
      interchange-sender-unknown (9),
    user-defined-reason (10)
      } (0..ub-reason-code)

-- Forwarding Notification Diagnostic Reason Codes from a user

   FNUserDiagnosticField ::= INTEGER (1..ub-reason-code)
   -- Contains reason passed by user when value of fn-user-basic-code is
   -- user-defined-reason. Additional information may be indicated in
   -- fn-supplementary-information.

-- Forwarding Notification Reason Codes from a PDAU

   FNPDAUReasonCodeField ::= SEQUENCE {
     fn-pdau-basic-code    [0] FNPDAUBasicCodeField,
     fn-pdau-diagnostic    [1] FNPDAUDiagnosticField OPTIONAL }

-- Forwarding Notification Basic Reason Codes from a PDAU

   FNPDAUBasicCodeField ::= INTEGER {
     unspecified (0),
     forwarded-for-physical-rendition-and-delivery (1)
     } (0..ub-reason-code)

-- Forwarding Notification Diagnostic Codes from a PDAU

   FNPDAUDiagnosticField ::= INTEGER (1..ub-reason-code)
```

A physical delivery access unit (PDAU) (see 15.4) is only able to generate NNs and FNs. PNs shall not be generated even if requested. If any notification (PN, FN, NN or any combinations of them) is requested and passing of Responsibility is allowed by the originator, the PDAU shall generate an FN with appropriate Forwarded Reason Code ("forwarded-for-physical-rendition-and-delivery") when it has determined that it can render the EDIM for physical delivery. If any notification (PN, FN, NN or any combinations of them) is requested and passing of responsibility is not allowed by the originator, the PDAU shall not render the EDIM for physical delivery and shall generate an NN if so requested. If no notifications are requested, the PDAU shall attempt to render the EDIM for physical delivery, regardless of whether passing Responsibility is allowed or not.

9.4.3 FN supplementary information

The FN Supplementary Information field may be used to return further information to the EDIN recipient to clarify the Forwarded Notification.

NOTE – The EDI Supplementary Information Field is defined in 8.3.2.

9.4.4 Forwarded notification extensions

The Forwarded Notifications Extensions allows for future extensions to the FN.

```
   FNExtensionsField ::= SET OF FNExtensionsSubField
```

FNExtensionsSubField ::= ExtensionField

There are no extensions to the FN defined in this part of ISO/IEC 10021.

Extensions shall not be critical in FNs.

Section 2 - EDI environment and abstract service definition

10 Primary object types

The environment in which EDI Messaging takes place can be modeled as an abstract object which is hereafter referred to as the EDI Messaging Environment (EDIME).

```
   edime OBJECT ::= id-ot-edime
```

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

```
edime-refinement REFINE edime AS
  edims
    origination          [S] PAIRED WITH edimg-user
    reception            [S] PAIRED WITH edimg-user
  edimg-user RECURRING
  ::= id-ref-primary
```

The lesser objects are referred to as the primary objects of EDI Messaging. They include a single, central object, the EDI Messaging System, EDIMS, and numerous peripheral objects called EDI Messaging System users (users).

The structure of the EDIME is depicted in figure 4.

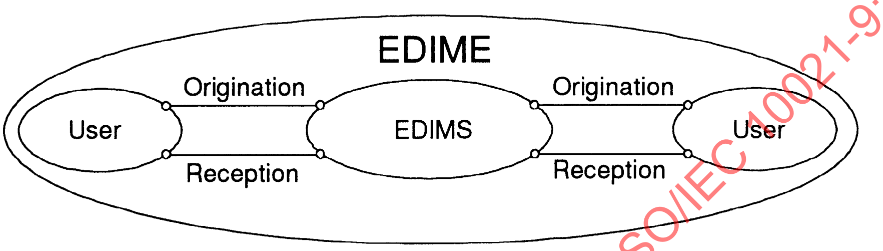

Figure 4 - The EDI messaging environment

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 EDI messaging user

An EDI Messaging user (EDIMG user) is typically a computer process or application that engages in EDI Messaging. Such processes or applications are referred to by the term "user" in this part of ISO/IEC 10021. A user originates, receives, or both originates and receives Information Objects of the types defined in clause 6.

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

The EDIME comprises any number of Users.

NOTE – EDI messaging is typically an activity between information processing systems. These are referred to as EDI applications. This does not preclude the possibility of human interaction with the information processing systems which are performing EDI, or more direct interaction of a human user with the EDIMS. The terms "user" and "EDIMG user" may be regarded as synonyms for EDI applications within this part of ISO/IEC 10021. For brevity, the term "user" is used throughout the rest of this part of ISO/IEC 10021 with the meaning of "EDIMG user".

10.2 EDI messaging system

The EDI Messaging System (EDIMS) is the object by means of which all users communicate with one another in EDI Messaging.

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

The EDIME comprises exactly one EDIMS.

11 Primary port types

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

Standard of a management port may be the subject of future standardization.

NOTE – In clause 15 to follow, the EDIMS 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 EDIMS Abstract Service.

11.1 Origination port

An Origination Port is the means by which a single user conveys to the EDIMS messages containing Information Objects of the types defined in clause 6. Through such a port the user originates EDI Messages and EDI Notifications. In addition, the user may originate probes through such a port.

```
origination PORT
  CONSUMER INVOKES {
    OriginateProbe,
    OriginateEDIM,
    OriginateEDIN }
  ::= id-pt-origination
```

The EDIMS supplies one Origination Port to each user [with the exception of indirect users served by PDAUs (see 15.4)].

11.2 Reception port

A Reception Port is the means by which the EDIMS conveys to a single user messages containing Information Objects of the types defined in clause 6. Through such a port the user receives EDI Messages and EDI Notifications. In addition, the user may receive reports through such a port.

```
reception PORT
  SUPPLIER INVOKES {
    ReceiveReport,
    ReceiveEDIM,
    ReceiveEDIN }
  ::= id-pt-reception
```

The EDIMS supplies one Reception Port to each user.

12 Abstract operations

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

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

The purpose of the EDIMS Abstract Service Definition is not to prescribe the interface between the EDI user and the EDI-UA, but rather to clarify the meaning and intended use of the Information Objects of clause 6. 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 EDIMS 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 EDIMS Abstract Service involves neither abstract bind nor abstract unbind operations.

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

NOTE – In clause 15 to follow, the EDIMS 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 EDIMS Abstract Service.

12.1 Origination abstract operations

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

12.1.1 Originate probe

The Originate Probe abstract operation originates a probe concerning (a class of) messages whose contents are EDIMs.

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

This abstract operation has the following arguments:

a) *Envelope:* A probe submission envelope, whose make-up the MTS Abstract Service defines. The UA supplies all but the following envelope components, which the user provides:

   1) The desired per-message options (i.e., per-message indicators 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:* An instance of the class of EDIM whose deliverability is to be probed.

This abstract operation has the following results:

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

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

12.1.2 Originate EDIM

The Originate EDIM abstract operation originates a message whose content is an EDIM.

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

This abstract operation has the following arguments:

a) *Envelope:* A Message Submission Envelope, whose make-up the MTS Abstract Service defines. 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. The OR-names supplied in the envelope shall have the same values as the OR-names in the RecipientsField, when present, of the EDIM heading which identify the corresponding recipients.

b) *Content:* The EDIM being originated.

   1) If application to application security services are required, the user shall supply the value for the EDI Application Security Elements field.

The EDIM 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.

12.1.3 Originate EDIN

The Originate EDIN abstract operation originates a message whose content is an EDIN.

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

A user may, if notifications are requested, invoke an Originate EDIN abstract operation to indicate to the UA that it should accept, refuse or forward Responsibility for the subject EDIM. The exact type of EDIN to be generated (PN, NN or FN) is determined from the Content argument.

An EDIN shall be originated only by an actual recipient of the subject EDIM of whom an EDIN is requested by means of the EDI Notification Request field of the subject EDIM's Recipient field.

A user may delegate the task of generating EDINs 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, whose make-up the MTS Abstract Service defines. 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 and deferred delivery shall be prohibited, priority shall be that of the subject EDIM.

   2) The OR-names of the preferred recipient and the per-recipient options (i.e., explicit conversion and extensions) desired. The preferred recipient of the EDIN is the originator of the subject EDIM or, if present, the OR-name indicated in the EDIN Receiver field.

b) *Content:* The EDIN being originated.

   1) If application to application security services are required, the user shall supply the value for the EDI Application Security Elements field.

The EDIN 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 EDIMS and performed by the user.

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

12.2.1 Receive report

The Receive Report abstract operation receives a report.

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

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

a) A message whose content was an EDIM that was originated with the Originate EDIM abstract operation or by forwarding.

b) A message whose content was an EDIN that was originated as a result of a previously received message. The EDIN could be any of PN, NN or FN.

c) A probe concerning a message whose content was an EDIM that was originated with the Originate Probe abstract operation.

This abstract operation has the following arguments:

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

e) *Undelivered-object:* The content of the message whose status is being reported. An EDIM or EDIN.

If the report was provoked by a previous Originate Probe abstract operation invocation, this conditional argument shall be absent. If the report was provoked by a previous Originate EDIM abstract operation invocation, the argument shall be present if, and only if, content return was requested. Otherwise (for example, if the report was provoked by an EDIN), the argument shall be absent.

This abstract operation has no results.

12.2.2 Receive EDIM

The Receive EDIM abstract operation receives a message whose content is an EDIM.

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

This abstract operation has the following arguments:

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

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

This abstract operation has no results.

When the received EDIM contains an EDIM Body Part (that is, when the original EDIM 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 8.3.2 for the nested structure of a forwarded EDIM and 17.3.3 for rules related to Heading fields).

12.2.3 Receive EDIN

The Receive EDIN abstract operation receives a message whose content is an EDIN. The EDIN is provoked by an EDIM originated with the Originate EDIM abstract operation.

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

This abstract operation has the following arguments:

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

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

This abstract operation has no results.

13 Abstract errors

The abstract errors that may be reported in response to the invocation of the abstract operations available at the Origination Port and Reception Port 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 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 EDIMS Abstract Service, defined above, the EDIMS shall transparently extend to each user the other MS (see ISO/IEC 10021-5 | CCITT Recommendation X.413) and MTS (see ISO/IEC 10021-4 | CCITT Recommendation X.411) capabilities identified below. (The enumeration of these capabilities necessarily anticipates the fact, stated in clause 15, that MSs and the MTS are among the EDIMS' component parts.)

The following additional capabilities shall be provided:

a) *Submission:* Capabilities of the MS' or MTS' Submission Port not embodied in the EDIMS Abstract Service, e.g., the ability to cancel delivery of a previously originated message whose content is an EDIM (but not an EDIN), if deferred delivery was selected.

b) *Delivery:* Capabilities of the MTS' Delivery Port not embodied in the EDIMS 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' or MTS' Administration Port.

d) *Retrieval:* The capabilities of the MS' Retrieval Port.

In addition to the above and as a local matter, the EDIMS may provide to users additional capabilities neither defined nor limited by this part of ISO/IEC 10021. Among such capabilities are those of the Directory.

NOTE – The required capabilities above are excluded from the formal definition of the EDIMS 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 EDIMS can be modeled as comprising lesser objects which interact with one another by means of (additional) ports.

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

These lesser objects are referred to as the secondary objects of EDI Messaging. They include a single, central object, the MTS, and numerous peripheral objects: EDI messaging system user agents (EDI-UA), EDI messaging system message stores (EDI-MS), telematic agents (TLMA), and physical delivery access units (PDAU). Specification of the protocol for the TLMA may be the subject for future standardization.

The structure of the EDIMS is depicted in figure 5. As shown by the figure, EDI-UAs and PDAUs are the instruments by means of which the EDIMS provides the EDIMS Abstract Service to users.

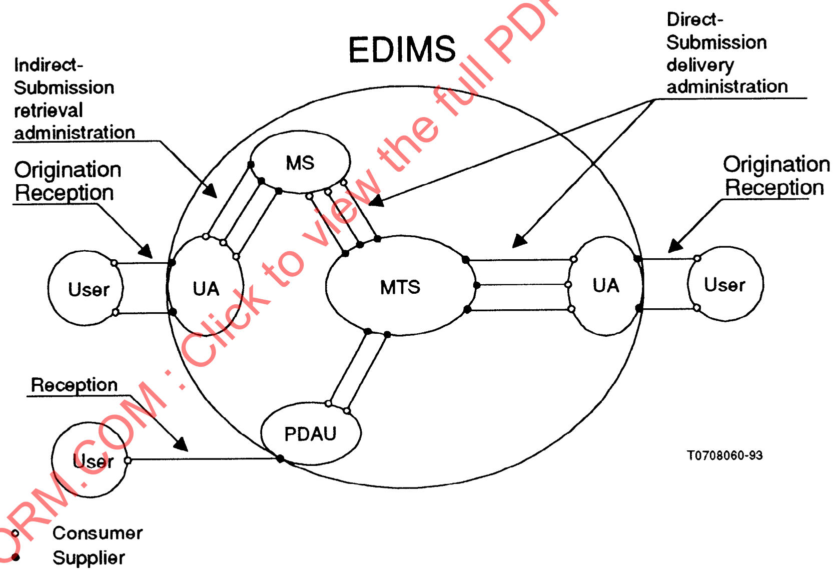

Figure 5 - The EDI messaging system

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., PDAU), and specific logical configurations of the MS. The latter are identified in ISO/IEC 10021-2 I CCITT Recommendation X.402.

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

15.1 EDI user agent

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

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

The EDIMS comprises any number of EDIMS UAs.

NOTE – As noted above, the term *user agent* (UA) is used throughout this part of ISO/IEC 10021 with the meaning of EDI-UA.

15.2 EDI message store

An EDI message store (EDI-MS) is an MS tailored so as to better assist a single UA engaged in EDI 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.

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

The EDIMS comprises any number of EDIMS MSs.

NOTE – As noted above, the term message store (MS) is used throughout this part of ISO/IEC 10021 with the meaning of EDI-MS.

15.3 Telematic agent

A telematic agent (TLMA) is an AU that helps a single indirect user engage in EDI Messaging from a Telematic terminal, along with that terminal and the network that joins the two. A TLMA helps the user originate, receive, or both originate and receive messages containing Information Objects of the types defined in clause 6.

Specification of the protocol for this AU may be the subject for future standardization.

15.4 Physical delivery access unit

In the present context, a Physical Delivery Access Unit (PDAU) helps any number of indirect users engage in EDI Messaging by means of a Physical Delivery System (PDS). It helps them receive (but not originate) messages containing Information Objects of the types defined in clause 6.

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

The EDIMS comprises any number of PDAUs.

A PDAU consumes import and export ports. However, since those ports are not formally defined (in ISO/IEC 10021-4 | CCITT Recommendation X.411), they are not included in the formal definition of PDAU above.

If notifications are requested, the PDAU shall generate one of the following:

- an FN with appropriate reason code if the PDAU determines that it can render and deliver the EDIM,
- a NN with appropriate reason code if the PDAU determines that it cannot render or deliver the EDIM.

The use of the PDAU shall be subject to the requirements of the security policy in force.

15.5 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 EDIMS comprises a single MTS.

The use of TLMA may be restriced by the security policy in force.

16 Secondary port types

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

The capabilities embodied in one Submission, one Retrieval, and one Administration Port constitute the MS Abstract Service. They are defined in ISO/IEC 10021-4 | CCITT Recommendation X.413.

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

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

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 probes concerning, and 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

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

An MS supplies one Administration Port to its UA.

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

16.5 Import port

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

The MTS supplies one Export Port to each AU.

Section 3 - Procedures

17 User agent operation

A UA must employ the MTS in a particular way in order to (correctly) provide the EDIMS 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. The operation of a TLMA is beyond the scope of this part of ISO/IEC 10021.

NOTE – The purpose of what follows is not to dictate or constrain the implementation of a real UA unnecessarily, but rather to specify the effect to be achieved.

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 operations of the MTS Abstract Service (which, for what follows, are unqualified as to their source):

a) Probe Submission

b) Message Submission

In response to the invocation of these abstract operations, a UA reports abstract errors as appropriate. Specification of the precise circumstances under which each abstract error should be reported is beyond the scope of this part of ISO/IEC 10021.

17.1.1 Originate probe

A UA shall perform the Originate Probe abstract operation by invoking Probe Submission with the arguments indicated below, and by returning to its user the results indicated below.

The arguments of Probe Submission shall be as follows:

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

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

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

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

   The components of this argument that constitute per-recipient fields shall be as specified by Originate Probe's Envelope argument.

The results of Originate Probe shall be as follows:

b) *Submission-identifier:* Probe Submission's "probe-submission-identifier" result.

c) *Submission-time:* Probe Submission's "probe-submission-time" result.

The UA shall ignore all properties of Originate Probe's Content argument other than those mentioned above.

How the UA employs Probe Submission's Content-identifier and Content-correlator are local matters.

17.1.2 Originate EDIM

A UA shall perform the Originate EDIM abstract operation by invoking Message Submission with the arguments indicated below, and by returning to its user the results indicated below.

The arguments of Message Submission shall be as follows:

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

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

2) Content-type and Original-encoded-information-types: Determined from Originate EDIM'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 Security Service, and when Notification Security is requested, the UA shall generate and submit the content-integrity-check Security Argument as defined in 8.2.1.1.1.28 of ISO/IEC 10021-4 | CCITT Recommendation X.411.

5) If "proof/non-repudiation of Content Originated" is requested, the UA shall submit the message using the "message-origin-authentication-check", or the "content-integrity-check" (possibly in the message token), according to the security policy in force.

The components of this argument that constitute per-recipient fields shall be as specified by abstract operation Originate EDIM's Envelope argument.

To prevent an unknown number of EDINs from being sent to the original originator of a message in case of forwarding, "DL-expansion-prohibited", if available, may be set to TRUE if any of PN, NN or FN are requested.

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

1) If "proof/non-repudiation of EDI Notification Request" notification is requested, the UA shall set the EDI Notification Security field accordingly for each recipient as required.

2) If "proof/non-repudiation of Content Received Request" notification is requested, the UA shall set the EDI Reception Security field accordingly for each recipient as required.

3) If "Application Security Element" is requested, the end-to-end application security value shall be conveyed in the EDI Application Security Elements field.

NOTE – In case of the use of a notarizing function, the *non-repudiation of content* service is provided implicitly, and is not reflected in any protocol elements.

The results of Originate EDIM 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 Message Submission's Content-identifier and Content-correlator are local matters.

The inclusion of Message Submission's Extensions result among Originate EDIM's results is proper and may be the subject for future standardization.

17.1.3 Originate EDIN

A UA shall perform the Originate EDIN 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.

A user may delegate the task of generating EDINs 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 accept Responsibility at will, but shall accept Responsibility when the EDIM is made available to the user, or when it forwards an EDIM with content changed (in this context, "content changed" means that the forwarding UA adds or removes body parts from the forwarding EDIM, in accordance with 8.3.2. The term forwarding EDIM is defined in 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 not explicitly mentioned below shall be as specified by Originate EDIN's Envelope argument:

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

2) Content-type and Original-encoded-information-types: Determined from the EDIN 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 EDIM.

NOTE – Subject EDIM is defined in 17.3.3.

6) Per-message-indicators: *notification-type* shall be set to "type 1" for PN, to "type 2" for NN and to "type 3" for FN.

b) *Content:* Determined from Originate EDIN's Content argument (identified as a PN, NN or FN) as specified in 19.1.

1) If, in the subject EDIM, EDI Reception Security is set to "non-repudiation" and EDI Notification Security is set to "non-repudiation" and the "content-integrity-check" security argument is present in the delivery envelope of the subject EDIM, then the "content-integrity-check" security argument is copied into the Content Integrity Check field of the EDIN. The UA shall submit the EDIN 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 EDIM, EDI Reception Security is set to "proof" and EDI Notification Security is set to "proof" and the "content-integrity-check" security argument is present in the delivery envelope of the subject EDIM, then the "content-integrity-check" security argument is copied into the Content Integrity Check field of the EDIN. The UA shall submit the EDIN with the security element "content-integrity-check" (possibly in the message token) or a "message-origin-autentication-check" (depending on the security policy in force).

3) If, in the subject EDIM, EDI Reception Security is set to "non-repudiation" and EDI Notification Security is set to "non-repudiation" and the "content-integrity-check" security argument is not present in the delivery envelope of the subject EDIM, then the Content of the subject message shall be copied into the Original Content field of the EDIN. The UA shall submit the EDIN 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).

4) If, in the subject EDIM, EDI Reception Security is set to "proof" and EDI Notification Security is set to "proof" and the "content-integrity-check" security argument is not present in the delivery envelope of the subject EDIM, then the Content of the subject message shall be copied into the Original Content field of the EDIN. The UA shall submit the EDIN 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).

5) If, in the subject EDIM, EDI Notification Security is set to "proof" the UA shall submit the EDIN 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.

6) If, in the Subject EDIM, EDI Notification Security is set to "non-repudiation" the UA shall submit the EDIN 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.

7) If the MTS does not support secure messaging and if the EDI Reception/Notification security services are requested, the EDIN shall contain an appropriate Reason Code.

The content-integrity-check shall always be checked for validity by the recipient UA before generating the EDIN.

The results of Originate EDIN 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 Message Submission's Content-identifier and Content-correlator are local matters.

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' 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 no 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.

How the UA employs the Content-identifier and Content-correlator components of Report Delivery's Envelope argument are local matters.

17.2.2 Receive EDIM

When the MTS invokes Message Delivery at a UA's Delivery Port, and its Content argument encodes an EDIM as specified in 19.1, the UA may invoke the Receive EDIM 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 an EDIM).

NOTE – Under some circumstances, for example, when the delivered message is forwarded, the UA might not invoke the Receive EDIM abstract operation.

17.2.3 Receive EDIN

Whenever the MTS invokes Message Delivery at a UA's Delivery Port, and its Content argument encodes an EDIN as specified in 19.1, the UA shall invoke the Receive EDIN 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 Responsibility, refusal of Responsibility and forwarding.

A user may instruct its UA to accept or refuse Responsibility 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 EDIM more than once. Mechanisms for detecting such duplicate receptions are not required, but may be a matter of local implementation by the UA. If they exist, and notifications are requested, the UA shall originate an NN. If they do not exist, and notifications are requested, the UA shall originate a PN or FN, 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 EDIM at the UA's Reception Port.

e) The message contains an EDIM (rather than an EDIN).

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.

17.3.1 Acceptance of responsibility

A UA shall accept Responsibility when a message is successfully passed from the UA to the user. The UA shall follow the procedures below for each candidate message with respect to whose content the following condition holds:

a) The EDIM requests a PN by means of the EDI Notification Request field of the appropriate Recipients Sub Field in the EDIM's Recipients field.

The UA may forward a message for which it has accepted Responsibility. See also 17.3.3 on forwarding.

17.3.1.1 Construction of PN

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

The PN shall also have the following common fields:

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

b) *EDIN Originator:* The OR-name of the UA which submits the EDIN. If the UA is a preferred recipient of the subject EDIM, the OR-name shall be precisely that which is the value of the Recipient field in the subject EDIM.

c) *First Recipient:* The OR-name of the UA which the originator of the EDIM specified as recipient in the Heading, or, if present, the First Recipient field in the EDIN Receiver field. If the EDIN Receiver field is not present, the First Recipient OR-name is the OR-name of the UA creating the EDIN, 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 delivery envelope (see 8.3.1.1.1.4 of ISO/IEC 10021-4 | CCITT Recommendation X.411). 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 ISO/IEC 10021-4 | CCITT Recommendation X.411).

d) *Notification Time:* The current date and time.

e) *EDI Notification Security elements:* Security elements in any type of notification shall follow the rules of 17.1.3.

f) *EDIN Initiator:* Shall be set according to 9.1.6.

17.3.1.2 Submission of PN

The UA shall submit the PN 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 EDIN abstract operation with the following exceptions:

   1) Priority: As specified by Message Delivery Envelope argument.

   2) Per-message-indicators: A local matter, except that *conversion-prohibited* shall be among the values specified and *notification-type* shall be set to "type 1".

   3) Per-recipient-fields: A single field whose Recipient-name component shall be the Originator-name component of Message Delivery Envelope argument, or if the EDIN Receiver field is present, the EDIN Receiver as specified by the originator of the message.

   NOTE – If the OR-name in the EDIN Receiver field is not valid, then the UA cannot submit the EDIN. Procedures to be followed in this case are a local matter.

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

17.3.2 Refusal of responsibility

A UA shall refuse to accept Responsibility when a message cannot be successfully passed from the UA to the user. A UA may refuse to accept Responsibility when forwarding was unsucessful [see **c)** of 17.3.3.1]. The UA shall follow the procedures below for each candidate message under the following conditions:

a) The EDIM requests an NN of the UA's user by means of the EDI Notification Requests field of the appropriate Recipients Sub Field in the EDIM's Recipients field.

b) The EDIM is not successfully forwarded onward, or not successfully passed to the user of this UA.

NOTE – See also 17.3.3 on forwarding.

17.3.2.1 Construction of NN

The UA shall construct an NN if, and only if, one is requested by means of the EDI Notification Requests field of the subject EDIM's Recipients field and in accordance with 17.3.2.

The NN shall have the common fields prescribed for Construction of PN (see 17.3.1.1).

The NN shall also have the following fields:

a) *Negative Notification Reason Code:* The reason why Responsibility for the EDIM was refused.

b) Optionally, NN Supplementary Information that adds information to the reason given.

17.3.2.2 Submission of NN

The UA shall submit the NN (if any) above by invoking Message Submission. Its Envelope argument shall be as prescribed for Acceptance of Responsibility (see 17.3.1) except that the *notification-type* in the "per-message-indicators" shall be set to "type 2", its Content argument determined from the NN as specified in 19.1.

NOTE – If the OR-name in the EDIN Receiver field is not valid, then the UA cannot submit the EDIN. Procedures to be followed in this case are a local matter.

17.3.2.3 Handling of received EDIM

The received EDIM for which the UA refuses Responsibility shall not be made available to the user, nor shall it be forwarded.

17.3.3 EDI Forwarding

The procedures defined in this paragraph describe EDI Forwarding.

NOTE – For brevity, the term "forwarding" is used in this part of ISO/IEC 10021 as a synonym for "EDI 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. A message shall not be forwarded when Responsibility for that message has been refused.

Additionally, if the user has a MS, forwarding may be performed by the MS under the control of the UA by means of the forwarding request extension (see 18.8.2), or the MS may perform EDI auto-forwarding (see 18.6).

In order to forward an EDIM, the UA creates a new EDIM with a new Heading and in the Primary Body Part encapsulates the received EDIM (Heading and Body) and optionally the envelope of the received message using the body part type EDIM Body Part (see 8.3.2).

Figure 6 illustrates forwarding with an example.

The term subject EDIM refers to the received EDIM that is being forwarded. The term forwarding EDIM refers to the new EDIM that is being created, and that will include all or part of the subject EDIM, in accordance with 8.3.2. The term forwarded EDIM refers to the outermost EDIM Body Part of the forwarding EDIM, constituting all or part of the subject EDIM.

Unless overridden by specific rules below, or by the requirements of the security policy in force, the following general rules apply to the creation of the Heading fields of the forwarding EDIM:

- All mandatory Heading fields and any optional fields whose values are changed with respect to the values present in the subject EDIM shall be present.
- Heading fields whose values are unchanged shall be copied from the subject EDIM Heading to the forwarding EDIM Heading if the field is present in the subject EDIM Heading and the value in the field is other that the value specified as DEFAULT in 8.2.
- Other Heading fields need not be copied.

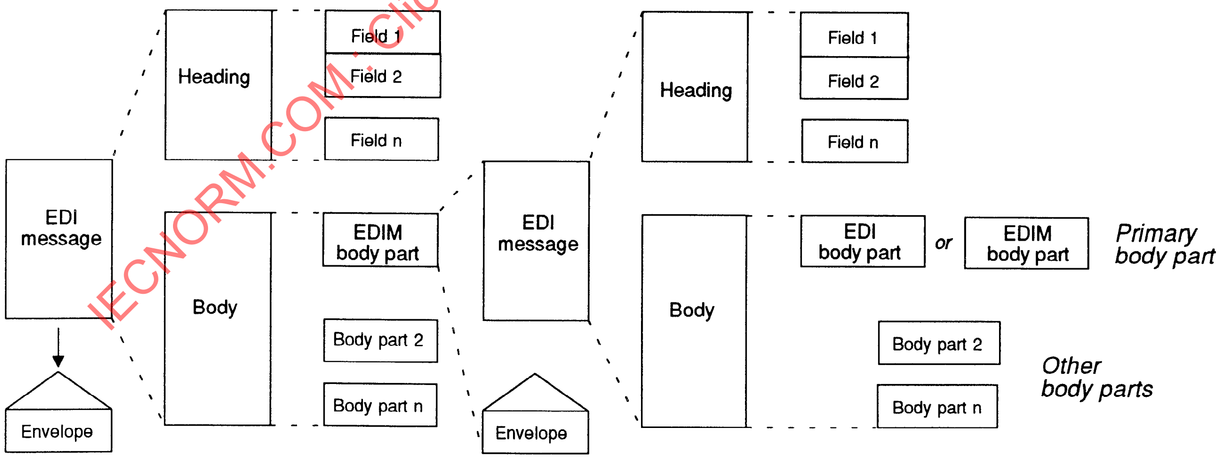

Figure 6 - Forwarding

EDI Forwarding may take two forms:

a) Forwarding of message and Responsibility forwarded.

b) Forwarding of message and Responsibility accepted.

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

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

17.3.3.1 Forwarding of message and responsibility forwarded

Forwarding a message without accepting Responsibility of the message 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 EDIM shall be included if security notifications are requested.

b) If passing of Responsibility is allowed by the originator, the EDI Notification Request field is forwarded unchanged with the new message to one, and only one, of the recipients of the new message. The value of the Responsibility Forwarded field shall be set to TRUE.

c) If the forwarding fails (for example, 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 refuse Responsibility (see 17.3.2).

d) If the EDI Notification Requests field of the subject EDIM's Recipients field requests FN, an FN EDIN shall be sent back to the specified EDIN Receiver, or to the originator of the EDIM, if no EDIN Receiver is specified.

The delivery envelope of the received message shall be included in the new EDIM if the received EDIM's Primary Body Part is not a Forwarded EDIM.

It is possible to forward a message more than once, and a message may be forwarded to multiple recipients, in accordance with the rules above.

The originator of a message may prohibit passing of Responsibility by setting the Responsibility Passing Allowed field to the value FALSE. In this case, if the UA cannot accept Responsibility and NN Notification is requested, the UA shall submit an NN EDIN with appropriate reason code. If the UA cannot accept Responsibility and NN Notification is not requested, then no EDIN shall be submitted.

17.3.3.2 Forwarding of message and responsibility accepted

Forwarding a message and accepting Responsibility of 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. The Heading of the subject EDIM shall remain unchanged.

NOTE 1 – 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.

The following rules apply with respect to removal of body parts:

1) A forwarded EDIM Body Part shall not be removed;

2) "removed-EDI-body" shall be inserted where an EDI Body Part has been removed (see 8.3.2);

3) Body Part Place Holders shall be inserted where other body parts have been removed (see 8.3.2);

4) the Incomplete Copy Indicator field of the forwarding EDIM shall be set to "TRUE" if body parts are removed.

b) Responsibility Forwarded shall not be requested (that is, the field shall not be present).

c) If the EDI Notification Requests field of the subject EDIM's Recipients field requests Positive Notification, a PN EDIN shall be sent back to the specified EDIN Receiver, or to the originator of the EDIM, if no EDIN Receiver is specified.

d) A Forwarded Notification, FN, shall not be sent back to the originator of the message.

NOTE 2 – By scanning the successive nested Headings of an EDIM that contains a forwarded EDIM, the final recipient UA can determine from the setting of the Responsibility Forwarded field at which point in the forwarding chain Responsibility was accepted.

17.3.3.3 Prevention of loops

The UA shall suppress forwarding if the EDIM to be forwarded itself contains a forwarding EDIM that the UA previously created. That is to say, forwarding shall be suppressed whenever the forwarding EDIM appears (directly) in a body part of the EDIM to be forwarded, or (nested) in a body part of the EDIM that appears in such a body part.

The UA shall consider itself to have created the forwarding EDIM above if, and only if, the OR-name component of a This EDIM Field in a forwarded EDIM matches the OR-name of the UA's user.

NOTE – Forwarding an EDIM of the kind described above would constitute an EDI Forwarding "loop".

17.3.3.4 Construction of forwarding EDIM

The UA shall construct a forwarding EDIM whose Primary Body Part comprises a body part of type EDIM Body Part.

The Heading shall have the following components:

a) *This EDIM:* New value generated.

b) *Originator:* OR-name of the forwarding user.

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

If Responsibility is not accepted, the following rules relating to the components of the EDIM Heading apply:

d) *EDIN Receiver Field:* shall be present and shall contain all optional fields. If the subject EDIM contains an EDIN Receiver Field, the fields of the EDIN Receiver Field of the forwarding EDIM shall have the values of the fields of the EDIN Receiver Field of the subject EDIM. If optional fields are missing from the EDIN Receiver Field of the subject EDIM, or if the subject EDIM does not contain an EDIN Receiver Field, then the missing fields of the EDIN Receiver Field of the forwarding EDIM shall have the following values:

   1) Edin-receiver-name: Originator of subject EDIM.

   2) Original-edim-identifier: This EDIM field of subject EDIM.

   3) First-recipient: OR-name of the UA to which the subject EDIM was first sent by the original originator. This is the OR-name of the forwarding UA (which is performing the first forwarding operation), unless the MTA has performed redirection. In case of redirection, the correct First Recipient OR-name must be obtained from the Intended Recipient Name field of the delivery envelope (see 8.3.1.1.1.4 of ISO/IEC 10021-4 | CCITT Recommendation X.411).

e) *EDI Notification Request (sub-field of Recipients):* The UA may forward the EDIM to several recipients by simply adding recipients to the Recipients field. The UA shall set identical EDI Notification Requests for one, and only one, of the recipients. One, and only one, of the UAs to whom the subject EDIM is forwarded will receive the EDI Notification Requests contained in the subject EDIM.

f) *Expiry Time:* may be set to a value different to the value indicated in the subject EDIM.

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

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

h) *EDIN Receiver Field:* may be absent or present. If present, it shall contain only the following value:

   1) Edin-receiver-name: OR-name of the desired EDIN Receiver.

i) Other fields may be added (including EDI Notification Requests).

In both cases other fields apart from those especially mentioned above may, but need not, be copied from the Heading of the subject EDIM to the Heading of the forwarding EDIM (except that the Original EDIM Identifier and First Recipient sub-fields of the EDIN Receiver Field shall not be present).

The Primary Body Part shall be of type EDIM Body Part and shall have the following components:

j) *Parameters:* Specified or omitted as a local matter.

k) *Data:* The EDIM to be forwarded.

17.3.3.5 Submission of forwarded EDIM

The UA shall submit the forwarded EDIM 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 EDIM as specified in 19.2 and 19.4.

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

4) Priority: As specified by Message Delivery Envelope argument.

5) Per-message-indicators and Extensions: A local matter.

6) Deferred-delivery-time: Omitted.

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

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

17.3.3.6 Construction of FN

The UA shall construct an FN if, and only if, one is requested by means of the EDI Notification Requests field of the subject EDIM's Recipients field and the user is not willing to accept Responsibility for the message and forwards the request for notification.

The FN shall have the common fields as prescribed for construction of PN (see 17.3.1.1).

The FN shall have the following forwarding fields:

a) *Forwarded To:* the OR-name of the recipient to whom the request for notification was forwarded.

b) *Forwarded Reason Code:* The reason why the subject message was forwarded.

c) Optionally, FN Supplementary Information that adds information to the reason given.

17.3.3.7 Submission of FN

The UA shall submit the FN (if any) above by invoking Message Submission. The Message Submission Envelope argument shall be as prescribed for acceptance of Responsibility (see 17.3.1) except that the *notification-type* in the "per-message-indicators" should be set to "type 3", its Content argument determined from the FN as specified in 19.1.

NOTE – If the OR-name in the EDIN Receiver field is not valid, then the UA cannot submit the EDIN. Procedures to be followed in this case are a local matter.

Section 4 - Message store

18 Message store operation

ISO/IEC 10021-5 | CCITT Recommendation X.413 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 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 shall not be construed as mandating the provision of an MS or the services it provides. Use of the word "shall" with respect to MS specifications shall be construed as mandating the specifications of an MS, if one is provided.

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

An MS may optionally offer additional support for the EDI Messaging specific attribute types and auto actions, which would qualify it as an EDI Messaging specific MS (EDI-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 ISO/IEC 10021-5 | CCITT Recommendation X.413. Attention should be given to the following points when using the MS for EDI Messaging.

18.1.1 Abstract-bind argument

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

a) *Fetch-restrictions:* The name of the object identifier for the EDI content type is "*id-mct-pedi*", the value is defined in annex A.

b) *Allowed-EITs:* The names of the object identifiers so far standardized in this part of ISO/IEC 10021 are defined in annex A. See also 19.4.

18.2 Abstract-bind result

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

- Available-auto-actions.

NOTE – The use of the general auto action "auto-forward" is not defined for use with EDIMs. Instead the EDI Messaging specific auto actions should be used, covering "edi-forward-with-responsibility-accepted" and "edi-forward-with-responsibility-not-accepted".

18.3 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 an EDIM or EDIN that is delivered to it.

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

c) The MS shall assign sequence numbers to the messages in the hierarchy formed by a forwarding EDIM and its forwarded EDIMs. The lowest number shall be assigned to the outermost level of the hierarchy.

The general (content independent) attributes that may occur in a stored-messages information-base are documented in ISO/IEC 10021-5 | CCITT Recommendation X.413. All content-independent MS attributes can be used for the content defined in this part of ISO/IEC 10021. The EDI specific attributes for stored-messages are defined in 18.7. All general attribute types classified as "mandatory" in table 1 of ISO/IEC 10021-5 | CCITT Recommendation X.413 shall be supported.

18.3.1 Mapping of an MHS message in MS

NOTE – In what follows, reference is made to an "MHS message". This is not be confused with the term "*message*", which refers to an EDIM.

When an EDIM or EDIN gets delivered into the 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 Interchange Length etc. It then generates attributes based on protocol elements in the MHS Envelope, in the Heading and one attribute containing the whole EDI Interchange. The attribute EDI Body Part Type signals which EDI standard has been used. Similarly, other body parts will be mapped into one or several additional attributes.

Figure 7 describes how an MHS message with an EDIM is mapped into a corresponding MS entry.

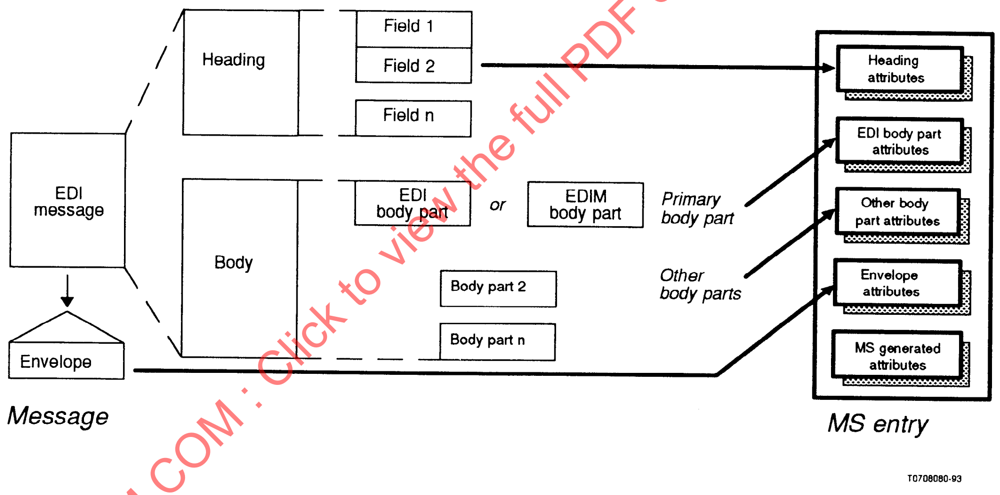

Figure 7 - MHS message with EDIM - Mapping in MS

18.3.2 Mapping of forwarding messages in MS

A forwarding EDIM shall be mapped into the Message Store as one main entry and one or more linked child entries. The final child entry shall contain the original EDIM (with its interchange and any additional body parts).

The MS Structure of a forwarding message such as the message in figure 6 is depicted in figure 8.

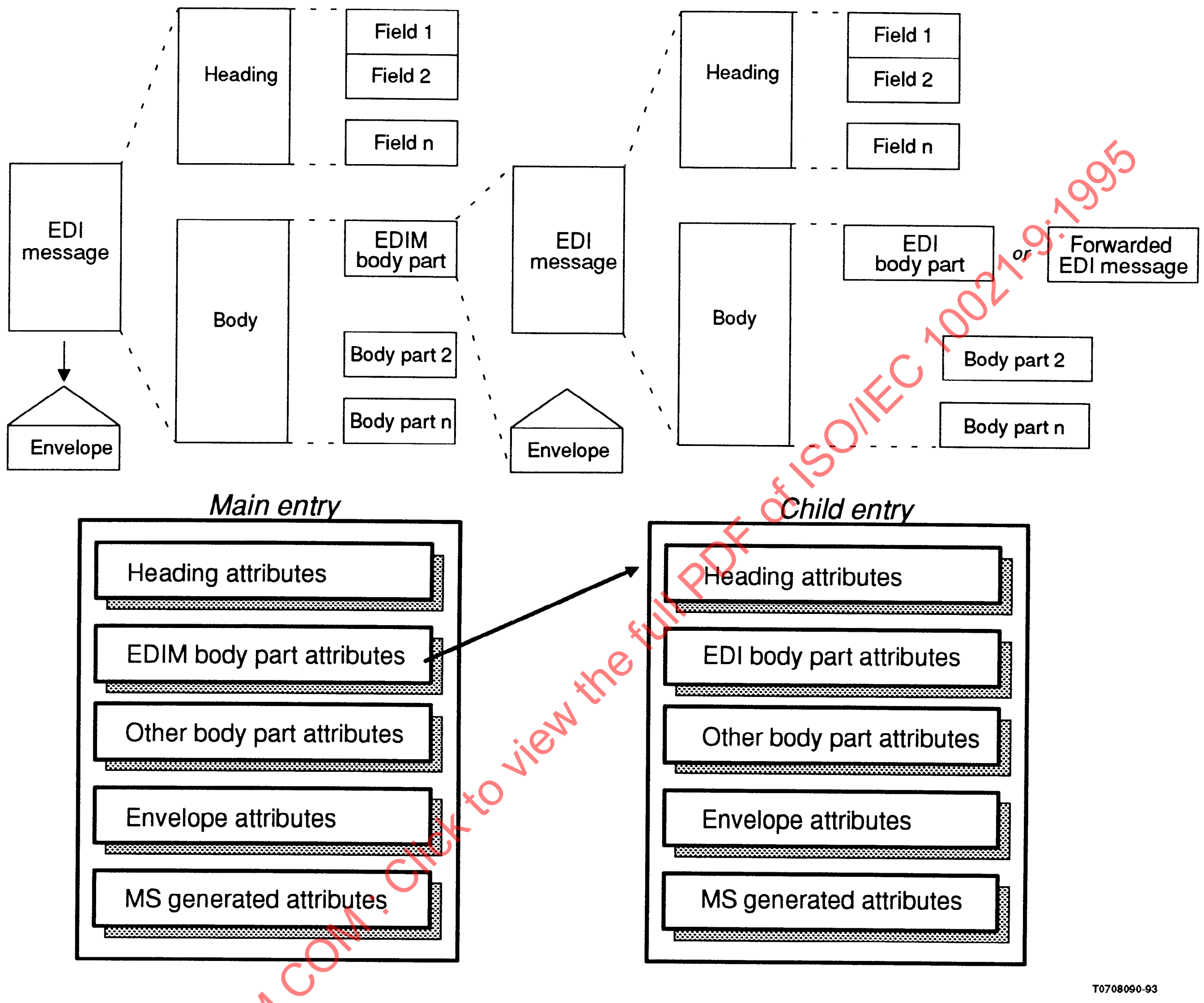

Figure 8 - Forwarding message in MS

18.4 Maintenance of attributes

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

a) For each EDIM or EDIN it holds, the MS shall support the attributes of 18.7 as specified therein.

b) For each EDIM 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 on it and the definition of that auto-action causes a change of entry-status.

c) For each EDIN 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 on it and the definition of that auto-action causes a change of entry-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 an EDIM or EDIN that is delivered to the MS shall have as value the Object Identifier *id-mct-pedi* (see annex A).

18.5 Negative notification

When it discards an EDIM while performing the Delete abstract operation of the MS Abstract Service, the MS shall submit an NN if one is requested and the EDIM's MS-status attribute has the value listed.

18.6 Auto action types

The concept of auto actions is described in subclause 6.5 and clause 12 of of ISO/IEC 10021-5 | CCITT Recommendation X.413. This introduces two general auto action types, which can potentially be used for all content-types. However, the "auto-forward" auto action is not defined for the EDIM content-type. Instead, a specific auto action for EDI Forwarding is defined below.

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

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

The EDI-auto-forward auto action is described in the following. The operation of this auto action may be affected by the implementation of a security policy.

The EDI-auto-forward auto action is defined below together with its abstract syntax using the AUTO-ACTION macro as defined in 6.5 of ISO/IEC 10021-5 | CCITT Recommendation X.413.

The EDI-auto-forward allows EDIMs to be forwarded as follows:

– forwarding-with-responsibility-not-accepted, which means that the EDI responsibility is forwarded. See a) of 17.3.3;

– forwarding-with-responsibility-accepted, which means that the EDI responsibility is accepted. See b) of 17.3.3.

As specified in 17.3.3, if no notifications are requested EDI-auto-forwarding may take place, and is equivalent to "forwarding-with-responsibility-accepted".

If EDI Security Requests are present, then the EDI-auto-forward actions defined above may be prohibited, subject to the security policy in force. If EDI Security Requests are present then the EDI-auto-forward action "forwarding-with-responsibility-accepted" shall not be performed.

The EDI-auto-forward allows one or more sets of EDI-auto-forward-registration-parameters to be registered with the MS, each identified by its registration-identifier. Each EDI-auto-forward-registration-parameter specifies criteria to determine whether it applies to a delivered EDIM, and if so, a copy of the message is EDI-auto-forwarded using the Message-submission abstract operation. The delivered EDIM may be automatically deleted afterwards. If a delivered EDIM matches more than one set of criteria, see 18.8.1 for a description of the interactions.

Below is the ASN.1 definition of the "edi-auto-forward" AUTO ACTION:

```
edi-auto-forward      AUTO-ACTION
  REGISTRATION PARAMETER IS EDIAutoForwardRegistrationParameter
  ::= id-act-edi-auto-forward
    EDIAutoForwardRegistrationParameter ::= SEQUENCE {
      filter                   [0] Filter OPTIONAL,
      edi-supplementary-info         [1] EDISupplementaryInfo OPTIONAL,
      delete-after-forwarding        [2] BOOLEAN DEFAULT FALSE,
      edi-forwarding-mode            CHOICE {
        forwarding-with-responsibility-not-accepted [3] ForwardWithRespNotAccepted,
        forwarding-with-responsibility-accepted     [4] ForwardWithRespAccepted }
```

NOTE – The data type Filter is defined in 8.1.2.1 of ISO/IEC 10021-5 | CCITT Recommendation X.413. The data type Forward With Resp Not Accepted is defined in 18.6.1 of this part of ISO/IEC 10021 and Forward With Resp Accepted is defined in 18.6.2 of this part of ISO/IEC 10021.

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

a) *Filter:* This is a set of criteria which a new entry representing a delivered EDIM 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 forwarding.

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

b) *EDI-supplementary-info:* This parameter can contain text to be included in the EDIN "fn-supplementary-information" field of an EDIN and in the other-parameters field of a forwarded EDIM.

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.

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

   1) forwarding-with-responsibility-not-accepted,

   2) forwarding-with-responsibility-accepted,

The remaining parameters are described separately for the two cases below.

Support of the EDI-auto-forward auto-action by an MS, or UA accessing an MS, requires that it supports registration of the auto-forward-registration-paremeter via the Register MS abstract operation.

18.6.1 Forwarding with responsibility not accepted

The forwarding-with-responsibility-not-accepted case enables the MS abstract service provider to automatically forward, with EDI Responsibility forwarded, any EDIM (with notification requests) that has been delivered into the stored-messages information base. The use of this auto action shall be subject to the requirements of the security policy in force. The MS shall follow the rules in 17.3.3.1. Appropriate values are added in the EDI Notification Indicator attribute.

The following limitations apply to forwarding-with-responsibility-not-accepted, when compared to the general rules for forwarding contained in 17.3.3:

a) The "forwarding-with-responsibility-not-accepted" auto action type shall only be performed once for a particular EDIM by the same MS.

b) Only one recipient shall be specified for the forwarding auto action.

```
ForwardWithRespNotAccepted ::= SET {
  COMPONENTS OF PerMessageAutoForwardFields, -- from envelope PerMessageFields
  per-recipient-field                   [3] PerRecipientAutoForwardFields,
  notification-argument                 [4] NotificationArguments OPTIONAL }

NotificationArguments ::= SET {
  COMPONENTS OF PerMessageAutoForwardFields, -- from envelope PerMessageFields
  per-recipients-field                  [3]  NotificationPerRecipientAutoForwardFields }
```

```
NotificationPerRecipientAutoForwardFields      ::= SET {
  recipient-name           RecipientName,
  explicit-conversion      IMPLICIT ExplicitConversion OPTIONAL,
  extensions               IMPLICIT PerRecipientMessageSubmissionExtensions DEFAULT {} }
```

NOTE – The data types Per Message Auto Forward Fields and Per Recipient Auto Forward Fields are defined in 12.1 of ISO/IEC 10021-5 | CCITT Recommendation X.413.

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

c) *Per Message Auto Forward Fields:* This is a set of arguments registered to be used for each Message Submission abstract operation (see 8.2.1.1.1 of ISO/IEC 10021-4 | CCITT Recommendation X.411). 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 "conversion-with-loss-allowed", 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 it is registered with the value "conversion-with-loss-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 it 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 argument is not registered, its presence as Message Submission arguments depends upon the presence of the corresponding message delivery arguments, its value being transformed where appropriate: 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 shall not be registered. These are: proof-of-submission-request, original-encoded-information-types and content-type.

d) *Per Recipient Auto Forward Fields:* This is a set of arguments registered to be used for each Message Submission abstract operation (see 8.2.1.1.1 of ISO/IEC 10021-4 | CCITT Recommendation X.411). Any argument which is not registered, not mandatory, and not specifically mentioned below, will be absent from each message submission.

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.

e) *Notification-argument:* This contains a subset of the same parameters as in c) and d) above, but the actual values can differ from the values in the forwarded EDIM. Notification-argument is used in the Message Submission abstract operation of any EDINs generated by an MS.

18.6.2 Forwarding with responsibility accepted

The forwarding-with-responsibility-accepted case enables the MS abstract service provider to automatically forward, with Responsibility accepted, any EDIM that has been delivered into the stored-messages information base. The use of this auto action shall be subject to the requirements of the security policy in force. The MS shall follow the rules in 17.3.3.2. Appropriate values are added in the EDI Notification Indication attribute.

The following limitations apply to forwarding-with-responsibility-accepted, when compared to the general rules for forwarding contained in 17.3.3:

a) The MS shall construct and forward an EDIM whose primary body part comprises a body part of type EDIM body part as described in 17.3.3.4, however no body parts shall be removed or added, the original delivery envelope shall be included and the components of the original Heading shall be copied to the Heading of the forwarding EDIM according to the rules in 17.3.3 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.

```
ForwardWithRespAccepted ::= SET {
  COMPONENTS OF PerMessageAutoForwardFields, -- from envelope PerMessageFields
  per-recipients-field              [3] SEQUENCE SIZE (1..ub-recipients) OF
                                    PerRecipientAutoForwardFields,
  notification-argument             [4] NotificationArguments OPTIONAL,
  new-edin-receiver-name            [5] RecipientField OPTIONAL,
  per-recipient-heading-fields      [6] SEQUENCE SIZE (1..ub-recipients) OF
                                    NextRecipientFields OPTIONAL }

NextRecipientFields::= SEQUENCE {
  next-recipient                       [0] RecipientField,
  next-recipient-action-request        [1] ActionRequestField DEFAULT {id-for-action},
  next-recipient-edi-notification-requests-field [2] EDINotificationRequestsField
  OPTIONAL,
  next-responsibility-passing-allowed [3] ResponsibilityPassingAllowedField DEFAULT
  FALSE }
```

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

b) *Per Message Auto Forward Fields:* The description is the same as in **c)** of 18.6.1.

c) *Per Recipient Auto Forward Fields:* The description is the same as in **d)** of 18.6.1.

Multiple recipients are allowed for this case.

d) *Notification-argument:* The description is the same as in e) of 18.6.1.

e) "new-edin-receiver-name" to replace "edin-receiver-name" in EDIN Receiver Field.

f) *HeadingFields:* New values may be defined for:

1) "next-recipient" to replace "recipient" in Recipients Sub Field. This field is mandatory;

2) "next-recipient-action-request" to replace "recipient-action-request" in Recipients Sub Field;

3) "next-recipient-edi-notification-requests-field" to replace "recipient-edi-notification-requests-field" in Recipients Sub Field;

4) "next-responsibility-passing-allowed" to replace "responsibility-passing-allowed" in Recipients Sub Field.

18.7 Message store attributes

As described in ISO/IEC 10021-5 | CCITT Recommendation X.413, 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 e.g. EDI Messaging kind.

The following defines the MS attributes specific to EDI Messaging. EDI specific attributes are defined below.

All of the attributes defined in this part of ISO/IEC 10021, except those corresponding to extended body part types (which cannot be enumerated), are listed alphabetically, for reference, in the first column of table 1. This table records their presence in a delivered message entry. None of them appears in a delivered report entry. Additional, unnamed attributes are described in 18.7.4.5. Table 2 describes how the EDI attributes are generated.

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

NOTE – See 5.3 and 5.4 for an elaboration of the legend of the tables.

Table 1 - Summary of EDI specific MS attribute types

Attribute	Single/ Multi valued	Support level by MS and UA	Presence in delivered EDIM	Presence in delivered PN	Presence in delivered NN	Presence in delivered FN	Available for list, alert	Available for summarize
acknowledgement-request-for-this-recipient	S	O	P	–	–	–	Y	N
action-request-for-this-recipient	S	O	P	–	–	–	Y	N
application-reference	S	O	C	–	–	–	Y	N
authorization-information-for-this-recipient	S	O	C	–	–	–	Y	N
body	S	M	P	–	–	–	N	N
communications-agreement-id-for-this-recipient	S	O	C	–	–	–	Y	N
cross-referencing-information	M	O	C	–	–	–	Y	N
date-and-time-of-preparation	S	M	C	–	–	–	Y	N
edi-application-security-elements	S	O	C	–	–	–	Y	N
edi-application-security-extentions	M	O	C	–	–	–	Y	N
edi-body-part	S	M	P	–	–	–	N	N
edi-bodypart-type	S	M	P	–	–	–	Y	Y
edi-message-type	M	M	C	–	–	–	Y	N
edi-notification-indicator	M	O	C–	–	–	–	Y	N
edi-notification-requests-for-this-recipient	S	O	C	–	–	–	Y	N
edi-notification-security-for-this-recipient	S	O	C	–	–	–	Y	N
edi-reception-security-for-this-recipient	S	O	C	–	–	–	Y	N
edim-body-part	S	O	C	–	–	–	N	N
edim-synopsis	S	O	P	–	–	–	N	N
edims-entry-type	S	M	P	P	P	P	Y	Y
edin-initiator	S	O	–	P	P	P	Y	N
edin-originator	S	O	–	P	P	P	Y	N
edin-receiver	S	O	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
fn-extensions	M	O	–	–	–	C	Y	N
fn-reason-code	S	O	–	–	–	P	Y	N
fn-supplementary-information	S	O	–	–	–	C	Y	N
forwarded-to	S	O	–	–	–	P	Y	N
heading	S	M	P	–	–	–	N	N
heading-extensions	M	O	C	–	–	–	Y	N
incomplete-copy	S	O	P	–	–	–	Y	N
interchange-control-reference-for-this-recipient	S	M	C	–	–	–	Y	N
interchange-length	S	O	P	–	–	–	Y	N

Table 1 ***(concluded)***

Attribute	Single/ Multi valued	Support level by MS and UA	Presence in delivered EDIM	Presence in delivered PN	Presence in delivered NN	Presence in delivered FN	Available for list, alert	Available for summarize
interchange-sender	S	M	C	–	–	–	Y	N
interchange-recipient-for-this-recipient	S	M	C	–	–	–	Y	N
interchange-sender	S	M	C	–	–	–	Y	N
message-data	S	O	C	–	–	–	N	N
message-parameters	S	O	C	–	–	–	N	N
nn-extensions	M	O	–	–	C	–	Y	N
nn-reason-code	S	O	–	–	P	–	Y	N
nn-supplementary-information	S	O	–	–	C	–	Y	N
notification-security-elements	S	O	–	C	C	C	Y	N
notification-time	S	O	–	P	P	P	Y	N
notifications-extensions	M	O	–	C	C	C	Y	N
obsoleted-edims	M	O	C	–	–	–	Y	N
originator	S	O	C	–	–	–	Y	N
pn-extensions	M	O	–	C	–	–	Y	N
pn-supplementary-information	S	O	–	C	–	–	Y	N
processing-priority-code-for-this-recipient	S	O	C	–	–	–	Y	Y
recipient-extensions-for-this-recipient	M	O	C	–	–	–	Y	N
recipient-reference-for-this-recipient	S	O	C	–	–	–	Y	N
related-messages	M	O	C	–	–	–	Y	N
responsibility-forwarded	S	O	P	–	–	–	Y	Y
responsibility-passing-allowed for-this-recipient	S	O	P	–	–	–	Y	N
service-string-advice	S	O	C	–	–	–	Y	N
subject-edim	S	M	–	P	P	P	Y	N
syntax-identifier	S	M	C	–	–	–	Y	Y
test-indicator-for-this-recipient	S	O	P	–	–	–	Y	Y
this-edim	S	M	P	–	–	–	Y	N
this-recipient	S	O	C	–	–	–	Y	N

Table 2 - Generation of the EDI specific MS attribute types

Attribute-type-name	Source parameters	Source generated by	Generation rules
acknowledgement-request-for-this-recipient	acknowledgement-request	MD	The attribute-value is the value of the parameter in the recipient-sub-field for this recipient. If the source parameter is missing, an attribute with the default value shall be generated.
action-request-for-this-recipient	action-request	MD	The attribute-value is the value of the parameter in the recipient-sub-field for this recipient. If the source parameter is missing, an attribute with the default value shall be generated.
application-reference	application-reference	MD	The value of the parameter is the attribute value.
authorization-information-for-this-recipient	authorization-information	MD	The attribute-values is the value of the parameter in the recipient-sub-field for this recipient.
body	body	MD	The value of the parameter is the attribute value.
communications-agreement-id-for-this-recipient	communications-agreement-id	MD	The attribute-values is the value of the parameter in the recipient-sub-field for this recipient.
cross-referencing-information	cross-referencing-information	MD	A value is generated from each value of the SET.
date-and-time-of-preparation	date-and-time-of-preparation	MD	The value of the parameter is the attribute value.
edi-application-security-elements	edi-application-security-elements	MD	The value of the parameter is the attribute value.
edi-application-security-extensions	edi-application-security-extensions	MD	A value is generated from each value of the SET.
edi-body-part	edi-body-part	MD	The value of the parameter is the attribute value.
edi-bodypart-type	edi-bodypart-type	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.
edi-message-type	edi-message-type	MD	A value is generated from each value of the SET.
edi-notification-indicator	NONE	MS	A value is added when an EDIN is submitted from the MS.
edi-notification-requests-for-this-recipient	edi-notification-request	MD	The attribute-value is the value of the parameter in the recipient-sub-field for this recipient.
edi-notification-security-for-this-recipient	edi-notification-security	MD	The value of the parameter is the attribute value.
edi-reception-security-for-this-recipient	edi-reception-security	MD	The value of the parameter is the attribute value.

Table 2 *(continued)*

Attribute-type-name	Source parameters	Source generated by	Generation rules
edim-body-part	NONE	MS	The value is the sequence-number of the entry created for the forwarded EDIM.
edin-synopsis	see subclause 18.7.1.2	MS	see subclause 18.7.1.2.
edims-entry-type	InformationObject and edin	MS	If the information object is an EDIM, the value is set to "edim". If the information object is an EDIN, the value is set according to the type of the EDIN.
edin-originator	edin-originator	MD	The value of the parameter is the attribute value.
edin-receiver	edin-receiver	MD	The value of the parameter is the attribute value.
expiry-time	expiry-time	MD	The value of the parameter is the attribute value.
externally-defined-body-part-types	additional-body-parts	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	first-recipient	MD	The value of the parameter is the attribute value.
fn-extensions	fn-extensions	MD	A value is generated from each value of the SET.
fn-reason-code	fn-reason-code	MD	The value of the parameter is the attribute value.
fn-supplementary-information	fn-supplementary-information	MD	The value of the parameter is the attribute value.
forwarded-to	forwarded-to	MD	The value of the parameter is the attribute value.
heading	heading	MD	The value of the parameter is the attribute value.
heading-extensions	heading-extensions	MD	A value is generated from each value of the SET.
incomplete-copy	incomplete-copy	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.
interchange-control-reference-for-this-recipient	interchange-control-reference	MD	The attribute-values is the value of the parameter in the recipient-sub-field for this recipient.
interchange-length	NONE	MS	The value is the number of octets occupied by the source parameter.
interchange-recipient-for-this-recipient	interchange-recipient	MD	The attribute-value is the value of the parameter in the recipient-sub-field for this recipient.

Table 2 *(concluded)*

Attribute-type-name	Source parameters	Source generated by	Generation rules
interchange-sender	interchange-sender	MD	The value of the parameter is the attribute value.
message-data	data	MD	The value of the parameter is the attribute value.
message-parameters	message-parameters	MD	The value of the parameter is the attribute value.
nn-extensions	nn-extensions	MD	A value is generated from each value of the SET.
nn-reason-code	nn-reason-code	MD	The value of the parameter is the attribute value.
nn-supplementary-information	nn-supplementary -information	MD	The value of the parameter is the attribute value.
notification-security-elements	notification-security-elements	MD	The value of the parameter is the attribute value.
notification-time	notification-time	MD	The value of the parameter is the attribute value.
notifications-extensions	notifications-extensions	MD	A value is generated from each value of the SET.
obsoleted-edims	obsoleted-EDIMs	MD	A value is generated from each value of the SEQUENCE.
originator	originator	MD	The value of the parameter is the attribute value.
pn-extensions	pn-extensions	MD	A value is generated from each value of the SET.
pn-supplementary-information	pn-supplementary-information	MD	The value of the parameter is the attribute value.
processing-priority-code-for-this-recipient	processing-priority-code	MD	The attribute-value is the value of the parameter in the recipient-sub-field for this recipient.
recipient-extensions-for-this-recipient	recipient-extensions	MD	A value is generated from each value of the SET in the recipient-sub-field for this recpient.
recipient-reference-for-this-recipient	recipient-reference	MD	The attribute-value is the value of the parameter in the recipient-sub-field for this recipient.
related-messages	related-messages	MD	A value is generated from each value of the SEQUENCE.
responsibility-forwarded	responsibility-forwarded	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.
responsibility-passing-allowed-for-this-recipient	responsibility-passing-allowed	MD	The attribute-value is the value of the parameter in the recipient-sub-field for this recipient. If the source parameter is missing, an attribute with the default value shall be generated.
service-string-advice	service-string-advice	MD	The value of the parameter is the attribute value.
subject-edim	subject-edim	MD	The value of the parameter is the attribute value.
syntax-identifier	syntax-identifier	MD	The value of the parameter is the attribute value.
test-indicator-for-this-recipient	test-indicator	MD	The attribute-value is the value of the parameter in the recipient-sub-field for this recipient. If the source parameter is missing, an attribute with the default value shall be generated.
this-edim	this-EDIM	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.

18.7.1 Summary attributes

Some attributes summarize an EDI Messaging information object. These attributes are defined and described below.

18.7.1.1 EDIMS entry type

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

```
edims-entry-type ATTRIBUTE
   WITH ATTRIBUTE-SYNTAX EDIMSEntryType
   MATCHES FOR EQUALITY
   SINGLE VALUE
   ::= id-sat-edims-entry-type

EDIMSEntryType ::= ENUMERATED {
   edim (0),
   pn (1),
   nn (2),
   fn (3) }
```

This attribute may assume any one of the following values:

a) *edim:* The information object is an EDIM.

b) *pn:* The information object is a PN.

c) *nn:* The information object is an NN.

d) *fn:* The information object is an FN.

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

18.7.1.2 EDIM synopsis

The EDIM Synopsis attribute gives the structure, characteristics, size, and processing status of an EDIM at the granularity of individual body parts. This attribute is created when an EDIM is delivered to the MS.

```
edim-synopsis ATTRIBUTE
   WITH ATTRIBUTE-SYNTAX EDIMSynopsis
   SINGLE VALUE
   ::= id-sat-edim-synopsis
```

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

```
EDIMSynopsis ::= 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 (i.e. body-parts other than a forwarded EDIM). This enables the synopsis of a forwarding EDIM to encompass the body parts of each forwarded EDIM (recursively), as well as those of the forwarding EDIM itself.

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

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

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

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

a) *Number:* The sequence number that the MS assigns to the entry that the Message body part represents. This component is generated when a child-entry is created.

b) *Synopsis:* The synopsis of the EDIM that forms the content of the message that the body part represents. This component is generated when a child-entry is created.

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) *Type:* This value is generated when the entry is created. If the Non-message Body Part is an edi-body-part, the value is the object identifier value contained in the edi-bodypart-type attribute contained in this entry. If it is a removed-edi-body, the value is set to "*id-syn-removed*" (See annex A). If it is a place-holder, the value is set to "*id-syn-place-holder*" (again, see annex A). If it is an external-body-part, the value is set to the Direct-reference component of the body part's Data component.

d) *Parameters:* This value is generated if the Non-message Body Part is an external-body-part. It contains that body part's Parameter component, which may describe the body part's format and control parameters.

e) *Size:* This value is created when the entry is created. The value is set to the size in octets of the encoding of the Encoding component of the body part's Data component when the Basic Encoding Rules of ISO/IEC 8825 | CCITT Recommendation X.209 are followed. If those rules permit several (e.g., both primitive and constructed) encodings of the component, the size may reflect any one of them.

f) *Processed (*default *false):* An indication of whether or not the body part has been conveyed to the UA by means of the MS Fetch abstract operation, or has been processed by an auto-action and the definition of that auto-action causes a change of entry-status. This value is set to the default value when the EDIM is delivered to the MS and is updated as described in 18.4.

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

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

18.7.2 EDI notification indicator

The EDI Notification Indicator attribute contains information about whether any EDI Notifications have been sent by the MS in response to an EDIM, and if so, which type of EDI Notifications were sent. The MS creates this attribute for each new EDIM and maintains the attribute values, depending on the auto-actions performed.

NOTE - When notifications are generated by the UA, the values of this attribute are not affected.

```
edi-notification-indicator ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX EDINotificationIndicator DEFAULT (0)
  MATCHES FOR EQUALITY
  MULTI VALUE ::=
  id-sat-edi-notification-indicator

EDINotificationIndicator ::= ENUMERATED {
   no-notification-sent (0),
   pn-sent              (1),
   nn-sent              (2),
   fn-sent              (3) }
```

Each value of this attribute may assume one of the following values:

a) *no-notification-sent:* This is the initial value set by the MS when a new MS entry is created for the EDIM.

b) *pn-sent:* This value means that the MS has generated and sent a Positive Notification (PN) in response to a request for a PN.

c) *nn-sent:* This value means that the MS has generated and sent a Negative Notification (NN) in response to a request for an NN.

d) *fn-sent:* This value means that the MS has generated and sent a Forwarded Notification (FN) in response to a request for an FN.

18.7.3 Heading attributes

Some attributes are derived from the Heading of an EDIM. These attributes are defined and described below.

18.7.3.1 Heading

The Heading attribute is the (entire) Heading of an EDIM.

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

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

18.7.3.2 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 8.2 for semantics.

```
this-edim ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX ThisEDIMField
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-hat-this-edim

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

edin-receiver ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX EDINReceiverField
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-hat-edin-receiver

responsibility-forwarded ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX ResponsibilityForwarded
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-hat-responsibility-forwarded

edi-bodypart-type ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX EDIBodyPartType
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-hat-edi-bodypart-type

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

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

```
   related-messages ATTRIBUTE
     WITH ATTRIBUTE-SYNTAX RelatedMessageReference
     MATCHES FOR EQUALITY
     MULTI VALUE
     ::= id-hat-related-messages

   obsoleted-edims ATTRIBUTE
     WITH ATTRIBUTE-SYNTAX ObsoletedEDIMsSubfield
     MATCHES FOR EQUALITY
     MULTI VALUE
     ::= id-hat-obsoleted-edims

   edi-application-security-element ATTRIBUTE
     WITH ATTRIBUTE-SYNTAX EDIApplicationSecurityElement
     MATCHES FOR EQUALITY
     SINGLE VALUE
     ::= id-hat-edi-application-security-element

   edi-application-security-extensions ATTRIBUTE
     WITH ATTRIBUTE-SYNTAX EDIApplicationSecurityExtension
     MATCHES FOR EQUALITY
     MULTI VALUE
     ::= id-hat-edi-application-security-extensions

   cross-referencing-information ATTRIBUTE
     WITH ATTRIBUTE-SYNTAX CrossReferencingInformationSubField
     MATCHES FOR EQUALITY
     MULTI VALUE
     ::= id-hat-cross-referencing-information
```

Fields from EDIFACT Interchange:

```
   edi-message-type ATTRIBUTE
     WITH ATTRIBUTE-SYNTAX EDIMessageTypeFieldSubField
     MATCHES FOR EQUALITY
     MULTI VALUE
     ::= id-hat-edi-message-type

   service-string-advice ATTRIBUTE
     WITH ATTRIBUTE-SYNTAX ServiceStringAdviceField
     MATCHES FOR EQUALITY
     SINGLE VALUE
     ::= id-hat-service-string-advice

   syntax-identifier ATTRIBUTE
     WITH ATTRIBUTE-SYNTAX SyntaxIdentifierField
     MATCHES FOR EQUALITY
     SINGLE VALUE
     ::= id-hat-syntax-identifier

   interchange-sender ATTRIBUTE
     WITH ATTRIBUTE-SYNTAX InterchangeSenderField
     MATCHES FOR EQUALITY
     SINGLE VALUE
     ::= id-hat-interchange-sender

   date-and-time-of-preparation ATTRIBUTE
     WITH ATTRIBUTE-SYNTAX DateAndTimeOfPreparationField
     MATCHES FOR EQUALITY ORDERING
     SINGLE VALUE
     ::= id-hat-date-and-time-of-preparation

   application-reference ATTRIBUTE
     WITH ATTRIBUTE-SYNTAX ApplicationReferenceField
     MATCHES FOR EQUALITY SUBSTRINGS
     SINGLE VALUE
     ::= id-hat-application-reference
```

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 an EDIM whose Heading contains the field or sub-field whose name the attribute bears.

18.7.3.3 Recipient sub-field

Some attributes bear the names of Recipient fields and have sub-fields of those fields as their values. See 8.2.3 for semantics.

```
this-recipient ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX RecipientField
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-rat-this-recipient

action-request-for-this-recipient ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX ActionRequestField
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-rat-action-request-for-this-recipient

edi-notification-requests-for-this-recipient ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX EDINotificationRequests
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-rat-edi-notification-requests-for-this-recipient

edi-notification-security-for-this-recipient ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX EDINotificationSecurity
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-rat-edi-notification-security-for-this-recipient

edi-reception-security-for-this-recipient ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX EDIReceptionSecurity
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-rat-edi-reception-security-for-this-recipient

responsibility-passing-allowed-for-this-recipient ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX ResponsibilityPassingAllowedField
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-rat-responsibility-passing-allowed-for-this-recipient

-- Fields from EDIFACT interchange

interchange-recipient-for-this-recipient ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX InterchangeRecipientField
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-rat-interchange-recipient-for-this-recipient

recipient-reference-for-this-recipient ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX RecipientReferenceField
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-rat-recipient-reference-for-this-recipient
```

```
interchange-control-reference-for-this-recipient ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX InterchangeControlReferenceField
  MATCHES FOR EQUALITY SUBSTRINGS
  SINGLE VALUE
  ::= id-rat-interchange-control-reference-for-this-recipient
processing-priority-code-for-this-recipient ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX ProcessingPriorityCodeField
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-rat-processing-priority-code-for-this-recipient

acknowledgement-request-for-this-recipient ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX AcknowledgementRequestField
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-rat-acknowledgement-request-for-this-recipient

communications-agreement-id-for-this-recipient ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX CommunicationsAgreementIdField
  MATCHES FOR EQUALITY SUBSTRINGS
  SINGLE VALUE
  ::= id-rat-communications-agreement-id-for-this-recipient

test-indicator-for-this-recipient ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX TestIndicatorField
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-rat-test-indication-for-this-recipient
-- END Fields from EDIFACT

-- Fields from ANSIX12 ISA

authorization-information-for-this-recipient ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX AuthorizationInformationField
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-rat-authorization-information-for-this-recipient

-- END Fields from ANSIX12 ISA
```

Extensions:

```
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 an EDIM whose Heading Recipients field contains the field whose name the attribute bears. It shall maintain one attribute value for each sub-field.

18.7.4 Body attributes

Some attributes are derived from the Body of an EDIM. These attributes are defined and described below.

18.7.4.1 Body

The Body attribute is the (entire) Body of an EDIM.

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

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

18.7.4.2 Body analyses

Some attributes have as their values information about the body parts contained in the body of the message.

The interchange length attribute is created by the Message Store when it receives an EDIM. Its value indicates the length of the EDI Interchange carried in the Primary Body Part of the message.

```
interchange-length ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX InterchangeLength
  MATCHES FOR ORDERING
  SINGLE VALUE
  ::= id-bat-interchange-length

InterchangeLength ::= INTEGER
```

The Interchange Length gives the number of octets occupied by the EDI Interchange.

18.7.4.3 Primary body parts

Some attributes bear the names of the Primary Body Part types and have such body parts as their values. See 8.3.1 for semantics.

```
edi-body-part ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX EDIBodyPart
  SINGLE VALUE
  ::= id-bat-edi-body-part
```

An MS holds each forwarded EDIM (i.e., each Message body part) as an information object in its own right, separate from the forwarding EDIM. (stored as a separate child entry in the stored-messages information base). That information object, of course, is a message whose content is an EDIM. The EDIM Body Parts attribute below, therefore, has as its values the sequence numbers the MS assigns to those messages. See 8.3.2 for semantics.

```
edim-body-part ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX SequenceNumber -- sequence number of the forwarded EDIM
  entry.
  SINGLE VALUE
  ::= id-bat-edim-body-part
```

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 EDIM. It shall maintain one attribute value for such a body part.

Some attributes bear the names of the Parameters and Data components of an EDIM 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 an EDIM whose Body contains an EDIM Body Part.

18.7.4.4 Externally defined body part types

The Externally Defined Body Part Types attribute identifies the externally defined body part types represented in an EDIM.

```
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 an EDIM whose Body contains one or more Additional 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 ISO/IEC 10021-7 | CCITT Recommendation X.420.

NOTES

1 - When a new extended body part type and a new EIT are defined and have a one to one relationship, then the same Object Identifier may be used for the data component of the EIT.

2 – Some Externally Defined body part types are defined in ISO/IEC 10021-7 | CCITT Recommendation X.420. Values are as specified in 7.3.12 of ISO/IEC 10021-7 | CCITT Recommendation X.420.

18.7.4.5 Externally defined body parts

Some attributes, unnamed, have as their values the Encoding components (see 8.3.3) of the ASN.1 Externals that constitute the Data components of the Externally Defined Body Part.

To each externally defined body part type there correspond two attributes. The first attribute is denoted by the object identifier that is the Direct-reference component (again, see 8.3.3) of the External that constitutes the Data component of a body part of that type. The content of this first attribute contains the Body Part Reference and the Externally Defined Body Part encoded as an EDIM-Externally Defined Body Part (see 8.3.3).

The second attribute is denoted by the object identifier that is the Direct-reference component of the External that constitutes the Parameters component of a body part of that type. The content of this second attribute contains the Body Part Reference and the Parameters component encoded as follows:

```
EDIExternallyDefinedBodyPartParameterAttribute ::= SEQUENCE {
  body-part-reference    [0] BodyPartReference OPTIONAL,
  parameter              [1] ExternallyDefinedParameters }
```

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

NOTE – The externally defined body part attributes cannot be enumerated in practice because the externally defined body part types cannot be so enumerated.

The Externally Defined Body Part Types attribute determines the Externally Defined body part types for a particular EDIM.

18.7.5 Notification attributes

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

18.7.5.1 Common fields

Some attributes bear the names of Common Fields and have those fields as their values. See 9.1 for semantics.

```
subject-edim ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX SubjectEDIMField
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-nat-subject-edim

edin-originator ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX EDINOriginatorField
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-nat-edin-originator

first-recipient ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX FirstRecipientField
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-nat-first-recipient
```

```
notification-time ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX NotificationTimeField
  MATCHES FOR EQUALITY ORDERING
  SINGLE VALUE
  ::= id-nat-notification-time

notification-security-elements ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX SecurityElementsField
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-nat-notification-security-elements

edin-initiator ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX EDINInitiatorField
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-nat-edin-initiator
```

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 an EDIN that contains the field or sub-field whose name the attribute bears.

18.7.5.2 Positive notification fields

Some attributes bear the names of PN EDIN fields and have those fields as their values. Some attributes bear the names of notification fields and have sub-fields of the PN fields of a notification as their values. See 9.2 for semantics.

```
pn-supplementary-information ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX EDISupplementaryInformation
  MATCHES FOR EQUALITY SUBSTRINGS
  SINGLE VALUE
  ::= id-nat-pn-supplementary-info

pn-extensions ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX PNExtensionsSubField
  MATCHES FOR EQUALITY
  MULTI VALUE
  ::= id-nat-pn-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 PN contains the field whose name the attribute bears. It shall maintain one attribute value for each sub-field.

18.7.5.3 Negative notification fields

Some attributes bear the names of NN EDIN fields and have those fields as their values. Some attributes bear the names of notification fields and have sub- fields of the NN fields of a notification as their values. See 9.3 for semantics.

```
nn-reason-code ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX NNReasonCodeField
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-nat-nn-reason-code

nn-supplementary-information ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX EDISupplementaryInformation
  MATCHES FOR EQUALITY SUBSTRINGS
  SINGLE VALUE
  ::= id-nat-nn-supplementary-info
```

```
nn-extensions ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX NNExtensionsSubField
  MATCHES FOR EQUALITY
  MULTI VALUE
  ::= id-nat-nn-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 an NN that contains the field whose name the attribute bears. It shall maintain one attribute value for each field or sub-field.

18.7.5.4 Forwarded notification fields

Some attributes bear the names of FN EDIN fields and have those fields as their values. Some attributes bear the names of notification fields and have sub-fields of the FN fields of a notification as their values. See 9.4 for semantics.

```
forwarded-to ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX ForwardedTo
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-nat-forwarded-to

fn-reason-code ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX FNReasonCodeField
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-nat-fn-reason-code

fn-supplementary-information ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX EDISupplementaryInformation
  MATCHES FOR EQUALITY SUBSTRINGS
  SINGLE VALUE
  ::= id-nat-fn-supplementary-info

fn-extensions ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX FNExtensionsSubField
  MATCHES FOR EQUALITY
  MULTI VALUE
  ::= id-nat-fn-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 an FN that contains the field whose name the attribute bears. It shall maintain one attribute value for each field or sub-field.

18.8 Procedures for an EDI MS

The procedures for a general MS are specified in clauses 14 and 15 of ISO/IEC 10021-5 | CCITT Recommendation X.413. This reference gives complementary information for MS systems that also explicitly support EDI Messaging.

18.8.1 Additional procedures for message delivery

How the MS consumes the MTS abstract service is described in clause 14 of ISO/IEC 10021-5 | CCITT Recommendation X.413. The following text describes additional information about the procedures needed for EDI Messaging.

If EDI Security Requests are present, then the EDI-auto-forward actions defined above may be prohibited, subject to the security policy in force. If EDI Security Requests are present then the EDI-auto-forward action (forwarding-with-responsibility-accepted) shall not be performed.

The following procedures supplement 14.1 of ISO/IEC 10021-5 | CCITT Recommendation X.413:

a. If EDI auto forwarding criteria are registered by the Register-MS abstract operation, the new entry shall be matched against the criteria registered. The matching shall always proceed starting with the registration having the lowest registration identifier and perform the following auto actions:

   – registrations against the "forward-with-responsibility-accepted" auto action.

If this results in forwarding being performed, it is possible that one or several forwardings may be performed for this EDIM.

- registrations against "forward-with-responsibilty-not-accepted" auto-action.

  If this results in a forwarding being performed, no further EDI forwarding actions shall be performed for this EDIM by the same EDI-MS.

b. If an auto action registration stipulates that the EDIM is to be deleted after forwarding, no further forwarding auto-action can take place.

c. The appropriate notification shall be returned for the first auto- forwarding that is performed for the EDIM.

d. When an EDIN is submitted, a value reflecting the type of the EDIN shall be added to the "edi-notification-indicator" attribute.

e. If an EDI auto-forwarding does not succeed, eg. through a non-delivery, an NN EDIN may be returned to the originator if an FN was previously sent.

f. The entry-status shall be set to "processed" when the MS has performed an EDI-auto-forward auto-action on a delivered message, and the EDIM has not been deleted.

18.8.2 Manual forwarding

When a EDI MS supports forwarding, it shall support the manual forwarding of a message using the forwarding-request extension defined in ISO/IEC 10021-5 | CCITT Recommendation X.413. The UA may submit an EDIM, and identify by means of the forwarding-request extension a received EDIM which is to be forwarded as the primary body part of the submitted EDIM. The MS shall construct an EDIM Body Part from the specified stored-message. The constructed EDIM Body Part replaces the primary body part supplied by the UA.

NOTE - The syntax of the Body of an EDIM does not allow the UA to omit the primary body part from the submitted EDIM. However, the value supplied by the UA services no purpose and may have any value; for example, it may be an EDIM-body-part of length “zero”.

19 Message contents

As has already been seen, various secondary objects (e.g., UAs) have occasion to convey the Information Objects of clause 6 as the contents of messages, as well as to convey probes concerning such 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 the EDI Messaging protocol ($\mathbf{P}_{\mathbf{edi}}$).

19.1 Content

A secondary object that submits a message containing an EDIM or EDIN 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 ISO/IEC 8825 | CCITT Recommendation X.209.

19.2 Content type

A secondary object that submits a message containing an EDIM or EDIN shall assign the integer value **35** to the Content Type.

19.3 Content length

A secondary object that submits a probe concerning a message containing an EDIM or EDIN 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 an EDIM or an EDIN) when the Basic Encoding Rules of ISO/IEC 8825 | CCITT Recommendation X.209 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 an EDIM or EDIN shall specify the Encoded Information Types (EITs) of the message as follows.

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

In the case of an EDIM, the EITs shall be the logical union of the EITs of the EDIM's body parts specified in accordance with the following rules:

a) *EDI Body Part:* The EIT of the EDI Body Part shall have the same value as the Heading field edi-bodypart-type, or its default value if absent.

b) *EDIM Body Part (Forwarded Message):* The EITs (if any) of an EDIM 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 EDI Body Part Type shall be indicated in the external EITs.

A secondary object that submits a message containing an EDIM to an MTA that conforms to ISO/IEC 10021-4 | CCITT Recommendation X.411 in its 1988 version shall use the union of the object identifiers from EDI Body Part Type (see 8.2.6 and annex A) for all "original-encoded-information-types".

A secondary object that submits a message containing an EDIM to an MTA that conforms to CCITT Recommendation X.411 in its 1984 version shall use the "*undefined*" bit of the "built-in-encoded-information-types" (called "basic-encoded-information-type" in CCITT Recommendation X.411 in its 1984 version), as no other indication is possible for the EITs defined in 8.2.6 in an MTA that conforms to CCITT Recommendation X.411 in its 1984 version. The "external-encoded-information-type" field shall not be present.

NOTE – The following reduced functionality has to be considered when a secondary object submits a message containing an EDIM to an MTA that conforms to CCITT Recommendation X.411 in its 1984 version or when such messages are relayed through such an MTA. The delivering MTA cannot make a comparison of which EITs, and hence primary EDI body part types, the UA is prepared to accept for delivery (otherwise it would not perform the delivery at all). In addition, the security features of an MTA that conforms to ISO/IEC 10021-4 | CCITT Recommendation X.411 in its 1988 version cannot be used.

20 Port realization

How an MS or the MTS concretely realizes the secondary ports it supplies is specified in 6.1 of ISO/IEC 10021-6 | CCITT Recommendation X.419.

How a UA, TLMA, or AU concretely realizes the primary ports it supplies is beyond the scope of this part of ISO/IEC 10021.

21 Conformance

The requirements a secondary object (excluding the MTS) and its implementor shall meet when the latter claims the former's conformance to this part of ISO/IEC 10021 are identified below. A number of the conformance requirements distinguish 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, EDIM Body Part type or Externally Defined Body Part type if, and only if, it accepts, preserves, and emits, in full accord with this part of ISO/IEC 10021, that particular Heading field or extension, or EDIM Body Part type or

Externally Defined Body Part type, whenever a user calls upon it to convey an EDIM 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, EDIM Body Part type or Externally Defined Body Part type if, and only if, it accepts, preserves, and emits, in full accord with this part of ISO/IEC 10021, that particular Heading field or extension, or EDIM 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 it to convey to the user an EDIM containing them.

A PDAU supports nothing upon origination because it is not a supplier of the Origination Port.

21.2 Statement requirements

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

a) The Heading fields for which he claims conformance.

b) The body part types for which he claims conformance.

c) In the case of a UA with MS or MS, the EDI Messaging-specific MS attributes for which it claims conformance.

d) In the case of a UA with MS or MS, whether it supports the EDI Messaging-specific auto action.

e) In the case of an AU, whether it supports an Import Port or Export Port, or both.

21.3 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) A UA with MS or MS shall support the EDI Messaging-specific MS attributes for which conformance is claimed, but including as a minimum those designated mandatory in 18.7. In addition, it shall support the mandatory attributes identified in Table 1 of ISO/IEC 10021-5 | CCITT Recommendation X.413.

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 type of 19.2.

e) An AU shall be able to import and export such messages as appropriate.

f) An MS, or a UA accessing an MS, shall conform to at least one of the MS Access protocols specified in ISO/IEC 10021-6 | CCITT Recommendation X.419.

21.4 Dynamic requirements

A UA, MS or AU shall satisfy the following dynamic requirements:

a) A UA or MS shall follow the rules of operation specified in clauses 17 or 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 the content type of 19.2 and EITs as specified in 19.4.

Annex A

(normative)

Reference definition of object identifiers

The annex defines for reference purposes various object identifiers cited in the ASN.1 modules of subsequent annexes. It uses ASN.1.

All Object Identifiers this part of ISO/IEC 10021 assigns are assigned in this annex. The annex is definitive for all but those for ASN.1 modules, the object EDIMS application (EDIME) itself and the EDI use of Directories. The definitive assignments for the former occur in the modules themselves; other references to them appear in IMPORT statements. For the EDI use of Directories object identifiers, this annex only defines a base object identifier. This annex also specifies a root identifier for identifying character repertoires encoded per ISO/IEC multi–part standard 8859.

```
EDIMSObjectIdentifiers {joint-iso-ccitt mhs-motis(6) edims(7) modules(0)
                                     object-identifiers(0) }

DEFINITIONS IMPLICIT TAGS ::=
BEGIN

-- Prologue

-- Exports everything

IMPORTS;    --  nothing --


ID ::= OBJECT IDENTIFIER

-- EDI Messaging (definitive)

id-edims ID ::= { joint-iso-ccitt mhs-motis(6) edims(7) } -- This is definitive


-- Categories
id-mod                     ID ::= {id-edims  0} -- modules
id-edi                     ID ::= {id-edims  1} -- reserved
id-ot                      ID ::= {id-edims  2} -- object types
id-pt                      ID ::= {id-edims  3} -- port types
id-ref                     ID ::= {id-edims  4} -- refinements
id-sat                     ID ::= {id-edims  5} -- summary attributes
id-hat                     ID ::= {id-edims  6} -- heading attributes
id-rat                     ID ::= {id-edims  7} -- recipient attributes
id-bat                     ID ::= {id-edims  8} -- body attributes
id-nat                     ID ::= {id-edims  9} -- notification attributes
id-mct                     ID ::= {id-edims 10} -- message content types
id-bp                      ID ::= {id-edims 11} -- edi body part types
id-nt                      ID ::= {id-edims 12} -- not used
id-for                     ID ::= {id-edims 13} -- edi action idicator types
id-act                     ID ::= {id-edims 14} -- edi auto-action indentifier types
id-dir                     ID ::= {id-edims 15} -- edi use of directory
id-syn                     ID ::= {id-edims 16} -- edi synopsis type

-- 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-edi-directory-cl-att                  ID ::= {id-mod  6}
id-mod-message-store-auto-actions            ID ::= {id-mod  7}
```

```
-- Object types
id-ot-edime                              ID ::= {id-ot 0}
id-ot-edimg-user                         ID ::= {id-ot 1}
id-ot-edims                              ID ::= {id-ot 2}
id-ot-edi-ua                             ID ::= {id-ot 3}
id-ot-edi-ms                             ID ::= {id-ot 4}
id-ot-pdau                               ID ::= {id-ot 5}

-- Port types
id-pt-origination                        ID ::= {id-pt 0}
id-pt-reception                          ID ::= {id-pt 1}

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

-- Message content type (for use by MS and the Directory)
id-mct-pedi                              ID ::= {id-mct 0} -- Pedi

-- EDI Body Part type (and P1 EIT)
id-bp-edifact-ISO646                     ID ::= {id-bp 0} -- ISO646 is equivalent
                                               -- to CCITT Recommendation T.50
id-bp-edifact-T61                        ID ::= {id-bp 1}
id-bp-edifact-octet                      ID ::= {id-bp 2}
id-bp-ansiX12-ISO646                     ID ::= {id-bp 3}
id-bp-ansiX12-T61                        ID ::= {id-bp 4}
id-bp-ansiX12-octet                      ID ::= {id-bp 5}
id-bp-ansiX12-ebcdic                     ID ::= {id-bp 6}
id-bp-untdi-ISO646                       ID ::= {id-bp 7}
id-bp-untdi-T61                          ID ::= {id-bp 8}
id-bp-untdi-octet                        ID ::= {id-bp 9}
id-bp-private-octet                      ID ::= {id-bp 10}
id-bp-undefined-octet                    ID ::= {id-bp 11}
id-bp-edifact-8859                       ID ::= {id-bp 12}   -- Root object-id for
                                         -- ISO/IEC 8859 character repertoires

-- EDI Action Request
id-for-action                            ID ::= {id-for 0} -- For action
id-for-copy                              ID ::= {id-for 1} -- copy, not original

-- EDIMG Specific Register Auto Actions
id-act-edi-auto-forward                  ID ::= {id-act 0}

-- EDIM Synopsis (MS)
id-syn-removed                           ID ::= {id-syn 0}
id-syn-place-holder                      ID ::= {id-syn 1}

-- MESSAGE STORE ATTRIBUTES
-- Summary attributes
id-sat-edims-entry-type                  ID ::= {id-sat 0}
id-sat-edim-synopsis                     ID ::= {id-sat 1}
id-sat-edi-notification-indicator        ID ::= {id-sat 2}

-- Heading attributes
id-hat-heading                           ID ::= {id-hat 0}
id-hat-this-edim                         ID ::= {id-hat 1}
id-hat-originator                        ID ::= {id-hat 2}
id-hat-recipients                        ID ::= {id-hat 3}
id-hat-edin-receiver                     ID ::= {id-hat 4}
id-hat-responsibility-forwarded          ID ::= {id-hat 5}
id-hat-edi-bodypart-type                 ID ::= {id-hat 6}
id-hat-incomplete-copy                   ID ::= {id-hat 7}
id-hat-expiry-time                       ID ::= {id-hat 8}
id-hat-related-messages                  ID ::= {id-hat 9}
id-hat-obsoleted-edims                   ID ::= {id-hat 10}
id-hat-edi-application-security-element  ID ::= {id-hat 11}
id-hat-edi-application-security-extensions ID ::= {id-hat 12}
id-hat-cross-referencing-information     ID ::= {id-hat 13}
id-hat-edi-message-type                  ID ::= {id-hat 14}
id-hat-service-string-advice             ID ::= {id-hat 15}
id-hat-syntax-identifier                 ID ::= {id-hat 16}
id-hat-interchange-sender                ID ::= {id-hat 17}
id-hat-date-and-time-of-preparation      ID ::= {iid-hat 18}
```

```
id-hat-application-reference                  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-action-request-for-this-recipient                 ID ::= {id-rat 1}
id-rat-edi-notification-requests-for-this-recipient      ID ::= {id-rat 2}
id-rat-responsibility-passing-allowed-for-this-recipient ID ::= {id-rat 3}
-- UNB EDIFACT Field Object Ids --
id-rat-interchange-recipient-for-this-recipient          ID ::= {id-rat 4}
id-rat-recipient-reference-for-this-recipient            ID ::= {id-rat 5}
id-rat-interchange-control-reference-for-this-recipient  ID ::= {id-rat 6}
id-rat-processing-priority-code-for-this-recipient       ID ::= {id-rat 7}
id-rat-acknowledgement-request-for-this-recipient        ID ::= {id-rat 8}
id-rat-communications-agreement-id-for-this-recipient    ID ::= {id-rat 9}
id-rat-test-indicator-for-this-recipient                 ID ::= {id-rat 10}
id-rat-notification-security-for-this-recipient          ID ::= {id-rat 11}
id-rat-edi-reception-security-for-this-recipient         ID ::= {id-rat 12}
id-rat-recipient-extensions-for-this-recipient           ID ::= {id-rat 13}
-- ANSIX12 ISA Field Object Ids --
id-rat-authorization-information-for-this-recipient      ID ::= {id-rat 14}

-- Body attributes
id-bat-body                                   ID ::= {id-bat 0}
id-bat-interchange-length                     ID ::= {id-bat 1}
id-bat-edi-body-part                          ID ::= {id-bat 2}
id-bat-edim-body-part                         ID ::= {id-bat 3}
id-bat-message-parameters                     ID ::= {id-bat 4}
id-bat-message-data                           ID ::= {id-bat 5}
id-bat-externally-defined-body-part-types     ID ::= {id-bat 6}

-- Notification attributes
id-nat-subject-edim                           ID ::= {id-nat 0}
id-nat-edin-originator                        ID ::= {id-nat 1}
id-nat-first-recipient                        ID ::= {id-nat 2}
id-nat-notification-time                      ID ::= {id-nat 3}
id-nat-notification-security-elements         ID ::= {id-nat 4}
id-nat-notification-extensions                ID ::= {id-nat 5}
id-nat-edin-initiator                         ID ::= {id-nat 6}
-- PN attributes
id-nat-pn-supplementary-info                  ID ::= {id-nat 7}
id-nat-pn-extensions                          ID ::= {id-nat 8}
-- NN attributes
id-nat-nn-reason-code                         ID ::= {id-nat 9}
id-nat-nn-supplementary-info                  ID ::= {id-nat 10}
id-nat-nn-extensions                          ID ::= {id-nat 11}
-- FN attributes
id-nat-forwarded-to                           ID ::= {id-nat 12}
id-nat-fn-reason-code                         ID ::= {id-nat 13}
id-nat-fn-supplementary-info                  ID ::= {id-nat 14}
id-nat-fn-extensions                          ID ::= {id-nat 15}

-- MESSAGE STORE ATTRIBUTES - END

END -- of EDIMSObjectIdentifiers
```

Annex B

(normative)

Reference definition of abstract information objects

This annex defines for reference purposes the abstract information objects of EDI Messaging. It defines a Body Part for EDIM that includes a body part reference number while importing the IPMS externally defined MACRO for specifying non-EDI body parts. It also defines an EDIM-EXTENSION MACRO that differs from IPMS.

```
EDIMSInformationObjects {joint-iso-ccitt  mhs-motis(6) edims(7) modules(0)
                                                     information-objects(2)  }

DEFINITIONS IMPLICIT TAGS ::=
BEGIN

-- Prologue

-- Exports everything

IMPORTS

-- EDIMS Upper bounds

ub-application-reference, ub-authorization-information,
  ub-authorization-information-qualifier, ub-communications-agreement-id,
  ub-edi-application-security-elements, ub-edi-message-type,
  ub-identification-code, ub-identification-code-qualifier,
  ub-interchange-control-reference, ub-local-reference,
  ub-processing-priority-code, ub-reason-code, ub-recipient-reference,
  ub-recipient-reference-qualifier, ub-routing-address, ub-syntax-identifier,
  ub-syntax-version
  ----
  FROM EDIMSUpperBounds {joint-iso-ccitt mhs-motis(6) edims(7)  modules(0)
    upper-bounds(5) }

-- EDIMS Object Identifiers

id-bp-edifact-ISO646, id-for-action
   -----
   FROM EDIMSObjectIdentifiers {joint-iso-ccitt mhs-motis(6) edims(7) modules(0)
          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) }

-- MTS Abstract Service

MessageDeliveryTime,  ORName, OtherMessageDeliveryFields, ContentIntegrityCheck,
Content
  ----
   FROM MTSAbstractService {joint-iso-ccitt mhs-motis(6) mts(3)  modules(0) mts-
          abstract-service(1) }

-- IPM Information Objects

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

```
--   END  Imports

-- ABSTRACT INFORMATION OBJECTS

-- Overview

InformationObject ::= CHOICE {
  edim            [0] EDIM,
  edin            [1] EDIN }

-- Common data types

-- EDIM Identifier

EDIMIdentifier ::= SET {
  user                                 [0] ORName,
  user-relative-identifier             [1] LocalReference }

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

-- Extensions

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

Criticality ::= BOOLEAN

-- EDIM Extension MACRO

EDIM-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

-- EDI Messages

EDIM ::= SEQUENCE {
  heading                              Heading,
  body                                 Body }

-- Heading Field Component Types

-- Interchange Recipient / Sender

-- Identification Code

IdentificationCode  ::= TeletexString (SIZE (1..ub-identification-code))

-- Identification Code Qualifier

IdentificationCodeQualifier  ::=
   TeletexString (SIZE (1..ub-identification-code-qualifier))

-- Routing Address

RoutingAddress ::= TeletexString (SIZE (1..ub-routing-address))
```

```
-- Heading Fields

Heading ::= SEQUENCE {
  this-EDIM                             [1] ThisEDIMField,
  originator                            [2] OriginatorField OPTIONAL,
  recipients                            [3] RecipientsField OPTIONAL,
  edin-receiver                         [4] EDINReceiverField OPTIONAL,
  responsibility-forwarded              [5] ResponsibilityForwarded DEFAULT FALSE,
  edi-bodypart-type                     [6] EDIBodyPartType DEFAULT {id-bp-edifact-ISO646},
  incomplete-copy                       [7] IncompleteCopyField DEFAULT FALSE,
  expiry-time                           [8] ExpiryTimeField OPTIONAL,
  related-messages                      [9] RelatedMessagesField OPTIONAL,
  obsoleted-EDIMs                       [10] ObsoletedEDIMsField OPTIONAL,
  edi-application-security-elements     [11] EDIApplicationSecurityElementsField OPTIONAL,
  cross-referencing-information         [12] CrossReferencingInformationField OPTIONAL,
  -- Begin Fields from EDIFACT Interchange
  edi-message-type                      [13] EDIMessageTypeField OPTIONAL,
  service-string-advice                 [14] ServiceStringAdviceField OPTIONAL,
  syntax-identifier                     [15] SyntaxIdentifierField OPTIONAL,
  interchange-sender                    [16] InterchangeSenderField OPTIONAL,
  date-and-time-of-preparation          [17] DateAndTimeOfPreparationField OPTIONAL,
  application-reference                 [18] ApplicationReferenceField OPTIONAL,
  -- End Fields from EDIFACT
  heading-extensions                    [19] HeadingExtensionsField OPTIONAL }

-- This  EDIM

ThisEDIMField ::= EDIMIdentifier

-- Originator

OriginatorField ::= ORName

-- Recipients

RecipientsField ::= SET OF RecipientsSubField

RecipientsSubField ::= SEQUENCE {
  recipient                            [1] RecipientField,
  action-request                       [2] ActionRequestField DEFAULT {id-for-action},
  edi-notification-requests-field      [3] EDINotificationRequestsField OPTIONAL,
  responsibility-passing-allowed       [4] ResponsibilityPassingAllowedField DEFAULT FALSE,
-- Begin Fields from EDIFACT UNB
  interchange-recipient                [5] InterchangeRecipientField OPTIONAL,
  recipient-reference                  [6] RecipientReferenceField OPTIONAL,
  interchange-control-reference        [7] InterchangeControlReferenceField OPTIONAL,
  processing-priority-code             [8] ProcessingPriorityCodeField OPTIONAL,
  acknowledgement-request              [9] AcknowledgementRequestField DEFAULT FALSE,
  communications-agreement-id          [10] CommunicationsAgreementIdField OPTIONAL,
  test-indicator                       [11] TestIndicatorField DEFAULT FALSE,
-- End Fields from EDIFACT UNB
-- Begin Fields from ANSIX12 ISA
  authorization-information            [12] AuthorizationInformationField OPTIONAL,
-- End Fields from ANSIX12 ISA
  recipient-extensions                 [13] RecipientExtensionsField OPTIONAL }

-- Recipient

RecipientField ::= ORName

-- Action Request

ActionRequestField ::= OBJECT IDENTIFIER

-- EDI Notification Requests
```

```
EDINotificationRequestsField ::= SEQUENCE {
  edi-notification-requests         [0] EDINotificationRequests DEFAULT {},
  edi-notification-security         [1] EDINotificationSecurity DEFAULT {},
  edi-reception-security            [2] EDIReceptionSecurity DEFAULT {}  }

EDINotificationRequests ::= BIT STRING {
  pn          (0),
  nn          (1),
  fn          (2) }(SIZE (0..ub-bit-options))

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

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

-- Interchange recipient

InterchangeRecipientField ::= SEQUENCE {
  recipient-identification          [0] IdentificationCode,
  identification-code-qualifier     [1] IdentificationCodeQualifier OPTIONAL,
  routing-address                   [2] RoutingAddress OPTIONAL }

-- Recipient reference

RecipientReferenceField ::= SEQUENCE {
  recipient-reference               [0] RecipientReference,
  recipient-reference-qualifier     [1] RecipientReferenceQualifier OPTIONAL }

RecipientReference   ::= TeletexString (SIZE (1..ub-recipient-reference))

RecipientReferenceQualifier    ::=
    TeletexString (SIZE (1..ub-recipient-reference-qualifier))

-- Recipient Extensions

RecipientExtensionsField ::= SET OF RecipientExtensionsSubField

RecipientExtensionsSubField ::= ExtensionField

-- EDIN receiver

EDINReceiverField ::= SEQUENCE {
  edin-receiver-name                [0] ORName,
  original-edim-identifier          [1] EDIMIdentifier OPTIONAL,
  first-recipient                   [2] FirstRecipientField OPTIONAL}

-- Responsibility Forwarded indication

ResponsibilityForwarded ::= BOOLEAN -- Default False

-- EDI Body Part Types - identifies EDI Standard, Character set and encoding

EDIBodyPartType :=   OBJECT IDENTIFIER -- default EDIFACT-ISO646

-- EDI message type

EDIMessageTypeField ::=   SET OF EDIMessageTypeFieldSubField

EDIMessageTypeFieldSubField ::= TeletexString (SIZE (1..ub-edi-message-type))

-- Responsibility Passing Allowed

ResponsibilityPassingAllowedField ::= BOOLEAN -- Default FALSE

-- Incomplete Copy

IncompleteCopyField ::= BOOLEAN  -- Default False

-- Expiry time
```

```
ExpiryTimeField ::= UTCTime

-- Related Messages

RelatedMessagesField ::= SEQUENCE OF RelatedMessageReference

RelatedMessageReference ::= CHOICE {
  edi-message-reference              [0] EDIMIdentifier,
  external-message-reference         [1] ExternalMessageReference }

ExternalMessageReference ::= EXTERNAL

-- Obsoleted EDIMs

ObsoletedEDIMsField ::= SEQUENCE OF ObsoletedEDIMsSubfield

ObsoletedEDIMsSubfield ::= EDIMIdentifier

-- EDI Application Security Elements

EDIApplicationSecurityElementsField ::= SEQUENCE {
  edi-application-security-element      [0] EDIApplicationSecurityElement OPTIONAL,
  edi-encrypted-primary-bodypart        [1] BOOLEAN OPTIONAL,
  edi-application-security-extensions   [2] EDIApplicationSecurityExtensions OPTIONAL }

EDIApplicationSecurityElement  ::=
    BIT STRING (SIZE (0..ub-edi-application-security-elements))

EDIApplicationSecurityExtensions     ::= SET OF EDIApplicationSecurityExtension

EDIApplicationSecurityExtension      ::= ExtensionField

-- Cross Referencing Information

CrossReferencingInformationField   ::= SET OF CrossReferencingInformationSubField

CrossReferencingInformationSubField ::= SEQUENCE {
  application-cross-reference         [0] ApplicationCrossReference,
  message-reference                   [1] MessageReference OPTIONAL,
  body-part-reference                 [2] BodyPartReference }

ApplicationCrossReference ::= OCTET STRING

MessageReference     ::= EDIMIdentifier

-- Service String Advice

ServiceStringAdviceField ::= SEQUENCE {
  component-data-element-separator    [0] ComponentDataElementSeparator,
  data-element-separator              [1] DataElementSeparator,
  decimal-notation                    [2] DecimalNotation,
  release-indicator                   [3] ReleaseIndicator OPTIONAL,
  reserved                            [4] Reserved OPTIONAL,
  segment-terminator                  [5] SegmentTerminator }

ComponentDataElementSeparator  ::= OCTET STRING (SIZE (1))

DataElementSeparator           ::= OCTET STRING (SIZE (1))

DecimalNotation                ::= OCTET STRING (SIZE (1))

ReleaseIndicator               ::= OCTET STRING (SIZE (1))

Reserved                       ::= OCTET STRING (SIZE (1))

SegmentTerminator              ::= OCTET STRING (SIZE (1))

-- Syntax Identifier

SyntaxIdentifierField ::= SEQUENCE {
  syntax-identifier                  SyntaxIdentifier,
  syntax-version                     SyntaxVersion }
```

```
SyntaxIdentifier ::= TeletexString (SIZE (1..ub-syntax-identifier))

SyntaxVersion ::= PrintableString (SIZE (1..ub-syntax-version))

-- Interchange sender

InterchangeSenderField ::= SEQUENCE {
  sender-identification          [0] IdentificationCode,
  identification-code-qualifier  [1] IdentificationCodeQualifier OPTIONAL,
    address-for-reverse-routing  [2] RoutingAddress OPTIONAL } -- EDIFACT Routing
                                          Information

-- Date and Time of preparation

DateAndTimeOfPreparationField ::= UTCTime

-- Interchange control reference

InterchangeControlReferenceField ::=
    TeletexString (SIZE (1..ub-interchange-control-reference))

-- Application reference

ApplicationReferenceField ::= TeletexString (SIZE (1..ub-application-reference))

-- Processing Priority Code

ProcessingPriorityCodeField ::= TeletexString (SIZE (1..ub-processing-priority-code))

-- Acknowledgement Request

AcknowledgementRequestField ::= BOOLEAN -- default FALSE

-- Communications Agreement Id

CommunicationsAgreementIdField ::=
    TeletexString (SIZE (1..ub-communications-agreement-id))

-- Test indicator

TestIndicatorField ::= BOOLEAN -- default FALSE

-- Authorization Information

AuthorizationInformationField ::= SEQUENCE {
  authorization-information            [0] AuthorizationInformation,
  authorization-information-qualifier [1] AuthorizationInformationQualifier OPTIONAL }

AuthorizationInformation ::= TeletexString (SIZE (1..ub-authorization-information))

AuthorizationInformationQualifier ::= TeletexString (SIZE (1..ub-authorization-
information-qualifier))

-- Heading Extensions

HeadingExtensionsField ::= SET OF HeadingExtensionsSubField

HeadingExtensionsSubField ::= ExtensionField

-- EDIM body

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

```
PrimaryBodyPart ::= CHOICE {
  edi-body-part                  [0] EDIBodyPart,
  forwarded-EDIM                 [1] EDIMBodyPart }

OtherBodyParts ::= SEQUENCE OF EDIM-ExternallyDefinedBodyPart

-- EDI body part

EDIBodyPart ::= OCTET STRING

-- Forwarded EDIM body part

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

MessageParameters ::= SET {
  delivery-time                  [0] MessageDeliveryTime OPTIONAL,
  delivery-envelope              [1] OtherMessageDeliveryFields OPTIONAL,
  other-parameters               [2] EDISupplementaryInformation OPTIONAL }
-- MessageDeliveryTime and OtherMessageDeliveryFields shall both be present or both be
-- absent. EDISupplementaryInformation is used in the case of ms-auto-actions, see 18.6.

MessageData ::= SEQUENCE {
  heading                        Heading,
  body                           BodyOrRemoved }

BodyOrRemoved ::= SEQUENCE {
  primary-or-removed             PrimaryOrRemoved,
  additional-body-parts          AdditionalBodyParts OPTIONAL }

PrimaryOrRemoved ::= CHOICE {
  removed-edi-body               [0] NULL,
  primary-body-part              [1] EXPLICIT PrimaryBodyPart }

AdditionalBodyParts ::= SEQUENCE OF CHOICE {
  external-body-part    [0] EDIM-ExternallyDefinedBodyPart,
   place-holder         [1] BodyPartPlaceHolder } -- This type is for Body Part Removal

BodyPartPlaceHolder ::= EDIM-ExternallyDefinedBodyPart -- Only the data
-- portion of the Externally Defined Body shall be removed.
-- See text in 8.3.2.

-- EDIM Externally Defined Body Parts

EDIM-ExternallyDefinedBodyPart ::= SEQUENCE {
  body-part-reference            [0] BodyPartReference OPTIONAL,
  external-body-part             [1] ExternallyDefinedBodyPart -- from IPMS --}

BodyPartReference ::= INTEGER -- is unique within a EDIM

-- Supplementary Info

EDISupplementaryInformation ::= TeletexString (SIZE (1..ub-supplementary-info-length))

-- EDI Notifications (EDINs)

EDIN ::= CHOICE {
  positive-notification          [0] PositiveNotificationFields,
  negative-notification          [1] NegativeNotificationFields,
  forwarded-notification         [2] ForwardedNotificationFields }

-- Common fields

CommonFields ::= SEQUENCE {
  subject-edim                   [1] SubjectEDIMField,
  edin-originator                [2] EDINOriginatorField,
  first-recipient                [3] FirstRecipientField OPTIONAL,
  notification-time              [4] NotificationTimeField,
  notification-security-elements [5] SecurityElementsField OPTIONAL,
  edin-initiator                 [6] EDINInitiatorField,
  notifications-extensions       [7] NotificationExtensionsField OPTIONAL }
```

```
-- Subject EDIM Identifier

SubjectEDIMField ::= EDIMIdentifier

-- EDI Notification Originator

EDINOriginatorField ::= ORName

-- First Recipient

FirstRecipientField ::= ORName

-- Notification Time

NotificationTimeField ::= UTCTime

-- Security Elements

SecurityElementsField ::= SEQUENCE {
  original-content                      [0] Content OPTIONAL,
  original-content-integrity-check      [1] ContentIntegrityCheck OPTIONAL,
  edi-application-security-elements     [2] EDIApplicationSecurityElementsField OPTIONAL,
  security-extensions                   [3] SecurityExtensionsField OPTIONAL }

SecurityExtensionsField ::= SET OF SecurityExtensionsSubField

SecurityExtensionsSubField ::= ExtensionField

-- EDIN Initiator

EDINInitiatorField ::= ENUMERATED {
  internal-ua (0),
  external-ua (1),
  internal-ms (2)}

-- Notification Extensions

NotificationExtensionsField ::= SET OF NotificationExtensionsSubField

NotificationExtensionsSubField ::= ExtensionField

-- Positive Notification fields

PositiveNotificationFields ::= SEQUENCE {
  pn-common-fields                      [0] CommonFields,
  pn-supplementary-information          [1] EDISupplementaryInformation OPTIONAL,
  pn-extensions                         [2] PNExtensionsField OPTIONAL }

-- Positive Notification Extensions

PNExtensionsField ::= SET OF PNExtensionsSubField

PNExtensionsSubField ::= ExtensionField

-- Negative notification fields

NegativeNotificationFields ::= SEQUENCE {
  nn-common-fields                      [0] CommonFields,
  nn-reason-code                        [1] NNReasonCodeField,
  nn-supplementary-information          [2] EDISupplementaryInformation OPTIONAL,
  nn-extensions                         [3] NNExtensionsField OPTIONAL }

-- Negative Notification Reason Codes

   NNReasonCodeField ::= CHOICE {
  nn-ua-ms-reason-code                  [0] NNUAMSReasonCodeField,
  nn-user-reason-code                   [1] NNUserReasonCodeField,
  nn-pdau-reason-code                   [2] NNPDAUReasonCodeField }

-- Negative Notification Reason Codes from an EDI-UA or EDI-MS
```

```
NNUAMSReasonCodeField ::= SEQUENCE {
  nn-ua-ms-basic-code                 [0] NNUAMSBasicCodeField,
  nn-ua-ms-diagnostic                 [1] NNUAMSDiagnosticField OPTIONAL }

-- Negative Notification Basic Reason Codes from an EDI-UA or EDI-MS. These codes
-- are those specified in annex B of ISO/IEC 10021-8 | CCITT Recommendation F.435
--  for the element of service "EDI Notification Request".

NNUAMSBasicCodeField ::= INTEGER{
  unspecified (0),
  cannot-deliver-to-user (1), -- the EDI Interchange can not be passed on to the user
  delivery-timeout (2),  -- the EDI Interchange could not be passed on to the user
    -- within a specified time limit
  message-discarded (3),-- the UA/MS discarded the message before handoff to user
  subscription-terminated (4),  -- recipient's subscription terminated after delivery
    -- but before handoff to user
  forwarding-error (5),          -- EDI Forwarding was attempted, but failed.
  security-error (6)             -- security error
`-- Physical delivery errors are indicated by "cannot-deliver-to-user"
  } (0..ub-reason-code)

-- Negative Notification Diagnostic Codes from an EDI-UA or EDI-MS

   NNUAMSDiagnosticField ::= INTEGER {
  -- This field may be used to further specify the error signalled in
  -- nn-ua-ms-basic-code.  Additional information may be indicated
  --in nn-supplementary-information.

-- General diagnostic codes
  protocol-violation (1),
  -- used if the UA detects a protocol error
  edim-originator-unknown (2),
  edim-recipient-unknown (3),
  edim-recipient-ambiguous (4),
  -- used if the EDIM recipients or originator are not valid
  action-request-not-supported (5),
  -- used when the action requested by the recipient is not performed
  edim-expired (6),
  -- used when the expiry date of the received EDIM occurred before the subject EDIM
  -- was successfully passed to the user or forwarded by the EDI-UA
  edim-obsoleted (7),
  -- used when the EDIM Identifier of the received EDIM was contained in the Obsoleted
  -- EDIM field of a previously received EDIM.
  duplicate-edim (8),
  -- used when the same EDIM is received more than once from the same originator
  unsupported-extension (9),
  -- used if the EDIM contains an extension which is not supported by the UA
  incomplete-copy-rejected (10),
  -- used if the EDI-UA does not accept EDIMs with the Incomplete Copy Indication true
  edim-too-large-for-application (11),
  -- used if the EDIM cannot be delivered to the user due to length constraints;
 -- Forwarding error diagnostic codes
  forwarded-edim-not-delivered (12),
  -- used when an Non-Delivery Report is received for forwarded EDIM
  forwarded-edim-delivery-time-out (13),
  -- used when no Delivery Report is received within a given period
  forwarding-loop-detected (14),
  -- used if the UA receives an EDIM which contains a previously forwarded EDIM
  unable-to-accept-responsibility (15),
  -- used if the EDI-UA cannot accept or forward responsibility
  -- interchange header diagnostic codes
  interchange-sender-unknown (16), -- used when the UA does not recognize the
  -- interchange-sender of the EDI interchange
  interchange-recipient-unknown (17), -- used when the UA cannot find a valid
  -- interchange recipient in the Recipient Specifier
  invalid-heading-field (18),
  invalid-bodypart-type (19),
  invalid-message-type (20),
  invalid-syntax-id (21),
  -- Security error diagnostic codes
  message-integrity-failure (22),
  forwarded-message-integrity-failure (23),
  unsupported-algorithm (24),
  decryption-failed (25),
  token-error (26),
  unable-to-sign-notification (27),
  unable-to-sign-message-receipt (28),
  authentication-failure (29),
  security-context-failure (30),
  message-sequence-failure (31),
  message-security-labelling-failure (32),
  repudiation-failure (33),
  proof-service-failure (34)
  } (1..ub-reason-code)

-- Negative Notification Reason Codes from a user

NNUserReasonCodeField ::= SEQUENCE {
  nn-user-basic-code                [0]  NNUserBasicCodeField,
  nn-user-diagnostic                [1]  NNUserDiagnosticField OPTIONAL }

-- Negative Notification Basic Reason Codes from a user

NNUserBasicCodeField ::= INTEGER {
  unspecified (0),
  syntax-error (1),    -- used when the user discovers a syntax error within
                       --  the EDI interchange
  interchange-sender-unknown (2),
  interchange-recipient-unknown (3), -- used when the UA cannot find a valid
                                     -- interchange recipient in the Recipient Specifier
  invalid-heading-field (4),
  invalid-bodypart-type (5),
  invalid-message-type (6),
  functional-group-not-supported (7),
  subscription-terminated (8),  -- unknown to EDIMS-User service
  no-bilateral-agreement (9),
  user-defined-reason (10)
  } (0..ub-reason-code)

-- Negative Notification Diagnostic Codes from a user

NNUserDiagnosticField ::= INTEGER (1..ub-reason-code)
  -- Contains reason passed by user when the value of nn-user-basic-code is
  -- user-defined-reason. Additional information may be indicated
  -- in nn-supplementary-information.

-- Negative Notification Reason Codes from a PDAU

NNPDAUReasonCodeField ::= SEQUENCE {
  nn-pdau-basic-code                [0]  NNPDAUBasicCodeField,
  nn-pdau-diagnostic                [1]  NNPDAUDiagnosticField OPTIONAL }
```

```
-- Negative Notification Basic Reason Codes from a PDAU

NNPDAUBasicCodeField ::= INTEGER {
  unspecified (0),
  undeliverable-mail (1),                  -- used if the PDAU determines that it cannot
                                           -- perform physical delivery of the EDIM
  physical-rendition-not-performed (2)     -- used if the PDAU cannot perform
                                           -- the physical rendition of the EDIM
  } (0..ub-reason-code)

-- Negative Notification Diagnostic Codes from a PDAU

NNPDAUDiagnosticField ::= INTEGER { -- This field may be used to further specify
--  the error signalled in nn-pdau-basic-code. Additional information may
  --  be indicated in the nn-supplementary-information.
  undeliverable-mail-physical-delivery-address-incorrect (32),
  undeliverable-mail-physical-delivery-office-incorrect-or-invalid (33),
  undeliverable-mail-physical-delivery-address-incomplete (34),
  undeliverable-mail-recipient-unknown (35),
  undeliverable-mail-recipient-deceased (36),
  undeliverable-mail-organization-expired (37),
  undeliverable-mail-recipient-refused-to-accept (38),
  undeliverable-mail-recipient-did-not-claim (39),
  undeliverable-mail-recipient-changed-address-permanently (40),
  undeliverable-mail-recipient-changed-address-temporarily (41),
  undeliverable-mail-recipient-changed-temporary-address (42),
  undeliverable-mail-new-address-unknown (43),
  undeliverable-mail-recipient-did-not-want forwarding (44),
  undeliverable-mail-originator-prohibited-forwarding (45),
  physical-rendition-attributes-not-supported (31)
   } (1..ub-reason-code)

-- Negative Notification Extension Field(s)

NNExtensionsField ::= SET OF NNExtensionsSubField

NNExtensionsSubField ::= ExtensionField

-- Forwarded Notification Fields

ForwardedNotificationFields ::= SEQUENCE {
  fn-common-fields                     [0] CommonFields,
  forwarded-to                         [1] ForwardedTo,
  fn-reason-code                       [2] FNReasonCodeField,
  fn-supplementary-information         [3] EDISupplementaryInformation OPTIONAL,
  fn-extensions                        [4] FNExtensionsField OPTIONAL }

-- Forwarded To

ForwardedTo ::= ORName

-- Forwarded Reason Code

FNReasonCodeField ::= CHOICE {
  fn-ua-ms-reason-code                 [0] FNUAMSReasonCodeField,
  fn-user-reason-code                  [1] FNUserReasonCodeField,
  fn-pdau-reason-code                  [2] FNPDAUReasonCodeField }

-- Forwarding Notification Reason Codes from an EDI-UA or EDI-MS

FNUAMSReasonCodeField ::= SEQUENCE {
  fn-ua-ms-basic-code                  [0]  FNUAMSBasicCodeField,
  fn-ua-ms-diagnostic                  [1]  FNUAMSDiagnosticField OPTIONAL,
  fn-security-check                    [2]  FNUAMSSecurityCheckField DEFAULT FALSE }
```

```
-- Forwarding Notification Basic Reason Codes from an EDI-UA or EDI-MS

FNUAMSBasicCodeField ::= INTEGER {
  unspecified (0),
  onward-routing (1),
  -- used whenever the UA decides to re-route the subject EDIM for local reasons
  EDIM-recipient-unknown (2),
  EDIM-originator-unknown (3),
  forwarded-by-edi-ms (4)
   } (0..ub-reason-code)

-- Forwarding Notification Diagnostic Reason Codes from an EDI-UA or EDI-MS
FNUAMSDiagnosticField ::= INTEGER { -- This field may be used to further specify the
  --  error signalled in fn-ua-ms-basic-code. Additional information may be
  -- indicated in fn-supplementary-information.
  recipient-name-changed (1),
  recipient-name-deleted (2)
   } (1..ub-reason-code)

-- Forwarding Notification Security Check Codes from an EDI-UA or EDI-MS

-- This field may be used, with a value of TRUE, to indicate that all security
-- features present have been validated, with a value of FALSE, to indicate that
-- the security features have not been validated.

FNUAMSSecurityCheckField ::= BOOLEAN

-- Forwarding Notification Reason Codes from a user

FNUserReasonCodeField ::= SEQUENCE {
  fn-user-basic-code                [0]  FNUserBasicCodeField,
  fn-user-diagnostic                [1]  FNUserDiagnosticField OPTIONAL }

-- Forwarding Notification Basic Reason Codes from a user

FNUserBasicCodeField ::= INTEGER {
  unspecified (0),
  forwarded-for-archiving (1),
  forwarded-for-information (2),
  forwarded-for-additional-action (3),
  subscription-changed (4),
  heading-field-not-supported (5),
  bodypart-type-not-supported (6),
  message-type-not-supported (7),
  syntax-identifier-not-supported (8),
  interchange-sender-unknown (9),
  user-defined-reason (10)
  } (0..ub-reason-code)

-- Forwarding Notification Diagnostic Reason Codes from a user

FNUserDiagnosticField ::= INTEGER (1..ub-reason-code)
-- Contains reason passed by user when value of fn-user-basic-code is user-defined-
-- reason. Additional information may be indicated in fn-supplementary-information.

-- Forwarding Notification Reason Codes from a PDAU

FNPDAUReasonCodeField ::= SEQUENCE {
  fn-pdau-basic-code                [0] FNPDAUBasicCodeField,
  fn-pdau-diagnostic                [1] FNPDAUDiagnosticField OPTIONAL }

-- Forwarding Notification Basic Reason Codes from a PDAU

FNPDAUBasicCodeField ::= INTEGER {
  unspecified (0),
  forwarded-for-physical-rendition-and-delivery (1)
  } (0..ub-reason-code)

-- Forwarding Notification Diagnostic Codes from a PDAU
```

```
FNPDAUDiagnosticField ::= INTEGER (1..ub-reason-code)

-- Forwarded Notification Extensions

FNExtensionsField ::= SET OF FNExtensionsSubField

FNExtensionsSubField ::= ExtensionField

END -- of EDIMSInformationObjects
```

Annex C

(normative)

Reference definition of Message Store attributes

This annex defines for reference purposes the MS attributes specific to EDI Messaging. It uses the ATTRIBUTE macro of ISO/IEC 9594-2 | CCITT Recommendation X.501.

```
EDIMSMessageStoreAttributes {joint-iso-ccitt  mhs-motis(6) edims(7) modules(0)
                                        message-store-attributes(4) }

DEFINITIONS IMPLICIT TAGS ::=
BEGIN

-- Prologue

-- Exports everything.

    IMPORTS

-- EDIMS Object Identifiers
id-bat-body, id-bat-edi-body-part, id-bat-edim-body-part,
  id-bat-externally-defined-body-part-types, id-bat-interchange-length,
  id-bat-message-data, id-bat-message-parameters, id-hat-application-reference,
  id-hat-cross-referencing-information,
  id-hat-date-and-time-of-preparation, id-hat-edi-application-security-element,
  id-hat-edi-application-security-extensions, id-hat-edi-bodypart-type,
  id-hat-edi-message-type, id-hat-edin-receiver, id-hat-expiry-time,
  id-hat-heading, id-hat-heading-extensions, id-hat-incomplete-copy,
  id-hat-interchange-sender, id-hat-obsoleted-edims, id-hat-originator,
  id-hat-recipients, id-hat-related-messages, id-hat-sensitivity,
  id-hat-service-string-advice, id-hat-syntax-identifier, id-hat-this-edim,
  id-nat-edin-originator, id-nat-first-recipient, id-nat-fn-extensions,
  id-nat-fn-reason-code, id-nat-fn-supplementary-info, id-nat-forwarded-to,
  id-nat-nn-extensions, id-nat-nn-reason-code, id-nat-nn-supplementary-info,
  id-nat-notification-extensions, id-nat-notification-security-elements,
  id-nat-notification-time, id-nat-pn-extensions, id-nat-pn-supplementary-info,
  id-nat-subject-edim, id-rat-acknowledgement-request,
  id-rat-action-request-for-this-recipient,
  id-rat-authorization-information-for-this-recipient,
  id-rat-communications-agreement-id-for-this-recipient,
  id-rat-edi-notification-requests-for-this-recipient,
  id-rat-edi-reception-security-requests-for-this-recipient,
  id-rat-interchange-control-reference-for-this-recipient,
  id-rat-interchange-recipient-for-this-recipient, id-rat-processing-priority-code,
  id-rat-recipient-extensions-for-this-recipient, id-rat-this-recipient,
  id-rat-recipient-reference-for-this-recipient,
  id-rat-responsibility-passing-allowed-for-this-recipient,
  id-rat-test-indicator-for-this-recipient, id-sat-edim-synopsis,
  id-sat-edims-entry-type
    -----
  FROM EDIMSObjectIdentifiers {joint-iso-ccitt mhs-motis(6) edims(7) modules(0)
    object-identifiers(0) }

-- MS Abstract Service

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

-- EDIMS Information Objects

AcknowledgementRequestField, ActionRequestField, ApplicationReferenceField,
  AuthorizationInformationField, Body, BodyPartReference,
  CommunicationsAgreementIdField, CrossReferencingInformationSubField,
  DateAndTimeOfPreparationField, EDIApplicationSecurityElementsField, EDIBodyPart,
  EDIBodyPartType, EDIMessageTypeFieldSubField, EDINInitiatorField,
  EDINOriginatorField, EDINotificationRequestsField, EDINReceiverField,
  EDISupplementaryInformation, ExpiryTimeField,
  FirstRecipientField, FNExtensionsSubField, FNReasonCodeField,
  ForwardedTo, Heading, HeadingExtensionsSubField, IncompleteCopyField,
  InterchangeControlReferenceField,  InterchangeRecipientField,
  InterchangeSenderField, MessageData, MessageParameters, NNReasonCodeField,
  NNExtensionsSubField, NotificationExtensionsSubField, NotificationTimeField,
  ObsoletedEDIMsSubfield, OriginatorField, PositiveNotificationFields,
  PNExtensionsSubField, ProcessingPriorityCodeField, RecipientExtensionsSubField,
  RecipientField, RecipientReferenceField, RecipientsSubField, RelatedMessagesField,
  ResponsibilityForwarded, ResponsibilityPassingAllowedField, SecurityElementsField,
  ServiceStringAdviceField, SubjectEDIMField,
  SyntaxIdentifierField, TestIndicatorField, ThisEDIMField
    ----
  FROM EDIMSInformationObjects {joint-iso-ccitt mhs-motis(6)
                                   edims(7) modules(0) information-objects(2) }

-- IPMS Information Objects

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

-- Directory Information Framework

ATTRIBUTE
    ----
  FROM InformationFramework {joint-iso-ccitt ds(5) modules(1) informationFramework(1) };

-- END imports

-- MESSAGE STORE ATTRIBUTES

-- Summary Attributes

-- EDIMS Entry Type

edims-entry-type ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX EDIMSEntryType
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-sat-edims-entry-type

EDIMSEntryType ::= ENUMERATED {
  edim (0),
  pn (1),
  nn (2),
  fn (3) }

-- EDIM Synopsis

edim-synopsis ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX EDIMSynopsis
  SINGLE VALUE
  ::= id-sat-edim-synopsis

EDIMSynopsis ::= SEQUENCE OF BodyPartSynopsis
```

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

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

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

-- EDI Notification Indicator

edi-notification-indicator ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX EDINotificationIndicator DEFAULT (0)
  MATCHES FOR EQUALITY
  MULTI VALUE
  ::= id-sat-edi-notification-indicator

EDINotificationIndicator ::= ENUMERATED {
  no-notification-sent              (0),
  pn-sent                           (1),
  nn-sent                           (2),
  fn-sent                           (3) }

-- Heading Attributes

-- Heading

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

-- Heading Fields

this-edim ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX ThisEDIMField
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-hat-this-edim

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

edin-receiver ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX EDINReceiverField
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-hat-edin-receiver

responsibility-forwarded ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX ResponsibilityForwarded
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-hat-responsibility-forwarded

edi-bodypart-type ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX EDIBodyPartType
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-hat-edi-bodypart-type
```

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

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

related-messages ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX RelatedMessagesReference
  MATCHES FOR EQUALITY
  MULTI VALUE
  ::= id-hat-related-messages

obsoleted-edims ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX ObsoletedEDIMsSubfield
  MATCHES FOR EQUALITY
  MULTI VALUE
  ::= id-hat-obsoleted-edims

edi-application-security-element ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX EDIApplicationSecurityElement
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-hat-edi-application-security-element

edi-application-security-extensions ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX EDIApplicationSecurityExtension
  MATCHES FOR EQUALITY
  MULTI VALUE
  ::= id-hat-edi-application-security-extensions

cross-referencing-information ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX CrossReferencingInformationSubField
  MATCHES FOR EQUALITY
  MULTI VALUE
  ::= id-hat-cross-referencing-information

edi-message-type ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX EDIMessageTypeFieldSubField
  MATCHES FOR EQUALITY
  MULTI VALUE
  ::= id-hat-edi-message-type

service-string-advice ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX ServiceStringAdviceField
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-hat-service-string-advice

syntax-identifier ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX SyntaxIdentifierField
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-hat-syntax-identifier

interchange-sender ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX InterchangeSenderField
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-hat-interchange-sender
```

```
date-and-time-of-preparation ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX DateAndTimeOfPreparationField
  MATCHES FOR EQUALITY ORDERING
  SINGLE VALUE
  ::= id-hat-date-and-time-of-preparation

application-reference ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX ApplicationReferenceField
  MATCHES FOR EQUALITY SUBSTRINGS
  SINGLE VALUE
  ::= id-hat-application-reference

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 RecipientField
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-rat-this-recipient

action-request-for-this-recipient ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX ActionRequestField
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-rat-action-request-for-this-recipient

edi-notification-requests-for-this-recipient ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX EDINotificationRequests
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-rat-edi-notification-requests-for-this-ient

edi-notification-security-for-this-recipient ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX EDINotificationSecurity
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-rat-edi-notification-security-for-this-ient

edi-reception-security-for-this-recipient ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX EDIReceptionSecurity
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-rat-edi-reception-security-for-this-ient

responsibility-passing-allowed-for-this-recipient ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX nsibilityPassingAllowedField
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-rat-responsibility-passing-allowed-for-this-ient
-- Fields from EDIFACT interchange
interchange-recipient-for-this-recipient ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX InterchangeRecipientField
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-rat-interchange-recipient-for-this-recipient

recipient-reference-for-this-recipient ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX RecipientReferenceField
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-rat-recipient-reference-for-this-recipient

interchange-control-reference-for-this-recipient ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX InterchangeControlReferenceField
  MATCHES FOR EQUALITY SUBSTRINGS
  SINGLE VALUE
  ::= id-rat-interchange-control-reference-for-this-recipient
```